



DocShareTool

Un tool per la condivisione
di documenti in CoFFEE



LEAD 

COOPERATIVE FACE2FACE EDUCATIONAL ENVIRONMENT



ISISLab

Contenuti



- x Introduzione
 - x Cos'è CoFFEE?
 - x CoFFEE as RCAs
 - x Un tool in CoFFEE
- x DocShareTool
 - x Motivazioni
 - x Il tool
- x Spunti e conclusioni

Contenuti



- x Introduzione
 - x Cos'è CoFFEE?
 - x CoFFEE as RCAs
 - x Un tool in CoFFEE
- x DocShareTool
 - x Motivazioni
 - x Il tool
- x Spunti e conclusioni

CoFFEE



- **Cooperative Face-2-Face Educational Environment**
 - *“CoFFEE is a suite of applications to support collaborative problem-solving discussions in the classroom. Its main components are a series of tools for collaboration, shared work, individual work and communication. Around these core tools, several other components make it possible to plan, run or participate in a CoFFEE lesson (or session).”* (from <http://www.coffee-soft.org/>)
- **Realizzato nell'ambito del progetto LEAD**
 - *“The LEAD project contributes to the development of new understandings of learning processes by exploring the links between human learning, cognition and technologies. The goal of the LEAD project is to develop, implement and evaluate conceptual models, practical scenarios and associated networked-computing technologies for effective face-to-face problem-solving discussions”* (from <http://www.lead2learning.org/>)

Contenuti



- x [Introduzione](#)
 - ✓ Cos'è CoFFEE?
- x [CoFFEE as RCAs](#)
- x Un tool in CoFFEE
- x DocShareTool
 - x Motivazioni
 - x Il tool
- x Spunti e conclusioni

CoFFEE as RCAs (1)

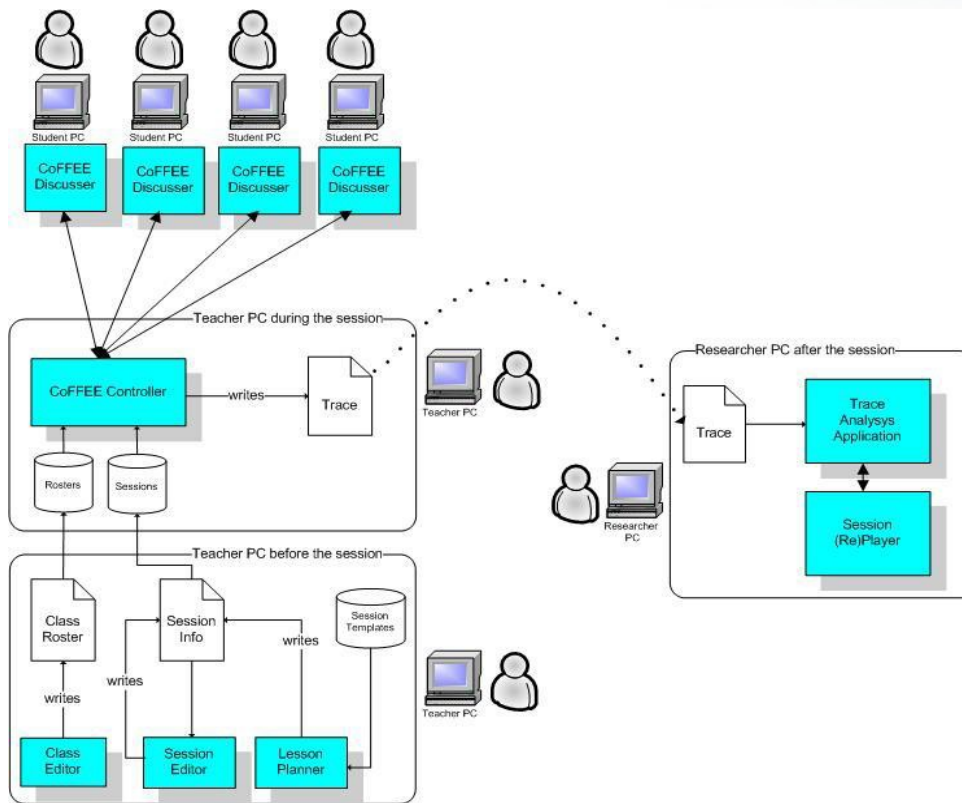


- L'architettura di CoFFEE:
 - Basata sul framework Eclipse RCP (Rich Client Platform):
 - È il core di Eclipse Platform, offerto per costruire applicazioni general-purpose
 - Contiene componenti per gestire il ciclo di vita dei plug-ins e l'interfaccia grafica
 - Una suite di Rich Client Applications:
 - CoFFEE Controller
 - CoFFEE Discusser
 - Session Editor
 - Lesson Planner
 - Class Editor
 - Replayer

