

ҲИСОБЛАШ ТЕХНИКАСИ БАЗАСИДАГИ ТЕХНОЛОГИК ЖАРАЁНЛАРНИ БОШҚАРУВ ТИЗИМЛАРИ

Ҳар қандай корхона фаолиятининг асосини ишлаб чиқариш жараёни ташкил этади. Шу сабабдан, корхона муваффақияти ана шу жараёни қандай ташкил этилишига боғлиқдир. Ишлаб чиқариш жараёнини тўғри йўлга қўйиш унинг самарадорлигини оширади, моддий харажатларни камайтиради, меҳнат сарфини қисқартириб, таннархни пасайтиради.

Ишлаб чиқариш жараёни - маҳсулот ишлаб чиқариш учун мазкур корхоналар ва меҳнат қуроллари ҳаракатларининг мажмуидир ёки бошқача қилиб тариф берилса, ишлаб чиқариш жараёни инсоннинг меҳнат қуроллари ёрдамида меҳнат буюмларига мақсадга мувофиқ таъсирдир.

Ишлаб чиқариш жараёни уч элементдан ташкил топади:

1. Меҳнат буюмлари (хом ашё, материаллар).
2. Меҳнат воситалари (жиҳозлар ва ускуналар).
3. Меҳнат (инсон фаолияти).

Ишлаб чиқариш жараёни технологик жараёнлар ва ёрдамчи жараёнлардан ташкил топади. Технологик жараён буюмлар ҳолатини ўзгартиришга қаратилган ишлаб чиқариш жараёнини бир қисмидир. Технологик жараён технологик операциялар йииндисидан иборат. Технологик операция иш ўрнининг ишлов берилаётган маҳсулотнинг ва ишчининг ўзгармаслиги билан характерланади. Ана шу элементлардан биронтасининг ўзгариши бир операция тугаб иккинчиси бошланганидан далолат беради.

Қўлланиладиган жиҳозларга кўра технологик операцияларнинг қуйидаги турларини ажратиш мумкин:

1. Қўл операциялари (машина ёрдамсиз қўлда бажарилади).
2. Машина қўл операциялари (ишчи ва машиналар ёрдамида бажарилади. Масалан: тикув машиналарида бажариладиган операциялар).
3. Машина операциялари (жараён ишчининг иштирокисиз бажарилади, ишчининг функцияси жиҳозни ишга тайёрлашдан иборатдир. Масалан: токорлик цанокларида бажариладиган операциялар).
4. Автоматлашган операциялар (ишчининг иштирокисиз амалга оширилади).

ИШЛАБ ЧИҚАРИШНИ ТАШКИЛ ЭТИШГА ҚЎЙИЛАДИГАН ТАЛАБЛАР

Корхона фаолиятининг самарадорлиги кўп жиҳатдан ишлаб чиқариш жараёнини қандай ташкил этилишига боғлиқ . Ишлаб чиқаришни ташкил этишга қуйидаги талаблар қўйилади:

- хусусий жараёнлар ўртасида узлуксизликни таъминлаш,
- ишлаб чиқариш қувватларида зарурий пропорционалликни таъминлаш.
- меҳнат буюмларини самарали ҳаракатини йўлга қўйиш,
- маҳсулот ишлаб чиқариш муддатини қисқартириш,
- ишлаб чиқариш воситаларидан ва ишчи кучидан самарали фойдаланиб бир маромда маҳсулот ишлаб чиқаришни таоминлашга эришиш,
- ишлаб чиқариш жараёнида харажатлар иқтисодига эришиш.

Ана шу талабларни бажаришда ишлаб чиқаришни самарали ташкил этишнинг принципларига риоя қилиш муҳим аҳамият касб этади. Кибернетика фанининг асосчиси, америкалик математик Н.Винер XVIII аср соатлар асри, XIX аср буғ машиналари асри, XIX аср эса алоқа ва бошқариш асри деб та'кидлаган эди.

ТЕХНОЛОГИК ЖАРАЁНЛАРНИ АВТОМАТЛАШТИРИЛГАН БОШҚАРИШ ТИЗИМЛАРИ (ТЖАБТ) РИВОЖЛАНИШИНИ БОСҚИЧЛАРИ

Саноат автоматлаштириши компьютер тизимларининг технологик жараёнларни автоматлаштирилган бошқариш тизимлари (ТЖАБТ) ривожланишини учта йирик босқичга ажратиш мумкин.

ТЖАБТ ни яратишнинг биринчи босқичи биринчи авлод ЭҲМ ларидан фойдаланиш билан боғлиқ, масалан, «Урал», «УМ-1», «Минск» каби ЭҲМ лар.

Иккинчи босқичда IBM, ЕС ЭҲМ, мини компьютерлар (СМ ЭҲМ ва б.) туридаги мейнфреймлар қўлланилган эди. Бу босқичларда бошқариш тизимлари марказлашган тузилишга эга бўлиб, кўпинча реал вақт режимида этарлича тезкорлик ва ишлашни таъминлай олмасди. Ҳа вақтдаги компьютерлар элемент базаси ва дастурий таъминоти мукамал бўлмагани сабабли ишончлилиги паст эди, шу сабабли кўпинча ишдан чиқар эди.

Микроэлектроникадаги муваффақиятлар, микропросессорларнинг пайдо бўлиши 80- йилларнинг бошларида бошқариш тизимининг тузилиш техникасида инқилобий ўзгаришларни амалга оширди, саноат ишлаб чиқарадиган компьютерлаштиришнинг ва автоматлаштиришнинг мутлақо янги техник воситаларини яратишнинг учинчи босқичини очиб берди.

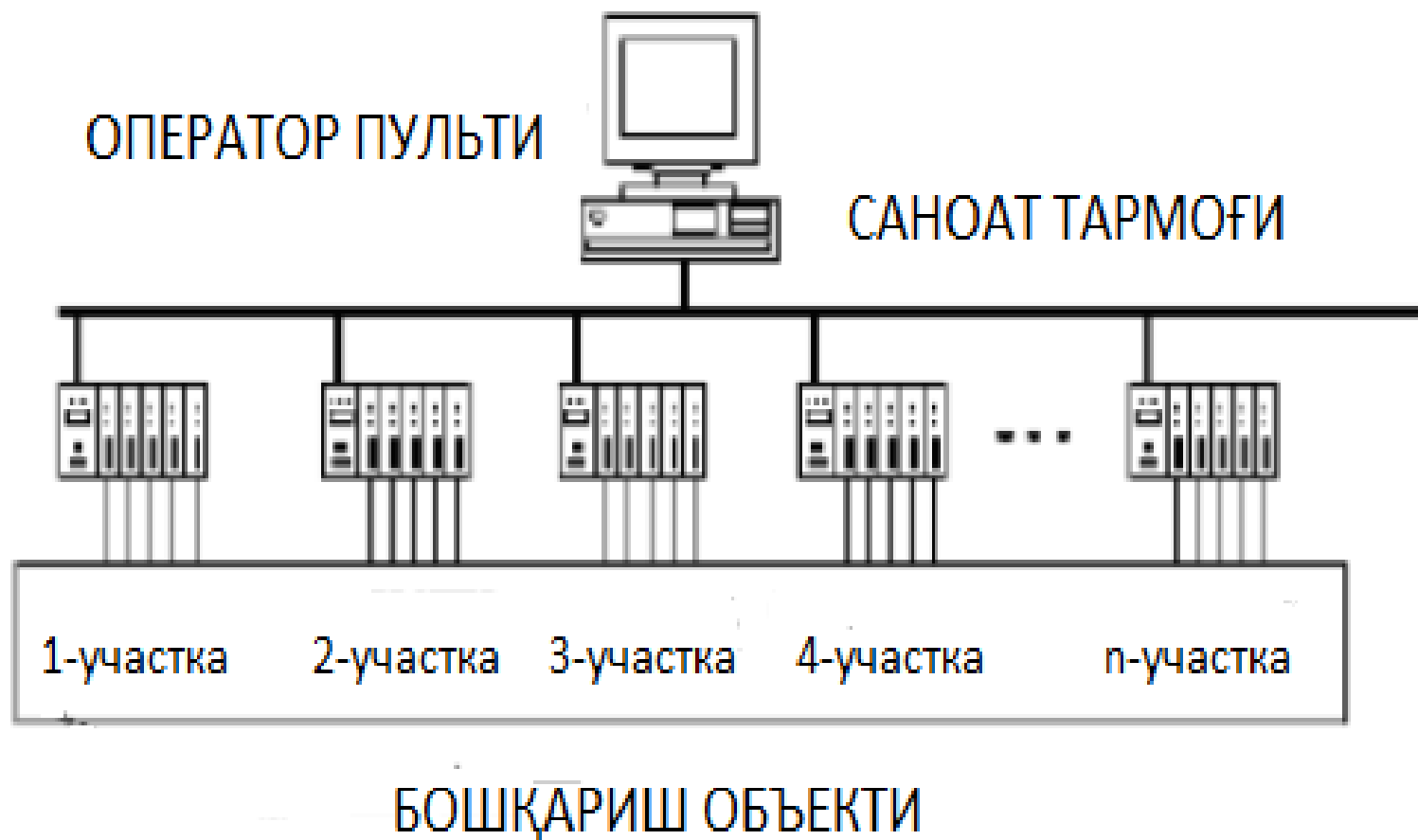
ДАСТУРИЙ-ТЕХНИК МАЖМУАЛАР

Микропросессорлар автоматлаштириш ва назоратнинг айрим воситалари таркибига кира бошлади. Айрим қурилмалар ўртасида ма'лумотларни рақамли узатиш, ҳисоблаш тармоғини бошқариш тизимларини қуришга асос бўлди. Ма'лумотларга ишлов беришнинг айрим қурилмалари орасидаги рақамли алоқани кўзда тутувчи янги тузилишдаги технологик жараённи бошқариш тизими марказлаштирилмаган - МТЖАБТ ёки тақсимланган – ТТЖАБ ХХ асрнинг 70- ва 80- йилларида жаҳондаги етакчи автоматлаштириш воситалари ишлаб чиқарувчилари ТЖАБТ ни қуриш учун дастурли-аппаратурали воситалар тўпламини ишлаб чиқара бошладилар. Бундай тўплamlарнинг асосий белгилари уларнинг мослашувчанлиги, ягона тизимда фаолият кўрсата олиш қобилиятига эгаллиги, интерфейсларнинг стандартлаштирилиши бутун ТЖАБТ ни фақат мазкур тўплаш воситаларидан қуришга имкон берувчи функционал тўлалик. Бундай воситалар тўплами дастурий-техник мажмуалар (ДТМ) номини олди.

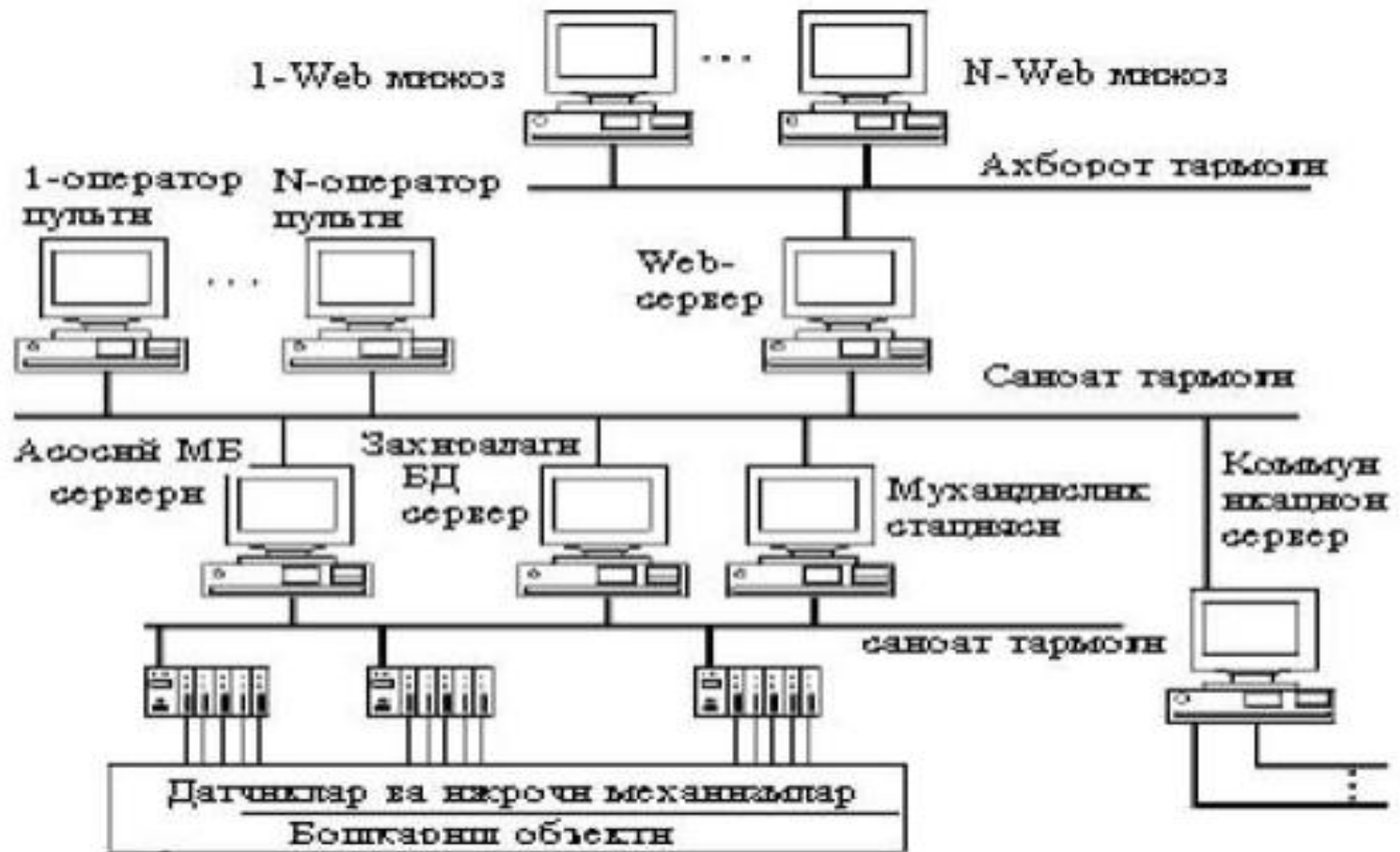
Ҳозирги пайтда кўпчилик технологик жараёнларни автоматлаштириш универсал микропроцессорли контроллер воситалари негизда амалга оширилмоқда, уларни дастурий-техник мажмуа (ДТМ) деб аталади. Дастурий-техник мажмуалар автоматлаштиришнинг микропроцессорли воситалари йиғиндисидан (микропроцессорли контроллерлар, объект билан алоқани ўрнатувчи мосламалари ОАЎМ), операторнинг дисплейли пуьтлари ва турли вазифаларни бажарувчи серверлар, саноат тармоқларидан иборат бўлиб, улар контроллерларнинг дастурий таъминотининг ва оператор дисплейли пуьтларининг санаб ўтилган компонентларини боғлашга имкон беради.

Бошқариш объектининг ахборот қувватининг (кирувчи-чиқувчи ўзгарувчилар миқдорининг) ортиши, бошқаришнинг юқори сатҳида ҳал этиладиган масалалар доирасининг кенгайиши, пухталиқ кўрсаткичларининг ортиши дастурий-техник мажмуаларнинг янада мураккаб тузилмаларининг пайдо бўлишига олиб келади.

