

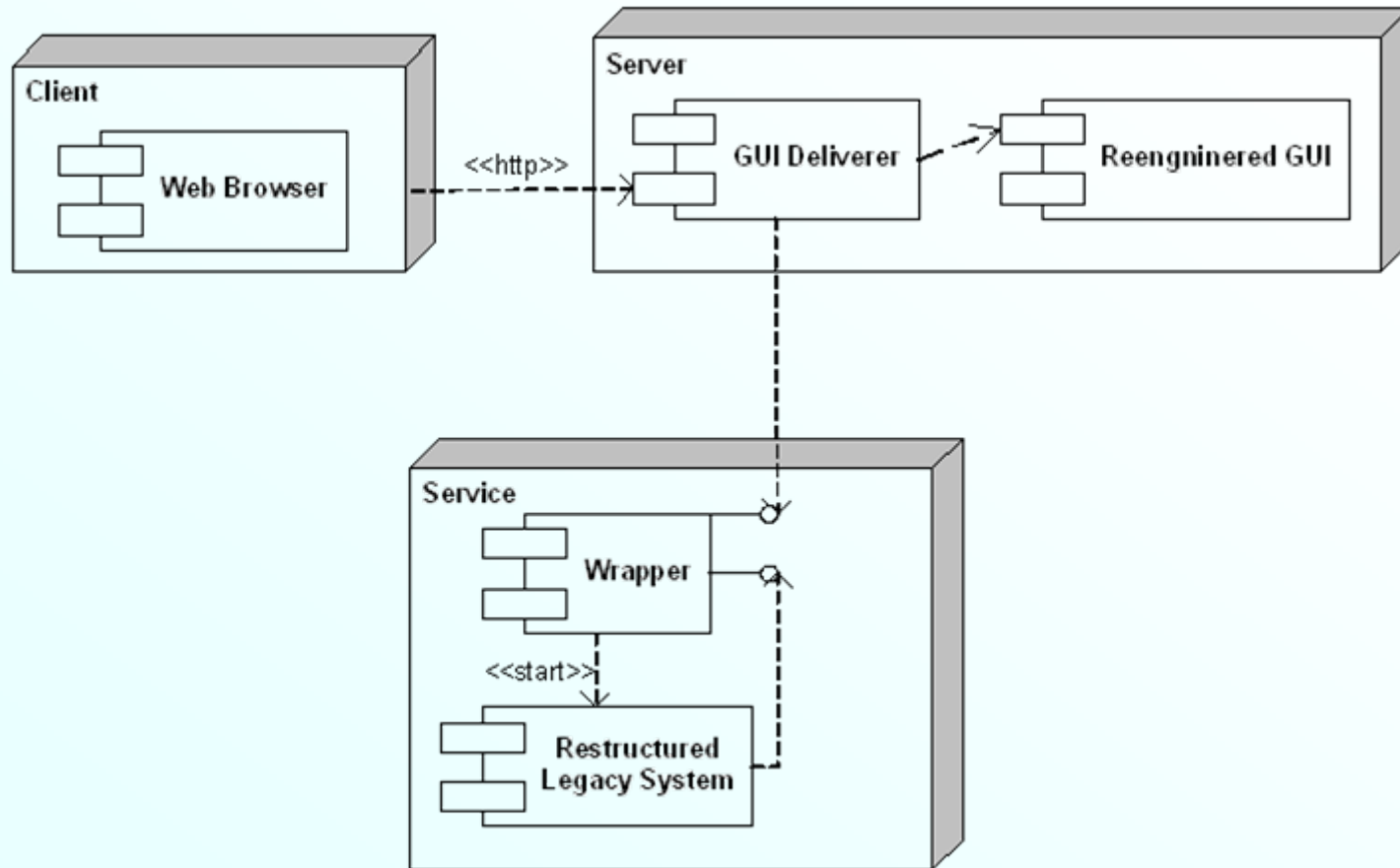
Esperimento controllato

Supervisor

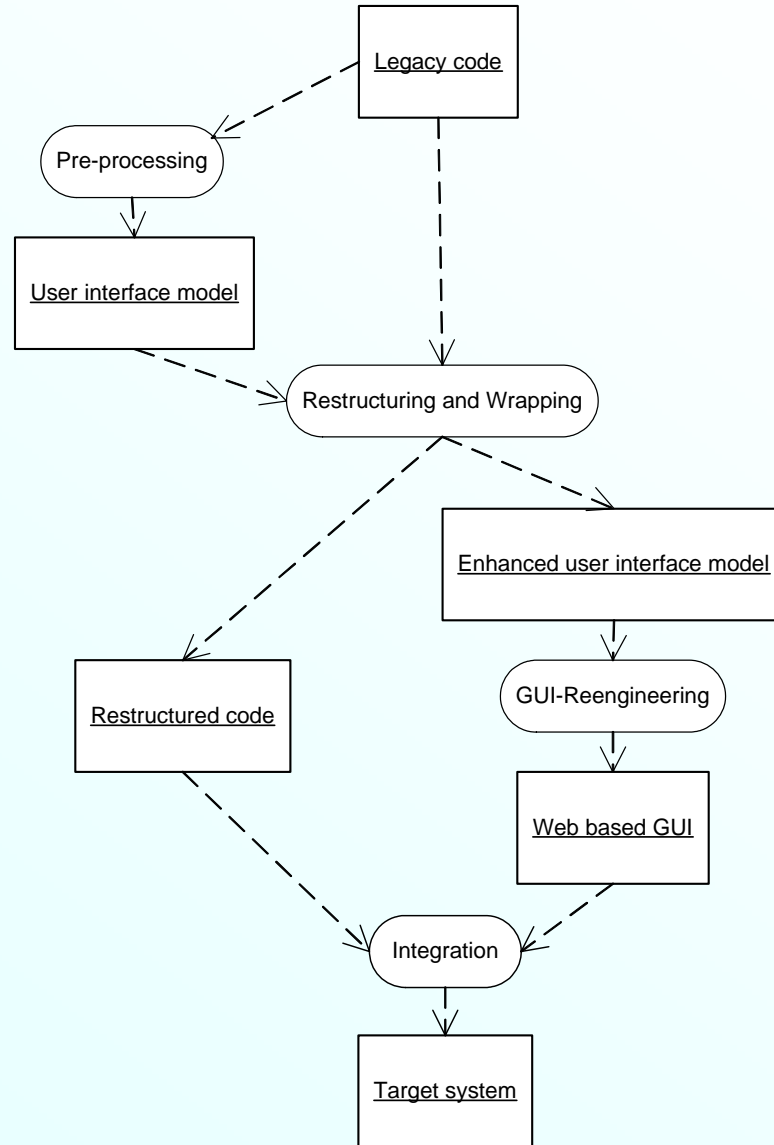
Giuseppe Scanniello
gscanniello@unisa.it

Massimo Colosimo
massimo.colosimo@gmail.com

Architettura Target



Strategia di Migrazione



Interfacce del Wrapper

Accesso al wrapper lato COBOL

E	Ordinal ^	Hint	Function
<input type="checkbox"/>	1 (0x0001)	9 (0x0009)	errore
<input type="checkbox"/>	2 (0x0002)	11 (0x000B)	setStatus
<input type="checkbox"/>	3 (0x0003)	10 (0x000A)	memvarCobol
<input type="checkbox"/>	4 (0x0004)	7 (0x0007)	crea_File
<input type="checkbox"/>	5 (0x0005)	6 (0x0006)	creaFile
<input type="checkbox"/>	6 (0x0006)	8 (0x0008)	displayCobol
<input type="checkbox"/>	7 (0x0007)	2 (0x0002)	Cblwait
<input type="checkbox"/>	8 (0x0008)	0 (0x0000)	Cblexit
<input type="checkbox"/>	9 (0x0009)	1 (0x0001)	Cblsync
<input type="checkbox"/>	10 (0x000A)	4 (0x0004)	Java_Prova_Exit
<input type="checkbox"/>	11 (0x000B)	3 (0x0003)	Java_Prova_Create
<input type="checkbox"/>	12 (0x000C)	5 (0x0005)	Java_Prova_Sync

Accesso al wrapper lato JAVA

Task senza l'utilizzo di MELIS

NO-MELIS: Pre-Processing

```
77 PLUTO pic X(20).  
77 PIPPONUM pic 9(2).  
.....
```

Dichiarazione variabile PLUTO di tipo STRINGA di 20 caratteri
