

tesi di laurea

Tecniche di riuso del software: applicazione ad un caso di studio reale

Anno Accademico 2005-2006

relatore

Ch.mo prof. Porfirio Tramontana

correlatore

Sig. Mario Polverino

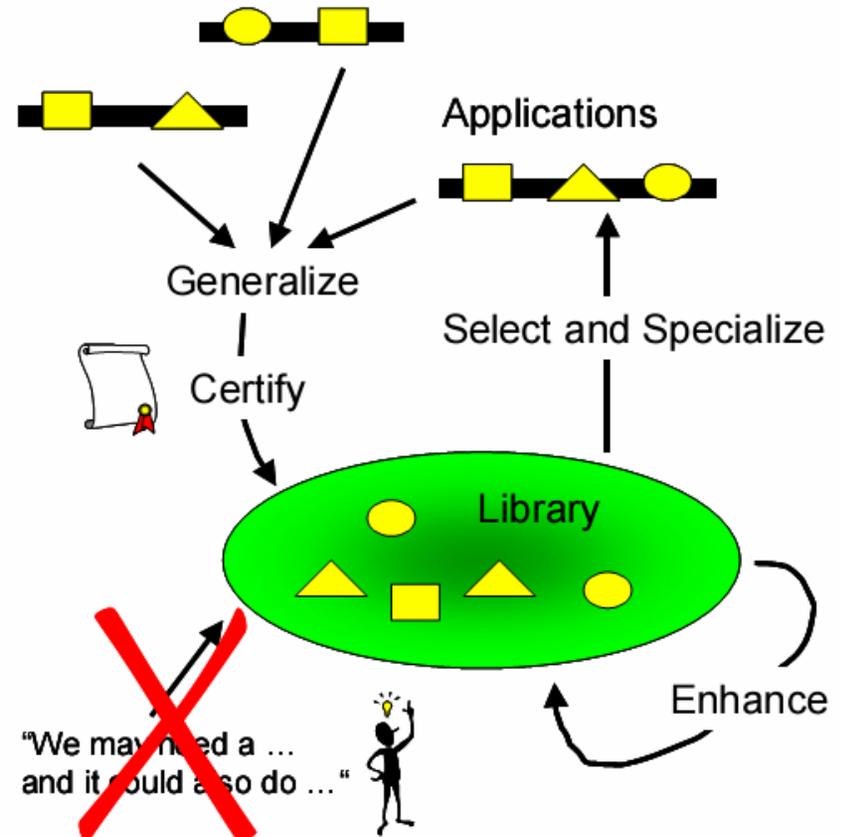
candidato

Luigi Gagliardo

Matr. 534/1767

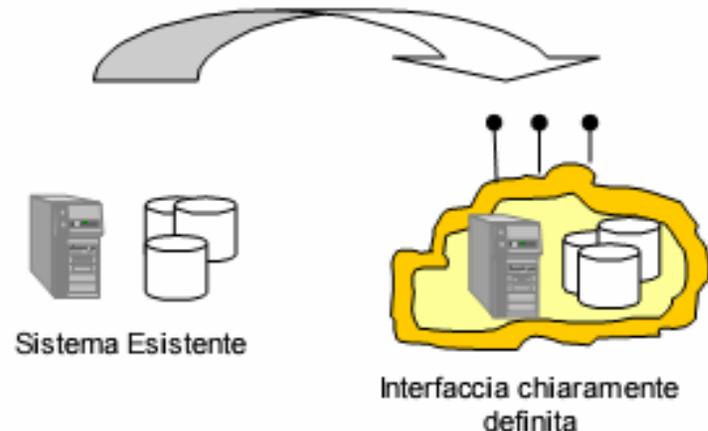
Riuso e riusabilità

- Per riuso si intende la possibilità di riusare “pezzi di software”
- Per riusabilità si intende la possibilità di sviluppare “pezzi di software” che siano riusabili
- L’obiettivo è di evidenziare i vantaggi e le tecniche del riuso, e discutere circa la riusabilità dei componenti
- Individuazione di componenti e operazioni atte alla componentizzazione di *Mail Hotel*, software sviluppato nell’Alpha Service