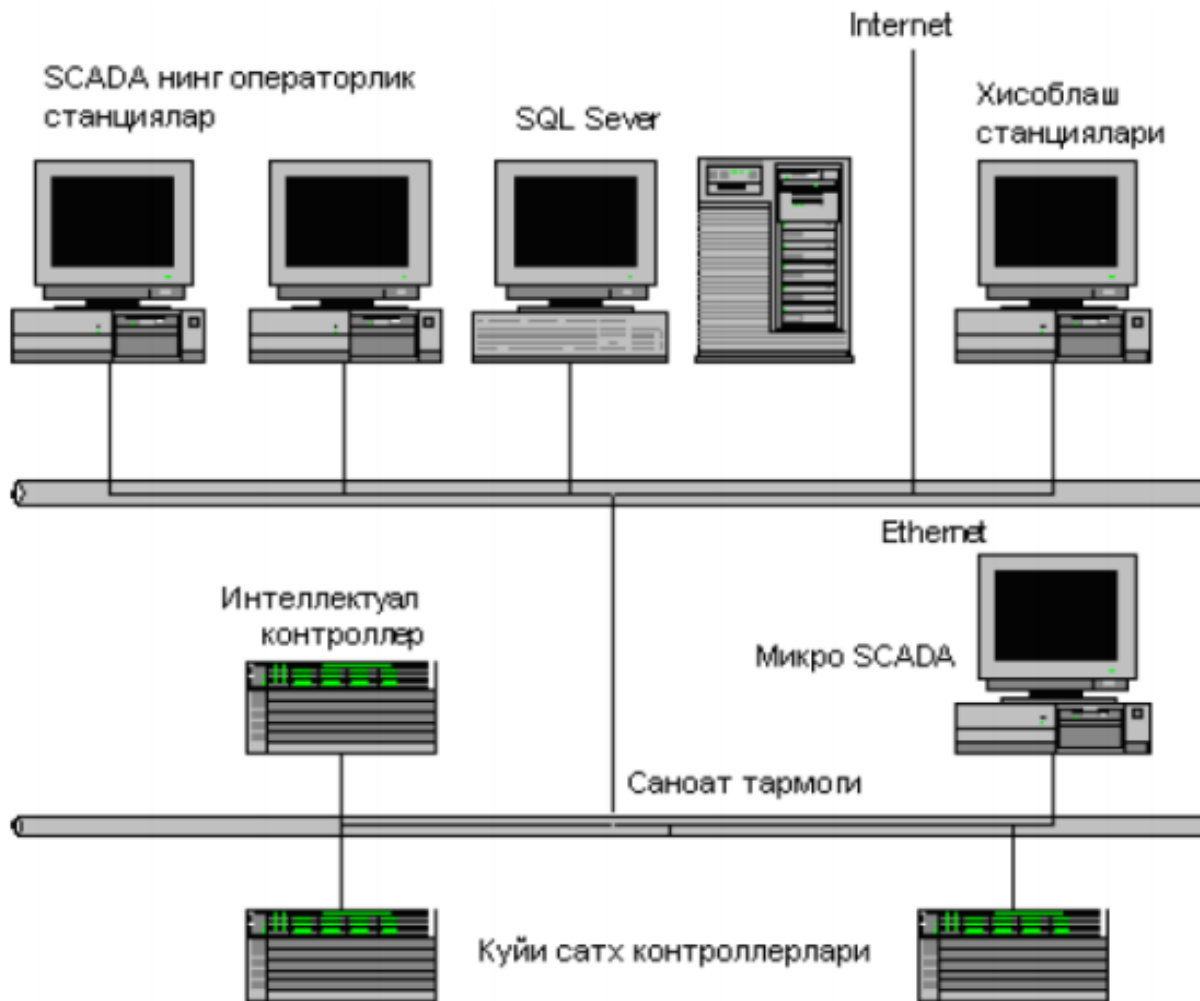
ДАСТУРИЙ-ТЕХНИК МАЖМУАНИ ТУЗИЛМАСИ



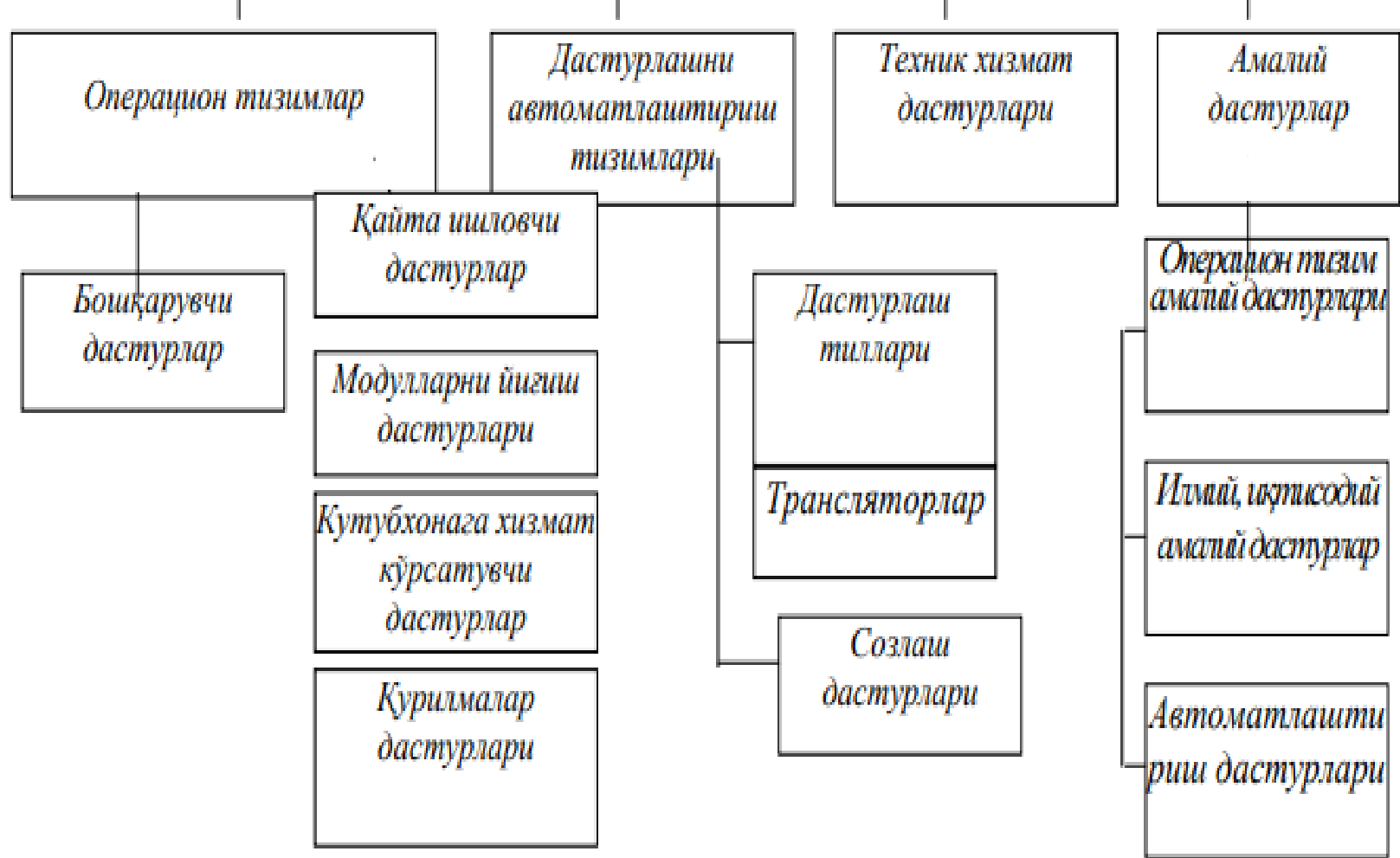
МУРАККАБ ТУЗИЛМАЛИ ДАСТУРИЙ-ТЕХНИК МАЖМУА



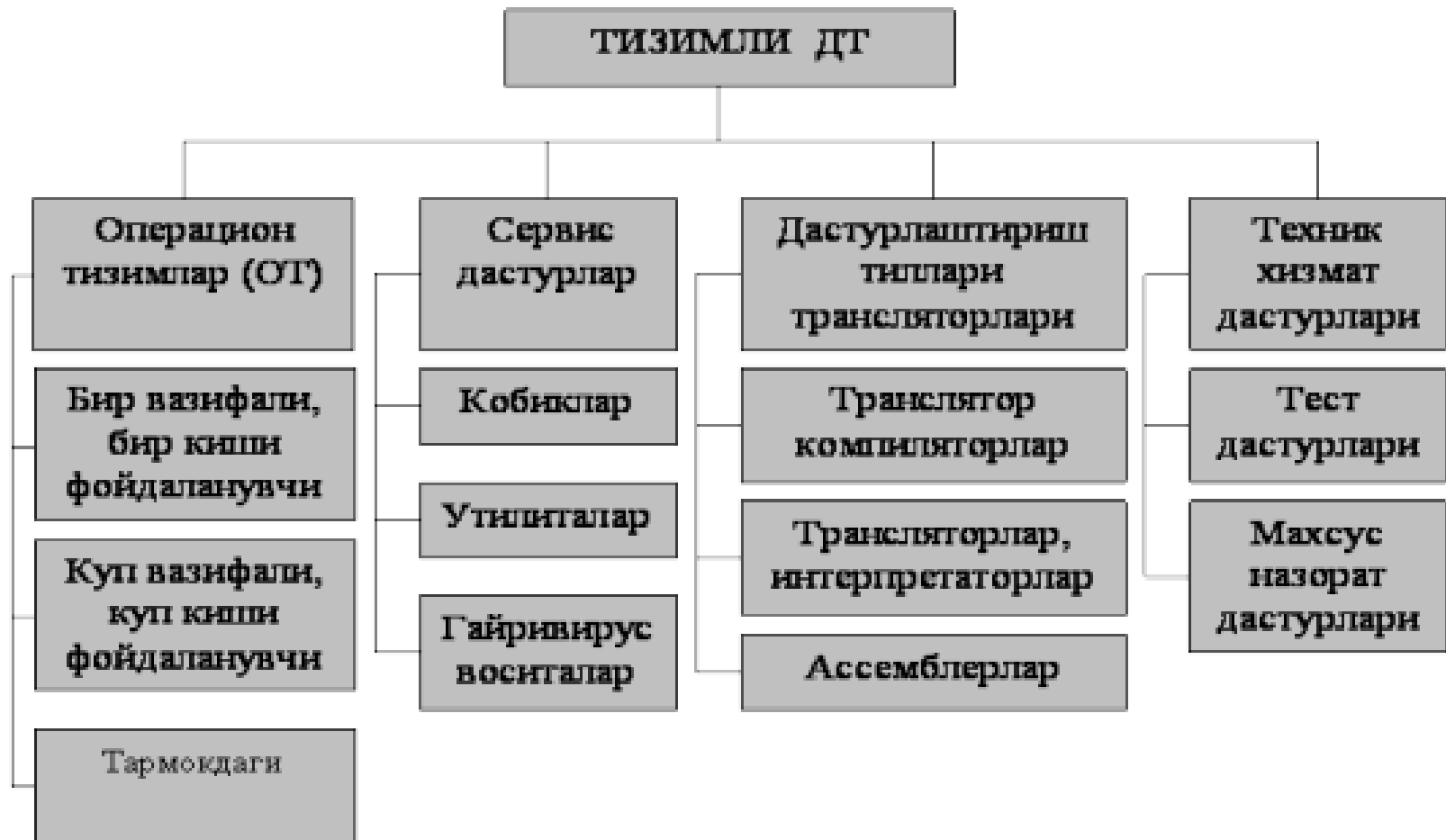
НАЗОРАТ ВА БОШҚАРИШ ТИЗИМИНИНГ УМУМИЙ СХЕМАСИ