Dichiarazione variabile PIPPONUM di tipo INTERO a due cifre

```
SCREEN SECTION.
```

```
01 SCREEN-1 , exception esci.  
02 label "ciao! "  
02 label PLUTO  
02 entry-field using PIPPONUM  
   after after-PIPPONUM.  
   before before-PIPPONUM.
```

LABEL

LABEL DINAMICA

ENTRY-FIELD creato usando la variabile PIPPONUM. E' presente una procedura AFTER ed una procedura BEFORE.

NO-MELIS: Restr&Wrapp

```
1 after-PIPPONUM SECTION.  
2 if PIPPONUM > 5 or PIPPONUM < 15  
3   move "CIAO" to ALERT  
4 exit.
```

call esterna alla routine ALERT

```
1 before-PIPPONUM SECTION.  
2 readFile(fileDB)  
3 exit.
```

lettura dati da un file

NO-MELIS: Restr&Wrapp

WORKING-STORAGE SECTION.

```
77 NOME-PROG PIC X(22) VALUE 'provaSW'.  
77 NOME-SCREEN PIC X(20).  
77 RET-DLL UNSIGNED-INT.  
77 SESSION PIC X(33).  
77 op-dll PIC X(7) VALUE 'isview'.  
77 ret-x PIC X.  
77 nome-file PIC X(40) VALUE 'c:\pass.txt'.
```

file che conterrà i dati
provenienti dalle
interfacce grafiche
reingegnerizzate

Variabili di supporto
per l'ENTRY-FIELD
PIPPONUM

```
77 name-PIPPONUM PIC X(100) VALUE 'PIPPONUM'.  
77 value-PIPPONUM PIC X(100).
```

Variabili di
supporto per la
LABEL DINAMICA
PLUTO

```
77 name-PLUTO PIC X(100) VALUE 'PLUTO'.  
77 value-PLUTO PIC X(100).
```