CoFFEE as RCAs (2)



Ecco le varie applications di CoFFEE:



Le applicazioni principali(core components) sono due:

- CoFFEE Controller (server)
- CoFFEE Discusser (client)

CoFFEE as RCAs (3)



- Aspetti comunicativi
 - ECF (Eclipse Communication Framework)
 - “ECF is a framework for supporting the development of distributed Eclipse-based tools and applications. It can be used to create other plugins, tools, or full Eclipse RCP applications that require asynchronous point-to-point or publish-and-subscribe messaging.” (from <http://www.eclipse.org/ecf/>)
 - Communication objects: containers and shared objects
 - I primi forniscono l'accesso al protocollo di comunicazione, mentre i secondi gestiscono il messaging.

Contenuti



- x [Introduzione](#)
 - ✓ Cos'è CoFFEE?
 - ✓ CoFFEE as RCAs
- x [Un tool in CoFFEE](#)
- x DocShareTool
 - x Motivazioni
 - x Il tool
- x Spunti e conclusioni

Un Tool in CoFFEE (1)

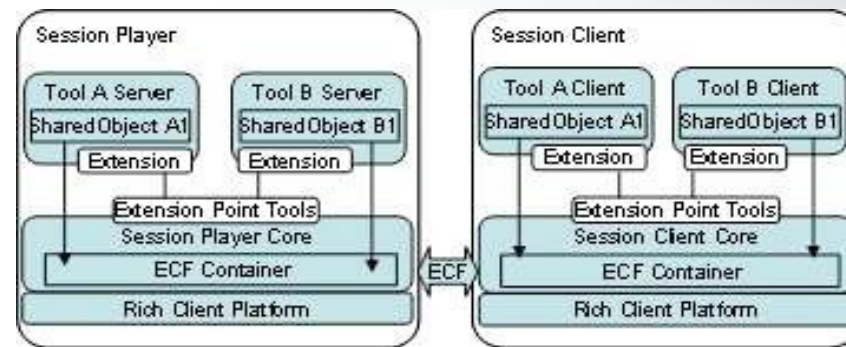


- Struttura di un tool
 - Composto da 3 plug-ins:
 - Server plug-in: “estende” il core component server (Controller)
 - Client plug-in: “estende” il core component client (Discusser)
 - Common plug-in: contiene le “shared libraries”
 - Service = <shared object, GUI, configuration>
 - Shared Object: componente di comunicazione
 - GUI: interfaccia grafica del tool associata al precedente Shared Object
 - Configuration: insieme di parametri per personalizzare l'aspetto o il comportamento del service

Un Tool in CoFFEE (2)



- La comunicazione fra client plug-in e server plug-in



- Vengono utilizzati un container per il CoFFEE Controller ed un container per il CoFFEE Discusser, mentre ogni tool utilizza gli shared objects.

Contenuti



- ✓ Introduzione
 - ✓ Cos'è CoFFEE?
 - ✓ CoFFEE as RCAs
 - ✓ Un tool in CoFFEE
- x [DocShareTool](#)
 - x [Motivazioni](#)
 - x Il tool
- x Spunti e conclusioni

Motivazioni



- Perchè abbiamo bisogno di un simile tool?



- Possibili scenari:
 - Compito in classe
 - Esercitazione in classe
- Uso generico
 - Scambio di files di tipo generico



Contenuti

- ✓ Introduzione
 - ✓ Cos'è CoFFEE?
 - ✓ CoFFEE as RCAs
 - ✓ Un tool in CoFFEE
- x [DocShareTool](#)
 - ✓ Motivazioni
 - x [Il tool](#)
- x Spunti e conclusioni

Il tool – Overview (1)



- Premesse
 - Di cosa abbiamo bisogno?
 - Una “zona comune”
 - Interfaccia grafica che permetta all'utente di scegliere quali files inviare/scaricare a/da una “zona comune”
 - Un meccanismo che permetta all'utente di sapere “dove si trova la zona comune”
 - Ed, infine, un meccanismo che permetta di “trasportare dati da un posto all'altro”
 - Traducendo...
 - Server FTP
 - SWT/JFace
 - Servizio di localizzazione
 - Protocollo di trasferimento file (FTP)

Il tool – Overview (2)



