

Важнейшие свойства и принципы в основе современных технологий.

“ Знания технологии зарождались вместе с потребностью сделать каменный топор и, с этой точки зрения, технология является прародительницей всех наук современности...”

(из студенческого юмора)

Постановка вопроса

Поскольку для самостоятельного создания вещей, представляющих пользу, необходимы определенные усилия, то, наследуя традиции предков, попытаемся разобраться в чем их суть. Для начала ответим на вопрос о том, что необходимо для получения любого полезного результата. Ответ достаточно прост, как правило, нам всегда будут необходимы:

Цель и ресурсы

При этом, наличие цели предполагает следующее:

Достижимая цель	Убежденность в полезности результата
	Наличие носителей цели (персонал)
	Позитивные оценки о принципиальной возможности достигнуть цель.
	Критерии и оценки достижения цели, а в идеале, и критерии оптимальности возможных путей ее достижения.

Ресурсы

Материальные ресурсы	Интеллектуальные ресурсы
	Инструментальные ресурсы
	Обрабатываемые ресурсы
	Энергетические ресурсы
	Финансовые ресурсы, + время

Хорошо, скажите Вы, цель и ресурсы у нас имеются, но это ведь не дает ответа на вопрос о том как достигнуть цель, то есть, нам необходимо обеспечить движение к цели. По существу, способы и средства построения и реализации такого движения и являются как предметом, так и результатом науки технологии, которая отвечает на свой основной вопрос:

Как сделать?

Наиболее обще на этот вопрос можно ответить следующим образом:

Опираясь на критерии, придумать и реализовать такую последовательность действий над доступными ресурсами, при которой цель будет достигнута или будет отслеживаться оптимальным образом.

По существу, приведенная формулировка подсказывает определение технологии как науки, а также определяет технологию как результат этой науки.

- Креативная часть ответа (“придумать”) определяет предмет технологии как науки.
- Исполнительная часть ответа (“последовательность действий”) определяет технологию как результат этой науки.

Рассмотрим отдельные детали приведенной постановки вопроса.

Цели:

Любые цели вырабатываются как результат поиска некоторой системой точек наивысшей устойчивости к самым разнообразным внешним и внутренним воздействиям. Формально, когда условия определяющие такую устойчивость осознаются, мы формулируем их совокупность в виде понятия «цель». Таким образом, цели неизбежно вырабатывается в процессе развития или самоорганизации систем.

Для того чтобы цель стала достижимой целью, (то есть попала в поле зрения технологии), она, опираясь на интеллектуальные ресурсы, должна быть сформулирована как совокупность критериев и путей ее достижения. Ответы о достижимости цели дают технологии двух классов:

- Технологии научных исследований
- Технологии проектирования

По существу, мы имеем дела с креативной или созидательной частью цели, когда ее реализация носит от части виртуальный характер.

Процесс реального создания системы достижения цели (в дальнейшем для простоты - технической системы) определяют группу технологий под общим названием:

- Технологии изготовления технических систем.

Окончательно цель может быть достигнута только тогда, когда вступают в действие технологии вида:

- Технологи эксплуатации технических систем
- Технологии производства.

Все названные классы и подклассы технологий охватывают грандиозный массив знаний и ресурсов, а следовательно, не могут быть рассмотрены в рамках одной учебной дисциплины. В тоже время их объединяет общая нить, которую можно сформулировать как “жизненный цикл технических систем”. Инструментом для анализа жизненного цикла систем, в самом общем случае, является системный анализ. Такой системный подход позволяет нам сформулировать основные принципы, характерных для технологий как науки так и результатов этой науки.

В первую очередь, подчеркнем одну из основных особенностей результатов науки технологии. Отвечая на вопрос «как сделать», мы не можем позволить себе отрицательного ответа, поскольку отрицательный ответ равнозначен отсутствию результата.

Таким образом, первым естественным определением технологии как результата будет является последовательность действий над ресурсами (и с помощью ресурсов) которая гарантировано обеспечивает достижение цели.

Такое определение подсказывает нам целесообразность первоочередного рассмотрения системных особенности названных действий, а также целесообразность их структурирования.

Итак, особенности действий:

Совместимость конкретного действия с конкретным видом ресурса.

Например, бесполезно ресурсу “чайник” предписывать выполнение действий по перемножению чисел, а ресурсу “компьютер” действие по кипячению литра воды – для этого чайников хватает ☺

Преимущественно последовательный характер действий во времени.

Поскольку в природе не существует неисчерпаемых ресурсов, а ресурсы необходимы для выполнения действий, всегда можно указать ограничение на количество действий, выполняемых одновременно. Однако, если ресурс допускает многократное использование (например, повторное использование некоторых инструментов) естественным является последовательное во времени использование только ресурса, а следовательно и преимущественно последовательный характер необходимых действий.