NO-MELIS: Restr&Wrapp

la funzione `PASSAGGIODATI-SCREEN-1` sostituisce la call `display screen-1`

** display screen-1*

```
perform PASSAGGIODATI-SCREEN-1.  
MOVE 'SCREEN-1' to NOME-SCREEN.  
MOVE x'00' to NOME-PROG (8:1).  
MOVE x'00' TO NOME-SCREEN (9:1).  
MOVE 'insert' to op-dll.  
MOVE x'00' to op-dll (7:1).  
set environment 'DLL_CONVENTION' to "1".  
call 'c:\ProvaSem.dll'.  
call 'displayCobol' using  
  by content SESSION  
  by content NOME-PROG  
  by content NOME-SCREEN  
  by content op-dll  
  giving RET-DLL.  
cancel 'c:\ProvaSem.dll'.  
set environment 'DLL_CONVENTION' to "0".
```

inserimento carattere di fine stringa `x'00'` nella posizione 8 perché 7 è il numero di lettere che compongono il nome "provaSW" dato al programma

chiamata al WRAPPER

by content: parametri funzione
giving: valore di ritorno

uso della funzione `displayCobol` del wrapper: in questo momento il sistema JAVA sta elaborando i dati appena inviati

la funzione `LETTURADATI-SCREEN-1` sostituisce la call `accept screen-1`

** accept SCREEN-1 on exception continue.*

```
perform LETTURADATI-SCREEN-1.
```

NO-MELIS: Restr&Wrapp

PASSAGGIODATI-SCREEN-1 SECTION.

```
set environment 'DLL_CONVENTION' to "1".
call 'c:\ProvaSem.dll'
call "creaFile" giving RET-DLL.
if RET-DLL not = 1
```

chiamata al WRAPPER

```
    * insert error
end-if.
```

carattere di fine stringa in posizione 9
perché PIPPONUM ha 8 lettere

```
move x'00' to name-PIPPONUM(9:1).
move 0 to RET-DLL.
move PIPPONUM to value-PIPPONUM.
move x'00' to value-PIPPONUM (100:1).
```

inserisco l'eventuale valore
di PIPPONUM nella
variabile di supporto

la funzione *memvarCobol*
permette di memorizzare nel
file var.txt una linea del tipo
PIPPONUM, valore

```
call 'memvarCobol' using
    by content name-PIPPONUM
    by content value-PIPPONUM giving RET-DLL.
if RET-DLL not = 1
    *insert error
end-if.
```

```
move x'00' to name-PLUTO(6:1).
move 0 to RET-DLL.
move PLUTO to value-PLUTO.
move x'00' to value-PLUTO (100:1).
```

la funzione *memvarCobol*
permette di memorizzare nel
file var.txt una linea del tipo
PLUTO, valore

```
call 'memvarCobol' using
    by content name-PLUTO
    by content value-PLUTO giving RET-DLL.
if RET-DLL not = 1
    *insert error
end-if.
```

```
.....
cancel 'c:\ProvaSem.dll'.
set environment 'DLL_CONVENTION' to "0".
EXIT.
```

chiusura comunicazione con
il wrapper e fine della
SECTION di supporto
PASSAGGIODATI

NO-MELIS: Restr&Wrapp

LETTURADATI-SCREEN-1 SECTION.

```
move 'N' TO ret-x.  
move '1' to name-PIPPONUM (9:1).  
call 'c:\leggInput.wob' using  
  nome-file  
  name-PIPPONUM
```

la funzione *leggInput* è stata appositamente creata per leggere dal file *pass.txt* i valori passati dal sistema JAVA. Prende in INPUT il nome del file ed il nome del campo e restituisce in

INPUT il suo valore ed
un flag booleano

! CONVERT

poiché PIPPONUM è di tipo intero, è necessario convertire il valore in arrivo da stringa ad intero con la clausola CONVERT