Компьютер дастурий таъминоти таркиби



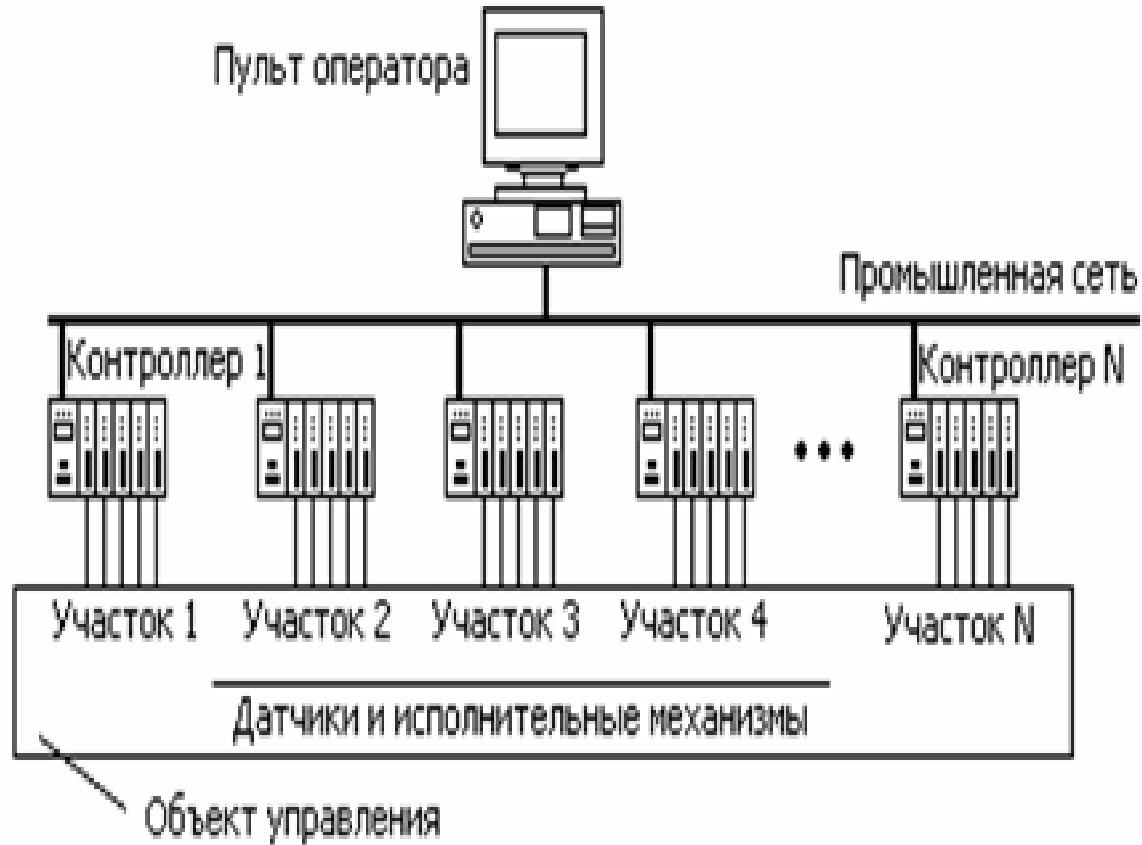
ДАСТУРИЙ ТАЪМИНОТНИНГ УМУМИЙ ЧИЗМАСИ



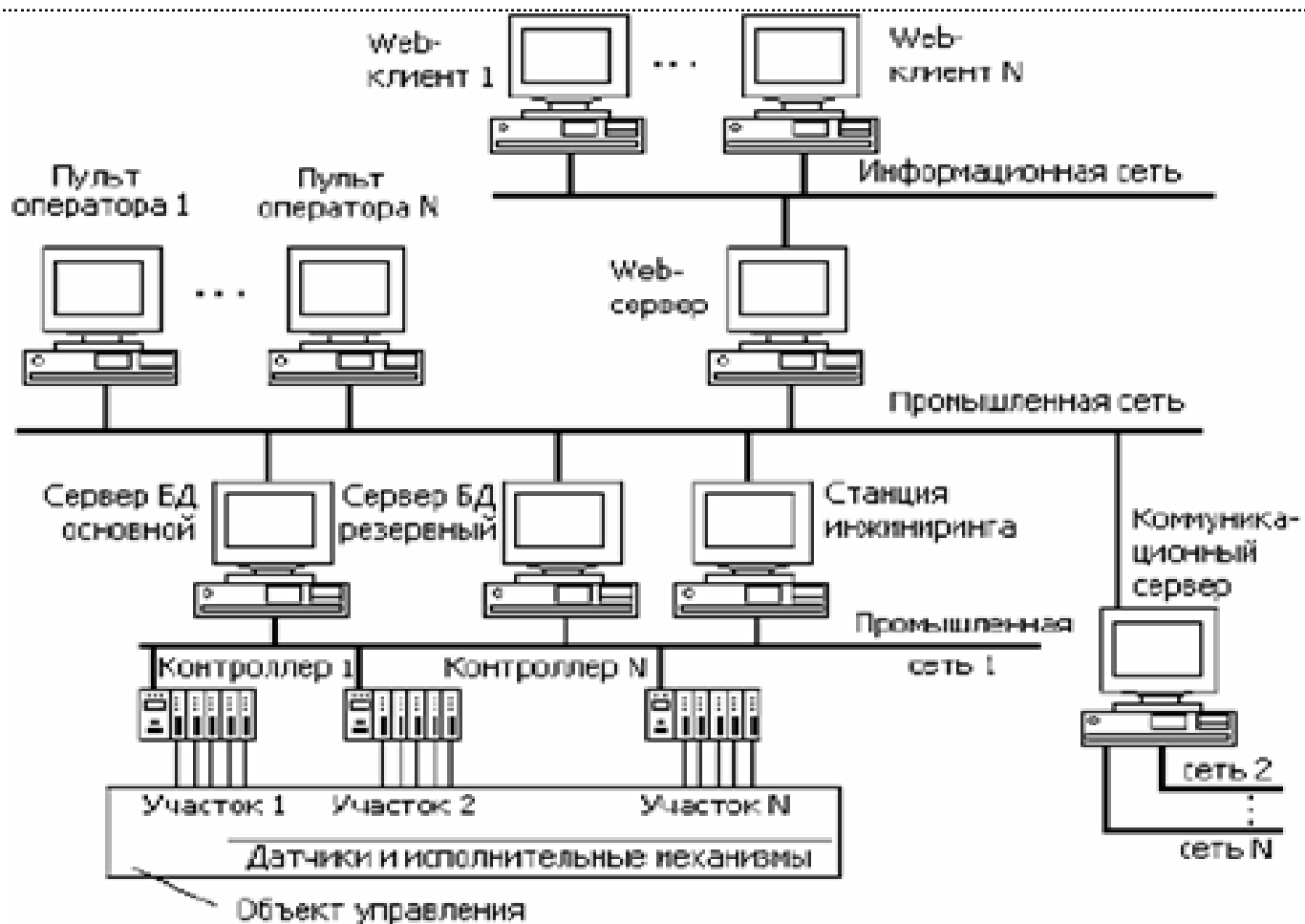
ДАСТУРИЙ ТАЪМИНОТНИНГ АСОСИЙ ГУРУҲЛАРИ

- операцион тизимлар – компьютер ва фойдаланувчи ўртасида мулоқот ўрнатиш, компьютер ва дастурларнинг ишини бошқариш учун мўлжалланган. Мисоллар: MS DOS, Windows XP, Vista, Linux, Unix, OS/2, Mac X ва бошқалар;
- матн муҳаррири – матн киритиш, таҳрирлаш, сақлаш ва очиш, чоп этиш, матнни форматлаш каби вазифаларни бажарадиган дастур. Мисоллар: MS Word, Lexicon, Wordpad, Notepad ва бошқалар;
- электрон жадваллар – жадвалга матн, рақам ва формула каби маълумотлар киритиб, улар устида ҳисоб-китоблар бажариш, диаграммалар яратиш имконини берадиган дастур. Мисоллар: Lotus, MS Excel ва бошқалар;
- маълумотлар базасини бошқариш тизими – маълумотларни махсус жадвалларга киритиб, тартибга солиш, керакли маълумотни излаш, маълумотлар омборини яратиш каби вазифаларни бажарадиган дастур. Мисоллар: MS Access, Foxpro, Clipper, Paradox, Oracle ва бошқалар;
- тақдимот яратиш – ҳар хил мавзуда намойиш ва тақдимот яратиш, уларга жадвал, расм, аудио, видео ва мультимедиа маълумотларни татбиқ этиш имкониятларига эга бўлган дастур. Мисоллар: MS PowerPoint, Macromedia Director ва ҳоказо. ва ҳ.к;
- график муҳаррир – растрли, векторли ва бошқа графикани яратиш, таҳрирлаш, ишлов бериш каби вазифаларни бажарадиган дастур. Мисоллар: Adobe Photoshop, Corel Draw, Macromedia Flash, Macromedia FreeHand, Adobe Illustrator, 3D Studio Max, Maya ва ҳ.к;
- мультимедиа дастурлари – мазкур дастурлар тоифаси ҳар хил турдаги мультимедиа ахборот файлларини (аудио, видео ва ҳоказо) очиш ва ижро этиш ва улар устида бошқа ҳар хил амалларни бажариш имкониятини яратувчи дастурлар. Мисоллар: Sound Forge, Cakewalk, WinAmp, Windows Media Player, DivX Player ва бошқалар;
- антивирус – компьютерни вируслардан сақлаш ва ҳимоялаш, қаттиқ диск, медиа воситаларни вирусга текшириш ва даволаш вазифаларини бажарадиган дастур. Мисоллар: Norton Antivirus, NOD32, McAfee Antivirus, Panda Antivirus, DrWeb, Антивирус Касперского ва бошқалар;
- архиватор – компьютердаги маълумотларни ихчам шаклга келтириш, яъни архивга жойлаштириш, чиқариш, янгилаш каби вазифаларни бажарадиган дастур. Мисоллар: ARJ, RAR, ZIP, WinZip, WinAce, WinRAR ва ҳ.к.;
- файл менежерлари – файл ва каталоглар (каталог, директорий) устида бир қатор амалларни бажариш имконини беради – яратиш, ўчириш, нусха кўчириш, номини ўзгартириш, кўчириш, таҳрирлаш, очиб кўриш ва ҳоказо. Мисоллар: Norton Commander, Dos Navigator, Far Manager, Volkov Commander, Total Commander ва бошқалар.
- Бундан ташқари ўйин, сканердан маълумот киритиш, маълумотнома ва энциклопедия, ўқитиш, таржимон ва бошқа турдаги дастурлар тоифалари ҳам мавжуд.

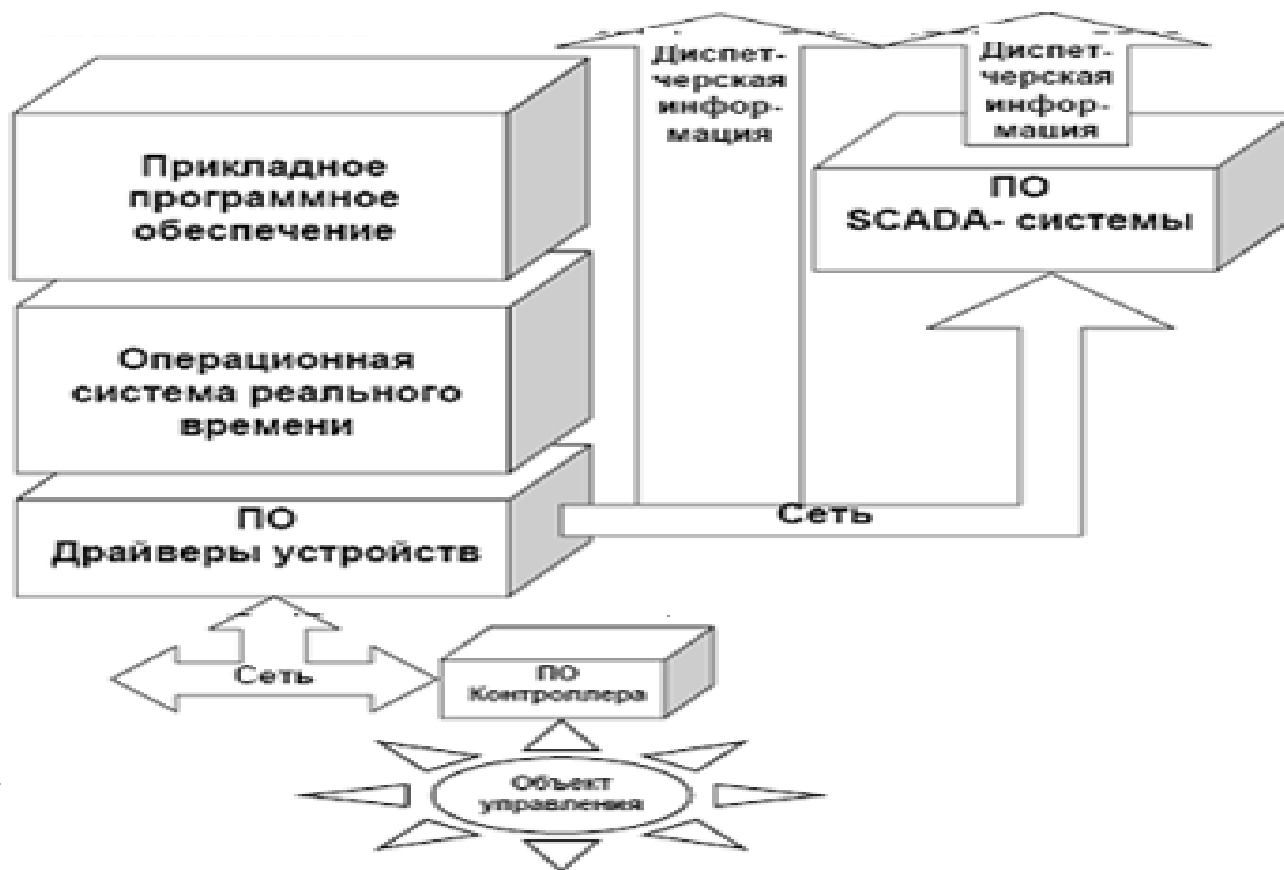
Простейшая структура САУ ТП



Структура САУ ТП



Структура программного обеспечения



Прикладное программное обеспечение, принципы программирования

С точки зрения инженера, создающего комплекс средств САУ ТП, - прикладное программное обеспечение можно разбить на следующие важнейшие группы:

дополнение к операционной системе (драйверы и т.п.);

программы управления, снятия (получения), передачи данных, обработки данных, планирования и т.п., то есть прикладные вычислительные задачи;

программное обеспечение локальных регуляторов. Эта часть программного обеспечения часто создаётся для специализированных микроконтроллеров и поэтому имеет свои особенности.

Важно также учитывать то обстоятельство, что для создания этих разнородных частей прикладного программного обеспечения используются совершенно разные методы программирования. Наиболее традиционной частью являются прикладные вычислительные задачи. Решать эти задачи стремятся традиционными методами и для этого стараются использовать программирование на языках высокого уровня, не упуская при этом из виду тот факт, что работа программы должна вестись в реальном времени. Обычно удаётся здесь обойтись программированием на языке C, C++, Pascal, привлекая для этого (по возможности быстрогодействия) интегрированные среды типа Visual C, Builder или Delphi.

Если это удаётся, то имеется возможность создания мощного современного программного обеспечения, удовлетворяющего всем требованиям к интерфейсам пользователя.

Непреодолимым барьером здесь может оказаться отсутствие требуемых средств разработки ПО для конкретной ОС РВ. В настоящее время для всех популярных ОС РВ имеются, по крайней мере, компиляторы языка C. Это существенно облегчает работу программиста.

ОСНОВНЫЕ ПОНЯТИЯ И ОПРЕДЕЛЕНИЯ

Информационная система (ИС) - организационно упорядоченная совокупность документов (массивов документов) и информационных технологий, в том числе с использованием средств вычислительной техники и связи, реализующих информационные процессы. Информационные системы предназначены для хранения, обработки, поиска, распространения, передачи и представления информации.

Автоматизированная (информационная) система (АС) - совокупность программных и аппаратных средств, предназначенных для хранения и/или управления данными и информацией и производства вычислений и управляемая человеком-оператором (в этом главное отличие автоматизированной системы от автоматической).

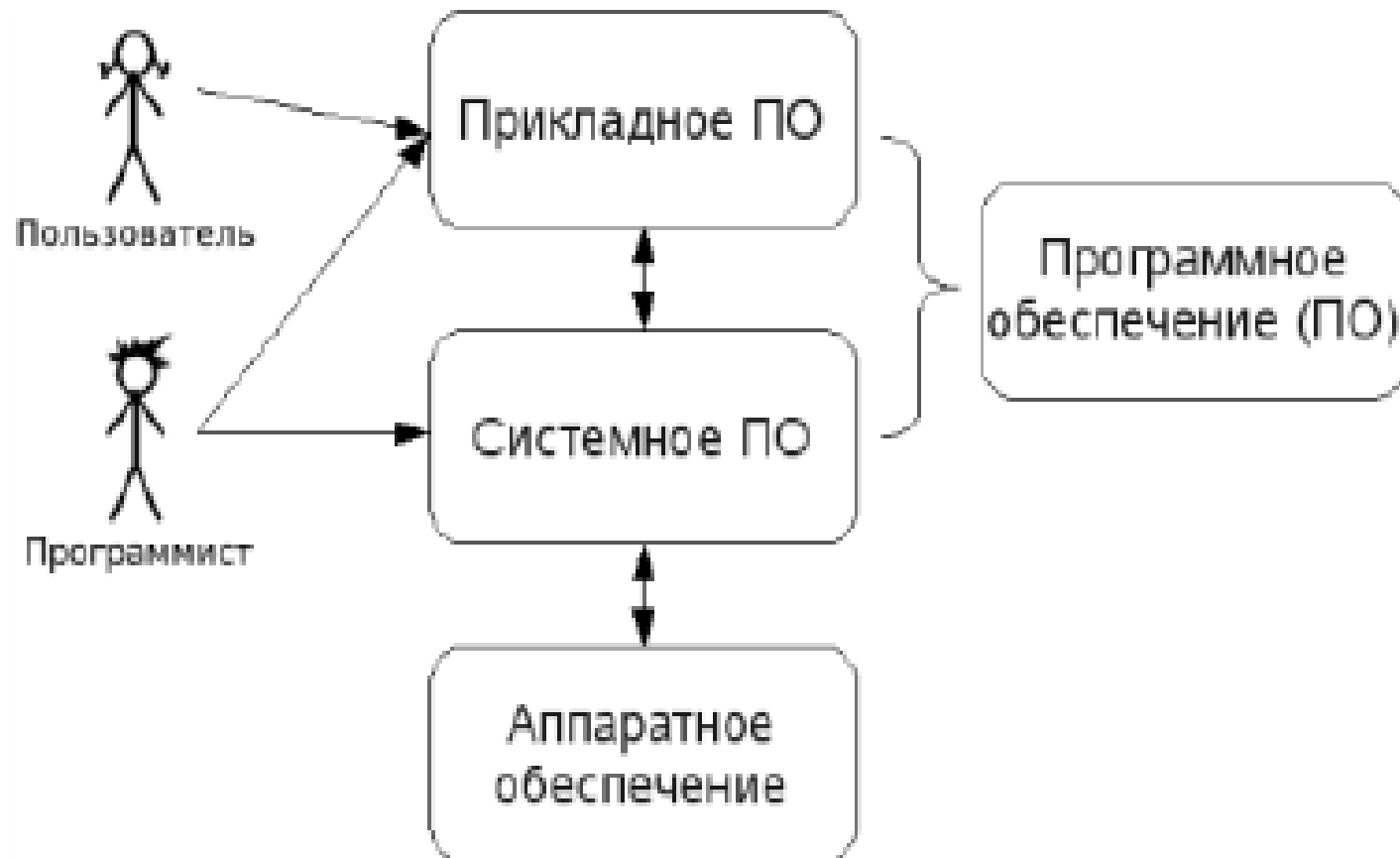
Аппаратное обеспечение ИС - комплекс электронных, электрических и механических устройств, входящих в состав информационной системы или сети.

Многоуровневое представление ИС - модель представления информационной системы в виде совокупности взаимосвязанных уровней, разделенных по функциональному назначению.

Программист - специалист, занимающийся разработкой и проверкой программ. Различают системных и прикладных программистов.

Пользователь - человек, принимающий участие в управлении объектами и системами некоторой предметной области и являющийся составным элементом автоматизированной системы.

МНОГОУРОВНЕВОЕ ПРЕДСТАВЛЕНИЕ ИС



ОСНОВНЫЕ ПОНЯТИЯ И ОПРЕДЕЛЕНИЯ

Рассмотрим несколько основных понятий и определений, которые помогут построить четкую и стройную классификационную картину.

Программа – последовательность формализованных инструкций, представляющих алгоритм решения некоторой задачи и предназначенная для исполнения устройством управления вычислительной машины.