Tecniche di riuso dei componenti e legacy

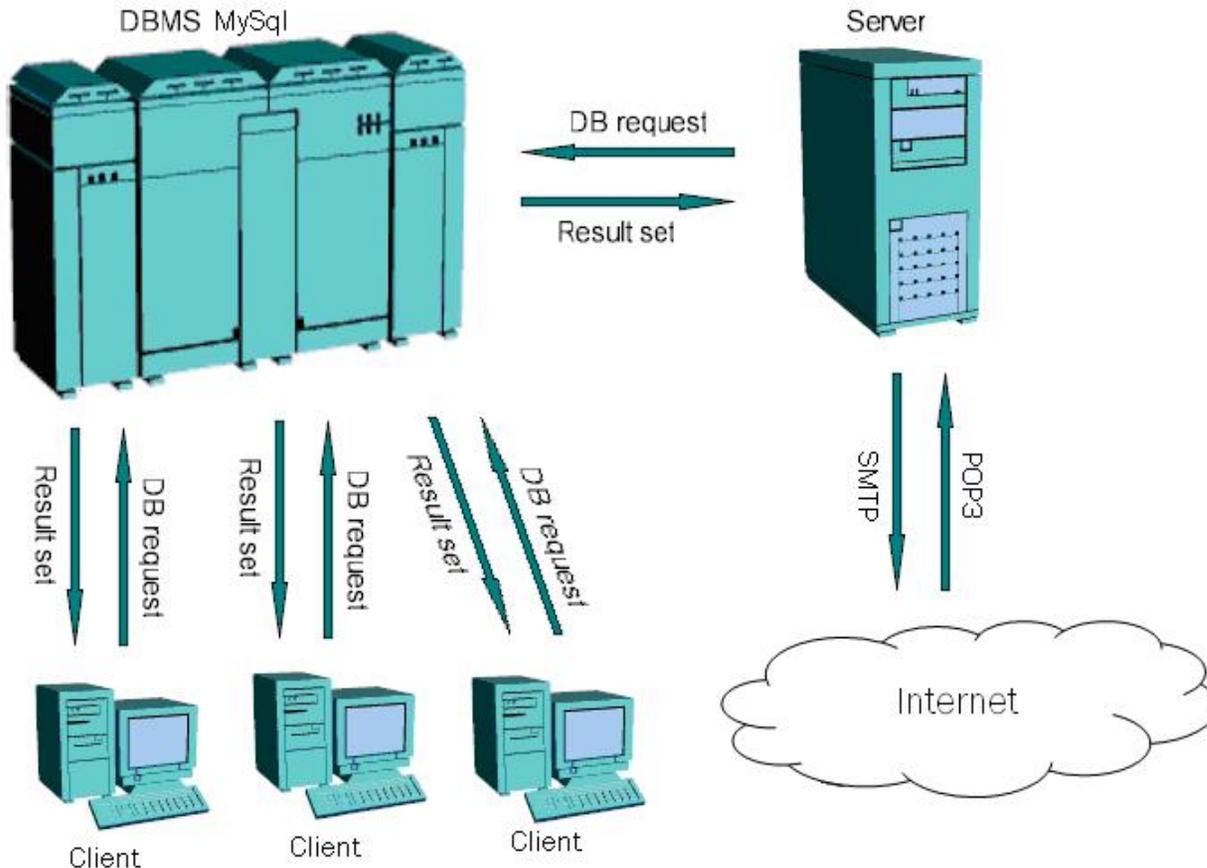
- Szyperski: “un componente è un’unità di composizione con interfacce specificate ed esplicite che può essere installato indipendentemente ed assemblato insieme ad altri componenti provenienti da terze parti”
- Sviluppo software CBD (Component-based-development)
 - Prelevare componenti da uno scaffale
 - Aggiungere componenti custom
 - Rispettare i principi di progettazione
 - Integrare i componenti
- Architettura COTS-based
 - Wrapper
 - Glue
 - Tailoring



Dai componenti ai servizi: *web services*

- **Funzionalità di business realizzata tramite un componente che rispetta un'interfaccia**
- **Unità elementare: riuso dei web services affine a quello delle funzioni**
- **Regole architettura SOA**
 - **Servizi ricercabili e recuperabili dinamicamente**
 - **Servizi autocontenuti, modulari, e stateless**
 - **Servizi con interfacce esplicite e indipendenti dall'implementazione**
 - **Servizi loosely-copled**
 - **Servizi accessibili trasparentemente rispetto all'allocazione**
 - **Servizi con interfacce a grana grossa**
 - **Servizi componibili**

