

OpenOffice.org Calc automation using ooRexx

Author: Michael Hinz h0253952

Advisor: ao. Univ.Prof. Dr.Rony G. Flatscher

Overview

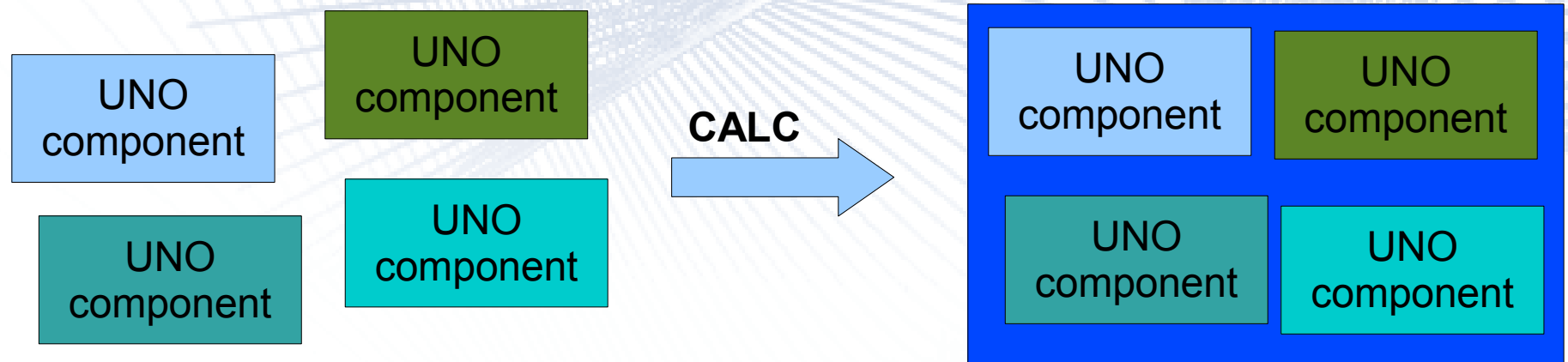
- OpenOffice.org
 - History
 - Architecture
- ooRexx
 - History
 - Advantages
- Overall Concept
- BSF4Rexx
- Example

OpenOffice.org - History

- Mid-80'ies: Star Division developed StarOffice
- 1999: Sun Microsystems took over StarOffice
- 2000: Open Source project
 - OpenOffice.org
 - October 2001: Build 638c
- 2003: OpenOffice.org 1.1
- 2006: Latest stable release - OpenOffice.org 2.0.2

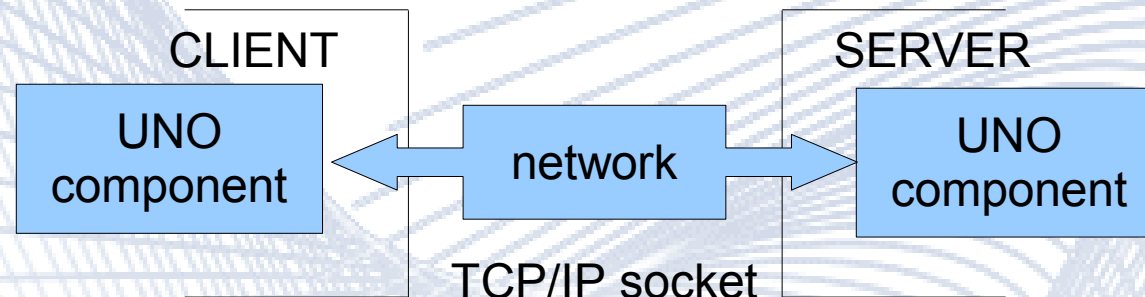
OpenOffice.org - Architecture

- Consists of UNO(Universal Network Object) components
- Single components can be combined to build an application