Программное обеспечение (ПО) – совокупность программ и данных, предназначенных для решения определенного круга задач и хранящиеся на носителях ЭВМ.

Прикладная программа, Приложение - Application program; Application

Прикладная программа – в широком смысле – программа или пакет прикладных программ, реализующие обработку данных в определенной области применения. Прикладная программа непосредственно выполняет функции, необходимые пользователю.

Прикладная программа – в узком смысле – программа, решающая проблему конечного пользователя.

Прикладное программное обеспечение – программное обеспечение, ориентированное на конечного пользователя и предназначенное для решения пользовательских задач.

Прикладное ПО состоит из:

- отдельных прикладных программ и пакетов прикладных программ, предназначенных для решения различных задач пользователей;
- автоматизированных систем, созданных на основе этих пакетов.

ОСНОВНЫЕ ЭТАПЫ РАЗВИТИЯ И СОВРЕМЕННОЕ СОСТОЯНИЕ

Современный пакет является сложной программной системой, включающей специализированные системные и языковые средства. В относительно короткой истории развития вычислительных ППП можно выделить 4 основных поколения (класса) пакетов. Каждый из этих классов характеризуется определенными особенностями входящих в состав ППП компонентов – входных языков, предметного и системного обеспечения.

Первое поколение. В качестве входных языков ППП первого поколения использовались универсальные языки программирования (Фортран, Алгол-60 и т.п.) или языки управления заданиями соответствующих операционных систем. Проблемная ориентация входных языков достигалась за счет соответствующей мнемоники в идентификаторах. Составление заданий на таком языке практически не отличалось от написания программ на алгоритмическом языке.

Второе поколение. Разработка ППП второго поколения осуществлялась уже с применением специализированных входных языков на базе универсальных языков программирования. Проблемная ориентация таких языков достигалась не только за счет использования определенной мнемоники, но также применением соответствующих языковых конструкций, которые упрощали формулировку задачи и делали ее более наглядной. Транслятор с такого языка представлял собой препроцессор (чаще всего макропроцессор) к транслятору соответствующего алгоритмического языка.

КЛАССИФИКАЦИЯ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

Любая классификация подразумевает выбор некоторого группировочного признака (или нескольких), на основании которого и производится отнесение объектов к тому или иному классу. Так, при классификации программного обеспечения по способу распространения можно выделить следующие категории (список не полный):

- Commercial Software - коммерческое (с ограниченными лицензией возможностями на использование), разрабатываемое для получения прибыли.
- Freeware - свободное ПО, распространяемое без ограничений на использование, модификацию и распространение.
- Shareware - условно-бесплатное ПО, с частичными ограничениями при работе в ознакомительном режиме (например, определенное количество запусков программы).
- Abandonware - «заброшенное» ПО, поддержка которого непосредственным разработчиком прекращена, но продолжается третьими лицами (например, партнерами или энтузиастами).
- Adware - ПО, в код которого включены рекламные материалы. Такое ПО распространяется бесплатно, но для отключения рекламных блоков необходима оплата.
- Careware - «благотворительное» ПО, оплату за которое разработчик (или распространитель) просит переводить на благотворительные нужды.

КЛАССИФИКАЦИЯ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

При классификации программного обеспечения **по назначению** в качестве критерия используют уровень представления ИС, на который ориентирована та или иная программа.

Соответственно **выделяют следующие классы ПО:**

Системное ПО – решает задачи общего управления и поддержания работоспособности системы в целом. К этому классу относят операционные системы, менеджеры загрузки, драйверы устройств, программные кодеки, утилиты и программные средства защиты информации;

Инструментальное ПО – включает средства разработки (трансляторы, отладчики, интегрированные среды, различные SDK и т.п.) и системы управления базами данных (СУБД);

Прикладное ПО – предназначено для решения прикладных задач конечными пользователями. Прикладное ПО является самым обширным классом программ, в рамках которого возможна дальнейшая классификация, например по предметным областям. В этом случае группировочным признаком является класс задач, решаемых программой.

ПРИМЕРЫ ПРИКЛАДНОГО ПО

Прикладное ПО - самый обширный класс программ, в рамках которого возможна дальнейшая классификация, например по предметным областям. В этом случае группировочным признаком является класс задач, решаемых программой. Приведем несколько примеров:

- Офисные приложения - предназначены для автоматизации офисной деятельности (текстовые редакторы и процессоры, электронные таблицы, редакторы презентаций и т.п.)
- Корпоративные информационные системы - бухгалтерские программы, системы корпоративного управления, системы управления проектами (Project Management), инструменты автоматизации документооборота (EDM-системы) и управления архивами документов (DWM-системы)
- Системы проектирования и производства - системы автоматизированного проектирования (САПР, CAD/CAM-системы), системы управления технологическими (SCADA) и производственными (MES) процессами
- Научное ПО - системы математического и статистического расчета, анализа и моделирования
- Геоинформационные системы (ГИС)
- Системы поддержки принятия решений (СППР)
- Клиенты доступа к сетевым сервисам (электронная почта, веб-браузеры, передача сообщений, чат-каналы, клиенты файлообменных сетей и т.п.)
- Мультимедийное ПО - компьютерные игры, средства просмотра и редактирования аудио- и видеоинформации, графические редакторы и вьюеры, анимационные редакторы и т.п.

КЛАССИФИКАЦИЯ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

В зависимости от степени интеграции многочисленные прикладные программные средства можно классифицировать следующим образом:

- отдельные прикладные программы;
- библиотеки прикладных программ;
- пакеты прикладных программ;
- интегрированные программные системы.

Отдельная прикладная программа пишется, как правило, на некотором высокоуровневом языке программирования (Pascal, Basic и т.п.) и предназначается для решения конкретной прикладной задачи.

Библиотека представляет собой набор отдельных программ, каждая из которых решает некоторую прикладную задачу или выполняет определенные вспомогательные функции (управление памятью, обмен с внешними устройствами и т.п.). Условно их можно разделить на библиотеки общего назначения и специализированные библиотеки.

Пакет прикладных программ (ППП) – это комплекс взаимосвязанных программ, ориентированный на решение определенного класса (из определенной проблемной области) задач.

Интегрированная программная система – это комплекс программ, элементами которого являются различные пакеты и библиотеки программ.

ПОНЯТИЕ ПАКЕТА ПРИКЛАДНЫХ ПРОГРАММ

ППП – это комплекс взаимосвязанных программ для решения определенного класса задач из конкретной предметной области.

На текущем этапе развития информационных технологий именно ППП являются наиболее востребованным видом прикладного ПО. Это связано с упомянутыми ранее особенностями ППП. Рассмотрим их подробнее:

-ориентация на решение класса задач. Одной из главных особенностей является ориентация ППП не на отдельную задачу, а на некоторый класс задач, в том числе и специфичных, из определенной предметной области.

-наличие языковых средств. Другой особенностью ППП является наличие в его составе специализированных языковых средств, позволяющих расширить число задач, решаемых пакетом или адаптировать пакет под конкретные нужды. Поддерживаемые языки могут быть использованы для формализации исходной задачи, описания алгоритма решения и начальных данных, организации доступа к внешним источникам данных, разработки программных модулей, описания модели предметной области, управления процессом решения в диалоговом режиме и других целей.

единообразие работы с компонентами пакета. Еще одна особенность ППП состоит в наличии специальных системных средств, обеспечивавших унифицированную работу с компонентами. К их числу относятся специализированные банки данных, средства информационного обеспечения, средства взаимодействия пакета с операционной системой, типовой пользовательский интерфейс и т.п.

ОСНОВНЫЕ ЭТАПЫ РАЗВИТИЯ И СОВРЕМЕННОЕ СОСТОЯНИЕ

Современный пакет является сложной программной системой, включающей специализированные системные и языковые средства. В относительно короткой истории развития вычислительных ППП можно выделить 4 основных поколения (класса) пакетов. Каждый из этих классов характеризуется определенными особенностями входящих в состав ППП компонентов – входных языков, предметного и системного обеспечения.

Первое поколение. В качестве входных языков ППП первого поколения использовались универсальные языки программирования (Фортран, Алгол-60 и т.п.) или языки управления заданиями соответствующих операционных систем. Проблемная ориентация входных языков достигалась за счет соответствующей мнемоники в идентификаторах. Составление заданий на таком языке практически не отличалось от написания программ на алгоритмическом языке.

Второе поколение. Разработка ППП второго поколения осуществлялась уже с применением специализированных входных языков на базе универсальных языков программирования. Проблемная ориентация таких языков достигалась не только за счет использования определенной мнемоники, но также применением соответствующих языковых конструкций, которые упрощали формулировку задачи и делали ее более наглядной. Транслятор с такого языка представлял собой препроцессор (чаще всего макропроцессор) к транслятору соответствующего алгоритмического языка.

ОСНОВНЫЕ ЭТАПЫ РАЗВИТИЯ И СОВРЕМЕННОЕ СОСТОЯНИЕ

Третье поколение. Третий этап развития ППП характеризуется появлением самостоятельных входных языков, ориентированных на пользователей-непрограммистов. Особое внимание в таких ППП уделяется системным компонентам, обеспечивающим простоту и удобство. Это достигается главным образом за счет специализации входных языков и включения в состав пакета средств автоматизированного планирования вычислений.

Четвертое поколение. Четвертый этап характеризуется созданием ППП, эксплуатируемых в интерактивном режиме работы. Основным преимуществом диалогового взаимодействия с ЭВМ является возможность активной обратной связи с пользователем в процессе постановки задачи, ее решения и анализа полученных результатов.

Большое внимание в настоящее время уделяется проблеме создания «интеллектуальных ППП». Такой пакет позволяет конечному пользователю лишь сформулировать свою задачу в содержательных терминах, не указывая алгоритма ее решения. Синтез решения и сборка целевой программы производятся автоматически. Такой способ решения иногда называют концептуальным программированием, характерными особенностями которого является программирование в терминах предметной области использование ЭВМ уже на этапе постановки задач, автоматический синтез программ решения задачи, накопление знаний о решаемых задачах в базе знаний.

ПЕРСПЕКТИВЫ РАЗВИТИЯ ПРИКЛАДНОГО ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

На сегодняшний день в качестве основных факторов, влияющих на функциональность ППП и сложность их разработки, можно отметить следующие:

- рост производительности персональных компьютеров;
- расширение классов решаемых задач;
- увеличение общего числа пользователей;
- значительное количество ранее созданного (наследованного) ПО;
- развитие Интернет и корпоративных сетей.

Разработка приложений с учетом этих факторов привела к появлению прикладных пакетов и интегрированных сред, которые по своим характеристикам выходят за рамки ППП четвертого поколения. К отличительным чертам ПО нового поколения следующие:

- интеграция компонентов прикладного пакета не только с приложениями пакета, но и с окружением;
- широкое использование отраслевых стандартов;
- использование инфраструктуры Интернет;
- платформено-независимость.

Перспективным направлением в развитии ППП является использование унифицированных форматов документов на основе открытых стандартов. Открытый стандарт – общедоступная спецификация свободная от лицензионных ограничений при использовании. Использование открытых форматов в ППП позволяет гарантировать возможность доступа к данным из любого совместимого приложения без оглядки на лицензионные права и технические спецификации.

СТРУКТУРА И КОМПОНЕНТЫ ППП

Несмотря на большое разнообразие конкретных пакетных разработок, можно выделить следующие основные компоненты ППП:

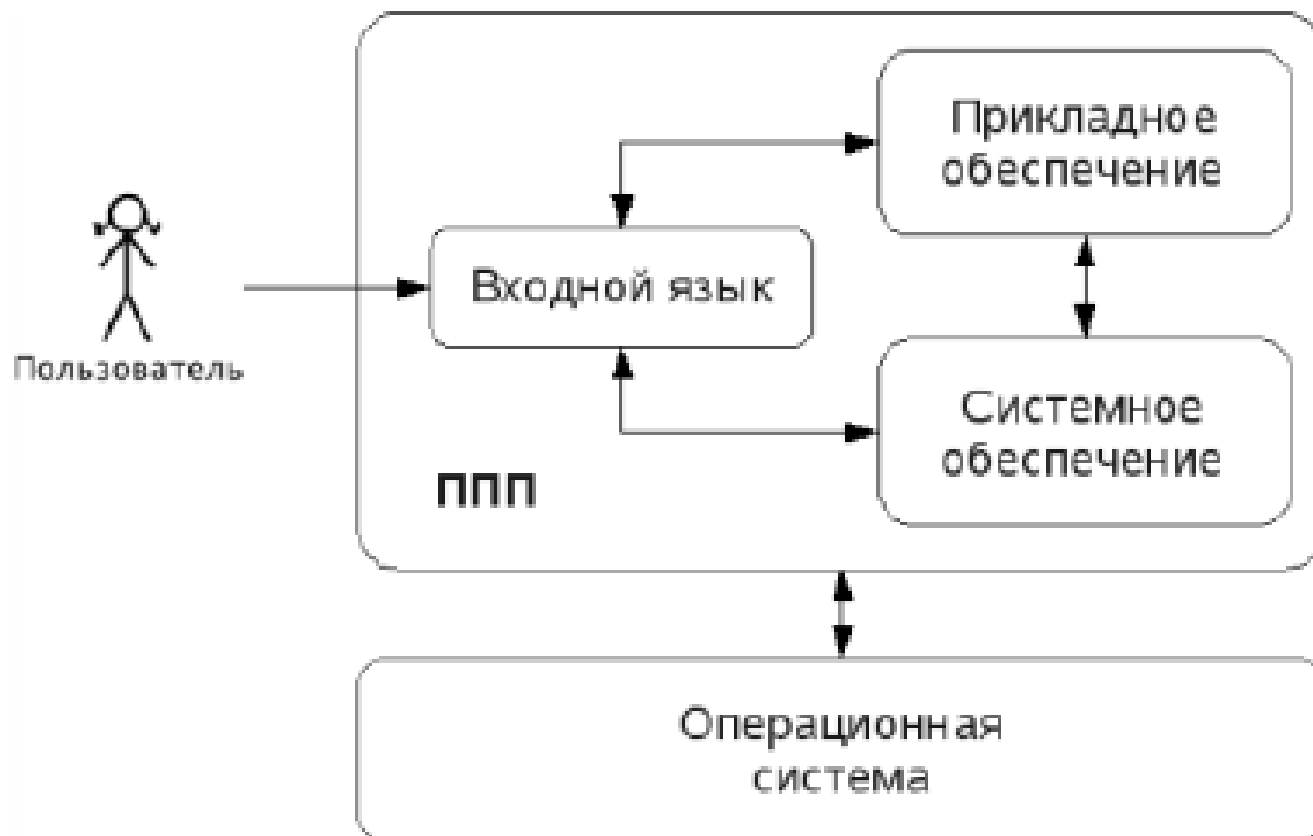
- входные языки;
- предметное обеспечение;
- системное обеспечение.

Важно отметить, что такое разбиение на составные элементы отражает в первую очередь функции, выполняемые программами ППП, а не структуру самих программ, которая зависит от индивидуальных особенностей конкретного пакета. В разных пакетах указанные компоненты могут быть развиты в различной степени или вовсе отсутствовать. Однако наиболее развитые ППП, как правило, обладают всеми этими компонентами, каждый из которых может иметь довольно сложную структуру.

В многочисленных работах, посвященных пакетной проблематике, из-за неустоявшейся терминологии нередко используются другие названия составных элементов ППП. Например, входной язык называют также языком заданий или языком управления. Для обозначения предметного обеспечения применяются термины «функциональное наполнение», «функциональная подсистема» или «тело пакета». Системное обеспечение часто называют системным наполнением, организующей или управляющей программой, а также процессором пакета.

Рассмотрим функции каждого из компонентов ППП.

