



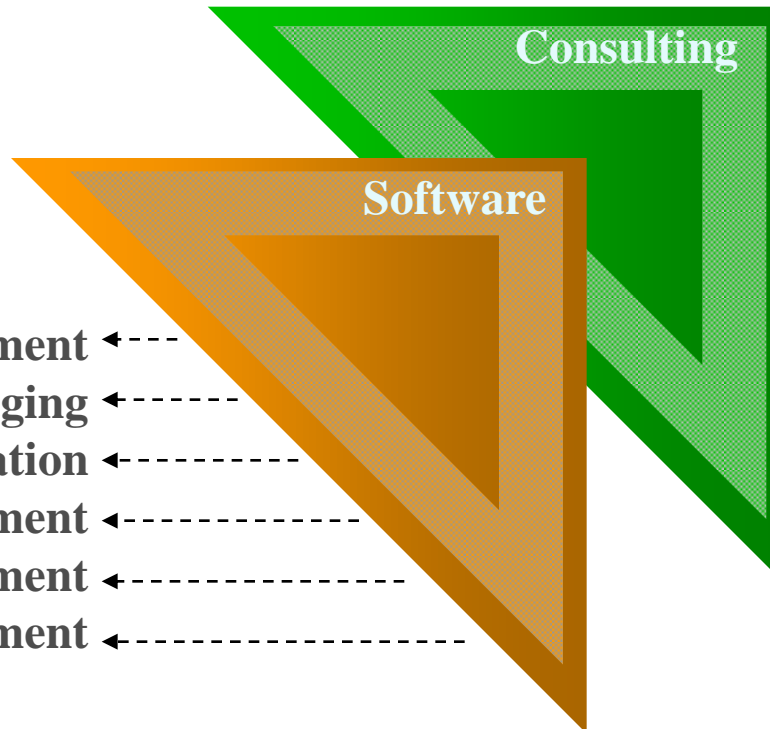
Anbindung von ISPF Anwendungen unter RDz

GSE SCM Tagung Winterthur
24. September 2009

Udo Partsch (SoforTe GmbH)
Christelle Dzomo (SoforTe GmbH)

- Vorstellung SoforTe
- Problemstellung
- Lösungen
 - Menu Manager
 - RDz Plugin Entwicklung
 - HATS
 - CARMA
- Bewertung
- Zusammenfassung

Ihr Spezialist
für
Softwareentwicklungsumgebungen
und
Projektmanagement



- > **Eclipse / RAD / RDz**
- > **Methoden & Tools**
- > **Projekt-Management**
- > **Zentrales
Metadatenrepository**
- > **Software-
Konfigurations-
Management**
- > **Migrationen**

- Enterprise Development <-----
- Teamroom, Messaging <-----
- Collaboration <-----
- Projekt-Management <-----
- Dokumenten-Management <-----
- Mail-Management <-----