Опосредованность большинства действий через инструментальные ресурсы.

Поскольку вершину известной нам иерархии саморазвивающихся систем занимают люди, то именно они являются носителями первичных целей, а следовательно и корневым источником, инициирующим цепочки действий по их достижению. В то же время, спектр физических и информационных действий, которые может выполнить человек, является ограниченным как по физическому составу, так и по параметрам. Поскольку сложные цели, как правило, требуют и более сложных действий, элементом реализации таких действий становятся инструменты. В общем случае такие инструменты можно определить, как управляемые со стороны человека преобразователи физических или информационных величин.

Например – “ для получения спектра сигнала необходим информационный инструмент «разложение в ряд Фурье». В этом случае в качестве инструмента выступает либо метод, либо алгоритм получения коэффициентов ряда. Другой пример – “для получения цилиндрической поверхности необходимого класса чистоты необходим токарный станок с резцом №4 при глубине резания 0.5 мм и продольной скорости не более 10 см/мин...”

Обратим внимание, что и метод (алгоритм) Фурье и токарный станок, являются инструментами, посредством которых, могут обрабатывать множество информационных или физических объектов. Также следует подчеркнуть, что отсутствие таких инструментов делает недостижимыми цели, ради которых они были созданы.

Возможность совершать действия с разными параметрами одним инструментом

Такое свойство инструмента прямо следует из принципа минимизации затрат, ибо исходя из обратного, нам бы понадобилось бесконечное число инструментов. Другими словами, даже простейший ряд из 3-4 молотков, пришлось бы заменить на тысячу, каждый молоток в которой был бы предназначен для удара определенной силы.

Таким образом, важно подчеркнуть, что инструменты выполняющие однотипные действия, должны позволять их выполнение с различными параметрами.

Обобщая этот (далеко не полный) список особенностей действий и инструментов, мы можем сформулировать следующее:

Определение:

Цепочки адресованных различным инструментам действий, некоторые из которых параметризуются через глобальные для всей цепочки предписания, будем называть технологическими процедурами, а формальное описание конкретных

действий в цепочке, будем называть технологическими инструкциями для конкретного инструмента.

Выше мы подчеркивали тесную связь инструмента и обрабатываемых инструментом ресурсов, теперь в эту связь мы ввели инструкции для действий совершаемых инструментом. Как следствие, это предполагает более внимательное рассмотрение основных системных особенностей по линиям этих связей.

Во первых, сразу отметим, что в результате такого рассмотрения, в поле нашего внимания должны попасть следующие технологии:

- Технологии организации процессов (как последовательности действий).
- Технологии транспортировки инструментального и обрабатываемого ресурсов.
- Технологии обработки (непосредственного выполнения действий).

Принципы организации процессов.

Очевидно, что для выполнения некоторой последовательности действий над обрабатываемым ресурсом, такой процесс необходимо организовать, т.е. спланировать действия, подготовить ресурсы и инструменты, выполнить спланированные действия, по завершению выполнения привести в исходное состояние многократно используемые ресурсы, а также утилизировать отходы. Все эти функции возлагаются на технологии организации процессов. Обобщенный вид структура подобной организации процесса нашла свое решение в реализациях систем управления вычислительным процессом и может быть представлена (с поправкой на введенную терминологию) следующей блок-схемой:



Очевидно, что исходной информацией для функции планирования являются заказанные технологические процедуры, карта состояния подчиненных ресурсов, а также различные события как вне, так и внутри области ответственности планировщика. Такая информационная картина позволяет планировщику разработать группы технологических процедур и передать их на исполнение диспетчеру технологического процесса.

В соответствии с планом, диспетчер пытается и, если это возможно, выполняет предписанные планировщиком технологические процедуры, а если текущая ситуация не позволяет выполнить ту или иную процедура, формирует событие, которое требует корректировки планов. Собственно говоря, любой генерал подтвердит вам, что даже идеальный план кампании устаревает после первого выстрела.

Таким образом, обратная связь от диспетчера к планировщику является залогом выхода из различных тупиковых ситуаций. Основными причинами таких тупиковых ситуаций является случайный характер качества (если речь идет об обрабатываемых ресурсах) и количества (если речь идет о свободных повторно используемых ресурсах) в структуре исходных ресурсов процесса. В любом случае, основными задачами диспетчера являются:

Загрузить технологическую (операционную) зону.

Под технологической (операционной) зоной будем понимать некоторое пространство, где будет проходить технологический процесс. Под загрузкой зоны доставку в тех. зону обрабатываемых и инструментальных ресурсов. В тех случаях, когда инструментальные ресурсы носят крупномасштабный физический характер, их, как правило, делают неотъемлемой частью технологической зоны (такие зоны традиционно называют технологическими участками).