СТРУКТУРА И ОСНОВНЫЕ КОМПОНЕНТЫ ППП



ВХОДНЫЕ ЯЗЫКИ

Входные языки представляют собой средство общения пользователя с пакетом. Как отмечалось ранее, развитый пакет может обладать несколькими входными языками, предназначенными для выполнения различных функций ориентированными на различные типы пользователей. Можно выделить следующие основные типы пользователей ППП:

разработчик ППП, осуществляющий его модификацию и развитие с учетом изменения круга пользователей, класса решаемых задач (появление новых типов задач, развитие численных методов, модификация форм проведения работ и т. д.), а также состава аппаратного и программного обеспечения ЭВМ;

ответственный за сопровождение, в функции которого входит поддержание пакета в работоспособном состоянии в условиях конкретной вычислительной системы (обеспечение сохранности программ и массивов данных, своевременное дублирование информационных файлов, выявление ошибок в программах пакета);

администратор, отвечающий за организацию доступа пользователей к пакету, содержимое базы данных, защиту информации от несанкционированного доступа;

конечный пользователь, применяющий пакет для решения конкретных прикладных задач.

Входные языки отражают объем и качество предоставляемых пакетом средств, а также удобство их использования. Таким образом, с точки зрения конечного пользователя именно входной язык является основным показателем возможностей ППП.

В качестве входных языков могут использоваться как универсальные, так и специализированные языки программирования. Например, в качестве входного языка разработчика ППП для написания прикладных и системных программ пакета обычно используется тот или иной универсальный язык программирования (Фортран, Паскаль). В то же время входной язык конечного пользователя в развитом пакете, как правило, является языком качественно более высокого уровня по сравнению с универсальными языками. Изобразительные средства такого языка учитывают особенности задач предметной области и специфику пользователей. Подобные языки называют проблемно-ориентированными, или предметно-ориентированными.

ПРЕДМЕТНОЕ ОБЕСПЕЧЕНИЕ

Конкретная прикладная деятельность характеризуется двумя факторами:

- 1) классом решаемых задач и используемых для этих целей методов,
- 2) дисциплиной работы, т.е. совокупностью правил, соглашений и технологических приемов, принятых при разработке, отладке, эксплуатации программ.

Предметное обеспечение представляет собой компонент пакета, отражающий особенности первого из этих факторов, т. е. особенности конкретной предметной области. Предметное обеспечение включает:

- программные модули, реализующие алгоритмы (или их отдельные фрагменты) решения прикладных задач;
- средства сборки программ из отдельных модулей.

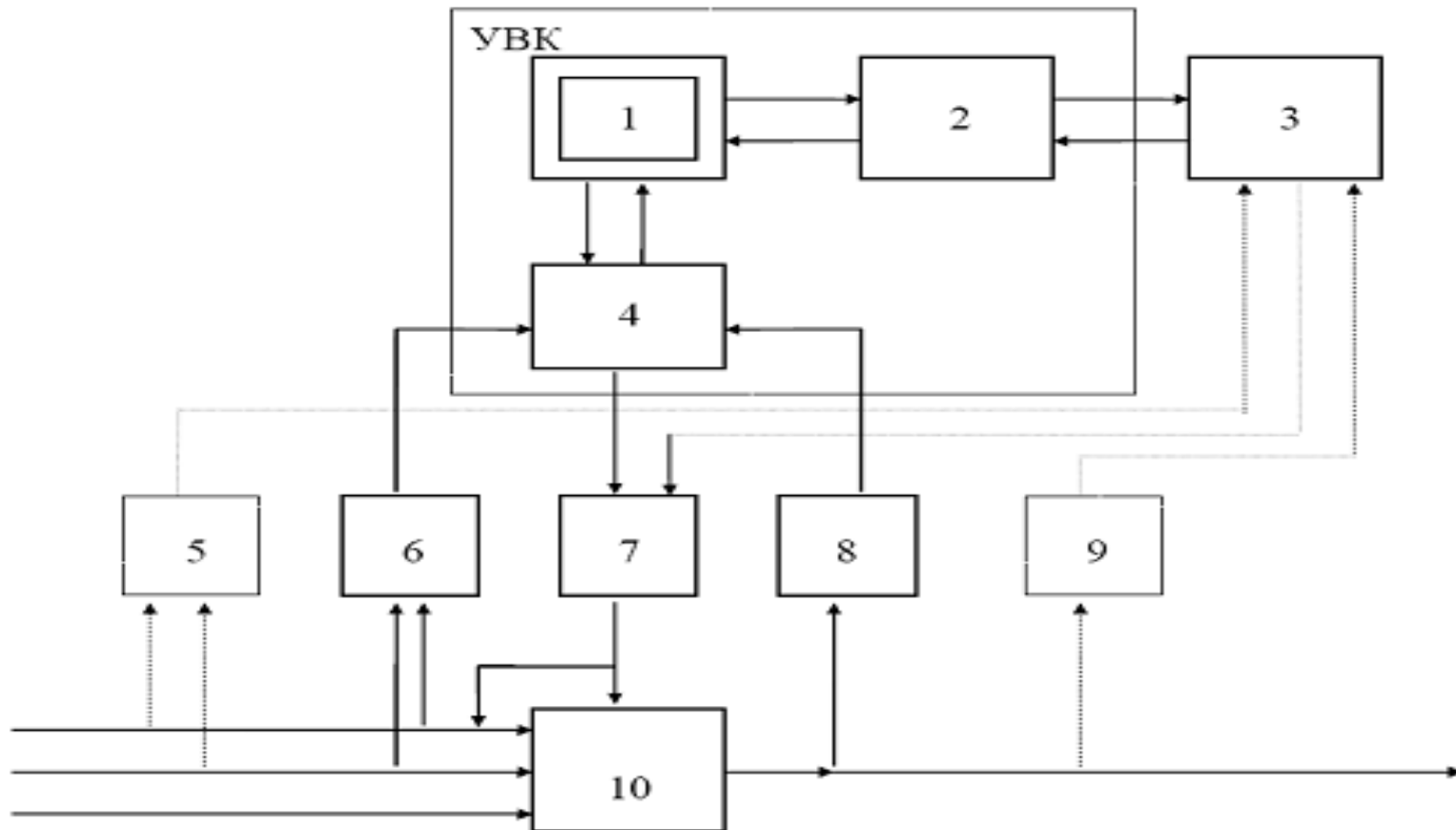
Определение состава библиотеки модулей и форм их взаимодействия между собой является одной из наиболее трудоемких задач при построении ППП. Ее решение предполагает проведение тщательного и квалифицированного модульного анализа используемых алгоритмов. Удачно проведенный модульный анализ в значительной степени влияет на полноту охвата предметной области, а также на возможность расширения класса решаемых задач. Таким образом, выделение модулей существенно зависит от специфики задач и используемых алгоритмов и, по сути дела, отражает принятый в пакет способ сборки программ.

СИСТЕМНОЕ ОБЕСПЕЧЕНИЕ

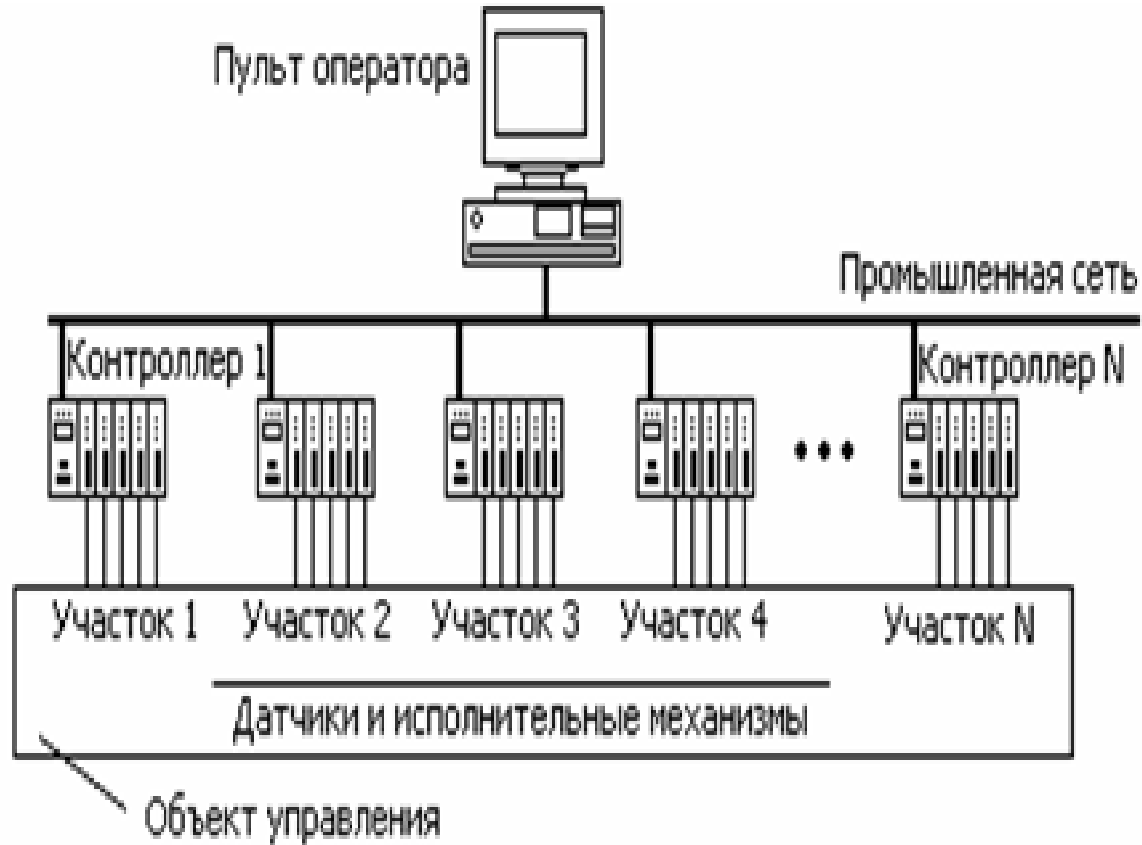
Системное обеспечение представляет собой совокупность системных средств (программы, файлы, таблицы и т. д.), обеспечивающих определенную дисциплину работы пользователя при решении прикладных задач. По своей роли в составе ППП и выполняемым функциям системное обеспечение по существу является специализированной операционной системой, определяющей операционное окружение пакета. Несмотря на многообразие способов реализации системного обеспечения в рамках конкретных пакетных разработок, можно выделить его следующие **основные компоненты**:

- монитор, управляющий процессом решения и взаимодействием всех компонентов ППП;
- трансляторы с входных языков;
- средства работы с данными;
- средства информационного обеспечения, реализующие выдачу разнообразной справочной информации как по запросам пользователей (о структуре и возможностях ППП, о допущенных ошибках и т.д.), так и по запросам различных компонентов пакета (например, сведения о свойствах модулей предметного обеспечения, необходимые планировщику вычислений);
- различные служебные программы, в том числе реализующие взаимодействие пакета с операционной системой (работа с внешней памятью, средства ввода/вывода, драйверы специализированных, внешних устройств и др.).

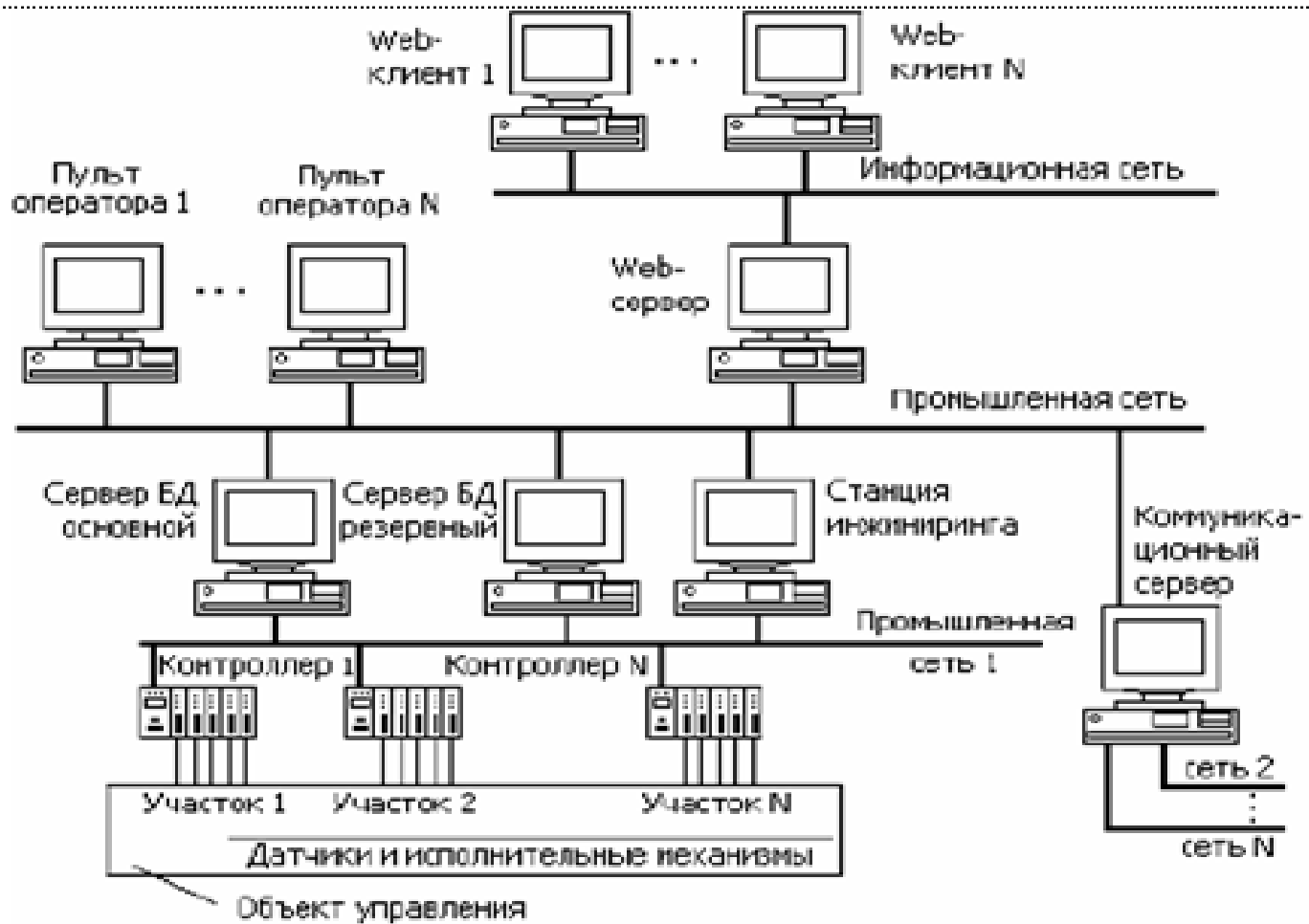
Структура САУ ТП



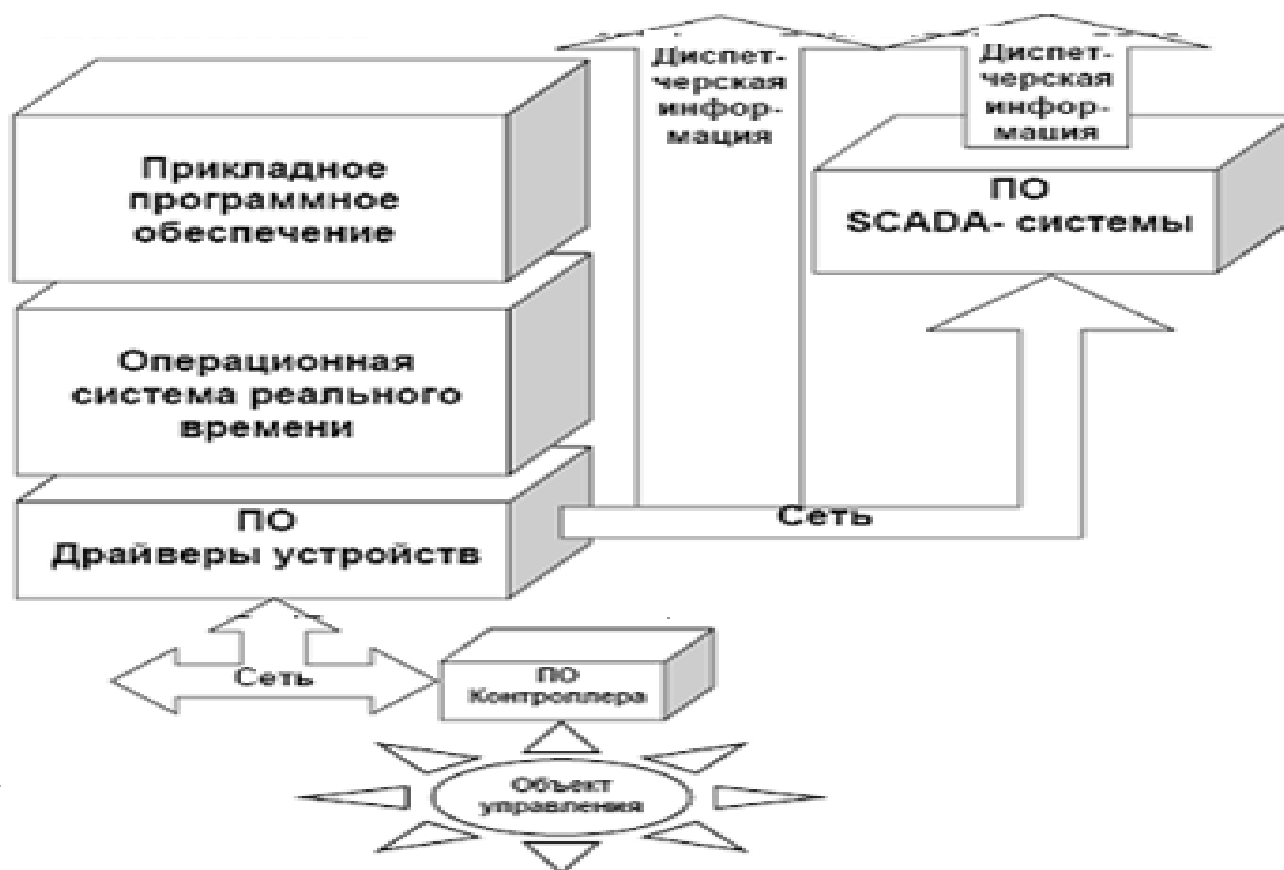
Простейшая структура САУ ТП



Структура САУ ТП



Структура программного обеспечения



Прикладное программное обеспечение, принципы программирования

С точки зрения инженера, создающего комплекс средств САУ ТП, - прикладное программное обеспечение можно разбить на следующие важнейшие группы:

дополнение к операционной системе (драйверы и т.п.);

программы управления, снятия (получения), передачи данных, обработки данных, планирования и т.п., то есть прикладные вычислительные задачи;

программное обеспечение локальных регуляторов. Эта часть программного обеспечения часто создаётся для специализированных микроконтроллеров и поэтому имеет свои особенности.

Важно также учитывать то обстоятельство, что для создания этих разнородных частей прикладного программного обеспечения используются совершенно разные методы программирования. Наиболее традиционной частью являются прикладные вычислительные задачи. Решать эти задачи стремятся традиционными методами и для этого стараются использовать программирование на языках высокого уровня, не упуская при этом из виду тот факт, что работа программы должна вестись в реальном времени. Обычно удаётся здесь обойтись программированием на языке C, C++, Pascal, привлекая для этого (по возможности быстрогодействия) интегрированные среды типа Visual C, Builder или Delphi.

Если это удаётся, то имеется возможность создания мощного современного программного обеспечения, удовлетворяющего всем требованиям к интерфейсам пользователя.

Непреодолимым барьером здесь может оказаться отсутствие требуемых средств разработки ПО для конкретной ОС РВ. В настоящее время для всех популярных ОС РВ имеются, по крайней мере, компиляторы языка C. Это существенно облегчает работу программиста.

РАЗНОВИДНОСТИ ППП

Пакеты прикладных программ (ППП) — это специальным образом организованные программные комплексы, рассчитанные на общее применение в определенной проблемной области и дополненные соответствующей технической документацией.

В зависимости от характера решаемых задач различают следующие разновидности ППП:

- 1) пакеты для решения типовых инженерных, планово-экономических, общенаучных задач;
- 2) пакеты системных программ;
- 3) пакеты для обеспечения систем автоматизированного проектирования и систем автоматизации научных исследований;
- 4) пакеты педагогических программных средств и другие.

ВИДЫ ППП

Выделяются следующие виды ППП:

- **проблемно-ориентированные**. Используются для тех проблемных областей, в которых возможна типизация функций управления, структур данных и алгоритмов обработки. Например, это ППП автоматизации бухучета, финансовой деятельности, управления персоналом и т.д.;
- **автоматизации проектирования** (или САПР). Используются в работе конструкторов и технологов, связанных с разработкой чертежей, схем, диаграмм;
- **общего назначения**. Поддерживают компьютерные технологии конечных пользователей и включают текстовые и табличные процессоры, графические редакторы, системы управления базами данных (СУБД);
- **офисные**. Обеспечивают организационное управление деятельностью офиса. Включают органайзеры (записные и телефонные книжки, календари, презентации и т.д.), средства распознавания текста;
- **настольные издательские системы** – более функционально мощные текстовые процессоры;
- **системы искусственного интеллекта**. Используют в работе некоторые принципы обработки информации, свойственные человеку. Включают информационные системы, поддерживающие диалог на естественном языке; экспертные системы, позволяющие давать рекомендации пользователю в различных ситуациях; интеллектуальные пакеты прикладных программ, позволяющие решать прикладные задачи без программирования.

Пакеты прикладных программ

**Пакеты
прикладных
программ
(ППП)**

Проблемно-ориентированные ППП

Методо-ориентированные ППП

ППП общего назначения

Интеллектуальные системы

Программы мультимедиа

Проблемно-ориентированные ППП

- **Средства автоматизации управления :**
 - ППП автоматизированного бухгалтерского учета (“1С:Бухгалтерия”);
 - ППП финансовой деятельности;
 - ППП управления персоналом (кадровый учет);
 - ППП управления производством и т.п.
- **Особенности**
 - Т.к. велики объемы хранимой информации, то высоки требования
 - к оперативности обработки данных,
 - к средствам администрирования данных БД.
 - Дружественный интерфейс для конечных пользователей.

Проблемно-ориентированные ППП

- **Средства автоматизированного проектирования**
- САПР – системы автоматизированного проектирования
- английская аббревиатура **CAD** – Computer-Aided Design

- ***Особенности***
 - высокие требования к технической части системы обработки данных,
 - наличие библиотек встроенных функций, объектов, интерфейсов с графическими системами и базами данных.

- ***Примеры***
 - Программа AutoCad фирмы AutoDesk является лидером среди инженерных графических пакетов.
 - P-CAD и OrCAD – предназначены для проектирования радиоэлектронных изделий.

Методо-ориентированные ППП

- **Программные продукты, обеспечивающие независимо от предметной области и функций информационных систем,**
 - *математические,*
 - *статистические*
 - *и другие методы решения задач.*
- **Наиболее распространены методы:**
 - *математического программирования,*
 - *решения дифференциальных уравнений,*
 - *имитационного моделирования,*
 - *исследования операций.*
- **Примеры пакетов:**
 - *пакет для обработки статистической информации StatGraphics*
 - *математический пакет Mathematica, MathCad*

ППП общего назначения

- Содержит широкий перечень программных продуктов, поддерживающих информационные технологии конечных пользователей.
- **Представители данного класса программных продуктов:**
 - *Текстовые процессоры (редакторы)*
 - *Графические пакеты*
 - *Электронные таблицы*
 - *СУБД*
 - *Интегрированные пакеты*
 - *Игры и т.п.*

Системы искусственного интеллекта

→ Программные продукты данного класса реализует *отдельные функции интеллектуальной деятельности человека.*

- **Основные компоненты систем ИИ:**

- база знаний,
- интеллектуальный интерфейс с пользователем
- программа формирования логических выводов.

- **Примеры:**

- экспертные системы MYCIN, Prospector, Лота
- программы–переводчики, средства проверки орфографии и распознавания текста (Stylus Lingvo Office, FineReader)
- шахматные программы

Программы мультимедиа

- Основное назначение программных продуктов мультимедиа – создание и использование *аудио- и видеоинформации* для расширения информационного пространства пользователя.
- **Основные сферы применения:**
 - *библиотечное информационное обслуживание,*
 - *прикладные обучающие системы,*
 - *развлекательные программы.*
- **Примеры:**
 - *базы данных компьютерных изображений произведений искусства,*
 - *библиотеки звуковых записей,*
 - *компьютерные игры,*
 - *библиотечные каталоги и фонды и т.п.*

Среды разработки программ

- В настоящее время бурно развиваются ***технологии создания программных продуктов.***
- **Это обусловлено:**
 - переходом на промышленную технологию производства программ,
 - стремлением к
 - *сокращению сроков разработки программ,*
 - *сокращению трудовых и материальных затрат на производство и эксплуатацию программ,*
 - *обеспечению гарантированного уровня качества программ.*
- ***Программотехника*** (software engineering) – технология разработки, отладки, верификации и внедрения программного обеспечения.

Среды разработки программ



РАЗНОВИДНОСТИ ППП

Пакеты прикладных программ (ППП) — это специальным образом организованные программные комплексы, рассчитанные на общее применение в определенной проблемной области и дополненные соответствующей технической документацией.

В зависимости от характера решаемых задач различают следующие разновидности ППП:

- 1) пакеты для решения типовых инженерных, планово-экономических, общенаучных задач;
- 2) пакеты системных программ;
- 3) пакеты для обеспечения систем автоматизированного проектирования и систем автоматизации научных исследований;
- 4) пакеты педагогических программных средств и другие.

ВИДЫ ППП

Выделяются следующие виды ППП:

- **проблемно-ориентированные**. Используются для тех проблемных областей, в которых возможна типизация функций управления, структур данных и алгоритмов обработки. Например, это ППП автоматизации бухучета, финансовой деятельности, управления персоналом и т.д.;
- **автоматизации проектирования** (или САПР). Используются в работе конструкторов и технологов, связанных с разработкой чертежей, схем, диаграмм;
- **общего назначения**. Поддерживают компьютерные технологии конечных пользователей и включают текстовые и табличные процессоры, графические редакторы, системы управления базами данных (СУБД);
- **офисные**. Обеспечивают организационное управление деятельностью офиса. Включают органайзеры (записные и телефонные книжки, календари, презентации и т.д.), средства распознавания текста;
- **настольные издательские системы** – более функционально мощные текстовые процессоры;
- **системы искусственного интеллекта**. Используют в работе некоторые принципы обработки информации, свойственные человеку. Включают информационные системы, поддерживающие диалог на естественном языке; экспертные системы, позволяющие давать рекомендации пользователю в различных ситуациях; интеллектуальные пакеты прикладных программ, позволяющие решать прикладные задачи без программирования.

ПРОБЛЕМНО-ОРИЕНТИРОВАННЫЕ ПАКЕТЫ

Используются в тех предметных областях, для которых возможна типизация функций управления, структур данных и алгоритмов обработки. Типичным примером является серия программ 1С:, позволяющая автоматизировать решение задач управления предприятием, например, 1С:Бухгалтерия, 1С: Предприятие, 1С: Кадры и т.д. К пакетам этого класса относятся и программы, реализующие дистанционное обучение, например пакет SunRay_BookOffice для создания и работы с электронным учебником.

СИСТЕМЫ АВТОМАТИЗАЦИИ ПРОЕКТИРОВАНИЯ (САПР)

Эти пакеты используются в работе конструкторов и технологов, связанных с разработкой чертежей, схем, диаграмм, т.е. с обработкой графических изображений.

Реализуют функции:

- коллективная работа в сети;
 - экспорт - импорт файлов различных форматов;
 - масштабирование объектов;
 - группировка объектов, передвижение, растяжка, поворот, разрезание, изменение размеров, работа со слоями;
 - перерисовка;
 - управление файлами;
 - использование чертежных инструментов, позволяющих рисовать кривые, эллипсы, линии произвольной формы, многоугольники и т.п.;
 - работа с цветом;
 - автоматизация отдельных процедур с использованием встроенного макроязыка.
- Примерами пакетов этого класса являются: AutoCAD (AutoDesk), DesignCAD, Grafic CAD Professional, DrawBase, Microstation, TurboCAD.

ПАКЕТЫ ОБЩЕГО НАЗНАЧЕНИЯ

Поддерживают компьютерные технологии конечных пользователей и включают текстовые и табличные процессоры (редакторы), графические редакторы, системы управления базами данных (СУБД), пакеты программ мультимедиа, пакеты демонстрационной графики.

- Текстовые процессоры (редакторы) позволяют готовить текстовые документы, которые могут включать и таблицы, и рисунки, и диаграммы. Примером пакетов этого класса являются MS Word, Блокнот, WordPad. Перечень выполняемых функций, например MS Word, очень широк и изучается студентами в лабораторном практикуме по информатике.

- Табличные процессоры (типичный пример - MS Excel) позволяют обрабатывать большие объемы числовой информации (не исключая при этом обычную символьную), формируя из данных таблицы. Можно сказать, что это очень мощные калькуляторы, хранящие в своей памяти огромные числовые массивы и позволяющие выполнять над ними различные арифметические и логические операции, формировать диаграммы и делать множество других операций, полезных для решения различных задач пользователя. Аналогично пакету MS Word, табличный процессор MS Excel изучается в лабораторном практикуме по информатике.

ПАКЕТЫ ОБЩЕГО НАЗНАЧЕНИЯ (продолжение)

Графические редакторы позволяют генерировать различные изобразительные объекты. Они делятся на 2 класса - растровой и векторной графики - в зависимости от того, какое внутреннее представление этих объектов в них поддерживается.

- **Редакторы растровой графики** используются для работы с фотографиями. Они кодируют фотоизображения в цифровую форму и позволяют выполнять над ними различные редактирующие операции (выделение фрагментов, перемещение, вырезание, копирование и т.д.). Примерами редакторов этого класса являются: Adobe Photoshop, Aldus Photo Styler, Picture Publisher, Photo Works Plus.

- **Редакторы векторной графики** используются для профессиональной работы, связанной с технической и художественной иллюстрацией с последующей цветной печатью. Они занимают промежуточное место между САПР и настольными издательскими системами. Включают инструментарий для создания графического объекта; средства манипулирования объектами; средства обработки текста в части оформления и модификации параграфов, работы со шрифтами; средства вывода на печать и настройки цвета. Примерами графических редакторов этого класса являются Corel Draw, Adobe Illustrator, Aldus Free Hand, Professional Draw.

- **Системы управления базами данных (СУБД)** используются для автоматизации процедур создания, хранения и извлечения электронных данных. Различаются способом организации данных, форматом, языком формирования запросов на операции с данными. Типичными примерами являются MS Access, Oracle, Paradox.

- **Пакеты программ мультимедиа** используются для отображения (воспроизведения) и обработки аудио- и видеоинформации. Включают, в частности, пакеты Director for Windows, Multimedia Viewer Kit, NEC MultiSpin.

- **Пакеты демонстрационной графики** - это конструкторы графических образов деловой информации, призванные в наглядной и динамической форме представлять результаты некоторых аналитических исследований. последовательность работы с такими пакетами включает шаги: разработка общего плана представления, выбор шаблона для оформления элементов, формирование и импорт элементов (текст, графика, таблицы, диаграммы, звуковые эффекты, видеоклипы). Примеры таких пакетов: Power Point, Harvard Graphics, WordPerfect Presentations.

ОФИСНЫЕ ПАКЕТЫ

Обеспечивают организационное управление деятельностью офиса.

Включают:

•**Органайзеры** используются для автоматизации процедур планирования использования различных ресурсов (времени, денег, материалов) как отдельного человека, так и всей фирмы или ее подразделений. Существуют 2 вида пакетов этого класса:

•**1) органайзеры для управления проектами.** используются для сетевого планирования и управления проектами. Позволяют спланировать проект любой величины и сложности, эффективно распределить людские, финансовые и материальные ресурсы, составить оптимальный график работ и проконтролировать его исполнение. К ним относятся Time Line, MS Project, CA - Super Project;

•**2) органайзеры для организации деятельности отдельного человека.** Это электронные секретари для эффективного управления деловыми контактами. Включают, в частности, Lotus Organizer, ACTI.