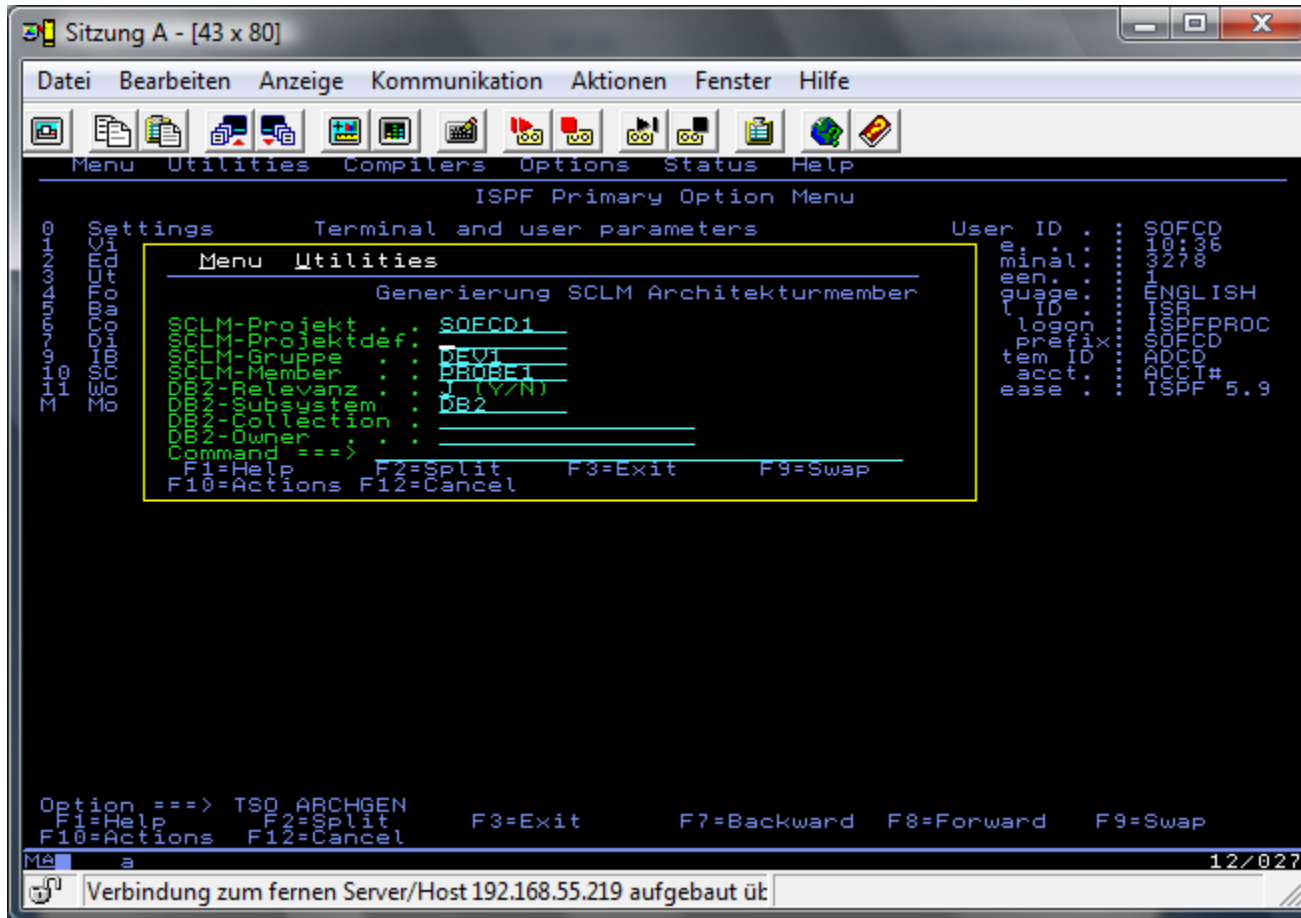
Ausgangssituation

- Fehlende Integration von kunden-bezogenen Applikationen/Funktionen im RDz
- Aufruf kunden-spezifischer ISPF Funktionen aus RDz nur über 3270-Emulation

Motivation

- Einheitliches Look and Feel
 - Schaffung eines schnellen Zugriffs dieser Funktionen
 - Höhere Produktivität
 - Akzeptanz

- Beispiel: Generierung SCLM Architekturdefinitionen
- Wie integrieren wir diesen ISPF Dialog unter RDz?



```
Sitzung A - [43 x 80]
Datei Bearbeiten Anzeige Kommunikation Aktionen Fenster Hilfe
Menu Utilities Compilers Options Status Help
ISPF Primary Option Menu
Settings Terminal and user parameters User ID . . . . . SOFCDC
me . . . . . 10:36
terminal . . . . . 3278
language . . . . . ENGLISH
l ID . . . . . ISR
logon . . . . . ISPFPROC
program . . . . . SOFCDC
term ID . . . . . ACDC
acct ID . . . . . ACCT#
ease . . . . . ISPF 5.9

Menu Utilities
Generierung SCLM Architekturmember
SOFCDC1
DEV1
PROBE1
(Y/N)
DB2
Command ==>
F1=Help F2=Split F3=Exit F9=Swap
F10=Actions F12=Cancel

Option ==> TSO ARCHGEN
F1=Help F2=Split F3=Exit F7=Backward F8=Forward F9=Swap
F10=Actions F12=Cancel
MA a 12/027
Verbindung zum fernen Server/Host 192.168.55.219 aufgebaut üt
```



SOFORTE
SOLUTIONS FOR TEAMS

Menu Manager

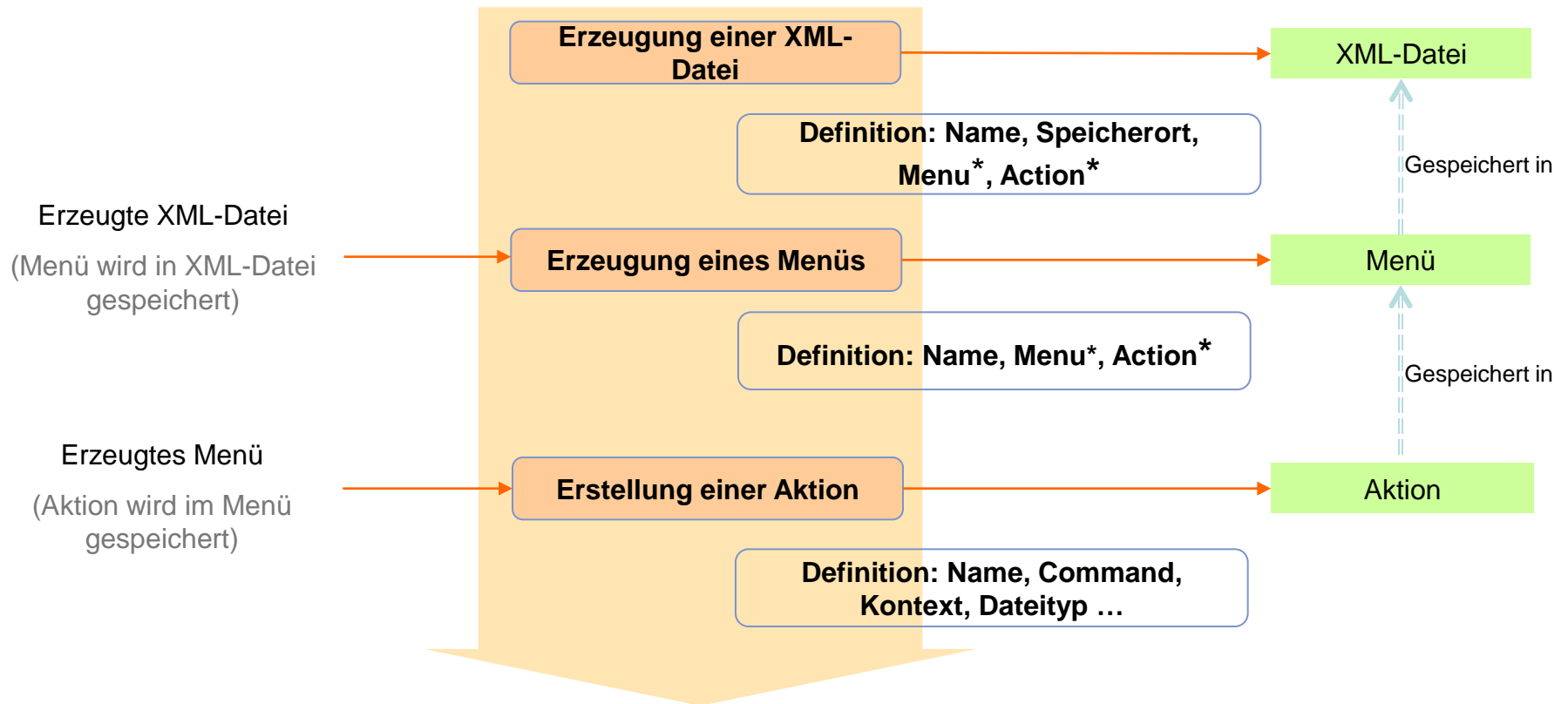
- Was ist Menu Manager?
 - grafisches Interface (Workbench Preference) zur
 - Erweiterung des Kontextmenüs
 - Erstellung von Popup Menüs
 - Verfügbar für
 - z/OS Project View
 - Lpex Editor

