~~Gotr~~

P. Nityasree
17MIS1007
SOFTWARE REUSE - DIGITAL ASSIGNMENT-3

④ Differences between waterfall model and Agilemodel.

Waterfall model	Agile model
→ Waterfall is linear sequential life cycle model	Agile is a continous iteration of development and testing in software development process
→ watefall is structured methodlogy	→ Agile is a flexibility model
→ water is sequential	→ Agile is a incremental
→ The methodoly comes after build phase	→ Agile concurrently with soft development
→ It has no slope of changing the req once the proj development starts	→ Agile changes in project development requirement

Advantages of waterfall :

→ It is one of easiest way to manage, works well for smaller projects.

→ Faster delivery of project

→ process and results are documented

Disadvantages of waterfall

→ not an ideal model for large size project

→ very difficult to move back to the making changes in prev phases.

Limitations Of agile

→ not useful for small projets, requires an expert to important decisions in the meeting

Agile	Waterfall
→ Prefers small but dedicated teams have higher degree of coordination	→ Team coordination/ Synchronization is very limited
→ products owner with team prepares requirements just about everyday	→ business analysis prepares the requirements befor the beginning of proj
→ description of proj details can be altered anytime during the SDLC process	→ detail description needs be implement the waterfall SDLC.

Reusability in Agile software development:

There are three ways or technologies discussed one by one below.

Component based development : CBD

→ CBD is a reusability approach that can be found in Microsoft .NET framework and J2EE component based software engineering (CBSE) process. Identifies not only candidate components.

→ Refactoring to Design patterns :

To provide a software system quality in terms of reusability, flexibility and extendibility, refactoring is significant section.

The new emerging approach that refactoring to design patterns in an applicaton increases reusability mainatability.

→ Refactoring has gained much more attention in the object oriented software development. Refactoring to patterns suggests that using patterns early in a design.

→ We improve designs with patterns by applying sequences of low level designs that can be transformed directly to code

Reusable architectures

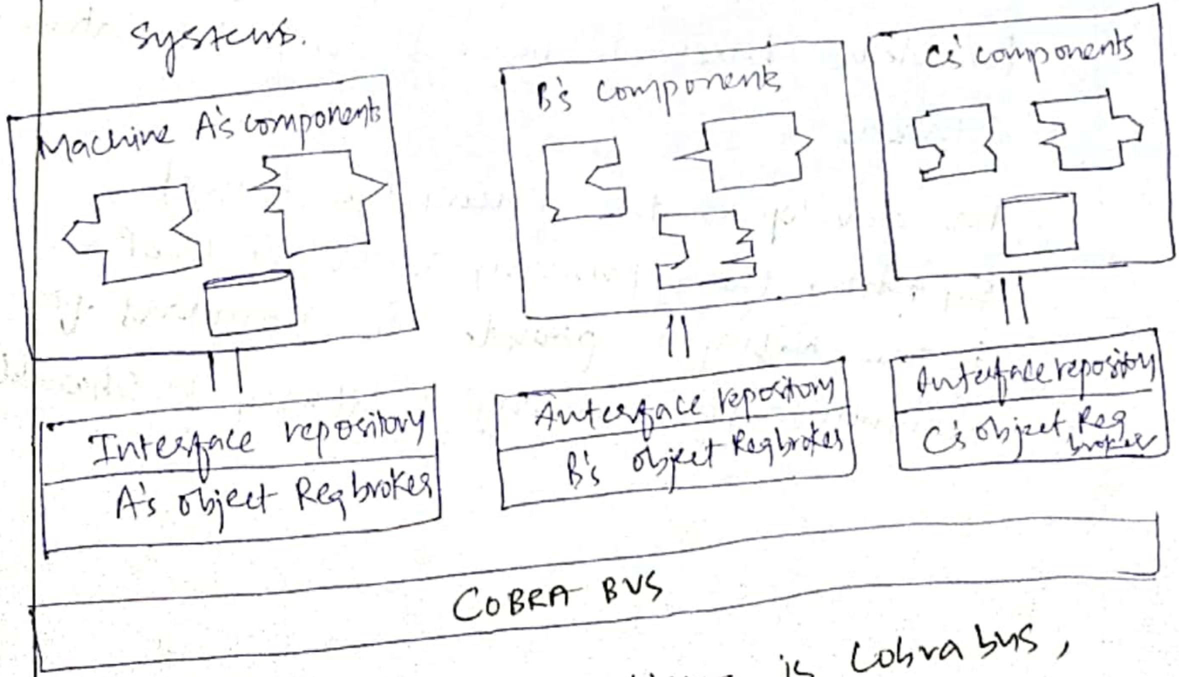
→ These can be developed from the reusable architectural patterns. They focus on how nonfunctional property reusability relates to the software architecture.