OpenOffice.org - Architecture 2

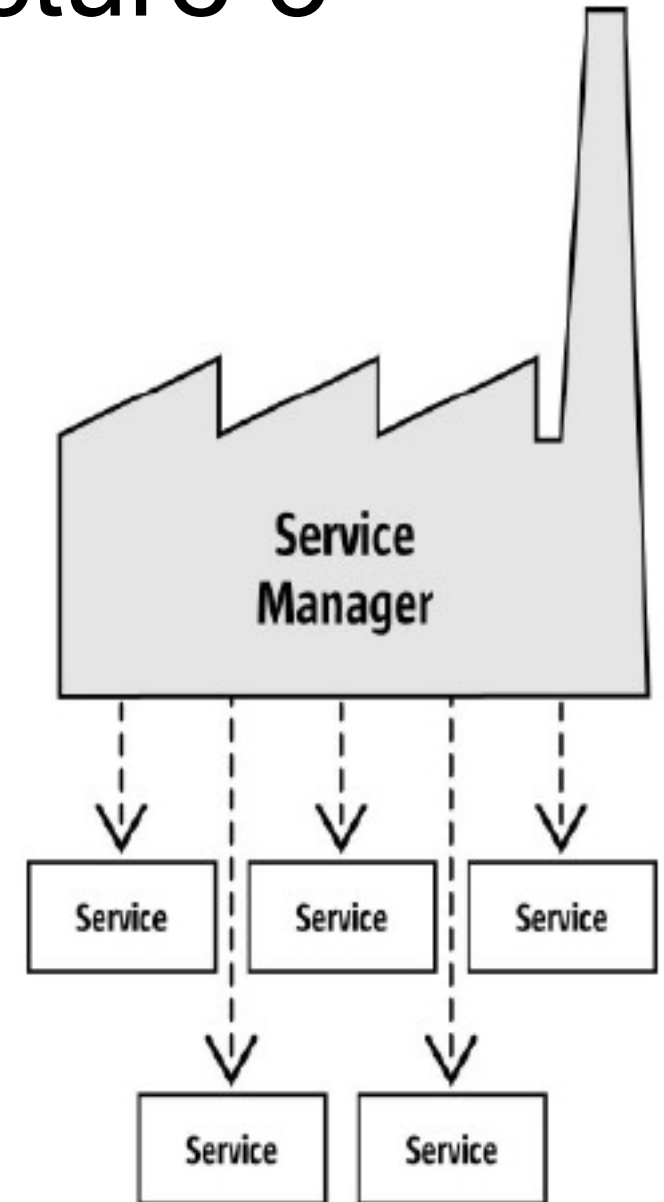
- Developed as a client server application
- Interacting over TCP/IP



- UNO components represents services
 - Additional services
 - Interfaces (**functions/methods**)
 - Properties

OpenOffice.org - Architecture 3

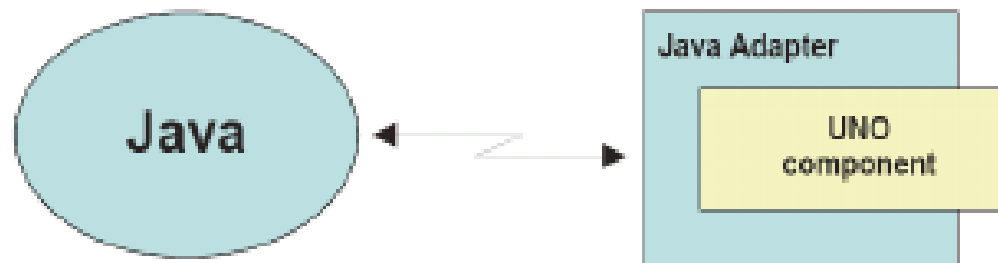
- Service Manager
 - Creates services
 - Exists in a component context
 - E.g. *com.sun.star.Desktop*
- Load documents, access loaded documents



from: *OOo Developer's Guide*(2006)

OpenOffice.org – Architecture 4

- Java Adapter



from: Rony G. Flatscher, *AUTOMATING OPENOFFICE.ORG WITH OOREXX: ARCHITECTURE, GLUING TO REXX USING BSF4REXX*(2005)