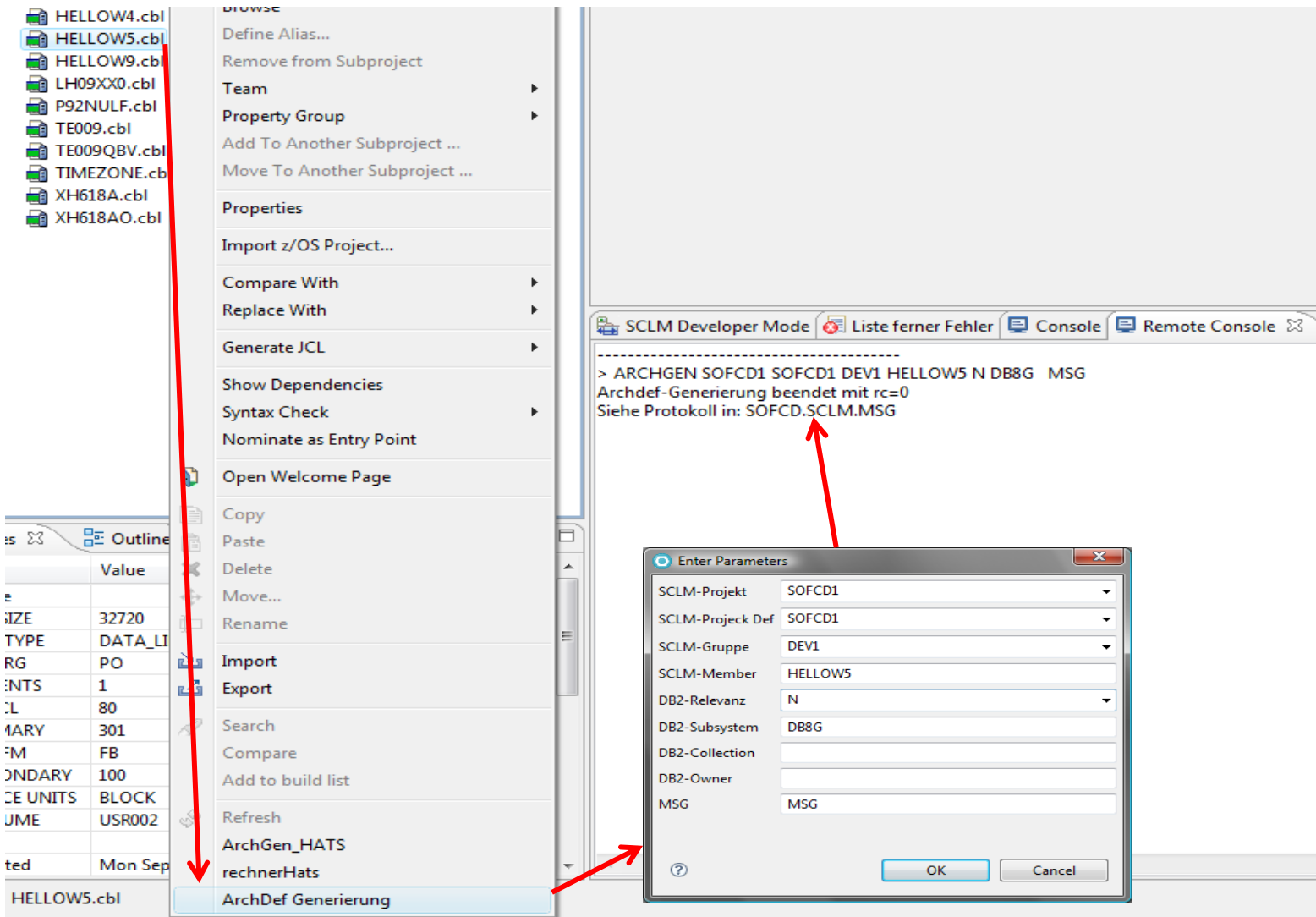
- RDz Konfiguration, gespeichert als XML-Datei

- Erstellte XML-Datei kann im RDz importiert oder verteilt werden

- Aktionen werden durch Konfiguration definiert und mit Dateitypen verbunden

Menu Manager: Vorgehensweise





The screenshot shows the SCLM Developer Mode interface. On the left, a file list contains several .cbl files, with 'HELLOW5.cbl' selected. A context menu is open over this file, listing various actions such as 'Define Alias...', 'Remove from Subproject', 'Team', 'Property Group', 'Add To Another Subproject ...', 'Move To Another Subproject ...', 'Properties', 'Import z/OS Project...', 'Compare With', 'Replace With', 'Generate JCL', 'Show Dependencies', 'Syntax Check', 'Nominate as Entry Point', 'Open Welcome Page', 'Copy', 'Paste', 'Delete', 'Move...', 'Rename', 'Import', 'Export', 'Search', 'Compare', 'Add to build list', 'Refresh', 'ArchGen_HATS', 'rechnerHats', and 'ArchDef Generierung'. The 'ArchDef Generierung' option is highlighted. Below the file list is an 'Outline' table with columns for 'Value' and 'Name'. The table contains the following data:

Value	Name
32720	SIZE
DATA_LI	TYPE
PO	RG
1	INTS
80	IL
301	MARY
FB	FM
100	ONDARY
BLOCK	CE UNITS
USR002	UME
Mon Sep	ted
HELLOW5.cbl	

On the right, the console window displays the following output:

```
> ARCHGEN SOFCD1 SOFCD1 DEV1 HELLOW5 N DB8G MSG  
Archdef-Generierung beendet mit rc=0  
Siehe Protokoll in: SOFCD.SCLM.MSG
```

An 'Enter Parameters' dialog box is open in the foreground, with the following fields:

- SCLM-Projekt: SOFCD1
- SCLM-Projekt Def: SOFCD1
- SCLM-Gruppe: DEV1
- SCLM-Member: HELLOW5
- DB2-Relevanz: N
- DB2-Subsystem: DB8G
- DB2-Collection: (empty)
- DB2-Owner: (empty)
- MSG: MSG

Red arrows indicate the flow of the process: from the 'HELLOW5.cbl' file in the file list, down to the 'ArchDef Generierung' menu item, and then to the 'Enter Parameters' dialog box. Another red arrow points from the 'MSG' field in the dialog box to the 'MSG' output in the console window.