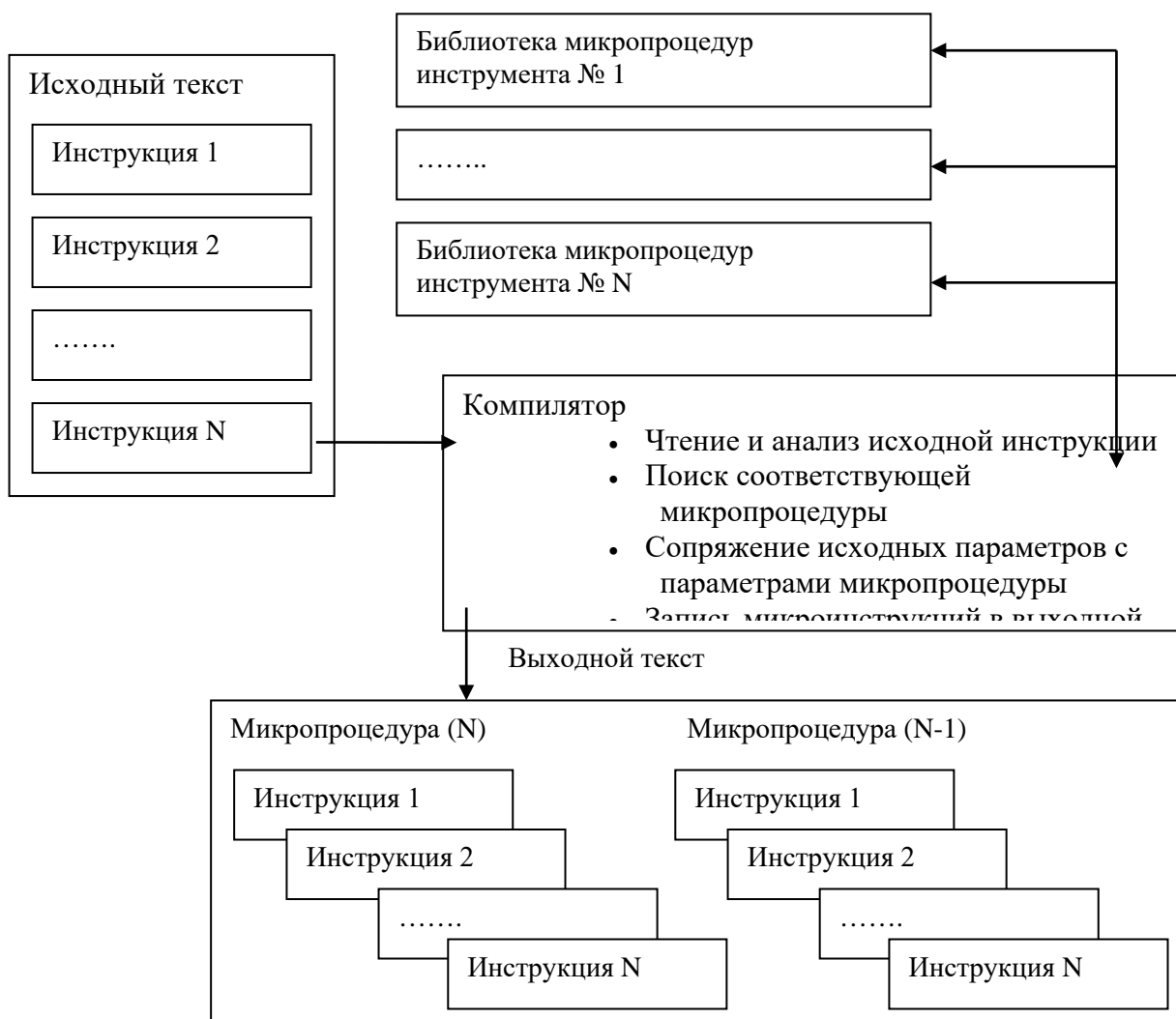
Очевидно, что название функции является результатом интерпретации диспетчером плана подготовки процесса и для их реализации требуют исполнения инструкций, формируемых транспортными технологиями.

Выполнить технологическую процедуру.

На этапе выполнения заказанной технологической процедуры, основной задачей диспетчера является размещение и отслеживание качества выполнения потока технологических инструкций (команд), которые адресованы инструментальным ресурсам в операционной зоне.