(Q) Analyze and illustrate how cobra technology can be reused with an example.

(Ans) The Object Management group approved version 2.0 of the Common object Request Broker architecture (COBRA). a Specification that describes how software components can interoperate across networks, languages, platforms. And the Cobra is primarly about specifying the component syntax and distributed implementations.

The COBRA standard should help in the distribute the cost of the building reuse libraries. Any developer who subscribes the standards can add to a world wide library of the COBRA Components.

The COBRA approach is emphasizing those ideas that are most applicable for knowledge based systems.



At the bottom of the figure is Cobra bus, representing the network of machines with COBRA compliant objects request brokers (ORBs)

As an example the interoporation across this bus, suppose there exists a knowledge bases of medical diseases, symptoms, literature citations that are linked to particular diseases.

Developer A, a c++ programmer on machine A wishes to build a special purpose statistical analysis program that looks at the relationship between the diseases.

Developer A happens to know that the developer B has already built an efficient cluster analysis routine. Thus developer A can make the remote calls to this cluster analysis routine.

And developer A is done, he or she registers the statistical analysis code as another COBRA component, so that the developers on machine C or elsewhere, can analyze the different between knowledge bases of diseases and literature citations.

For developers the knowledge based systems, the primary value of COBRA is the ability to provide the standard for communication among a library of sharable

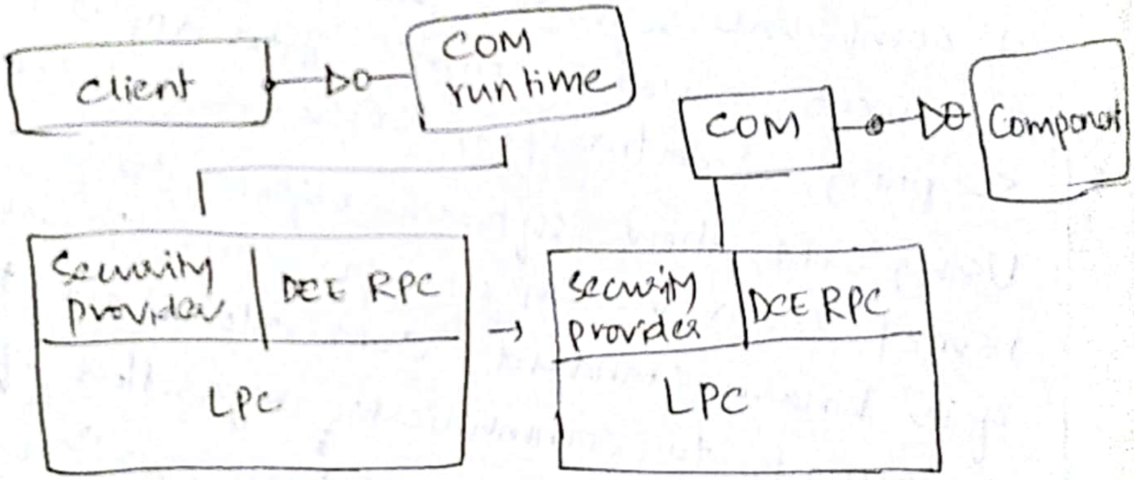
③. How COM/DCOM technology can be reused with an example.

The microsoft component object models(COM) is a computer software architecture that allows the developers to partition an application into multiple components that can be developed and installed independently each other COM is underlying architecture that forms the foundation of higher level software services. As OLE services (object linking and the Embedding). OLE services span various aspects of component software, including compound documents, custom controls, inter application scripting, data transfer, software interactions. Using COM allows software objects to be reused for a variety of application. Because, of its binary standard, COM allows any two components to communicate regardless of the language in which they were written. The Microsoft Distributed Component Object model (DCOM) is an extension of COM, and supports commonication among the objects residing on different computers such as LANS, WANS, and the internet. And with the DCOM these software objects can be reused over a distributed environment.