Vorteile:

- Einfach zu erstellen
- Einfache Verwendung von RDz Variablen im Popup Menüs
 - Benutzerinformationen z.B. userid
 - Dateiinformationen z.B. Membername
- Keine Java Programmierkenntnisse erforderlich

Nachteile:

- Läuft nur in Verbindung mit der z/OS Project View und m.E. Lpex Editor
- Eingeschränkte Funktionalitäten
 - Keine Eingabeprüfung client-seitig
 - Keine Aufbereitung der Funktionsausgabe
- ISPF Dialoganwendung muss angepasst werden
 - Parameterübergabe
 - ISPF Panel darf nicht aufgerufen werden

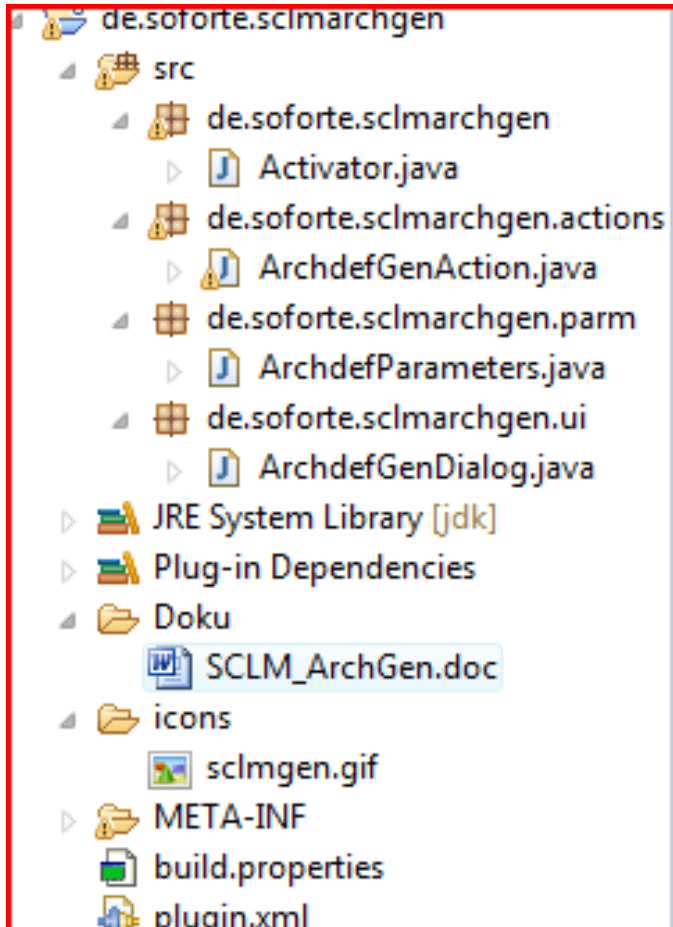


RDz Plugin-Entwicklung mit dem Ressource-API

- **Plugin-Entwicklung**
 - Action erstellen
 - GUI
 - Aufruf der ISPF Funktion
 - Fehlerbehandlung
 - Ausgabeaufbereitung

- **RDz Ressource API**
 - Java Programmierschnittstelle
 - Host Kommunikation
 - Aufruf von ISPF Funktionen

- **Plugin muss im RDz installiert werden**



```
ZOSSystemImage si =  
    SystemSelector.getInstance().getConnectedSystem();
```

```
setZOSUserID(si.getUserID());
```