Особо следует подчеркнуть, что технологические инструкции (как составные части технологической процедуры), являются результатом компиляции заказанной технологической процедуры на основе возможностей и параметров загруженного в технологическую зону инструментария. Поясним это:

Предположим, что в качестве заказанной технологической процедуры планировщик получал заказ на изготовление винта с резьбой М3. Пусть также параметры этой процедуры не требуют высокого качества изготовления головки винта. Таким образом, его изготовление можно реализовать либо токарной обработкой заготовки с последующей нарезкой резьбы, либо штамповкой изделия сразу специальной матрицей. В зависимости от загруженности инструментария планировщик может выбрать любой из вариантов реализации обработки. Однако, в этом случае, одни и те же исходные данные заказа он должен представить различными цепочками технологических инструкций (команд), которые ориентированы на различный вид и состав инструментария. Проще всего такую цепочку составить в том случае, если каждой единице инструментария будет прилагаться набор типовых технологических микропроцедур, которые данная единица инструментария может реализовать. Тогда заказанная технологическая процедура может быть составлена так, что ее отдельные инструкции можно заменять сходными технологическими микропроцедурами, входящими библиотеки возможностей взаимозаменяемых инструментов. Такой процесс называется компиляцией, а его структуру можно представить следующим рисунком:



Понятно, что скомпилированный заказ (полный выходной текст) будет содержать цепочки из технологических процедур (микропроцедур), адресованных различным инструментам или (что равнозначно) цепочки соответствующих инструкций (микроинструкций). С учетом дополнительных предписаний планировщика выходной текст или его сегменты могут размещаться диспетчером в инструментарий параллельным, последовательным или комбинированным (во времени) способом. Например, два робота могут монтировать две детали одновременно, если их рабочие области в зоне не пересекаются, а третья деталь должна монтироваться на первые две. Если скорости монтажа различны у нас возникают следующие события – “готовность к монтажу деталей №1 и №2”, “деталь №1 – смонтирована”, “деталь №2 – смонтирована”, “готовность к монтажу детали №3”, “деталь №3 – смонтирована”. Очевидно, что перечисленные события должны синхронизировать действия диспетчера по размещению очередной тех. процедуры в каждый инструмент и ее активации.

Разгрузить технологическую зону.

Данная процедура является зеркальной по отношению к задаче загрузки технологической зоны. При этом повторно (многократно) используемые ресурсы возвращаются в зону хранения свободных ресурсов, где если это необходимо, подвергаются профилактике или восстановлению.

Например: катализаторы очищаются и активируются, свободная память компьютера дефрагментируется, резцы станков затачиваются, приборы калибруются, и т.п.

Результирующие ресурсы выгружаются из технологической зоны и размещаются в буферных приемниках, складах или иных промежуточных зонах хранения. Отходы обработки выгружаются в зоны утилизации или переработки.

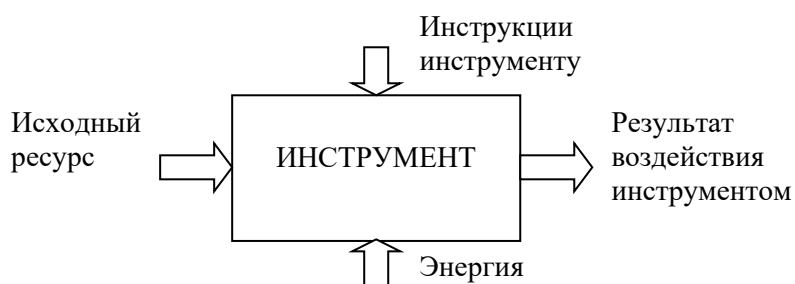
Подводя итог рассмотрению основных особенностей по линии инструменты – обрабатываемые ресурсы, следует отметить, что с ростом сложности инструментария и повышением гибкости в реконфигурации технологических зон, общее число элементарных технологических инструкций, подлежащих планированию и диспетчированию, имеет тенденции возрастать в геометрической прогрессии. Про этом всегда можно указать некоторый количественный или скоростной порог, при котором вначале становится невозможным диспетчирование процесса человеком-оператором, а потом и оперативное планирование.

Другими словами, мы показали возрастающую тенденцию вытеснения живого труда из области непосредственного управления процессами в области все более общих абстракций. Однако замена человека различными автоматами на практических установках технологий еще не объясняет, почему компьютерные технологии вынесены в отдельный и весьма специфический класс, присутствие которого сегодня наблюдается как основная часть любой другой технологии. Для ответа на этот вопрос проследим особенности, которые просматриваются в эволюции инструментария и принципах его интеграции в более сложные структуры.

Особенности эволюции инструментария.