Mail Hotel: overview



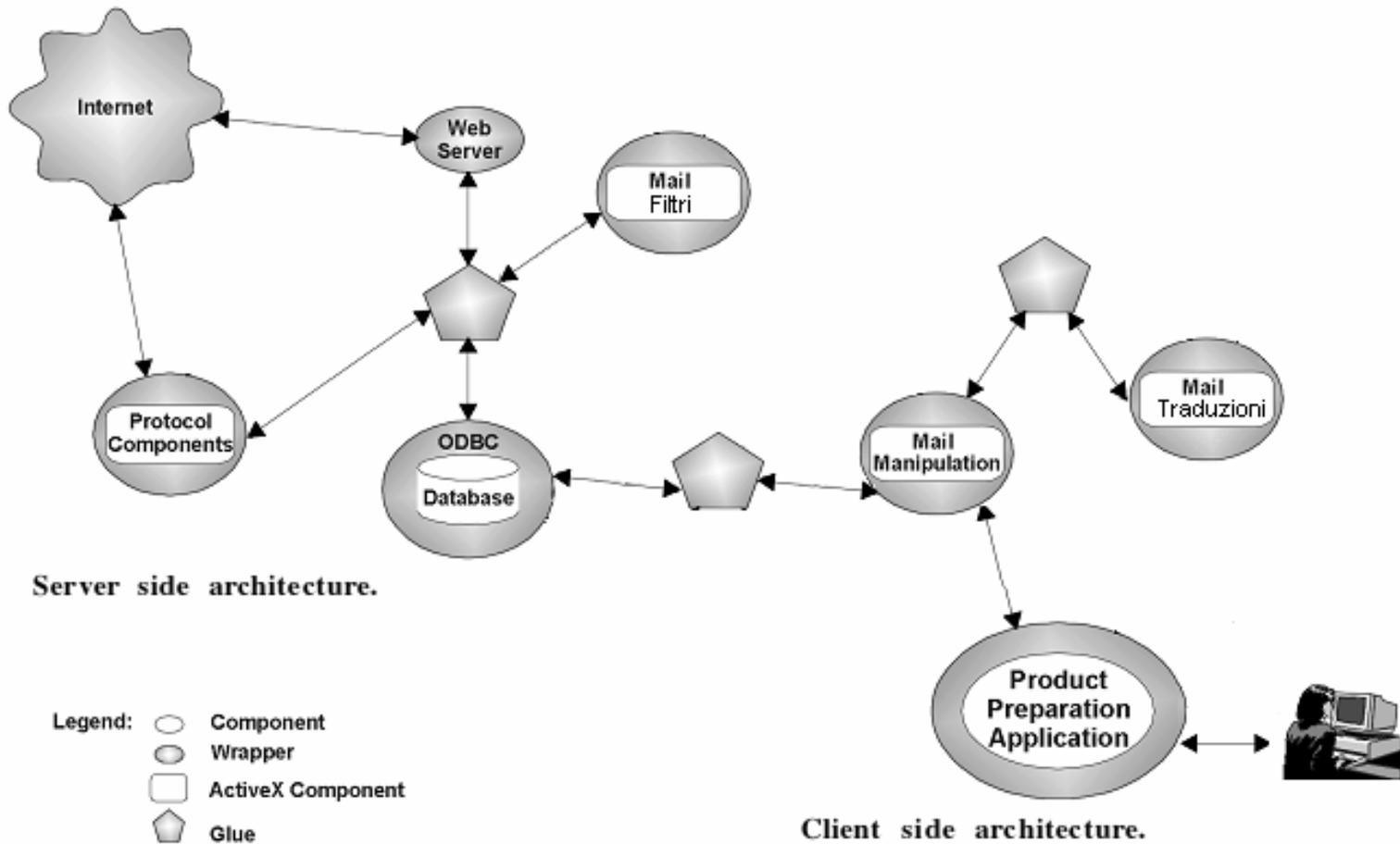
Requisiti

- Gestione e-mail e allegati
- Download e upload dei messaggi di posta
- Filtraggio delle e-mail
- Gestione delle prenotazioni e ospiti
- Gestione delle risposte delle e-mail
- Gestione degli operatori