•**Программы для распознавания символов** используются для перевода графических изображений букв и цифр в ASCII-коды этих символов. Применяются в сканерах. Примерами таких пакетов являются Fine Reader, Cunie Form, Tiger, Omni Page.

НАСТОЛЬНЫЕ ИЗДАТЕЛЬСКИЕ СИСТЕМЫ

Применяются для профессиональной издательской деятельности. Позволяют осуществлять электронную верстку широкого спектра основных типов документов типа информационного бюллетеня, цветной брошюры, каталога, справочника.

Позволяют решать задачи:

- компоновать (верстать) текст;
 - использовать всевозможные шрифты и выполнять полиграфические изображения;
 - осуществлять редактирование текста на уровне лучших текстовых процессоров;
 - обрабатывать графические изображения;
 - выводить документы полиграфического качества;
 - работать в сетях на разных платформах.
- Примерами таких пакетов являются: Corel Ventura, Page Maker, QuarkXPress, Frame Maker, MS Publisher, Page Plus, Compu Work Publisher.

СИСТЕМЫ ИСКУССТВЕННОГО ИНТЕЛЛЕКТА

Этот класс пакетов включает:

-информационные системы, поддерживающие диалог на естественном языке (естественно-языковой интерфейс);

-экспертные системы, позволяющие давать рекомендации пользователю в различных ситуациях;

-интеллектуальные пакеты прикладных программ, позволяющие решать прикладные задачи без программирования.

•**Естественно-языковой интерфейс** был наиболее привлекателен для общения с ЭВМ с момента ее появления. Это позволило бы исключить необходимость обучения конечного пользователя языку команд или другим приемам формулировки своих заданий для решения на компьютере, поскольку естественный язык является наиболее приемлемым средством общения для человека. Поэтому работы по созданию такого рода интерфейса начались с середины 20-го века. Однако, несмотря на весь энтузиазм исследователей и проектировщиков, эта задача не решена и по сей день из-за огромных сложностей, связанных с пониманием предложений естественного языка и связного текста в целом. Некоторые программные продукты, которые появлялись на рынке, носили скорее экспериментальный характер, имели множество ограничений и не решали задачу кардинально. Тем не менее, несмотря на кажущийся застой в этой сфере, данная проблема остается актуальной и по сей день и вошла в состав проблематики, связанной с проектом ЭВМ пятого поколения.

СИСТЕМЫ ИСКУССТВЕННОГО ИНТЕЛЛЕКТА (продолжение)

•**Экспертные системы** впервые появились в области медицины. Возникла идея интеграции знаний экспертов в области медицины или ее отдельных разделов в некоторую электронную форму, которая позволила бы начинающему врачу иметь своеобразного электронного советника при принятии решений по тому или иному врачебному случаю. Выбор области медицины объясняется слишком большой ценой ошибок, которые касаются жизни и здоровья людей. Постепенно от области медицины эта технология распространилась и на другие сферы деятельности человека, например, производство. Технология использования экспертных систем предполагает первоначальное "обучение" системы, т.е. заполнение ее конкретными знаниями из той или иной проблемной области, а потом уже эксплуатацию наполненной знаниями экспертной системы для решения прикладных задач. Эта идеология проявила себя в проекте ЭВМ пятого поколения в части привлечения конечного пользователя к решению своих задач и связана с проблемой автоформализации знаний.

•**Интеллектуальные пакеты прикладных программ позволяют**, аналогично экспертным системам, предварительно создавать базу знаний, включающую совокупность знаний из той или иной области деятельности человека, а затем решать практические задачи с привлечением этих знаний. Различие этих видов пакетов состоит в том, что экспертные системы, в отличие от интеллектуальных ППП, позволяют интегрировать знания из так называемых слабо формализуемых предметных областей, в которых сложно определить входные и выходные параметры задачи, а также невозможно сформировать четкий алгоритм ее решения. Кроме того, экспертные системы не формируют алгоритм решения задачи как в случае интеллектуальных ППП, а лишь выдают "советы" пользователю на основании его запроса.

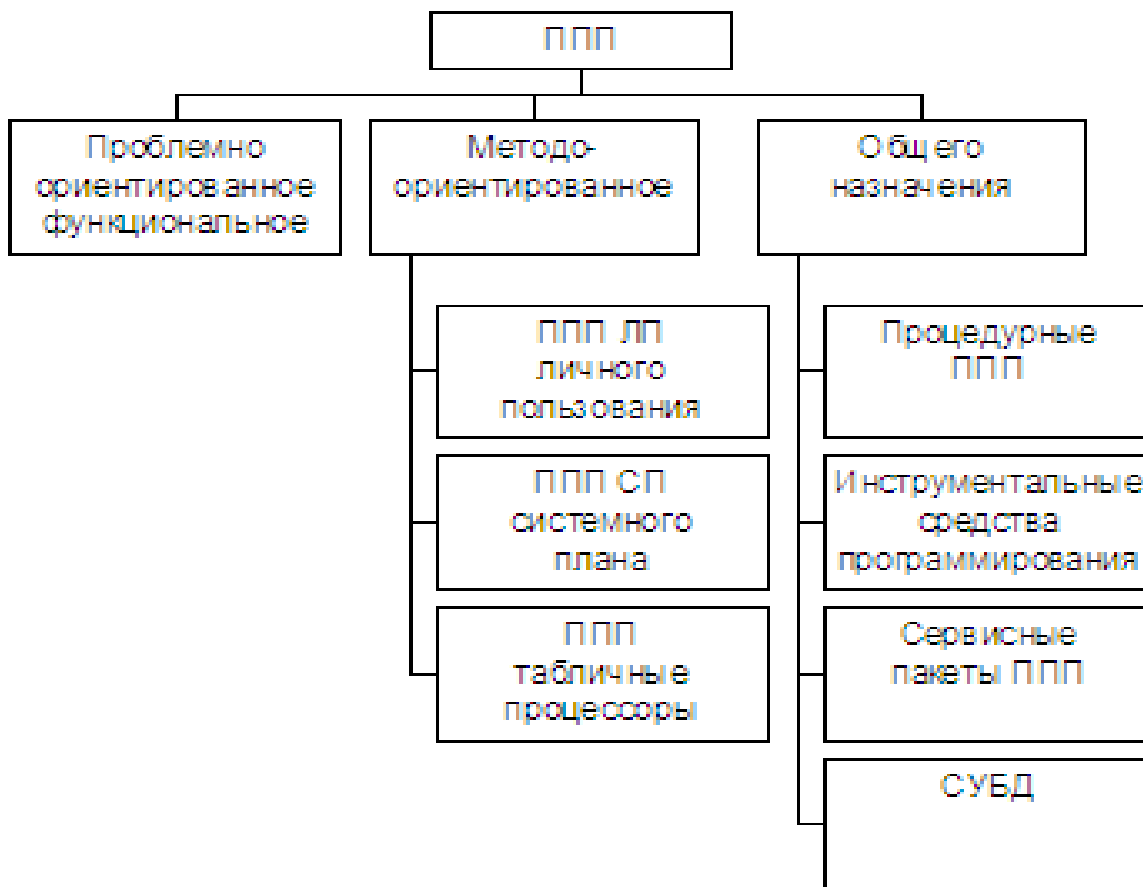
СВОЙСТВА ППП

ППП – комплекс программных средств и документов, предназначенных для реализации функционально завершенного алгоритма обработки данных. Он обеспечивает автоматизацию создания рабочих программ, автоматизацию процесса решения задач.

Характерные черты ППП (3 свойства) :

- Содержит набор готовых алгоритмических решений доводимых до конкретной машинной реализации;
- Содержит механизм настройки на параметры конкретного объекта применения;
- Пакет ПП должен предусматривать возможность дополнения его программами, привязывающими к специфике конкретного объекта, а также к изменившимся во времени условиям эксплуатации.

КЛАССИФИКАЦИЯ ППП (по ОБЛАСТИ ПРИМЕНЕНИЯ)



ДОСТОИНСТВА И НЕДОСТАТКИ ППП

Достоинства ППП:

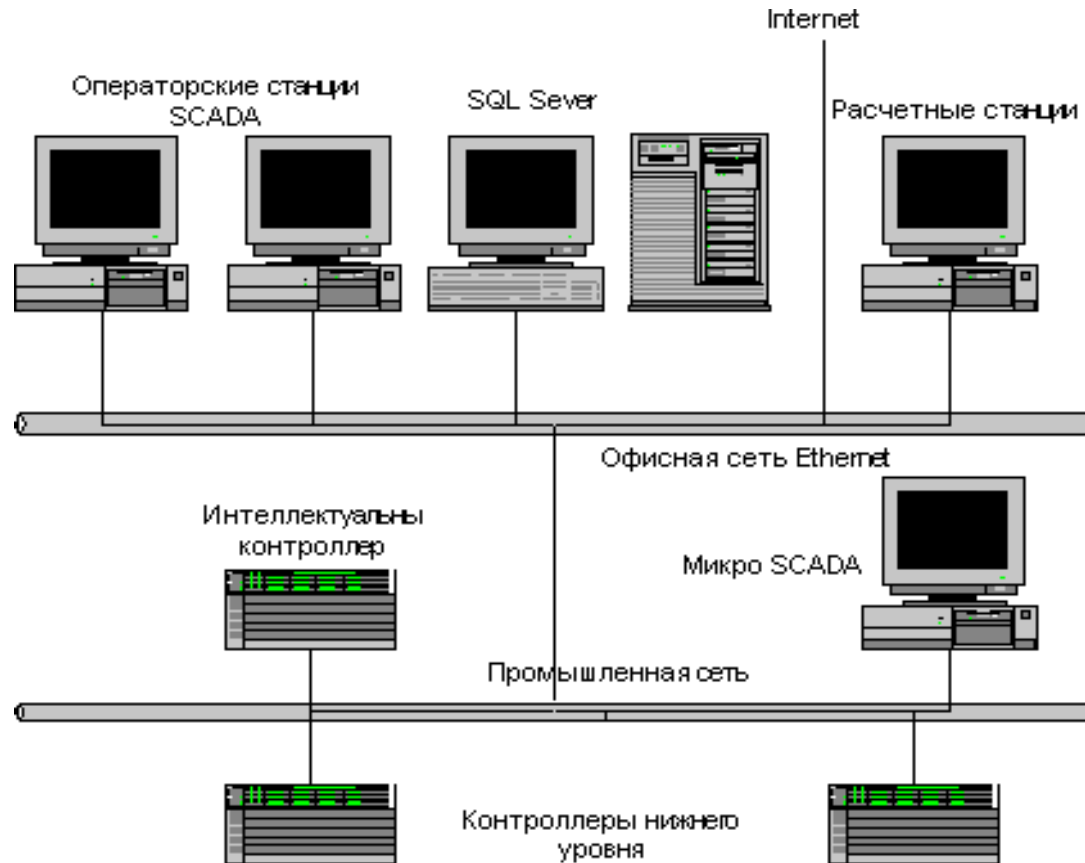
- Сокращение затрат на разработку (до нескольких десятков процентов, в среднем 20–30%);
- По сравнению с элементарными средствами, более высокая комплексная увязка решений;
- Более высокое качество документирования ПИ;
- Более высокая функциональная надежность;
- Наличие развитой системы сопровождения (набор сервисных услуг, которые поддерживают эксплуатацию у пользователя);
- ППП – средство передачи и обмена опытом между разработчиками и между конечными пользователями.

Недостатки ППП:

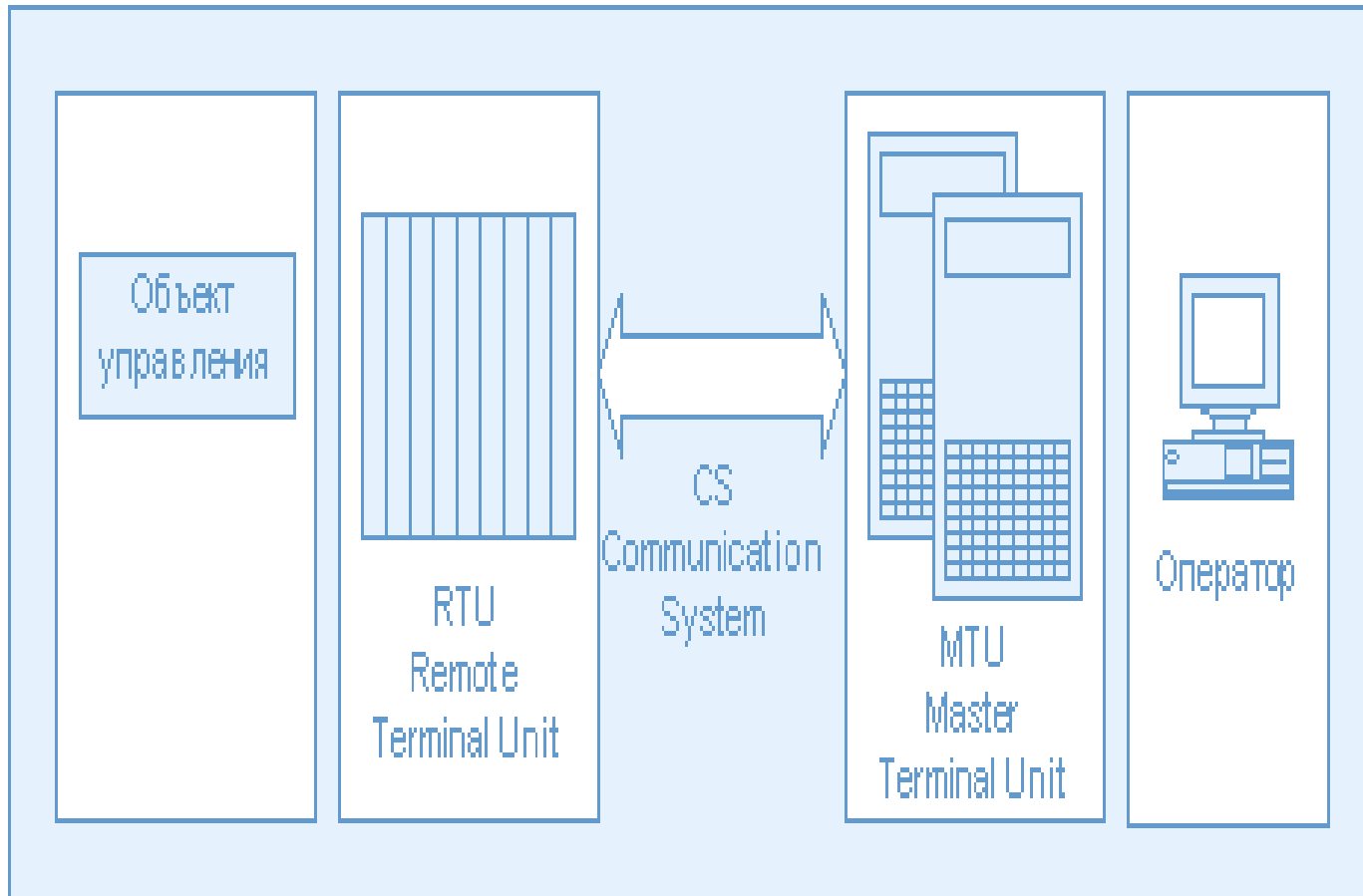
- Сложность освоения ППП;
- Большое разнообразие ППП по распространенным задачам затрудняет выбор. На сегодня отсутствуют объективные методы оценки ППП;
- Низкая степень системной увязки существующих ППП (в случае увязки нескольких конкретных программ по входам–выходам);
- Проблема наращивания и модификации;
- Малая функциональная полнота.

Обобщенная схема системы контроля и управления.