Самый первый инструмент придуманный человеком сочетал в себе приемник управления, источник энергии и преобразователь энергии в необходимую физическую величину. Конечно, данная дефиниция смотрится по крайней мере одиозно по отношению к обычной палке или камню взятых в руку, однако позволяет спроектировать эту структуру (приемник управления, источник энергии, преобразователь энергии в необходимую физическую величину) на все известные сегодня инструменты. Например, цифро-аналоговый преобразователь (ЦАП), который связывает управляющий код с необходимой физической величиной (физическим действием).

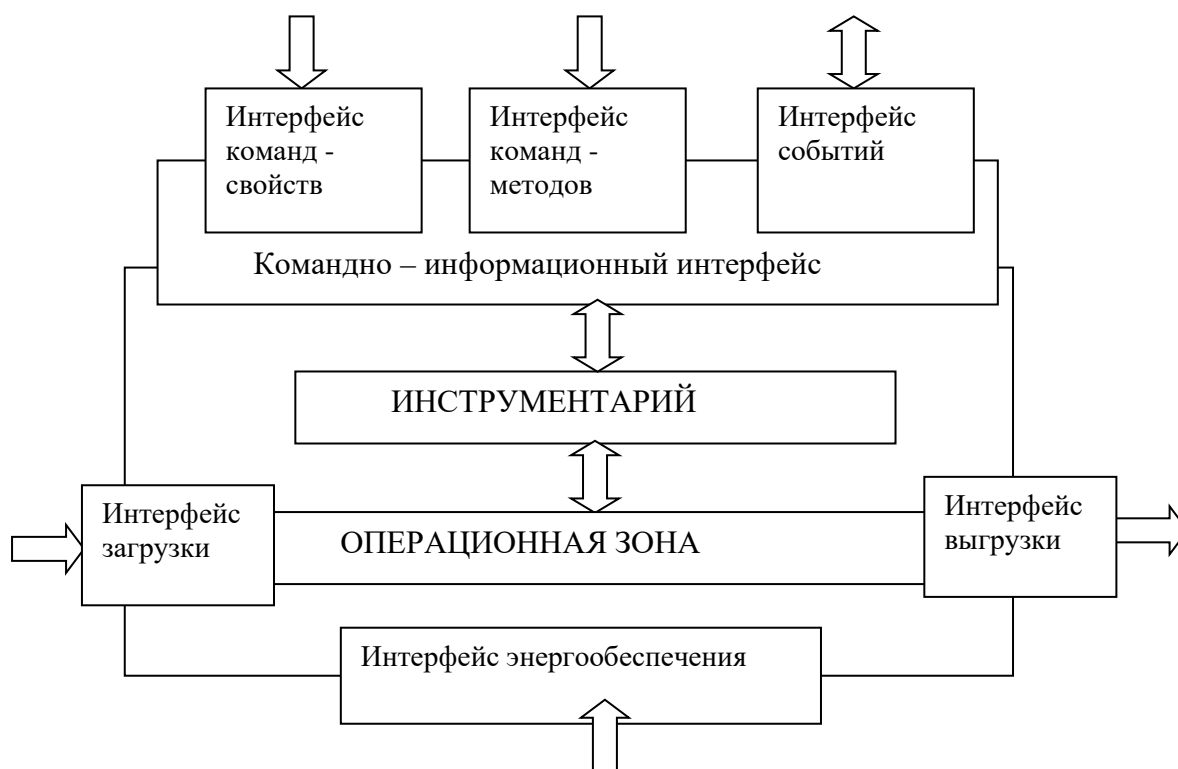
Исходя из сказанного, можно предложить самую обобщенную структуру для формализма «инструмент»:



Продолжая линию примера, можно заметить, что следующим инструментом являлся молоток или топорик. Другими словами, эволюция инструмента заключалась в соединении двух простейших инструментов палки и зубила в единое целое. Такой подход создания новых инструментов на основе их предшественников как принцип исповедуется и в современных разделах технологий конструирования. По мере развития сложности инструментария его эксплуатационный опыт позволил ряд других важных принципов:

Принцип инкапсуляции.

Суть этого принципа отражается в размещении всех составляющих инструмента, которые напрямую не взаимодействуют с оператором, в некоторую защищенную капсулу (корпус, оболочку). Очевидно, что подобная защита составляющих инструмента позволяет защитить его целостность и качество от случайных негативных воздействий. Развитие названных принципов в контексте формализации взаимодействия оператор-инструмент привело к тому, что на территории технологий программирования была сформулирована абстракция “инкапсулированный объект”. Данная абстракция определяет такой объект как совокупность инструментария, инкапсулированного в закрытую оболочку, на поверхность которой выносятся интерфейс объекта. Появление такого интерфейса (отражающего систему соглашений между лицами) позволяет структурировать характер взаимодействия инструмента-объекта с внешней средой. Поскольку с одной стороны, интерфейс – это инструменты реализации этих правил (например команды на выполнение действий) подобная формализация инструмента не могла быть проигнорирована технологией как наукой. Таким образом, технологическую зону можно формализовать как инструмент-объект, в интерфейсах которого формальным образом представлены командная часть (совокупность команд для загрузки зоны, самодиагностики и калибровки), физическая часть, которая определяет характер и параметров интерфейсов – загрузки выгрузки обрабатываемых ресурсов и энергообеспечения, а также интерфейсы для обмена событиями как со стороны внешней среды, так и со стороны инструмента-объекта. Сказанное позволяет форматным образом упорядочить взаимоотношения между технологическим процессом и его диспетчером в том случае, если строго потребовать, чтобы инструмент-объект проявил любую форму активности только как следствие получения команды от диспетчера.