- Use UNO components like native Java components

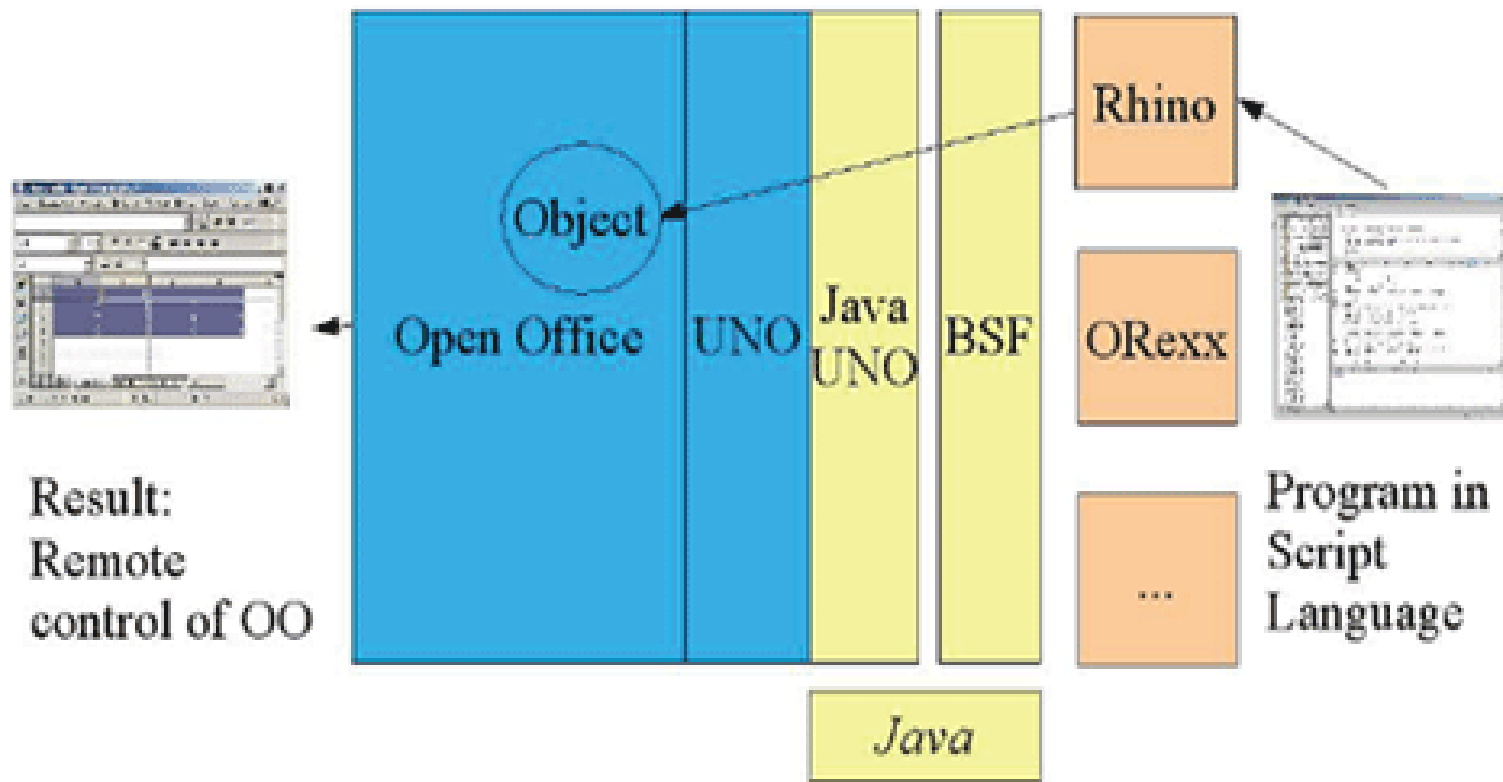
ooRexx - History

- 1979: Rexx (Restructured Extended Executer) developed by IBM employee Mike F. Cowlislow
- 1988: object-oriented version ObjectRexx
- 2005: Open Object Rexx
 - Open Source Project
 - Managed by Rexx Language Association(RexxLA)

ooRexx - Advantages

- English-like statements
- Fewer Rules
- Statements are interpreted not compiled
- Built-in Functions and Methods
- Typeless Variables
- Powerful String Handling
- Clear Error Messages