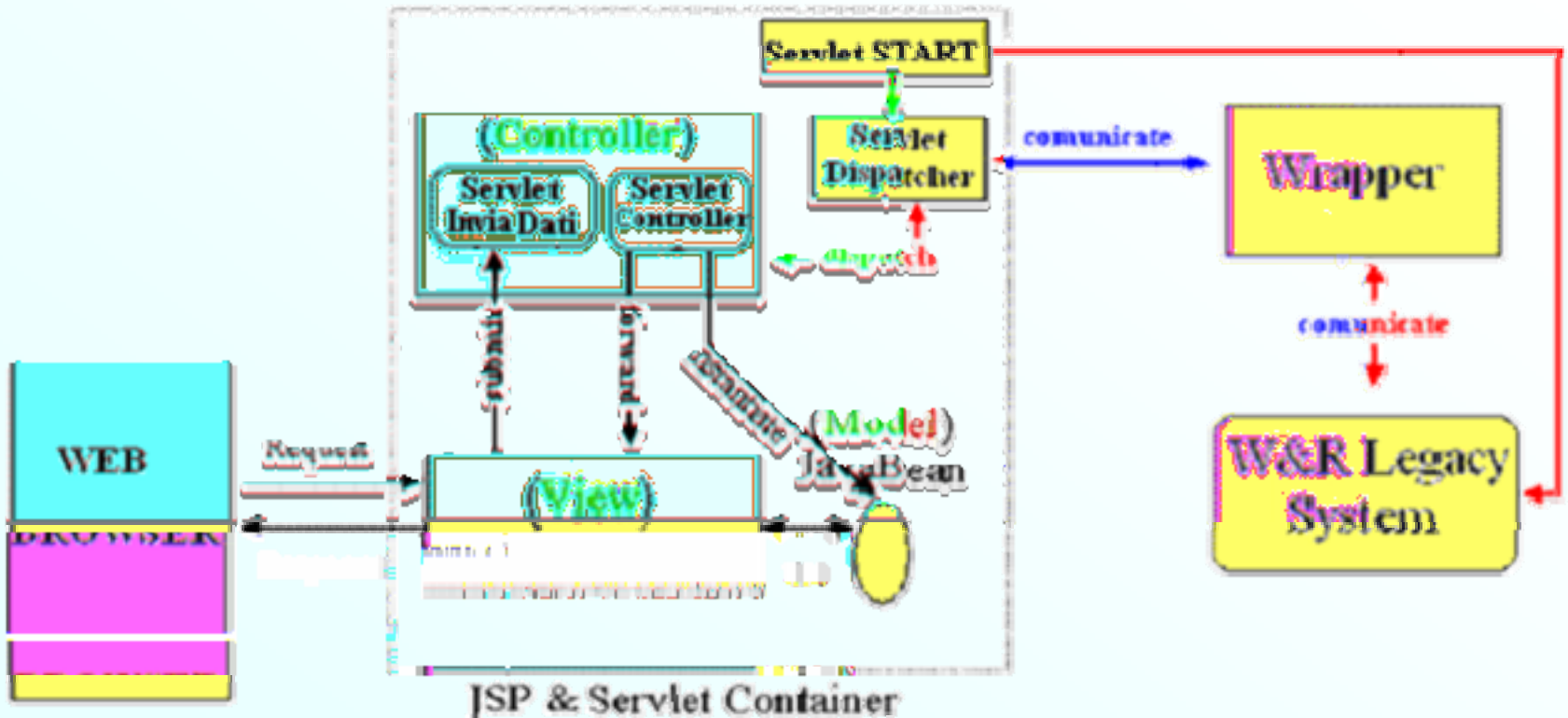
E' importante aggiungere 1 dopo il nome della variabile.

```
ret-x.  
if ret-x = 'S'  
  move value-PIPPONUM to PIPPONUM  
end-if.
```

```
move 'N' TO ret-x.  
move '1' to name-PLUTO (6:1).  
call 'c:\leggInput.wob' using  
  nome-file  
  name-PLUTO  
  value-PLUTO  
ret-x.  
if ret-x = 'S'  
  move value-PLUTO to PLUTO  
end-if.
```