В соответствии со сказанным, представим блок-схему инструмента-объекта, на которой структурируем некоторые особенности командного интерфейса. Для того, чтобы манипуляции с самым инструментом четко разграничивались с манипуляциями над ресурсами, которые обрабатываются, всю совокупность команд делят на две части – команды свойства и команды методы. Команды свойства позволяют реконфигурировать инструментарий внутри инструмента объекта или изменять параметры. Команды методы побуждают инструментарий выполнять действия над обрабатываемым ресурсом.

События. События направленные от инструмента-объекта по существу являются инструментом информации для диспетчера о том когда и насколько качественно завершилось выполнение очередной команды. Такие отчеты инструмента-объекта позволяют синхронизировать поток технологических инструкций, которые формирует диспетчер. Рассмотренная особенность настолько значительна что определила собой основной принцип организации вычислительного процесса в современных операционных системах.

События направленные из внешней среды к инструменту-объекту, как правило, носят характер аварийных оповещений. Такие оповещения равнозначны для инструмента-объекта требованию отключить командный интерфейс и продолжить работу по аварийному списку инструкций, который размещен внутри инструмента-объекта. Такой подход является особо значимым для тех. процессов, утрата управления которыми чревата разрушительными последствиями.

Принцип наследования.

Рассматривая принцип инкапсуляции, мы уже затрагивали основную идею конструирования – созданий новых инструментов путем интеграции более простых, но выдержавших испытание временем. Размещая такие инструменты внутри инструмента-объекта, мы должны задуматься – да, мы получим новые возможности и, соответственно, новые команды, но все ли команды инкапсулированных инструментов потеряют свою актуальность. Скорее всего, ответ будет отрицательным. Но когда возникает вопрос о том, как получить доступ к таким командам, ведь инструмент, который их выполняет, инкапсулирован. О Ответ очевиден – необходимо включить в командный состав нового инструмента-объекта такие важнейшие команды инкапсулированных инструментов и потребовать от объекта переадресовать их исполнителю. Но это лишь один из аспектов принципа, который называется наследованием. Предположим, что один из инкапсулированных объектов мог выполнять команду “изменить темп работы”. Очевидно, что если мы желаем наследовать эту команду в новом инструменте, то нам потребуется изменять темп работы всех инструментов, которые непосредственно связаны с инструментом, изменившим свои свойства.

Если конструируемый инструмент способен выполнить такую внутреннюю синхронизацию инкапсулятов, то мы можем допустить в его интерфейсе команду “изменить темп работы”, которая будет наследоваться от одного из инкапсулятов, однако будет корректно выполняться конструируемым инструментом в целом.

Такой подход, как принцип, получил название наследование.

В чем же конструктивность такого подхода? Оставляя за скобками некоторые особенности, ответим так – сохраняя старые формы представления (форматы) отдельных команд, мы не плодим новые сущности, и тем самым сохраняем возможность выполнять давно существующие тех. процедуры на новом оборудовании.

Принцип полиморфизма

Полиморфизм (поли – “много”, морфа – “форма”), или много способов реализации одной формы команды. Где и когда это необходимо? Приведем пример.

Пусть у нас есть инструменты, способные перемножать целые, действительные, комплексные числа, инкапсулированные в инструмент “перемножитель”.

Формируя состав командного интерфейса операции, мы можем избрать две стратегии: первая – представить в интерфейсе три команды, которые напрямую наследуются от каждого из инкапсулятов. При этом структура представления каждой из команд не требует изменения, но нам приходится эти команды видоизменять, без чего их невозможно отличить и переадресовать исполнителю. Такой подход резко расширяет множество команд.