Многие проекты автоматизированных систем контроля и управления (СКУ) для большого спектра областей применения позволяют выделить обобщенную схему их реализации, представленную на следующем рисунке



Основные структурные компоненты SCADA-системы



Основные структурные компоненты SCADA-системы

- Remote Terminal Unit (RTU) удаленный терминал, осуществляющий обработку задачи (управление) в режиме реального времени. Спектр его воплощений широк от примитивных. Конкретная его реализация определяется конкретным применением. Использование устройств низкоуровневой обработки информации позволяет снизить требования к пропускной способности каналов связи с центральным диспетчерским пунктом.
- Master Terminal Unit (MTU), Master Station (MS) диспетчерский пункт управления (главный терминал); осуществляет обработку данных и управление высокого уровня, как правило, в режиме мягкого (квази-) реального времени; одна из основных функций обеспечение интерфейса между человеком-оператором и системой (HMI, MMI). В зависимости от конкретной системы MTU может быть реализован в самом разнообразном виде от одиночного компьютера с дополнительными устройствами подключения к каналам связи до больших вычислительных систем (мэйнфреймов) и / или объединенных в локальную сеть рабочих станций и серверов. Как правило, и при построении MTU используются различные методы повышения надежности и безопасности работы системы.
- Communication System (CS) коммуникационная система (каналы связи), необходима для передачи данных с удаленных точек (объектов, терминалов) на центральный интерфейс оператора-диспетчера и передачи сигналов управления на RTU (или удаленный объект в зависимости от конкретного исполнения системы) [3].

Компоненты систем контроля и управления и их назначение

Как правило, это двухуровневые системы, так как именно на этих уровнях реализуется непосредственное управление технологическими процессами. Специфика каждой конкретной системы управления определяется используемой на каждом уровне программно - аппаратной платформой.

Нижний уровень - уровень объекта (контроллерный) - включает различные датчики для сбора информации о ходе технологического процесса, электроприводы и исполнительные механизмы для реализации регулирующих и управляющих воздействий. Датчики поставляют информацию локальным программируемым логическим контроллерам (PLC - Programming Logical Controller), которые могут выполнять следующие функции:

- сбор и обработка информации о параметрах технологического процесса;
- управление электроприводами и другими исполнительными механизмами;
- решение задач автоматического логического управления и др.

Информация с локальных контроллеров может направляться в сеть диспетчерского пункта непосредственно, а также через контроллеры верхнего уровня. В зависимости от поставленной задачи контроллеры верхнего уровня (концентраторы, интеллектуальные или коммуникационные контроллеры) реализуют различные функции.

Верхний уровень - диспетчерский пункт (ДП) - включает, прежде всего, одну или несколько станций управления, представляющих собой автоматизированное рабочее место (АРМ) диспетчера/оператора. Здесь же может быть размещен сервер базы данных, рабочие места (компьютеры) для специалистов и т. д. Часто в качестве рабочих станций используются ПЭВМ типа IBM PC различных конфигураций.

Станции управления предназначены для отображения хода технологического процесса и оперативного управления. Эти задачи и призваны решать SCADA - системы. SCADA - это специализированное программное обеспечение, ориентированное на обеспечение интерфейса между диспетчером и системой управления, а также коммуникацию с внешним миром.

Спектр функциональных возможностей SCADA

Спектр функциональных возможностей определен самой ролью SCADA в системах управления и реализован практически во всех пакетах:

- автоматизированная разработка, дающая возможность создания ПО системы автоматизации без реального программирования;
- средства исполнения прикладных программ;
- сбор первичной информации от устройств нижнего уровня;
- обработка первичной информации;
- регистрация алармов и исторических данных;
- хранение информации с возможностью ее пост-обработки (как правило, реализуется через интерфейсы к наиболее популярным базам данных);
- визуализация информации в виде мнемосхем, графиков и т.п.; возможность работы прикладной системы с наборами параметров, рассматриваемых как "единое целое" ("recipe" или "установки").

Micro-SCADA.

Micro-SCADA - это системы, реализующие стандартные (базовые) функции, присущие SCADA - системам верхнего уровня, но ориентированные на решение задач автоматизации в определенной отрасли (узкоспециализированные). В противоположность им SCADA - системы верхнего уровня являются универсальными.

Все компоненты системы управления объединены между собой каналами связи. Обеспечение взаимодействия SCADA - систем с локальными контроллерами, контроллерами верхнего уровня, офисными и промышленными сетями возложено на так называемое коммуникационное ПО. Это достаточно широкий класс программного обеспечения, выбор которого для конкретной системы управления определяется многими факторами, в том числе и типом применяемых контроллеров, и используемой SCADA - системой.

Большой объем информации, непрерывно поступающий с устройств ввода/вывода систем управления, предопределяет наличие в таких системах баз данных (БД). Основная задача баз данных - своевременно обеспечить пользователя всех уровней управления требуемой информацией. Но если на верхних уровнях АСУ эта задача решена с помощью традиционных БД, то этого не скажешь об уровне АСУ ТП. До недавнего времени регистрация информации в реальном времени решалась на базе ПО интеллектуальных контроллеров и SCADA - систем. В последнее время появились новые возможности по обеспечению высокоскоростного хранения информации в БД.

Бурное развитие Интернет не могло не привлечь внимание производителей программного продукта SCADA. Возможно ли применение Интернет - технологий в системах управления технологическими процессами? Если да, то какие решения предлагаются в настоящее время компаниями - разработчиками?

Программные продукты класса SCADA

Программные продукты класса SCADA широко представлены на мировом рынке. Это несколько десятков SCADA – систем, многие из которых нашли свое применение и в Узбекистане.

Наиболее популярные из них приведены ниже:

InTouch (Wonderware) - США;

Citect (CI Technology) - Австралия;

FIX (Intellution) - США;

Genesis (Iconics Co) - США;

Factory Link (United States Data Co) - США;

RealFlex (BJ Software Systems) - США;

Sitex (Jade Software) - Великобритания;

TraceMode (AdAstrA) - Россия;

Cimplicity (GE Fanuc) - США;

САРГОН (НВТ - Автоматика) - Россия.

SCADA

• **SCADA** ([аббр.](#) от [англ.](#) *Supervisory Control And Data Acquisition* — диспетчерское управление и сбор данных) — [программный пакет](#), предназначенный для разработки или обеспечения работы в реальном времени систем сбора, обработки, отображения и архивирования информации об объекте мониторинга или управления. SCADA может являться частью [АСУ ТП](#), [АСКУЭ](#), системы экологического мониторинга, научного эксперимента, автоматизации здания и т. д. SCADA-системы используются во всех отраслях хозяйства, где требуется обеспечивать операторский контроль за технологическими процессами в реальном времени. Данное программное обеспечение устанавливается на компьютеры и, для связи с объектом, использует драйверы ввода-вывода или [OPC/DDE](#) серверы. Программный код может быть как написан на языке программирования (например на [C++](#)), так и сгенерирован в среде проектирования.

• Иногда SCADA-системы комплектуются дополнительным ПО для программирования промышленных контроллеров. Такие SCADA-системы называются интегрированными и к ним добавляют термин *SoftLogic*.

• Термин «SCADA» имеет двоякое толкование. Наиболее широко распространено понимание SCADA как приложения^[1], то есть программного комплекса, обеспечивающего выполнение указанных функций, а также инструментальных средств для разработки этого программного обеспечения. Однако, часто под SCADA-системой подразумевают программно-аппаратный комплекс. Подобное понимание термина SCADA более характерно для раздела [телеметрия](#).

• Значение термина SCADA претерпело изменения вместе с развитием технологий автоматизации и управления технологическими процессами. В 80-е годы под SCADA-системами чаще понимали программно-аппаратные комплексы сбора данных в реальном времени. С 90-х годов термин SCADA больше используется для обозначения только программной части [человеко-машинного интерфейса](#) АСУ ТП.

Основные задачи, решаемые SCADA-системами

SCADA-системы решают следующие задачи:

- Обмен данными с «устройствами связи с объектом» (то есть с [промышленными контроллерами](#) и [платами ввода-вывода](#)) в реальном времени через драйверы.
- Обработка информации в реальном времени.
- Логическое управление.
- Отображение информации на экране монитора в удобной и понятной для человека форме.
- Ведение базы данных реального времени с технологической информацией.
- Аварийная сигнализация и управление тревожными сообщениями.
- Подготовка и генерирование отчетов о ходе технологического процесса.
- Осуществление сетевого взаимодействия между SCADA ПК.
- Обеспечение связи с внешними приложениями ([СУБД](#), [электронные таблицы](#), текстовые процессоры и т. д.). В системе управления предприятием такими приложениями чаще всего являются приложения, относимые к уровню [MES](#).

SCADA-системы позволяют разрабатывать [АСУ ТП](#) в клиент-серверной или в распределённой архитектуре.

Основные компоненты SCADA

SCADA—система обычно содержит следующие подсистемы:

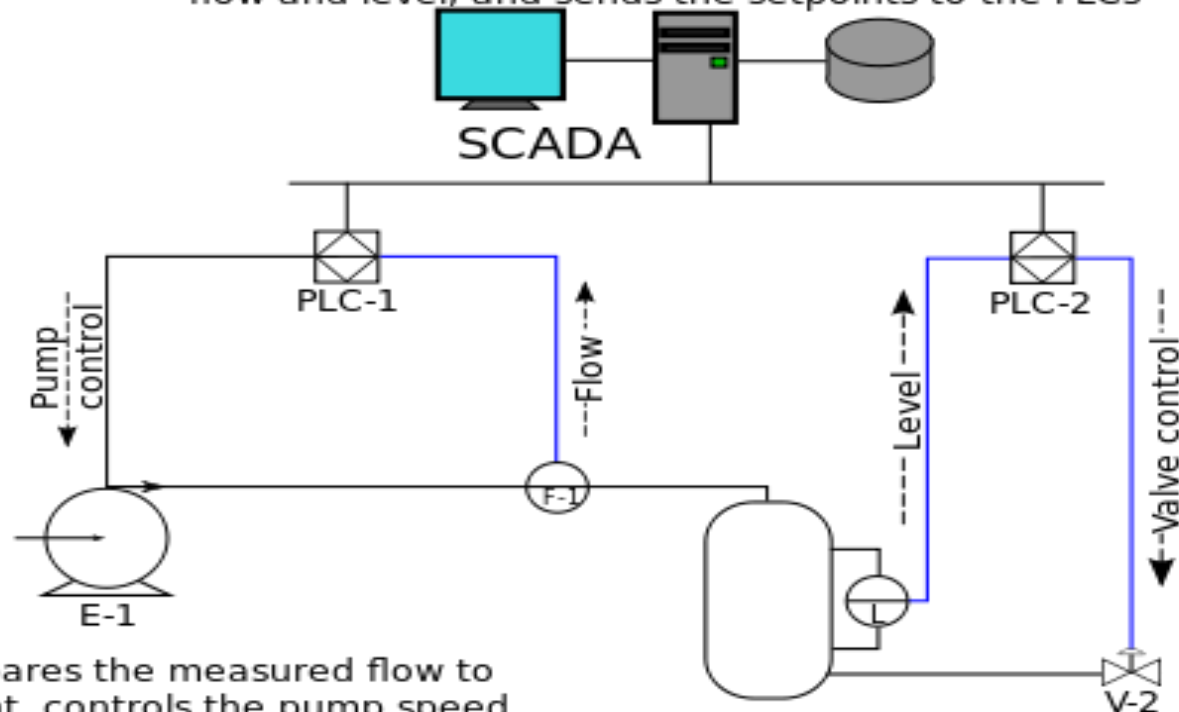
- Драйверы или серверы ввода-вывода — программы, обеспечивающие связь SCADA с [промышленными контроллерами](#), [счётчиками](#), [АЦП](#) и другими устройствами ввода-вывода информации.
- [Система реального времени](#) — программа, обеспечивающая обработку данных в пределах заданного временного цикла с учетом приоритетов.
- [Человеко-машинный интерфейс](#) ([HMI](#), [англ. Human Machine Interface](#)) — инструмент, который представляет данные о ходе процесса человеку оператору, что позволяет оператору контролировать процесс и управлять им.
- Программа-редактор для разработки человеко-машинного интерфейса.
- Система логического управления — программа, обеспечивающая исполнение пользовательских программ (скриптов) логического управления в SCADA-системе. Набор редакторов для их разработки.
- [База данных реального времени](#) — программа, обеспечивающая сохранение истории процесса в режиме реального времени.
- Система управления тревогами — программа, обеспечивающая автоматический контроль технологических событий, отнесение их к категории нормальных, предупреждающих или аварийных, а также обработку событий оператором или компьютером.
- Генератор отчетов — программа, обеспечивающая создание пользовательских отчетов о технологических событиях. Набор редакторов для их разработки.
- Внешние интерфейсы — стандартные интерфейсы обмена данными между SCADA и другими приложениями. Обычно [OPC](#), [DDE](#), [ODBC](#), [DLL](#) и т. д.

Концепции систем

- Термин SCADA обычно относится к централизованным системам контроля и управления всей системой, или комплексами систем, осуществляемого с участием человека. Большинство управляющих воздействий выполняется автоматически [УСО \(RTU\)](#) или [ПЛК \(PLC\)](#). Непосредственное управление процессом обычно обеспечивается RTU или PLC, а SCADA управляет режимами работы. Например, PLC может управлять потоком охлаждающей воды внутри части производственного процесса, а SCADA система может позволить операторам изменять уставки для потока, менять маршруты движения жидкости, заполнять те или иные ёмкости, а также следить за тревожными сообщениями (*алармами*), такими как — потеря потока и высокая температура, которые должны быть отображены, записаны, и на которые оператор должен своевременно реагировать. Цикл управления с обратной связью проходит через RTU или PLC, в то время как SCADA система контролирует полное выполнение цикла.
- Сбор данных начинается в RTU или на уровне PLC и включает показания измерительного прибора. Далее данные собираются и форматируются таким способом, чтобы оператор диспетчерской, используя [HMI](#), мог принять контролирующие решения — корректировать или прервать стандартное управление средствами RTU/PLC. Данные также могут быть записаны в архив для построения [трендов](#) и другой аналитической обработки накопленных данных.

Пример применения SCADA

The SCADA system reads the measured flow and level, and sends the setpoints to the PLCs



PLC1 compares the measured flow to the setpoint, controls the pump speed as required to match flow to setpoint

PLC2 compares the measured level to the setpoint, controls the flow through the valve to match level to setpoint

Архитектура SCADA-систем

В зависимости от сложности управляемого технологического процесса, а также требований к надёжности, SCADA-системы строятся по одной из следующих архитектур:

•Одиночные

При использовании данной архитектуры система состоит из одной или нескольких рабочих станций оператора, которые не "знают" друг о друге. Все функции системы выполняются на единственной (нескольких независимых) станции(ях).

Преимущества:

- простота.

Недостатки:

- низкая отказоустойчивость;
- не обеспечивается истинность данных (исторические данные могут отличаться между разными станциями),

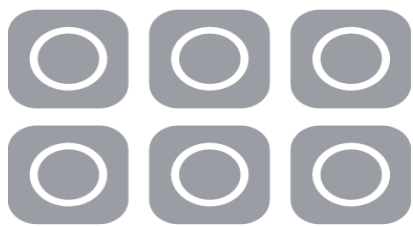
•Клиент-Серверные

В данном случае система выполняется на сервере, а операторы используют клиентские станции для мониторинга и управления процессом. Высоконадёжные системы строятся на базе двойного либо тройного резервирования серверов и дублирования клиентских станций оператора, дублирования сетевых подключений сервер-сервер и клиент-сервер. При данной архитектуре уже возможно разделение функций SCADA-системы между серверами. Например, сбор данных и управление ПЛК выполняется на одном сервере, архивирование данных - на втором, а взаимодействие с клиентами - на третьем.

•Виртуализация

Современные технологии виртуализации уже глубоко проникли в промышленную автоматизацию. Применение кластеров и виртуальных серверов с разделением функций SCADA-системы позволяет обеспечивать высокую отказоустойчивость, гибкое распределение вычислительных ресурсов, изолирование системы и сетевых подключений от постороннего сетевого трафика, безопасность данных. При использовании виртуализации клиентские станции уже не требуют полноценного ПК(Толстый клиент), достаточно тонкого клиента с подключением к виртуальному клиенту. Облачные вычисления также применяются в промышленной автоматизации либо автоматизации зданий.

Базовая схема SCADA



Sensors

Sends data to PLCs or RTUs



PLCs or RTUs

Feeds data to SCADA system



HMI/SCADA Panel View

Supervise and control from an operational terminal



Manual Inputs

Sends data to PLCs or RTUs



PLCs or RTUs

Feeds data to SCADA system



HMI/SCADA Computer

Supervise and control from a workstation