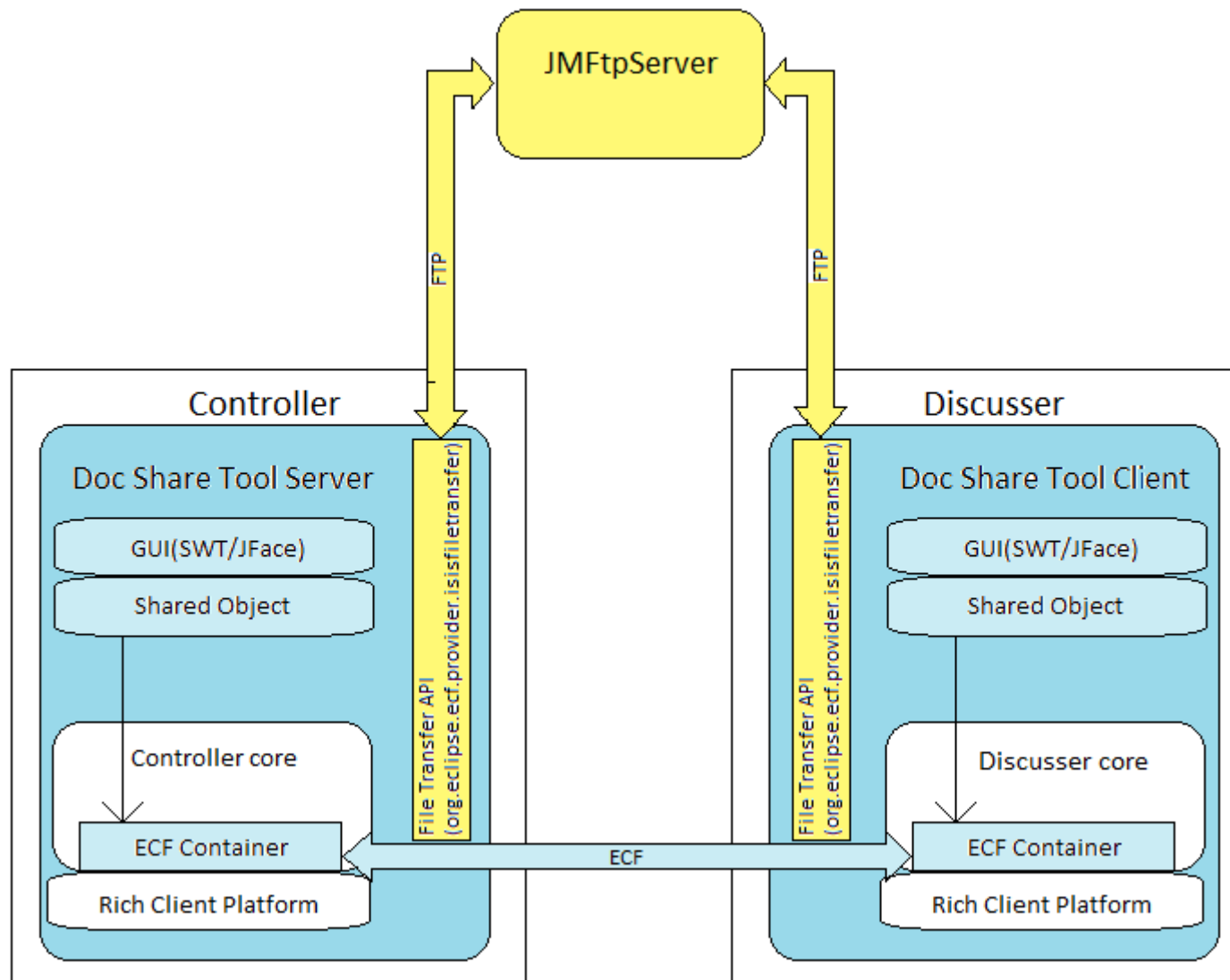
- Server FTP
 - JMFtpServer
 - Già presente in CoFFEE e realizzato da Donato Pirozzi. Le richieste al server vengono effettuate mediante l'utilizzo di un'implementazione della File Transfer API di ECF unitamente ad alcune classi java che forniscono operazioni non previste dalla suddetta API.
- SWT/JFace
 - *“SWT is an open source widget toolkit for Java designed to provide efficient, portable access to the user-interface facilities of the operating systems on which it is implemented.”* (from <http://www.eclipse.org/swt/>)

Il tool – Overview (3)



- Servizio di localizzazione
 - Basato su Shared Object e Container. La parte server del tool possiede indirizzo IP e porta del server ftp. La parte client del tool, all'avvio, fa richiesta di indirizzo e porta alla parte server, che provvede ad inviare al richiedente questa informazione.
- Protocollo FTP
 - Protocollo per il trasferimento di dati tra host basato su TCP
 - Specifica: RFC-959

II tool – Overview (4)



Il tool – Le classi (1)