Вторая стратегия предполагает связывание механизма распознавания с анализом свойств обрабатываемых ресурсов. Это позволяет получать необходимую для переадресации информацию автоматически внутри создаваемого инструмента. В этом случае в состав “перемножителя” войдет новый инкапсулят или “распознаватель типа ресурса”, а на командный интерфейс “перемножителя” будет вынесена всего одна команда. Итак, команда одна, а способов ее исполнения несколько. Причем, выбор конкретного способа привязывается к типу обрабатываемого ресурса. В приведенном нами примере тип обрабатываемого ресурса определяется по форме записи операндов, то есть загрузка ресурсов осуществляется через форму (но не структуру) записи самой команды. Однако, в любом случае, приведенная выше логика реализации остается неизменной и соответствующий ей принцип сохраняет наименование – полиморфизм.

На первый взгляд, полиморфизм является весьма специфичным принципом, характерным лишь для инструментария обработки информации. Однако это не так.

Достаточно вспомнить о том, что сегодня, к примеру, современные мультиметры не только автоматически распознают диапазон измерения, но и тип измеряемой величины (постоянный или переменный ток) и делает это по одной команде “измерить напряжение”. В случае этого примера связь между формой и способом ее реализации значительно прозрачнее.

Итак, подводя итог рассмотрения системных особенностей эволюции инструментария сформулируем следующее:

Современный инструмент – это иерархическая инкапсулированная структура, основанная на принципах наследования и полиморфизма.

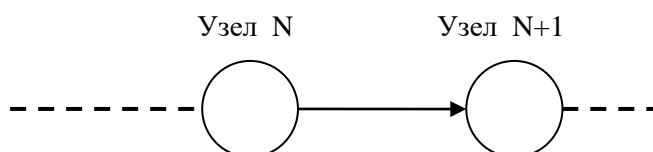
При этом следует подчеркнуть, что принцип наследования в основном определяет возможности реконфигурации самого инструмента, а полиморфизма-способности автоматической подстройки инструмента к свойствам обрабатываемого ресурса. Кроме того, формализация инструмента к виду инструмент-объект позволила вести нам такое важное понятие как “событие”. Опираясь на сказанное, гораздо проще пояснить принципы построения более сложных структур, которые объясняют отдельные инструмент – объекты (именуя их узлами обработки или просто узлами), в различные варианты обрабатывающих сетей.

Свойства и принципы в основе обрабатывающих сетей.

Очевидно, что простейшая обрабатывающая сеть состоит из цепочки узлов последовательно связанных по линии движения обрабатываемых ресурсов.

Если вспомнить один из принципов гидродинамики, гласящих что скорость потока постоянной плотности определяется через самое узкое его сечение, то, ассоциируя не сжимаемость вещества потока (постоянную плотность) с общим числом операций в цепочке, мы можем сформулировать следующий вывод: “скорость движения обрабатываемых ресурсов через строго последовательную цепочку узлов будет определяться самым медленным из них”.

При всей простоте, эта особенность позволяет сформулировать целый ряд полезных следствий:



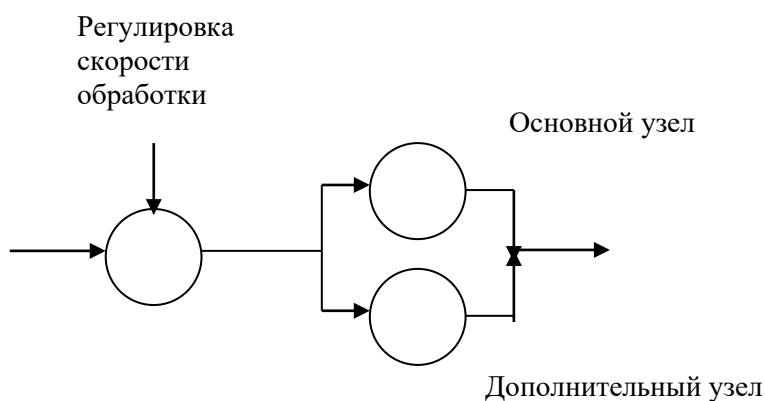
Следствие 1

Если скорость прохождения узла (N) потоком обрабатываемых ресурсов будет меньше скорости прохождения в узле (N+1), поток будет периодически разрываться, а узел (N+1) простаивать.

Следствие 2

Если скорость потока в узле (N) будет выше скорости в узле (N+1), линия от узла (N) к узлу (N+1) или меж узловой буфер будет аварийно переполняться. Это неизбежно, поскольку не существует как безграничных ресурсов, так и безграничных хранилищ.

На первый взгляд, выход из положения достаточно прост – пусть скорость потока в узле (N) будет равна скорости в узле (N+1). Для этого достаточно воспользоваться командой “изменить скорость обработки”, которую мы уже затрагивали при рассмотрении инкапсуляции инструментария, или, если это не помогает, запаралелить медленный узел, например (N+1) :



Более того, подход с выравниванием скоростей, позволяет сформулировать два логически прозрачных критерия для оптимизации работы цепочки:

- Количество обрабатываемого ресурса, ожидающего обработки в любом меж узловом буфере должно быть минимальным.
- Простой любого узла обрабатывающей сети должен быть минимальным.