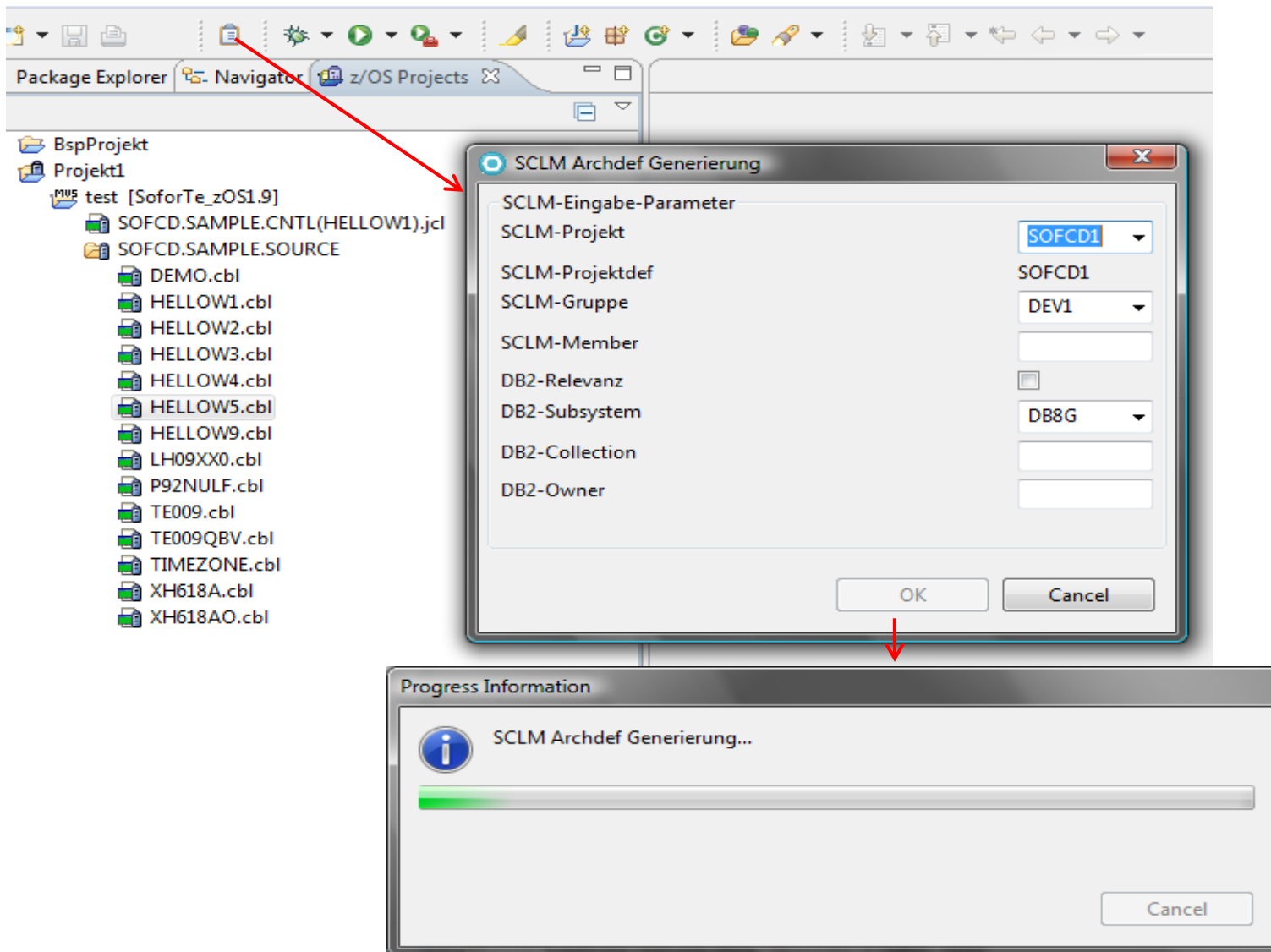
```
String pCommand = "ARCHGEN " + sclmProj + " "  
    + firstQual + " " + sclmGroup + " " + progName  
    + " " + db2Flag + " " + db2Sys+ " "+ owner + " "  
    + collID+ " " + msgDD;
```

```
_lastCommand = si.getCommandObject();
```

```
_lastCommand.setCommandString(pCommand);
```

```
_lastCommand.run();
```

Beispiel



Vorteile:

- Höchste Flexibilität
- Benutzerfreundlich
- Standard Eclipse Plugin Installationsverfahren

Nachteile:

- Sehr gute Plugin Kenntnisse erforderlich
- Hoher Programmieraufwand
- ISPF Dialoganwendung muss angepasst werden
 - Parameterübergabe
 - ISPF Panel darf nicht aufgerufen werden



HATS RCP

- HATS
 - Host Application Transformation Services

- Guifizierung beliebiger 3270-Anwendungen (IMS, CICS, ISPF, ...)

- Integration von 3270-Anwendungen in
 - Web Anwendung

 - **RCP (Rich Client Plattform)**
 - Bsp: Eclipse Look and Feel einer ISPF-Anwendung

 - Möglichkeit der Integration einer Hostanwendung mit anderen Desktop-Applikationen

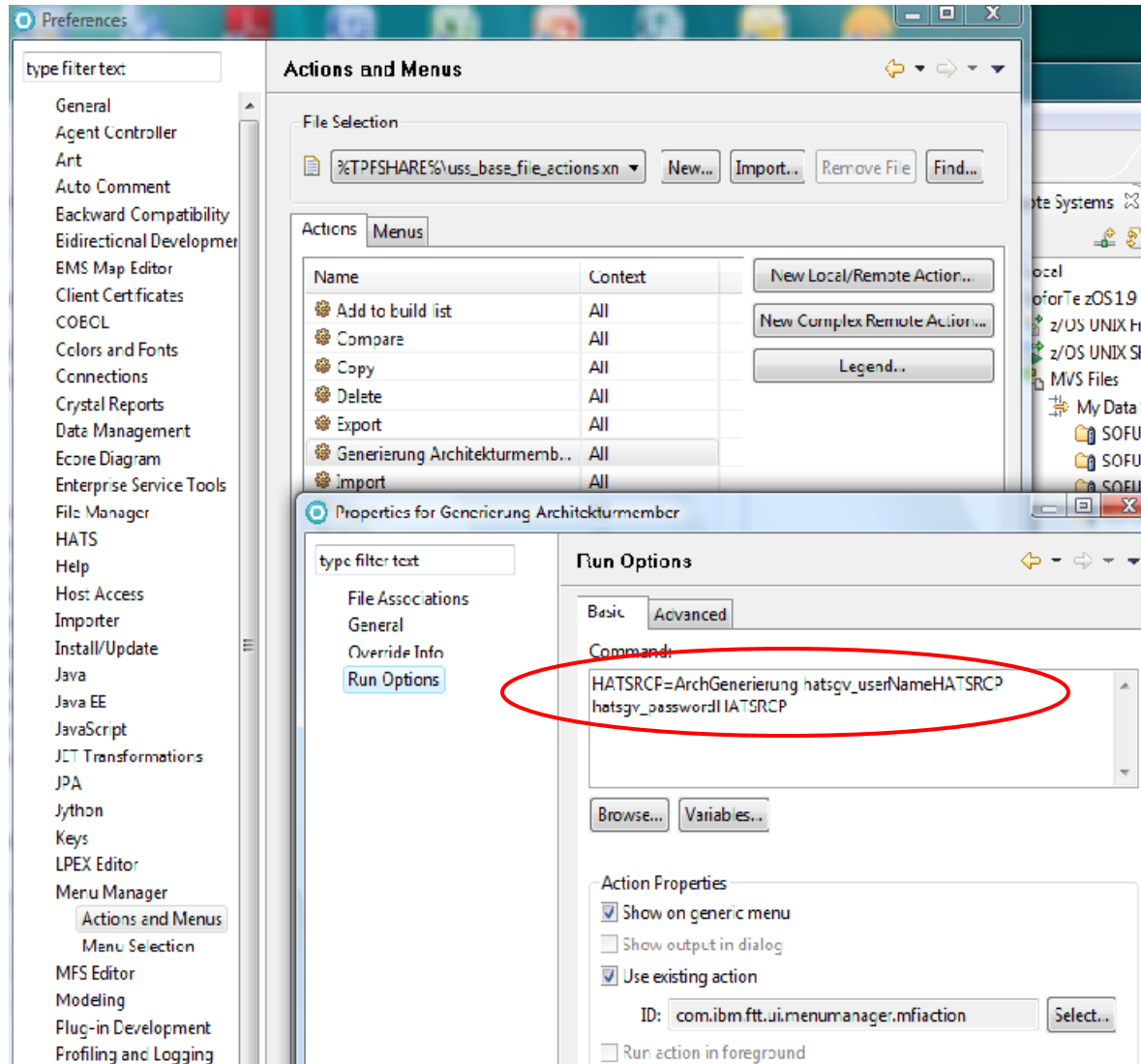
- Software-Voraussetzung für die Entwicklung von HATS/RDz Anwendung
 - RDz 7.5.1 mit RAD 7.5.2
 - HATS 7.5.0.1