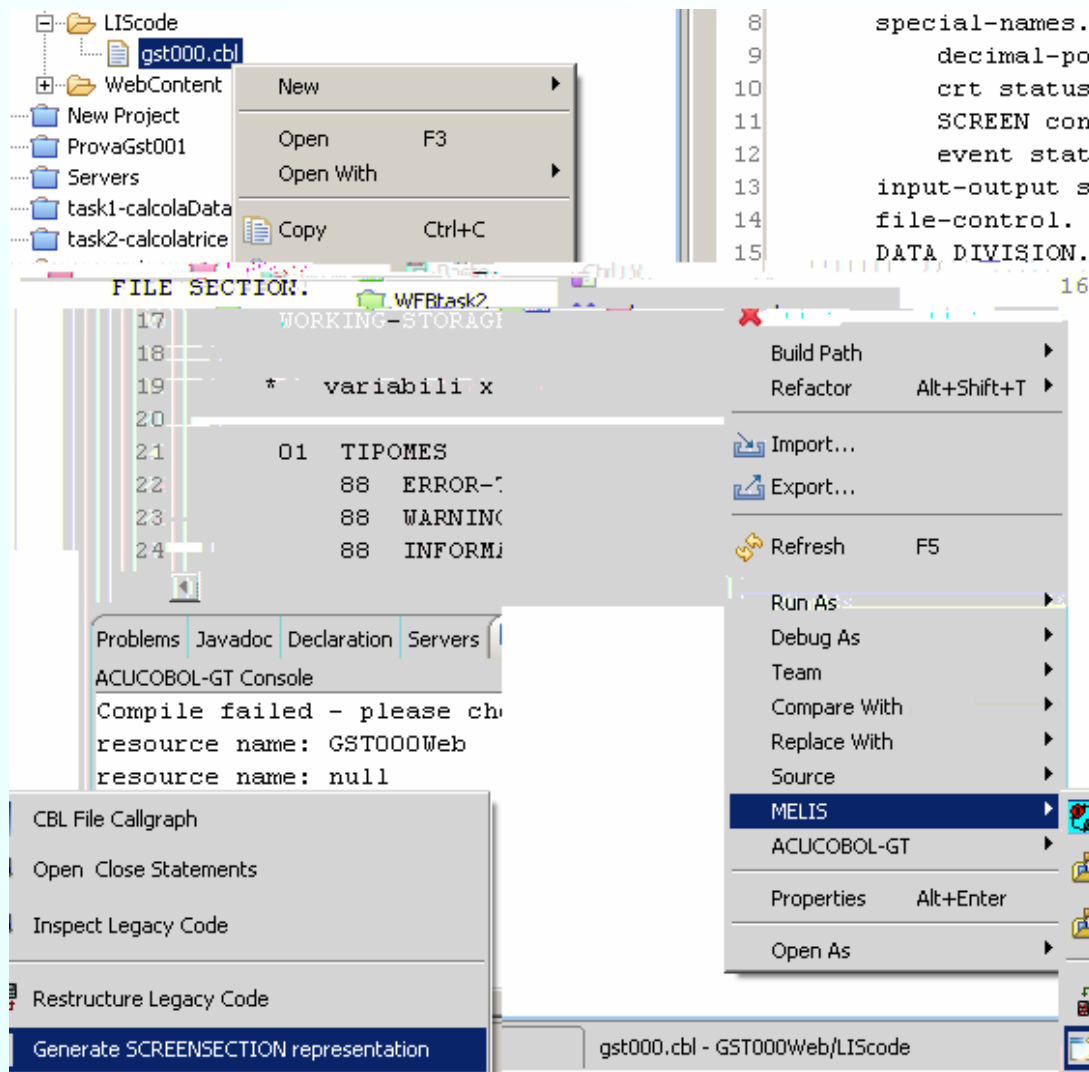
```
.....  
.....  
EXIT.
```

NO-MELIS: GUIReing



Task con l'utilizzo di MELIS

MELIS: Pre-Processing



MELIS: Restr&Wrapp

Procedure Section

- SCREEN_SECTION
 - SCREEN SCREEN-1
 - SCREEN SCREEN-2
 - LABEL - Ragione Sociale -
 - ENTRY-FIELD - CST000RAGSOC-
 - LABEL - Indirizzo -
 - ENTRY-FIELD - CST000ADOMFISC-
 - LABEL - Comune -
 - ENTRY-FIELD - CST000ACOMDF-
 - LABEL - C.A.P./Pr. Nord/Sud. -
 - ENTRY-FIELD - CST000SCAPDF-
 - ENTRY-FIELD - CST000APRODF-
 - ENTRY-FIELD - CST000SN-
 - After: CTR-NORDSUD
 - LABEL - Impresa (S/N) -
 - ENTRY-FIELD - CST000APERI-
 - LABEL - Telefono / Fax -
 - ENTRY-FIELD - C0005NUMTEL-
 - ENTRY-FIELD - C0005NUMFAX-
 - LABEL - Dom.Fis.Div.Sede S/N -
 - ENTRY-FIELD - CST000ATESTSD-
 - LABEL - Sede Div. (Via N.) -
 - ENTRY-FIELD - CST000ASEDDIV-
 - LABEL - Comune -
 - ENTRY-FIELD - CST000ASCOMSD-
 - LABEL - C.A.P. / Provincia -
 - ENTRY-FIELD - CST000SCAPSD-
 - ENTRY-FIELD - CST000APROSD-
 - LABEL - Data Nascita Ditta -
 - ENTRY-FIELD - CST000DATAND-
 - LABEL - Comune Nascita -
 - ENTRY-FIELD - CST000ACOMND-
 - LABEL - Natura Giuridica -
 - ENTRY-FIELD - CST000NATGIU-
 - LABEL - @des-natgiu-
 - LABEL - Se Pers.Fisica-Sesso -
 - ENTRY-FIELD - CST000ATSEXP-
 - LABEL - Data Nascita -

Visualizzazione Type for Screen Section

Orizzontale/verticale Split

Section Cobol Code

```
CTR-NORDSUD SECTION.  
CO01.  
006E44     IF CST000SN NOT = 'S' AND NOT = 'N'  
006E55     MOVE ' VALORE ERRATO ' TO MESSAG1  
006E5D     perform errore  
006E61     SET W-SCR-GOTO TO TRUE  
006E67     GO CO09.  
006E6B     DISPLAY SCREEN-2.  
  
CO09. EXIT.
```

LINE 4:Find PERFORM, check the statement!
LINE 5:Find SET, check the statement!
LINE 7:Find DISPLAY, check that is not a REFRESH!

JavaScript Suggestion Code

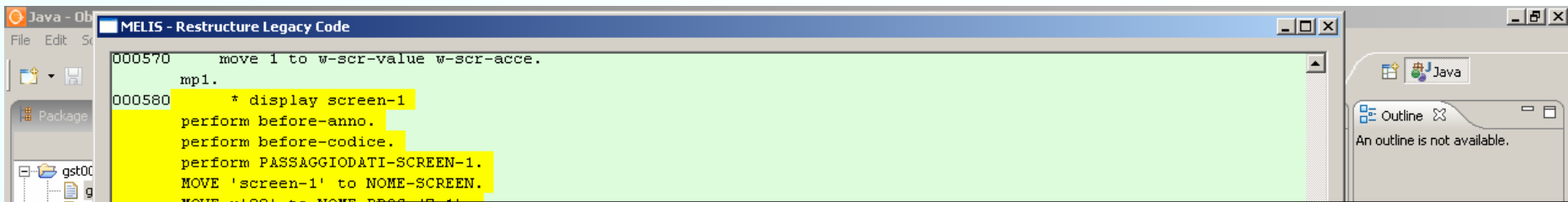
enable JS

```
function CTR-NORDSUD (withAlert,entryValue){  
  if(entryValue == 'S' || entryValue == 'N'){  
    return true;  
  }  
  if(withAlert){  
    alert("INSERIRE S/N!");  
  }  
}
```