- Il tool è suddiviso in 3 progetti:
 - `lead.docShareTool.client`
 - `lead.docShareTool.server`
 - `lead.docShareTool.common`
- Classi del client e del server identicamente strutturate, a parte qualche eccezione...
 - Il package `lead.docsharetool.configuration` con le relative classi si trova solo in `lead.docShareTool.server`

Il tool – Le classi (2)



- Progetto lead.docShareTool.server
 - Package lead.docsharetool.server
 - Activator.java
 - DocShareComposite.java
 - DocShareService.java
 - DocShareTool.java
 - Package lead.docsharetool.configuration
 - DocShareConfiguration.java
 - DocShareConfigurationDlg.java
 - DocShareConfigurator.java

Il tool – Le classi (3)



- Progetto lead.docShareTool.common
 - Package lead.docsharetool.common
 - Activator.java
 - DocShareToolConstants.java
 - RemoteFile.java
 - RequestAddressMessage.java
 - FTPAddressMessage.java
 - TransferUtils.java
- Ed ora un po' di codice...

Server plug-in



- Progetto lead.docShareTool.server
 - Package lead.docsharetool.server
 - Activator.java
 - DocShareComposite.java
 - [DocShareService.java](#) ←
 - DocShareTool.java
 - Package lead.docsharetool.configuration
 - DocShareConfiguration.java
 - DocShareConfigurationDlg.java
 - DocShareConfigurator.java

DocShareService.java

```
package lead.docsharetool.server;
import ...;
public class DocShareService extends LEADService {

    public DocShareService(String name, boolean isOwn,
                           ToolConfigurationProperties conf,
                           int stepNumber, ILEADContainer container) {
        super(name, isOwn, conf, stepNumber, LEADServer.getInstance());

        addMessageProcessor(this);
    }
    public synchronized boolean processToolMessage(LEADMessage m) {
        Object objmsg = m.getContent();
        if (objmsg instanceof RequestAddressMessage) {
            this.sendAddressMessage();
            return true;
        } else if ...
        ...
        return false;
    }
    public void sendAddressMessage() {
        String target = LEADServer.getInstance().getIdentifier();
        FTPAddressMessage sendable =
            new FTPAddressMessage(Activator.getFTPServerURL());
        LEADMessage leadmsg = new LEADMessage(this, target, sendable);
        leadmsg.setBroadcast(true);
        try {
            this.getMessageSender().sendMessage(leadmsg);
        } catch (IOException e) {
            e.printStackTrace();
        }
        ...
        ...
    }
}
```

Costruttore:

viene richiamato il costruttore della superclasse ed aggiunto un processore di messaggi a questo contesto.

processToolMessage(LEADMessage):

processa i messaggi ricevuti. Se il LEADMessage ricevuto contiene una richiesta di invio dell'indirizzo del server FTP...

...viene invocato il metodo

sendAddressMessage, che, recupera

l'id dello shared object container

del server, crea il messaggio con

l'indirizzo richiesto e lo invia al client..

Client plug-in



- Progetto lead.docShareTool.client
 - Package lead.docsharetool.client
 - Activator.java
 - DocShareComposite.java
 - [DocShareService.java](#)
 - DocShareTool.java





DocShareService.java

```
package lead.docsharetool.client;
import ...;
public class DocShareService extends LEADService {

    public DocShareService(String name, boolean isOwn,
                           ToolConfigurationProperties conf,
                           int stepNumber, ILEADContainer container) {
        super(name, isOwn, conf, stepNumber, LEADClient.getInstance());
        addMessageProcessor(this);
        this.sendRequestAddressMessage();
    }
    public void sendRequestAddressMessage() {
        String target = LEADClient.getInstance().getIdentifier();
        RequestAddressMessage sendable =
            new RequestAddressMessage();
        LEADMessage leadmsg = new LEADMessage(this, target, sendable);
        try {
            this.getMessageSender().sendMessage(leadmsg);
        } catch (IOException e) {
            e.printStackTrace();
        }
    }
    public synchronized boolean processToolMessage(LEADMessage m) {
        Object objmsg = m.getContent();
        if (objmsg instanceof FTPAddressMessage) {
            FTPAddressMessage msg = (FTPAddressMessage) objmsg;
            Activator.setFTPServerURL(msg.getAddress());
            return true;
        } else if ...
        ..
        ..
        return false;
    }
    ..
    ..
    ..
}
```

Costruttore:

viene richiamato il costruttore della superclasse, aggiunto un processore di messaggi a questo contesto ed inviato un messaggio per la richiesta dell'indirizzo del server ftp.

sendRequestAddressMessage():