- Mit HATS aus einer ISPF Anwendung eine RCP-Anwendung generieren
 - Bildschirmerkennung konfigurieren
 - Makros aufzeichnen
 - Logon -, Logoff- Verfahren , ...
 - Menu Navigation
 - 3270-Bildschirme ggf. für HATS-Anwendung optimieren:
 - GUI-Elemente integrieren
 - Felder ausblenden
 - Zusätzliche Felder / Schaltflächen aufnehmen

- Einbindungsmöglichkeiten unter RDz:
 - HATS-RCP-Anwendung über RDz-Menü-Manager einbinden
 - HATS-API in PlugIn-Entwicklung

- Spezielle Features
 - Automatisierte Navigation zum ISPF-Tool
 - LOGON-Informationen aus einer bestehenden Connection zum RSE können auch für LOGON der HATS-Anwendung verwendet werden
 - Aufruf einer RDz-Funktion (z.B. LPEX-Editor) aus einer HATS-Anwendung durch Programmierung möglich.

HATS-Anwendung im Menü-Manager definieren



The screenshot displays the 'Preferences' dialog box in IBM HATS, specifically the 'Actions and Menus' section. The 'File Selection' field is set to '%TPFSHARE%\uss_base_file_actions.xn'. The 'Actions' tab is active, showing a list of actions with their contexts. The 'Run Options' for the selected action 'Generierung Architekturmember' are shown in a separate dialog box, with the 'Command' field circled in red. The command is:

```
HATSRCP=ArchGenerierung hatsgv_userNameHATSRCP  
hatsgv_passwordHATSRCP
```

The 'Run Options' dialog also shows the 'Action Properties' section with the following settings:

- Show on generic menu
- Show output in dialog
- Use existing action
- ID: com.ibm.ftt.ui.menumanager.mfiaction
- Run action in foreground

Beispiel

The screenshot shows the SoforTe IDE interface. On the left, a file explorer lists several CBL files, with 'HELLOW5.cbl' selected. A context menu is open over this file, with 'ArchGen_HATS' highlighted at the bottom. Below the file list is an 'Outline' table with the following data:

Value	
SIZE	32720
TYPE	DATA_LIBRA
ORG	PO
ENTS	1
L	80
ARY	301
M	FB
INDARY	100
E UNITS	BLOCK
IME	USR002

The main workspace displays the 'ArchGenerierung' window, titled 'Generierung SCLM Architekturmember'. It contains a 'Disconnect' button and a list of configuration parameters for SCLM:

- SCLM-Projekt . . : SOFCD1
- SCLM-Projektdef. : [empty field]
- SCLM-Gruppe . . : DEV1
- SCLM-Member . . : HELLOW5
- DB2-Relevanz . . : N (Y/N)
- DB2-Subsystem . : DB2
- DB2-Collection . : COLL1
- DB2-Owner . . . : OWN1

Vorteile:

- Keine Anpassung der ISPF Anwendung erforderlich
- Eclipse Look and Feel erreichen ohne Programmierung
- Automatische Konvertierung von Funktion Keys in Toolbar Button
- Information Integration
 - Integration und Darstellung von Informationen aus verschiedenen ISPF Anwendungen
- Standard Eclipse Plugin Installationsverfahren

Nachteile:

- Mehrere HATS Anwendungen können nicht gleichzeitig aktiv sein (Anwendung beginnt mit TSO-Logon)
- Benutzer darf nicht unter 3270- ISPF angemeldet sein
- Höherer Test/ Wartungsaufwand nach Änderung einer Hostanwendung