Save Exit

gst0000.cbl - gst0000

MELIS: Restr&Wrapp

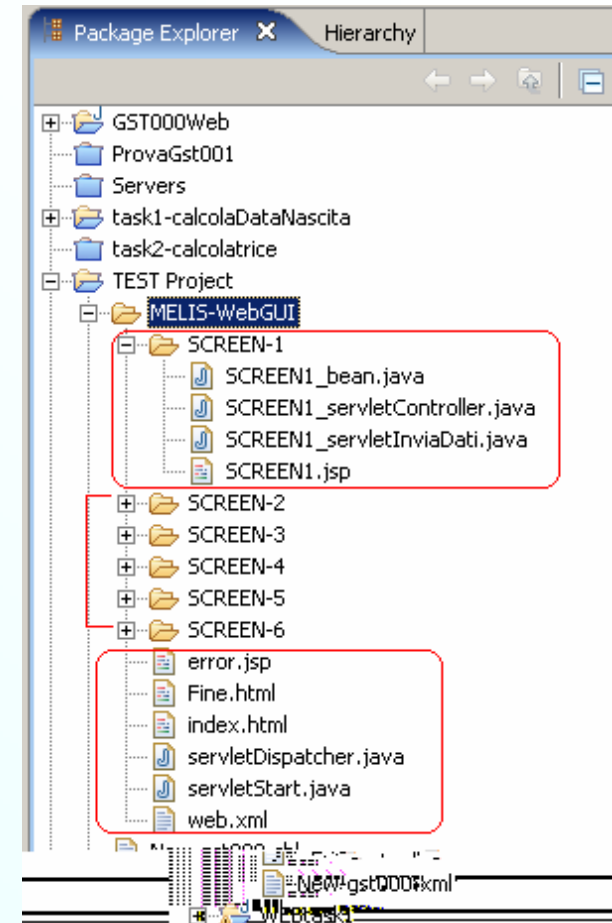
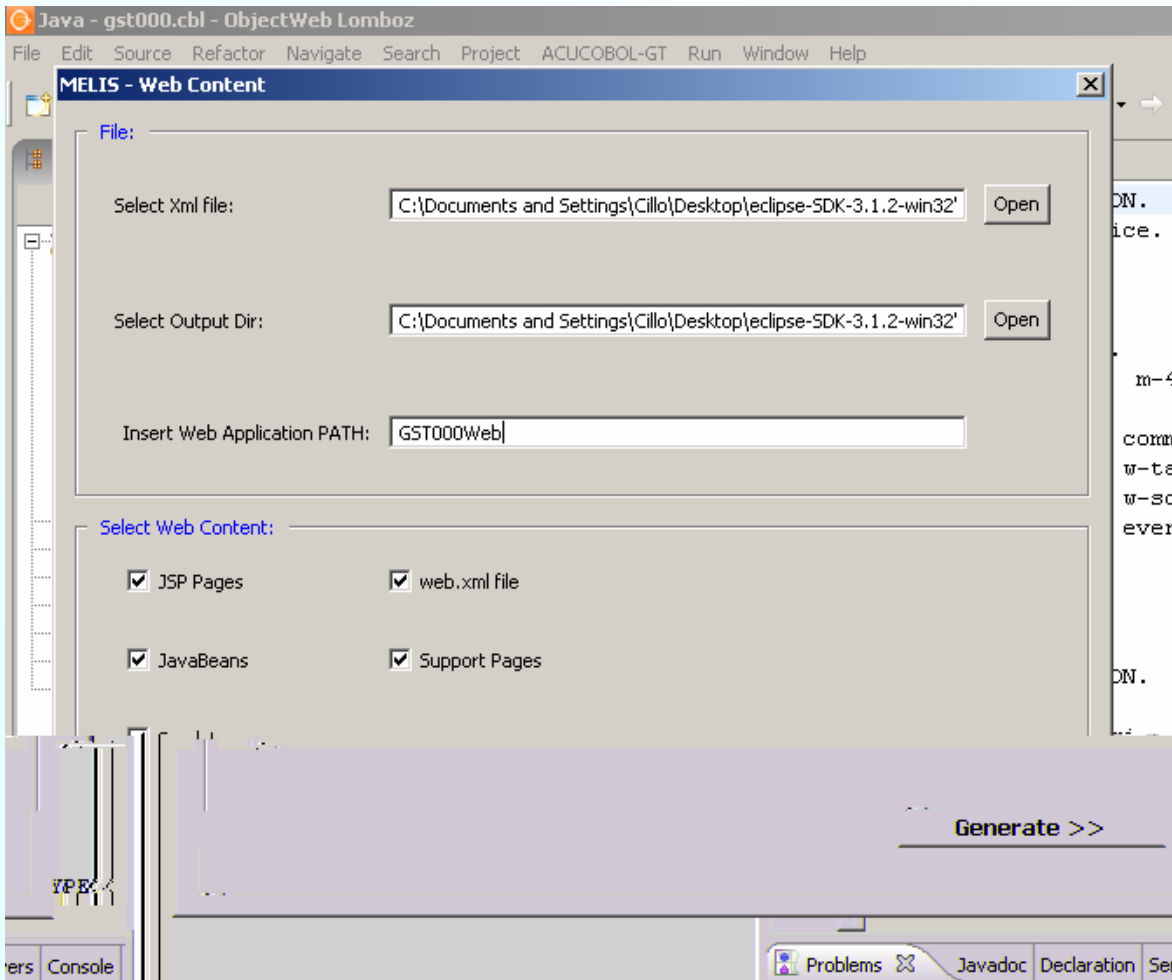