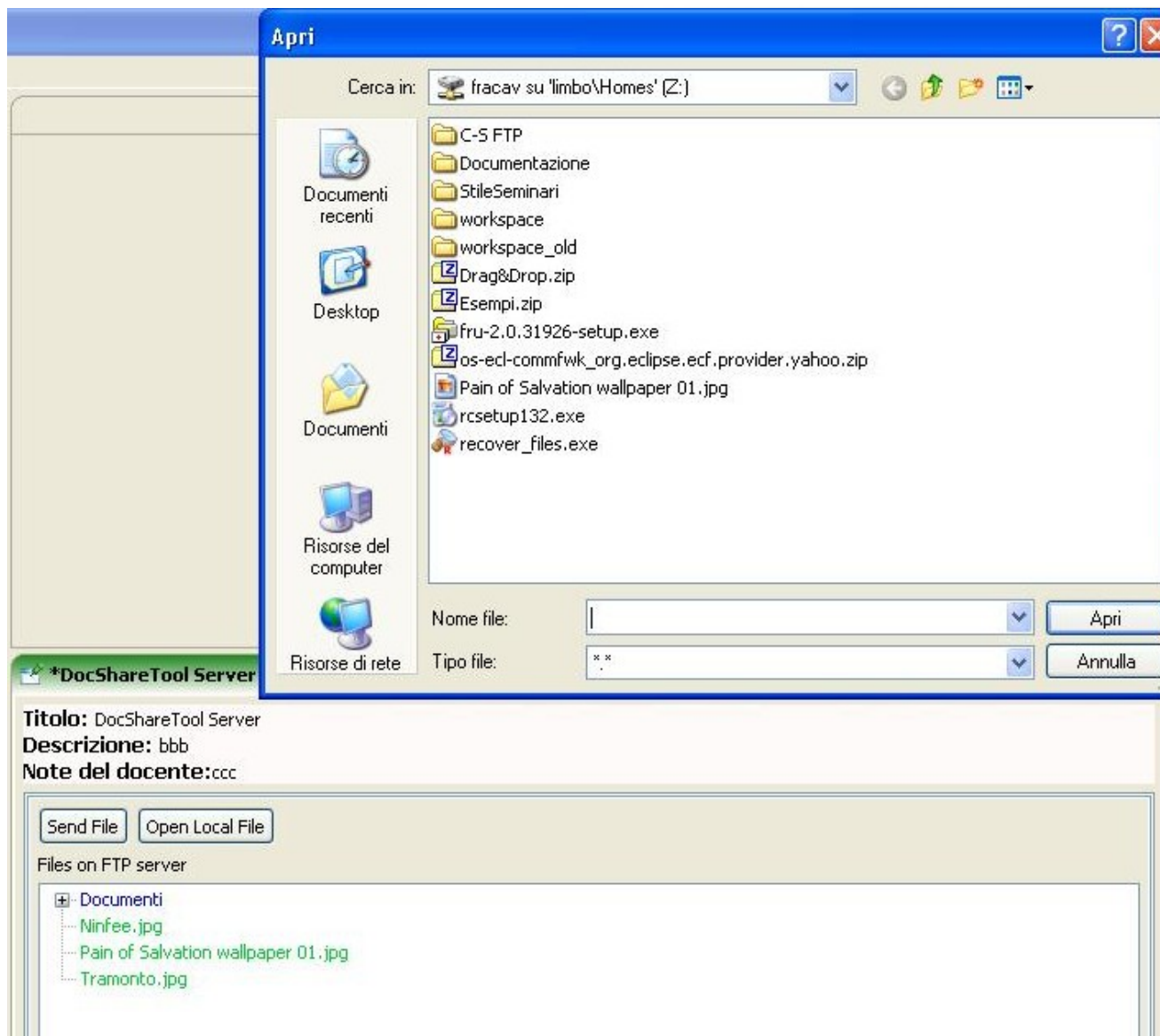
invia al server una richiesta per il recupero dell'indirizzo del server ftp.

processToolMessage(LEADMessage):

processa i messaggi ricevuti. Se il LEADMessage ricevuto contiene l'indirizzo del server FTP, quest'ultimo viene recuperato e memorizzato.



GUI – Tree View



- Bottoni:
 - Send File
 - Open Local File

Client plug-in



- Progetto lead.docShareTool.client
 - Package lead.docsharetool.client
 - Activator.java
 - [DocShareComposite.java](#) ←
 - DocShareService.java
 - DocShareTool.java

...vediamo come si invia un file...