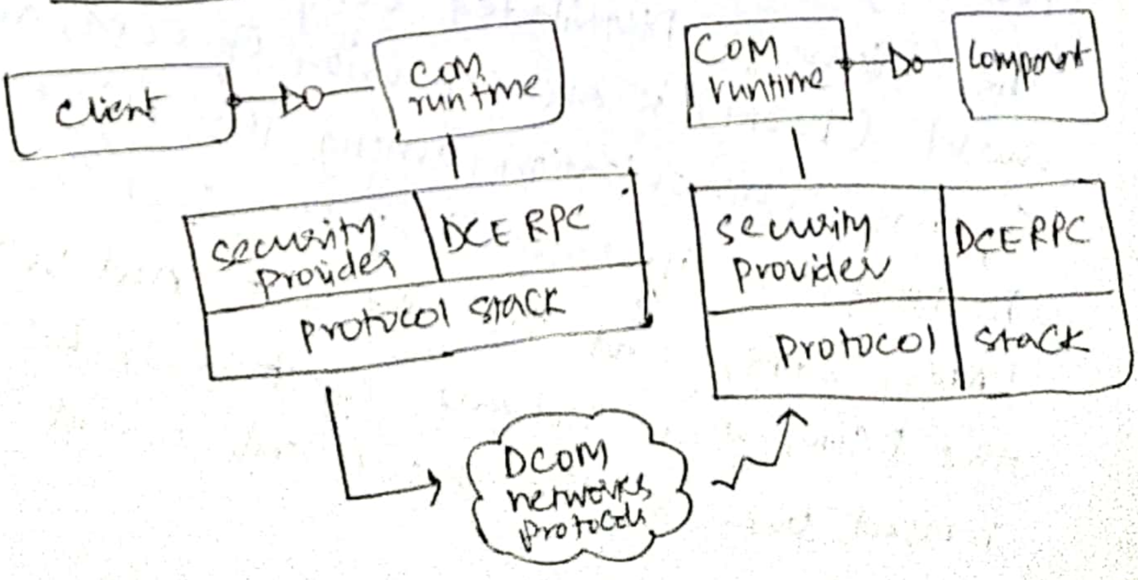
The COM objects or components are the individual modular software routines that can be reused within applications. COM objects that are reusable compiled binary objects, as opposed to reusable sections of code.

Creating an instance of COM object provides a reference through which you can access an objects functionality

COM architecture:



DCOM architecture



(1) Analyse the illustrate how javabeans Component technology is promoting the software reuse explain how all the three types of beans. [Entity, Session, Messaging] can be reused with an example.

Javabeans Components

These are the java Classes that can be easily reused and composed together into applications. Any Java class that follows certain design conventions is a Java beans Component. The java beans pages technology directly supports using java beans components with JSP lang elements. You can easily and initialize beans and get the set the values of properties.

Javabeans Component design conventions:

→ The javabeans can be a readwrite / readonly or write only which means a single value

→ A property doesnt have to be implemented by an instance variable. it must simply be accessible using public methods that confirm the following conventions.

→ For each readable property, the bean must have a method of form.

→ For each writable property, the bean must have a method of form.

Session beans:

These are the business process objects that perform actions. An action may be account, transferring funds, performing a

Calculation. Session beans consists of remote, home, and bean classes. ~~A client~~

Example: A client gets a referonce the real life objects on which its uses the remote interafce to invoke the bears methods.

Entity beans:

These are objects that represent life objects on sessions beans performs actions. Objects may include such as remote, account, employes

Ex: The Entity bean is exposed to the Client with remote interface, which the client uses invokes bears methods.

Message driven methods:

Message driven methods are messaging obj designed to route messages from clients to other Enterprise Javabeans.

Ex: A message driven bean differs from the session and the entity beans in that has no local remote. A MDB is not exposed to client at all.

The MDB implements two interfaces as

① Javax. ejb. Message Bean

② Javax. jms. Message Listener

Minimally, the MDB must implement the set Message driven Content, and the ejb Remote methos from the javax. ejb Message bean interface. In addition, the MDB must implement a container calls on the message interface.