Verifiche per l'identificazione

- **Verificare che all'interno del progetto esistano delle parti (componenti, classi, variabili) che appartengano allo stesso livello logico o area del dominio**
- **Verificare, dopo aver individuato le parti, che non esistano dipendenze ineliminabili con funzionalità (o dati) appartenenti ad altre aree del dominio**
- **Verificare che le parti candidate alla componentizzazione siano autosufficienti, ovvero che il componente sia indipendente dallo specifico contesto operativo**
- **Verificare che sia possibile definire un'interfaccia specifica, in modo da rendere più semplice l'assemblaggio del componente insieme ad altri**

Uno sguardo di insieme secondo Vidger



Componentizzazione di Mail Manipulation

- Inserire in un componente le classi `clsMail`, `clsMailPart`, i moduli per la codifica e decodifica dei messaggi MIME, e per la visualizzazione e creazione di e-mail.
- Eliminare le dipendenze che sussistono con il modulo che gestisce la comunicazione con il DB.
- Aggiungere nel componente una proprietà che identifichi l'oggetto mail visualizzato, in modo da poterlo manipolare e modificare a prescindere dall'interfaccia grafica.
- Implementare l'evento: `MailScritta` (in cui è possibile manipolare la mail che l'utente ha scritto).
- Fare in modo che il componente "comunichi" con l'ambiente esterno soltanto tramite lo scambio di stringhe in formato MIME: le funzioni devono accettare come parametri di ingresso stringhe in formato MIME e restituire in uscita stringhe in formato MIME.
- Inserire un metodo per la visualizzazione di una mail in formato MIME: ad esempio `public function VisualizzaMail(MailMIME As String)`
- Effettuare un wrapping con tecnologia OCX

Modello 3C di Tracz di Mail Manipulation

■ Concetto:

Il componente serve ad effettuare la lettura e la scrittura di e-mail con eventuali allegati

■ Contenuto:

Il componente effettua, per quanto riguarda la lettura, il parsing del messaggio in formato MIME, e “traduce” il suo contenuto, tramite i moduli aggiuntivi per la codifica e la decodifica del testo in quoted-printable (o UU) e degli allegati scritti in base64. Per quanto concerne la scrittura, il componente formatta i campi inseriti dall'utente secondo il formato MIME. È organizzato in una libreria per la decodifica base64, le classi clsMail e clsMailPart, e due moduli per la decodifica UU e quoted-printable

■ Contesto:

Il componente si può utilizzare su macchine dotate di sistema operativo Windows. È possibile usare il componente per leggere dei messaggi MIME: ad esempio da un file, oppure da una variabile di tipo stringa. È anche possibile utilizzare il componente in concomitanza con un modulo per la scrittura di file di testo, per salvare i messaggi di posta

Glue tra DB e Mail Manipulation

```
Dim ListaPOP3 As New Collection 'lista delle stringhe MIME ricevute
Dim ListaSMTP As New Collection 'lista delle stringhe MIME da inviare
Dim ListaMessaggiMIME As Collection 'lista di oggetti messaggi MIME
    '(stringaMIME + id)

Set ListaMessaggiMIME = CaricaMailNonInviolate 'carica gli oggetti MIME
    'da inviare

' carica nella listaSMTP la parte stringa degli oggetti MessMIME
Dim MessMIME As New clsMessaggioMIME
For Each MessMIME In ListaMessaggiMIME
    ListaSMTP.Add = MessMIME.Messaggio
Next

'POP3 before SMTP
Set ListaPOP3 = Download() 'Scarica le mail
Upload (ListaSMTP) 'Invia le mail ancora non inviate

'Salva le mail in arrivo
SalvaMailInArrivo ListaPOP3
```

- Funge da ponte
- Flusso di controllo

Conclusioni e sviluppi futuri

- L'obiettivo è evidenziare che lo sviluppo di software con il riuso rappresenta una risposta efficace ed efficiente alla richiesta di software di qualità
- Il riuso e la riusabilità sono obiettivi complicati
- La componentizzazione di *Mail Hotel* è un esempio di come effettuare un reengineering dei componenti
- Guardare a nuovi standard e tecnologie che permettano l'interoperabilità
- Implementare le modifiche progettate ed effettuare la migrazione tra VB6 e VB.net per una maggiore interoperabilità, portabilità e riusabilità
- Ed uno sguardo ai web services: il problema della semantica