DocShareComposite.java

```
package lead.docsharetool.client;
import ...;
public class DocShareComposite extends LEADToolUI {

    public DocShareComposite(Composite parent, DocShareService service,
                             boolean showArtifacts) {
        super(parent, showArtifacts);
        docShareService = service;
        init(this);
    }

    private void createSendButton(DocShareComposite composite,
                                  final Shell shell) {
        Button sendButton = new Button(composite, SWT.PUSH);
        sendButton.setText("Send File");
        sendButton.addSelectionListener(new SelectionAdapter() {
            public void widgetSelected(SelectionEvent event) {
                FileDialog fdlg = new FileDialog(shell);
                String path = fdlg.open();
                if(path != null) {
                    File file = new File(path);
                    IsisProtocolOutgoingAdapter adapter =
                        (IsisProtocolOutgoingAdapter)
                            TransferUtils.getOutgoingFileTransferContainerAdapter();
                    FileTransferNamespace name = new FileTransferNamespace();
                    ID id = null;
                    String pathUrl = "";
                    final TreeItem[] selection = filesTree.getSelection();
                    ...
                }
            }
        });
    }
}
```

Costruttore:

richiama il costruttore della superclasse, inizializza la variabile d'istanza docShareService ed inizializza l'interfaccia.

createSendButton:

crea il pulsante per l'invio dei files ed aggiunge un listener.

Viene aperta una finestra di dialogo per selezionare il file da inviare.

Viene recuperata un'istanza di una classe che implementa l'interfaccia `ISendFileTransferContainerAdapter`, creato un `FileTransferNamespace` e recuperata la selezione sull'albero.

DocShareComposite.java



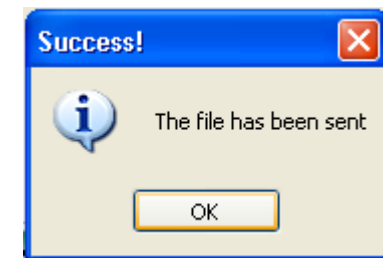
```
...
...
if(selection.length == 1){
    RemoteFile rfile = (RemoteFile)selection[0].getData();
    pathUrl = rfile.getPath().replaceAll("\\\\", "/") +
    (rfile.getPath().equals("")?"":"/");
    if(rfile.isDirectory())
        pathUrl += rfile.getName() + "/" + file.getName();
    else pathUrl += file.getName();
} else pathUrl = file.getName();
removableFiles.add(pathUrl.replaceAll("/", "\\\\"));
try {
    id = (ID)name.createInstance(new Object[]
        {Activator.getFTPServerURL() + pathUrl});
} catch (IDCreateException e) {
    e.printStackTrace();
}
if(adapter != null) {
    try {
        adapter.sendOutgoingRequest(id, file,
            new IFileTransferListener() {
        public void handleTransferEvent(IFileTransferEvent arg0)
        {
            if(arg0 instanceof IOutgoingFileTransferSendDoneEvent){
                final Display d =
                    PlatformUI.getWorkbench().getDisplay();
                d.asyncExec(new Runnable() {
                public void run() {
                    MessageBox messageBox = new MessageBox(shell,
                        SWT.ICON_INFORMATION | SWT.OK);
                    messageBox.setText("Success!");
                }
            });
            }
        }
    });
}
}
```

Calcolo dell'URL del file

Viene aggiunto il path
alla lista removableFiles
(non presente dal lato server!!!)