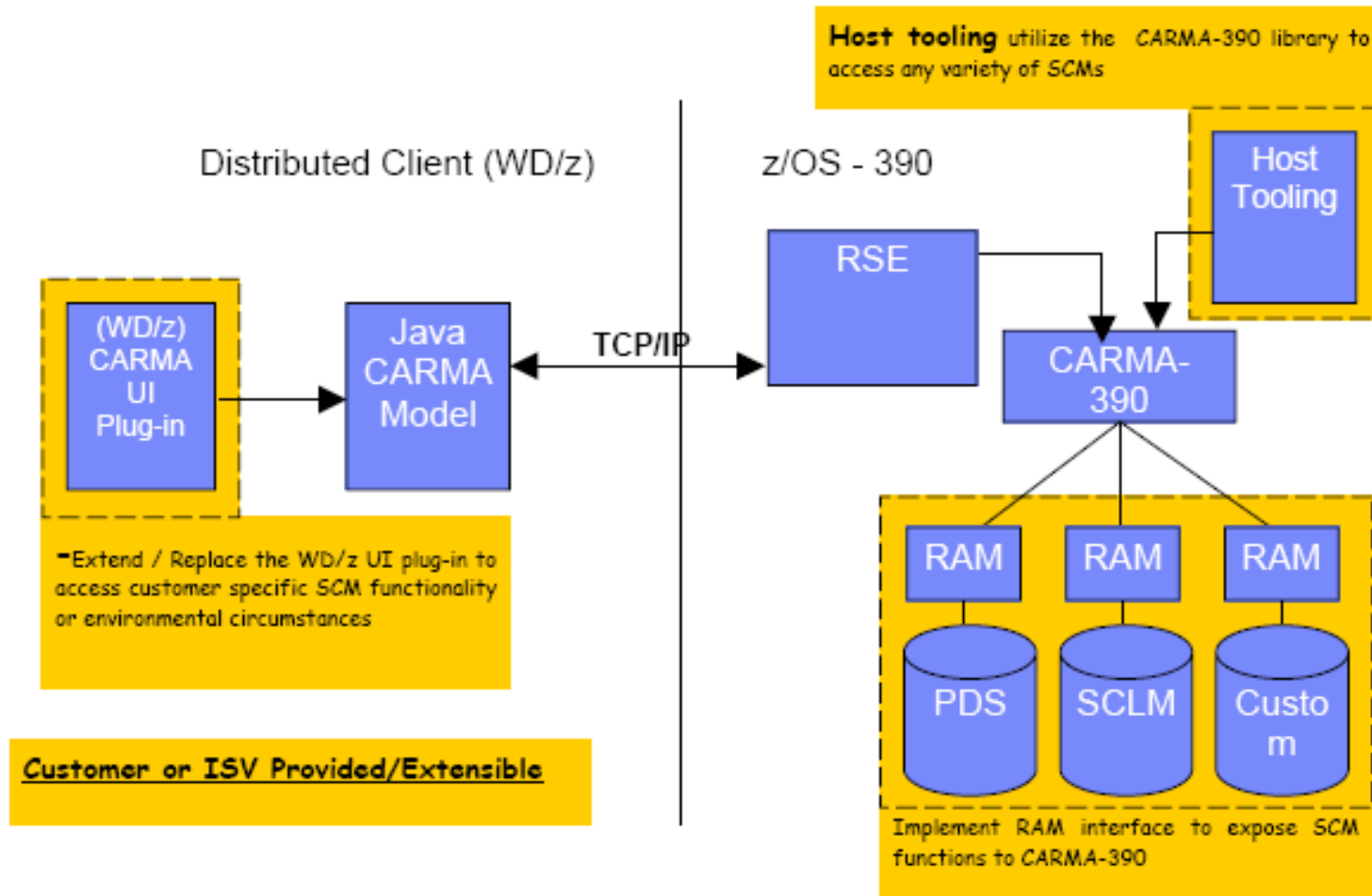
SOFORTE
SOLUTIONS FOR TEAMS

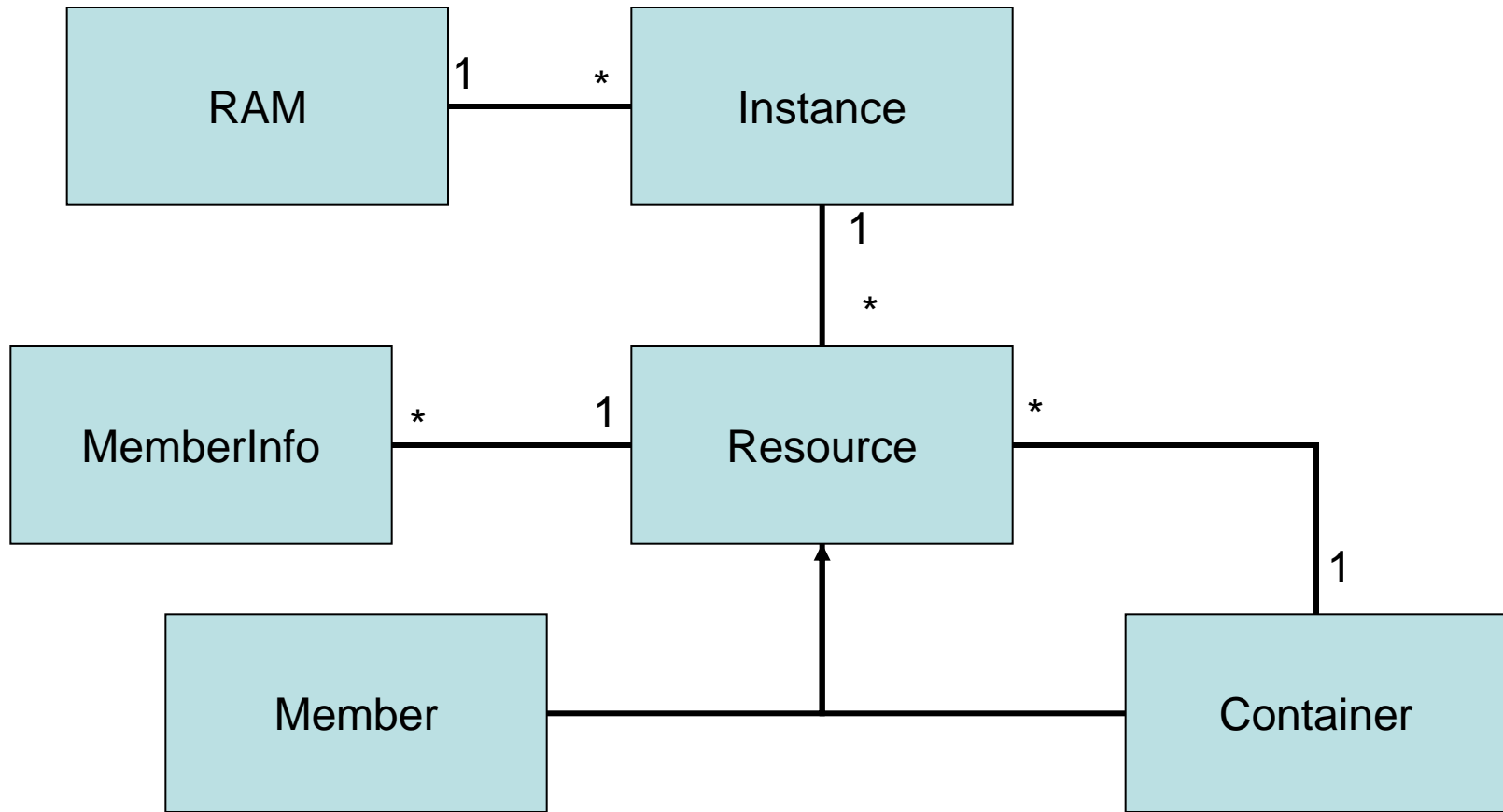
CARMA

➤ CARMA

- Common Access Repository Manager
SCM Framework als Teil von RDz
- Generische Schnittstelle zur Anbindung eines z/OS SCM-Systems an RDz
- CARMA stellt ein Framework zur Verfügung, das individuell angepasst werden kann.
- Wesentliche SCM-Funktionen sind in der Schnittstelle berücksichtigt.
- Schnittstelle kann um benutzer-definierte Funktionen ergänzt werden.

CARMA technische Architektur





➤ CARMA Server

ISPF-Task unter RDz/RSE

- Kein Batchjob / Initiator erforderlich
- Allokierung anpassbar
- Wiederverwendbarkeit existierender Host-Tools möglich (REXX etc.)

➤ Generischer Client unter RDz

- Stellt alle im RAM definierten und auskodierte Funktionen dynamisch zur Verfügung
- Ohne eigene Client-Programmierung nutzbar

➤ CARMA-Java-API

- Entwicklung eines eigenen Clients (Eclipse-Plugin)
- Erweiterung / Anpassung des generischen Clients
- Erlaubt die flexible Einbindung aller im RAM definierten Funktionen unter RDz

➤ RAM

- RAM Implementierung in C, aber ...
 - REXX-Aufruf für jede implementierte Funktion
 - Existierende SCM-Funktionen wiederverwenden / anpassen
- 2 VSAM Steuerdateien
 - Definition Funktionen, Funktionsparameter, ...

➤ CARMA Client

- Den existierenden generischen CARMA Client verwenden und ggf. anpassen / erweitern

CARMA Beispiel – Aufruf REXX Steuerprozedur Definition im C-Programm (RAM)

```
.....
int performAction(int actionID, char instanceID[256],
                  char memberID[256], void** params,
                  void*** customReturn, char error[256]) {
.....
    switch (actionID) {
.....
    case ARCHG_ACTION:
.....
        memcpy(fkt, "ArchG", 5);
        sprintf(buf, "%i %.24s %.256s %.256s %c %.8s %.8s %.8s",