Далее, применяя математические методы теории направленных графов и сетевого планирования, реализовать требования критериев четкое расписание действий каждого узла цепочки.

В идеальном случае так и делается. А что тогда означает случай реальный?

Реальный случай подразумевает учет событий, которые возникают в инструменте-объекте и носят случайный характер. Такие события в отличие от их детерминированных собратьев возникают тогда, когда средства полиморфизма реагируют на изменение качества или структуры обрабатываемого ресурса, когда возникают нештатные ситуации и т.п. Все эти события побуждают через фиксирующий их диспетчер активировать планировщик, который откорректирует необходимые тех. процедуры и, тем самым, построить план устранения неблагоприятной ситуации. Далее этот план попытается реализовать диспетчер и, если это не вызовет новых событий, требующих корректировки плана, процесс будет завершен детерминированным событием “обработка в рабочей зоне завершена”.

Такая цепочка из различных “если”, да еще инициируемая случайным образом, приводит к тому, что время исполнения технологической процедуры становится случайной величиной в которой нам детерминированным образом известна только начальная точка. Поскольку конструкторы технологии стремятся всяческим образом обеспечить главное условие технологии – гарантированное достижение поставленной цели, мода распределения вероятности времени выполнения операции будет близка к идеальному времени, а сам закон распределения будет иметь приблизительно такой вид:



Итак, поскольку основной параметр – время обработки приобретает случайный характер, нарисованная ранее идеальная картина для обрабатывающей сети значительно усложняется и становится предметом изучения таких научных разделов как теория массового обслуживания (ТМО) или при большем акценте на проблемы движения обрабатываемых ресурсов – теории очередей (ТО).

В последнее время к методам и результатам то стали подключать экологические методы, критерии и оценки. Такое обобщение сформировало новое научное направление – Логистику.

Очевидно, что вероятностный характер процессов, которые исследуют эти научные разделы, требует применения достаточно сложных, а в прикладном применении, трудоемких математических методов. Таким образом, оперативное планирование и диспетчирование обрабатывающих сетей еще более усиливает тенденцию вытеснения живого труда к самым высоким уровням иерархии управления. Сформулировав эту тенденцию при рассмотрении инструмент - объекта и усилив ее при рассмотрении основных особенностей обрабатывающих сетей, мы неизбежно приходим к выводу, что на равне с анализом движения обрабатывающих ресурсов (условно назовем их горизонтальными потоками) заметное значение также приобретает анализ потоков управления или вертикальных потоков , вершиной которых является основная цель создания технологии.

Сегодня проблематика анализа синтеза потоков управления разрабатывается целым рядом научных управлений, исследующих задачи автоматического и автоматизированного управления. Однако наиболее обобщенную реализацию результатов этих исследований мы можем сегодня наблюдать в системах, получивших название – операционные системы для организации вычислительных и управляемых процессов (ОС и ОС реального времени соответственно).

Почему компьютерные ОС оказались лидерами в эволюции управляющих систем? Ответ на этот вопрос становится очевидным при рассмотрении особенностей жизненного цикла любой системы, возьмем небольшой сегмент этого цикла:



Предполагая, что любая новая система может потребовать доработки по итогам ее опытной эксплуатации, сравним стоимость и время, необходимое для уменьшения некоторой функции как компьютерной программы и как аппаратного узла. Для тех, кто еще не догадался, сообщим что стоимость в финансовом и временном измерении будет на несколько порядков в пользу компьютерной программы. Поскольку доступный уровень финансовых и временных ресурсов определяет порог потенциальной сложности и функциональности систем (что также очевидно из сегмента создания системы), в эволюционной гонке начинает выигрывать программно – компьютерный подход.

Другими словами, компьютерные технологии становятся неотъемлемой частью практически всех технологий, а различные подходы свойства и принципы в основе операционных систем становятся предметом, требующим внимания. Тем самым, мы наше внимание привлекает принцип клиент – серверных отношений, который является характерным для современных операционных систем. Однако об этом, мы будем говорить уже в следующей части книги.

Литература:

1. Гради Буч. Объектно-ориентированный анализ и проектирование.pdf
2. Хинчин А.Я. Работы по математической теории массового обслуживания.1963.djvu
3. Клейнрок Л. Теория массового обслуживания.1979.djvu
4. Рыжиков Ю. Теория очередей и управление запасами.2001.pdf
5. Алесинская Т. Основы логистики. Общие вопросы логистического управления.2005.pdf