Creazione dell'ID

Richiesta di invio file



DocShareComposite.java

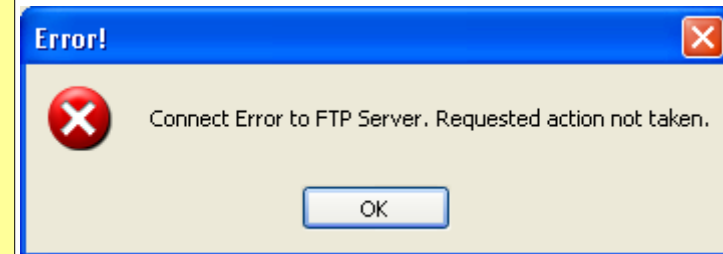


```
...
...
messageBox.setMessage("The file has been sent");
messageBox.open();

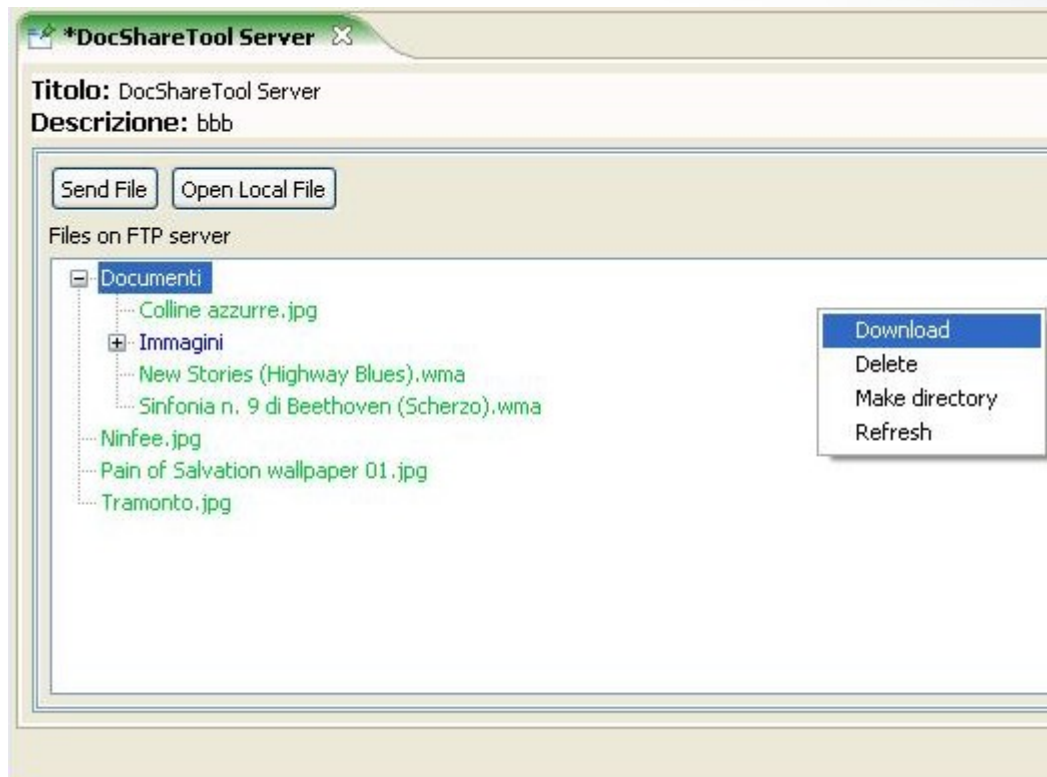
//aggiornamento della vista ad albero

} //end run
} //end new Runnable()
); //end asyncExec
} //end if
} //end handleTransferEvent
} //end new IfileTransferListener
); //end sendOutgoingRequest
} //end try
catch(final Exception e){
    final Display d = PlatformUI.getWorkbench()
                                .getDisplay();

    d.asyncExec(new Runnable(){
    public void run() {
        MessageBox messageBox = new MessageBox(shell,
                                                SWT.ICON_ERROR | SWT.OK);
        messageBox.setText("Error!");
        messageBox.setMessage(e.getMessage());
        messageBox.open();
    }
    }
    ...
    ...
    ...
```



GUI – Pop-up Menu



- Selezione multipla per:
 - Download
 - Delete
- Refresh

Client plug-in



- Progetto lead.docShareTool.client
 - Package lead.docsharetool.client
 - Activator.java
 - [DocShareComposite.java](#) ←
 - DocShareService.java
 - DocShareTool.java

...vediamo come si crea una directory...

DocShareComposite.java



```
private Menu createMenu(final Shell shell){
    ..
    ..
    final MenuItem mkdirItem = new MenuItem(menu, SWT.PUSH);
    mkdirItem.setText("Make directory");
    mkdirItem.setEnabled(false);
    mkdirItem.addSelectionListener(new SelectionListener() {
    public void widgetDefaultSelected(SelectionEvent e) {}
    public void widgetSelected(SelectionEvent e) {
        TreeItem[] selection = filesTree.getSelection();
        RemoteFile rfile = null;
        if(selection.length == 1)
            rfile = (RemoteFile)selection[0].getData();
        InputDialog dialog = new InputDialog(shell, "",
            "Insert the name of the new directory", "", null);
        dialog.open();
        String dirName = dialog.getValue();
        if(dirName != null) {
            String path = "";
            if(rfile != null) {
                path = rfile.getPath() +
                    + (rfile.getPath().equals(""))? "" : SEPARATOR;
                if(rfile.isDirectory())
                    path += rfile.getName() + SEPARATOR + dirName;
                else path += dirName;
            }else path = dirName;
            removableFiles.add(path);
            TransferUtils.makeDirectory(Activator.getFTPServerURL(), path);
            refresh();
        }
    }
});
```

Crea un menu item e lo aggiunge
al pop-up menu.

Recupera la selezione dall'albero,
chiede all'utente di inserire il
nome della directory da creare,
ne calcola il path ed invia
la richiesta al server ftp

Client plug-in



- Progetto lead.docShareTool.client
 - Package lead.docsharetool.client
 - Activator.java
 - [DocShareComposite.java](#) ←
 - DocShareService.java
 - DocShareTool.java

...vediamo come si rimuove un file o una directory...