                &rexxRes1, fkt, instanceID, memberID, db2Opt,
                db2Sys, db2Owner, db2CollId);

        rexxRes = execREXXProcedure("SCLMCAST", 1, buf);

        memcpy(rexxRes, rexxRes1, 300);
.....
        retc = atoi(rexxRes);
        memcpy(error, rexxRes+10, 256);

        break;        /* end of case ARCHG_ACTION */
    }
.....
```

CARMA Beispiel – Aufruf Archgen Funktion aus Steuer-REXX SCLMCAST

.....

```
When fkt = 'ARCHG' Then Do
  Parse Var rest instanceid memberid db2opt db2sys ,
           db2owner db2coll
  db2coll = Strip(db2coll)
  Call ISPF '0 SELECT CMD(SCLMCAAG 'instanceid' 'memberid' ',
           db2opt' 'db2sys' 'db2owner' 'db2coll' 'resadr')'
```

End

.....

CARMA Beispiel – REXX Prozedur SCLMCAAG zum Aufruf der Archgen Funktion

```
.....
Arg instanceID memberID db2opt db2sys db2owner collid
  resadr

instanceID = Strip(instanceID)
memberID = Strip(memberID)
user = USERID()
pref = Substr(memberID,1,3)

.....
grp = Substr(instanceID,4)
lp_strich = Lastpos('-',memberID)
type = Substr(memberID,4,lp_strich-4)
mem = Strip(Substr(memberID,lp_strich+1))
ADDRESS ISPEXEC "SELECT CMD(ARCHGEN "sclmproj" ",
  "sclmproj" "grp, mem" "db2opt" "db2sys" "db2owner,
  " "collid")"

Call Exit retc 0 errmsg
.....
```


	Aufwand	Flexibilität	Performanz	Techn. Skill
Menü Manager	gering	Gering	Gut	Gering
Plugin Entwicklung	Sehr hoch	Sehr hoch	Gut	Hoch (i.d.R. nicht vorhanden)
HATS RCP	Mittel	Mittel	Sehr gut	Mittel (i.d.R. nicht vorhanden)
CARMA	Hoch	Hoch	Gut	Hoch (überwiegend vorhanden)

- Menü Manager
 - Für einfache Tools
 - Project View
- RDz Plugin Entwicklung
 - Flexible Integration
- HATS
 - Für komplexe Host-Tools
 - Bei schlecht wartbarer Source
 - Oder keine Source verfügbar
- CARMA
 - Integration eines Host-SCM-Tools
 - Host Konfiguration
 - Host Programmierung im Vordergrund

THANK YOU