The screenshot shows a Java IDE window titled "MELIS - Restructure Legacy Code". The main editor area contains the following code:

```
000570   move 1 to w-scr-value w-scr-acce.  
      mp1.  
000580   * display screen-1  
      perform before-anno.  
      perform before-codice.  
      perform PASSAGGIODATI-SCREEN-1.  
      MOVE 'screen-1' to NOME-SCREEN.  
      NONE w-scr-acce MOVE PASSAGGIODATI-SCREEN-1
```

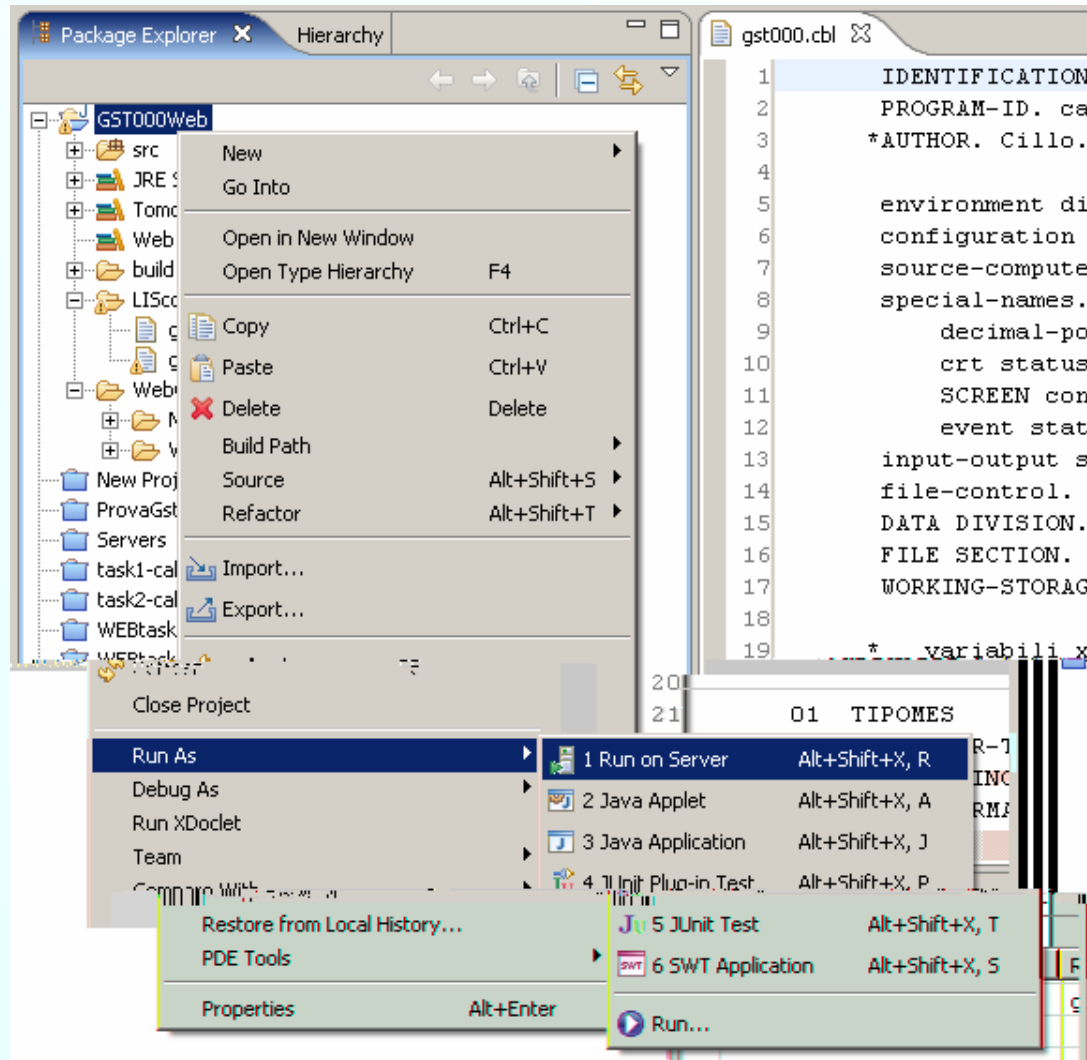
The code is displayed on a light green background. To the right of the editor, there is a "Java" tab and an "Outline" view which displays the message "An outline is not available." The IDE interface includes a menu bar (File, Edit, etc.), a package explorer on the left, and standard window controls.