DocShareComposite.java



```
    . .
    . .
private Menu createMenu(final Shell shell) {
    Menu menu = new Menu(fileTree);
    . .
    . .
    final MenuItem deleteItem = new MenuItem(menu, SWT.PUSH);
    deleteItem.setText("Delete");
    deleteItem.setEnabled(false);
    deleteItem.addSelectionListener(new SelectionListener() {
        public void widgetDefaultSelected(SelectionEvent e) { }
        public void widgetSelected(SelectionEvent e) {
            TreeItem[] selection = fileTree.getSelection();
            boolean toRefresh = false;
            for(int i = 0; i < selection.length; i++){
                RemoteFile rfile = (RemoteFile)selection[i].getData();
                String pathname = (rfile.getPath().equals("")) ? "" :
                    rfile.getPath() + SEPARATOR + rfile.getName();
                if(removableFiles.contains(pathname)) {
                    TransferUtils.delete(Activator.getFTPServerURL(),
                                            pathname, rfile.isDirectory());
                    removableFiles.remove(pathname);
                    toRefresh = true;
                }
            }
            if(toRefresh)
                refresh();
        }
    });
    . .
    . .
}
```

Recupera la selezione dall'albero,
per ogni file selezionato calcola
il path e, se può essere rimosso,
viene inviata la richiesta di
rimozione al server ftp.

Client plug-in



- Progetto lead.docShareTool.client
 - Package lead.docsharetool.client
 - Activator.java
 - [DocShareComposite.java](#) 
 - DocShareService.java
 - DocShareTool.java

...vediamo come si scaricano files e directories...

DocShareComposite.java



```
        : :  
        : :  
private Menu createMenu(final Shell shell) {  
    Menu menu = new Menu(filesTree);  
    final MenuItem downloadItem = new MenuItem(menu, SWT.PUSH);  
    downloadItem.setText("Download");  
    downloadItem.setEnabled(false);  
    downloadItem.addSelectionListener(new SelectionListener() {  
    public void widgetDefaultSelected(SelectionEvent e) {}  
    public void widgetSelected(SelectionEvent event) {  
        TreeItem[] selection = filesTree.getSelection();  
        for(int i = 0; i < selection.length; i++) {  
            RemoteFile rfile = (RemoteFile)selection[i].getData();  
            IsisProtocolRetrieveAdapter retrFileTransf =  
                (IsisProtocolRetrieveAdapter)TransferUtils.  
                    getRetrieveFileTransferFactory().newInstance();  
            downloadFile(shell, rfile, retrFileTransf);  
        }  
        MessageBox messageBox = new MessageBox(shell,  
                                                    SWT.ICON_INFORMATION | SWT.OK);  
        messageBox.setText("");  
        messageBox.setMessage("Done!");  
        messageBox.open();  
    }  
    });  
        : :  
        : :  
        : :  
        : :  
}
```

Recupera la selezione dall'albero,
per ogni file selezionato, viene
Invocato il metodo downloadFile.



DownloadFile() – pseudocodice

- Firma
 - `private void downloadFile(final Shell shell, RemoteFile rfile, IsisProtocolRetrieveAdapter retrFileTransfer);`
- Algoritmo:
 - Se rfile è un file, allora invia una richiesta di download mediante retrFileTransfer e termina;
 - Altrimenti(rfile è una directory), recupera tutti i files presenti nella directory e invoca ricorsivamente downloadFile per ogni file della directory.



Contenuti

- ✓ Introduzione
 - ✓ Cos'è CoFFEE?
 - ✓ CoFFEE as RCAs
 - ✓ Un tool in CoFFEE
- ✓ DocShareTool
 - ✓ Motivazioni
 - ✓ Il tool
- x Spunti e conclusioni

Spunti e Conclusioni (1)



- Riassumendo:
 - DocShareTool consente di effettuare:
 - Navigazione all'interno della root del server ftp
 - Download multiplo di cartelle e files
 - Invio di file
 - Rimozione multipla di files e cartelle
- Spunti:
 - Modifica della modalità di expand di una cartella nella vista ad albero (strettamente legato all'aggiornamento della vista dopo l'invio di un file)
 - Bottone per il recupero dei files all'avvio del tool

Spunti e Conclusioni (2)



- Ciò che manca...
 - Gestione dei gruppi
- Concludendo...
 - FileTransferTool: il lavoro su DocShareTool è stato inutile?
 - Ulteriori sviluppi futuri:
 - Interfaccia grafica avanzata
 - Utilizzo di server ftp diversi da JMFtpServer



GRAZIE PER L'ATTENZIONE!