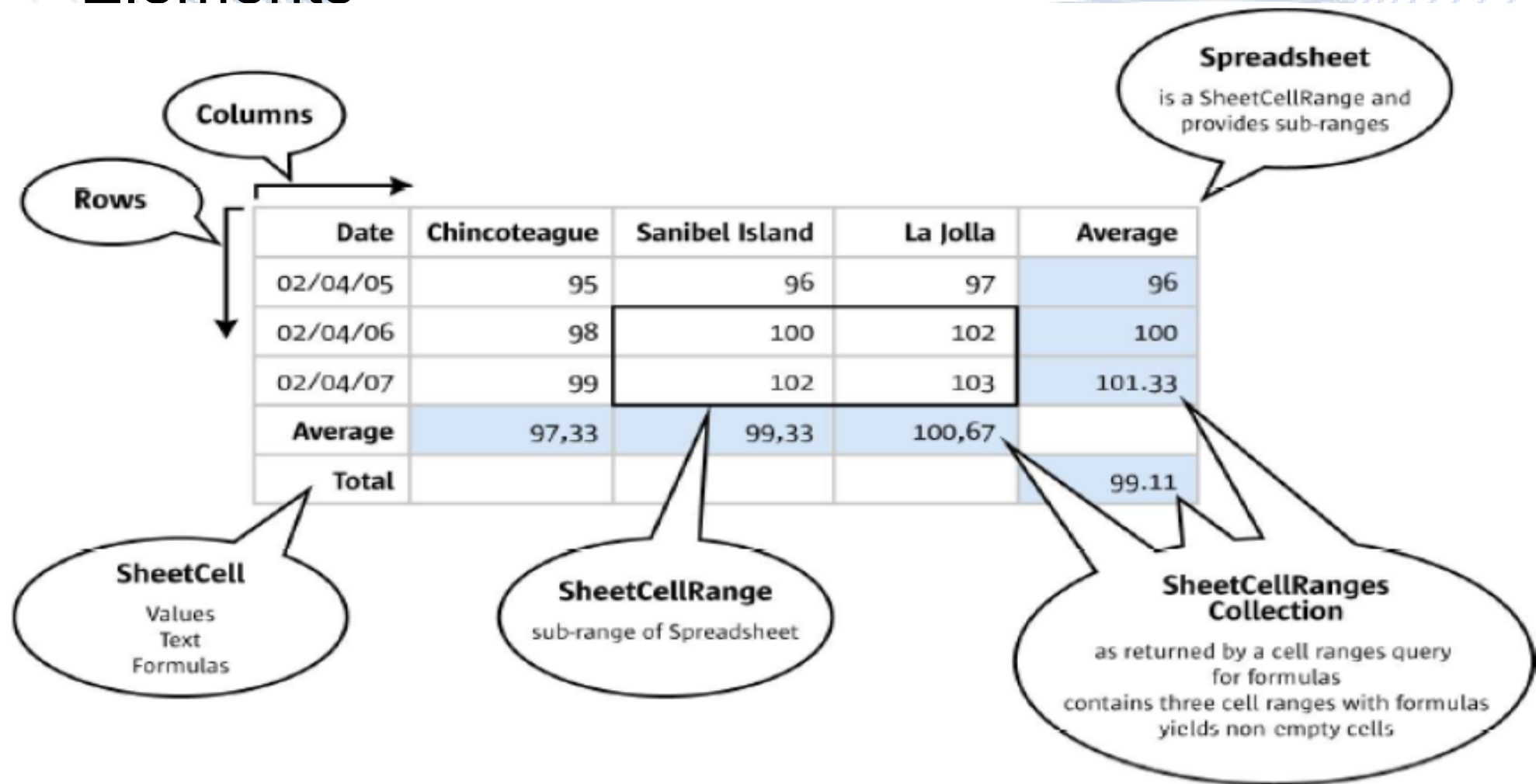
Overall Concept



from : Walter Augustin, Examples for Open Office Automation with Scripting Languages(2005)

Calc Component

- Elements



BSF4Rexx2

- BSF.CLS
 - Comouflages Java as Object Rexx
 - Public routines
- UNO.CLS
 - Easier communication with OOo
 - Automates commen steps

BSF4Rexx

- Bean Scripting Framework(BSF)
 - Developed by IBM in 1999
 - Enables scripting languages to be used in Java
 - Enables scripting languages to access Java objects
- BSF4Rexx
 - 2000: Essener Version by Prof. Rony G. Flatscher based on proof of concept (Peter Kalender)
 - 2003: Augsburger Version
 - Now: Vienna Version

Example

```
oDesktop = UNO.createDesktop()
```

```
xComponentLoader = oDesktop~XDesktop~XComponentLoader
```

```
url = "private:factory/scalc"
```

```
xCalcComponent = -
```

```
  xComponentLoader~loadComponentFromURL(url, -  
  "_default", 0, .UNO~noProps)
```

```
::requires UNO.CLS
```

Example HelloWorld

...

```
xSheet = xCalcComponent~XSpreadSheetDocument~getSheets~ -  
XIndexAccess~getByIndex(0)~XSpreadSheet  
CALL UNO.setCell xSheet, 0, 0, "HelloWorld!"
```

::requires UNO.CLS

Example - AutoFilter

```
...  
xDocument = xCalcComponent~XSpreadSheetDocument  
xSheet = xDocument~getSheets~XIndexAccess~getByIndex(0)~XSpreadSheet  
CALL sys$sleep 3  
CALL UNO.setCell xSheet, 0, 0, "Something"  
...  
xCellRange = xSheet~xCellRange~getCellRangeByPosition(0, 0, 5, 5)  
xAutoForm = xCellRange~XAutoFormatTable  
xAutoForm~autoFormat("Gelb")  
::requires UNO.CLS
```


Thank you for your attention!