MELIS: GUIreing

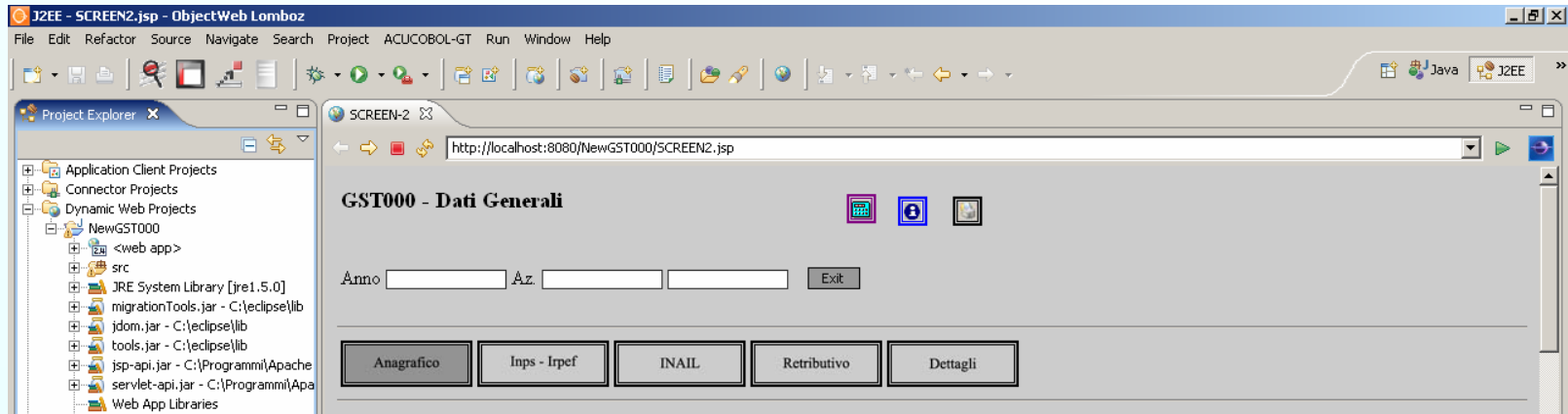


Eseguire l'applicazione migrata
(con e senza MELIS)

RUN dell'applicazione Migrata



RUN dell'applicazione Migrata



Informazioni aggiuntive

Wait & Exit

WAIT

```
set environment 'DLL_CONVENTION' to "1".  
call 'c:\ProvaSem.dll'.  
call 'Cblwait' using by content session.  
cancel 'c:\ProvaSem.dll'.  
set environment 'DLL_CONVENTION' to "0".
```

EXIT

```
set environment 'DLL_CONVENTION' to "1".  
call 'c:\ProvaSem.dll'.  
call 'Cblexit' using by content session.  
cancel 'c:\ProvaSem.dll'.  
set environment 'DLL_CONVENTION' to "0".
```

Sessione

PROBLEMI con la SESSIONE. A causa di alcuni problemi con la sessione, che funge da ID per istanziare correttamente il Wrapper, è necessario **inserire a mano nel codice COBOL**, subito dopo la MAIN PROGRAM SECTION, il seguente codice:

```
move 'ABCDEFGHIJKLMN OPQRSTUVWXYZ1234567890F' to SESSION.  
MOVE x'00' to SESSION(33:1).
```

Lo stesso problema si presenta **lato JAVA** per cui nelle *ServletStart* e *ServletDispatcher* va inserita la seguente stringa:

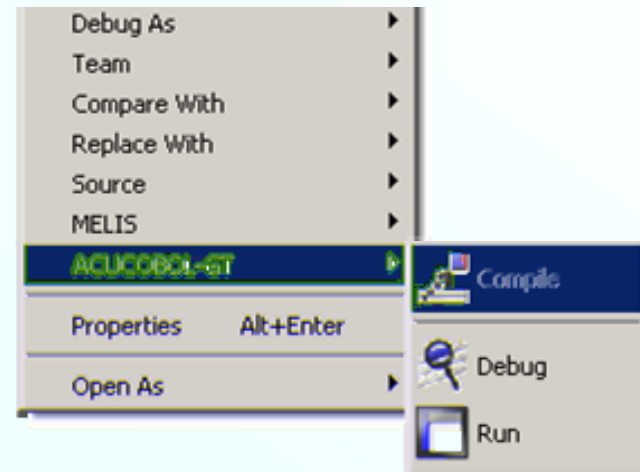
```
static final public String sessionID =  
"ABCDEFGHIJKLMN OPQRSTUVWXYZ1234567890F";
```

Buon lavoro!!!

....e grazie per la collaborazione

Remarks (1 di 3)

- Eseguire e compilare un programma COBOL prima e dopo la migrazione:
 - senza MELIS
 - path\compila.bat SORGENTECOBOL.cbl
 - path\run.bat SORGENTECOBOL.acu
 - con MELIS (per ulteriori dettagli guarda la guida fornita)



Remarks (2 di 3)

1. *Do you want insert support variables?: “Accept”*
2. *Add CHAINING parameter to PROCEDURE DIVISION?: “Refuse”*
3. *Destroy, Display (singole): “Refuse”*
4. *Display and Accept consecutivi: “Accept”*
5. *Section di supporto: “SI”*

```
comment 'DLL_CONVENTION' to "1".
\ProvaSem.dll'.

displayCobol' using by content SESSION
ent NOME-PROG
ent NOME-SCREEN
ent op-dll
RET-DLL.
c:\ProvaSem.dll'.
comment 'DLL_CONVENTION' to "0".

pt screen-1 on exception continue.
t-tasto = zero set cr-key to true.
```

PLAY statement with a DLL call?

Accept

Next

Refuse

Remarks (3 di 3)

- Icone sul desktop

