Tcl Europe 2006
Bergisch-Gladbach
Germany
26.-27. May 2006

teluno and friends

Wolfgang Großer
wolfgang@grosser-erding.de

What is tcluno?

- tcluno is a bundle of packages
- tcluno is a binding between Tcl and the OpenOffice.org UNO (Universal Network Objects) interface
- tcluno was a shared library and some tcl scripts providing access to the shared library

Which are tcluno's friends?

- tcluno consists of
  - tcluno_soffice
    - tcluno_swriter
    - tcluno_scalc
    - tcluno_simpress
    - tcluno_registry
    - tcluno_type_info
    - tcluno_interface_info
  - tclurtp (uno remote (text|tcl) protocol)
  - itcluno
  - unospection

What were the reasons for creating tcluno?

- customers need excel sheets
- customers have MS Windows os
- server has a linux os
- Rewriting a perl package to write binary excel format

What were the disadvantages of this approach?

- no graphics available
- code was not maintainable
- maximum file size is about 7 MB
- ... but it worked

Is this code maintainable?

```
private method _store_workbook {} {
        ...
        _store_style
        ...
}
private method _store_style {} {
        set header [binary format ${_int16Format}2 [list 0x0293 0x0004]]
        set data [binary format ${_int16Format}cc 0x8000 0x00 0xff]
        _append $header$data
}
```

Compare Java code and Tcl code

Java Code

```
com.sun.star.sheet.XDataPilotTablesSupplier
    xDPSupp =
    (com.sun.star.sheet.XDataPilotTablesSupplier)
    UnoRuntime.queryInterface(com.sun.star.shee
    t.XDataPilotTablesSupplier.class, xSheet);

com.sun.star.sheet.XDataPilotTables xDPTables =
    xDPSupp.getDataPilotTables();

com.sun.star.sheet.XDataPilotDescriptor xDPDesc
    = xDPTables.createDataPilotDescriptor();

com.sun.star.table.CellRangeAddress
    aSourceAddress =
    createCellRangeAddress(xSheet, "A10:C30");

xDPDesc.setSourceRange(aSourceAddress);

com.sun.star.container.XIndexAccess xFields =
    xDPDesc.getDataPilotFields();
```

Tcl Code

```
set xDPSupp
    [itcluno::OfficeUtilities::queryInterface  $xSheet
    com.sun.star.sheet.XDataPilotTablesSupplier]


set xDPTables [$xDPSupp getDataPilotTables]


set xDPDesc [$xDPTables
    createDataPilotDescriptor]

set aSourceAddress [getCellRangeAddress
    $xSheet A10:C30]


$xDPDesc setSourceRange $aSourceAddress

set xFields [$xDPDesc getDataPilotFields]
```

Why creating tclurtp?

- first version of tcluno was a linux only version
- some requests showed the need of a windows version
- no development environment for MS Windows
- Arnulf Wiedemann decided to make a tcl only version for the communication between tcl and OpenOffice.org
- this should provide a MacOS version as well

Why creating itcluno?

- OpenOffice.org has hundreds of
  - classes and interfaces (1600)
  - methods
  - properties
- They are hard to remember
- Most of them are rarely used

a first example

- connect to a desktop

```
set localContext [::tcluno::getComponentContext 1]

set localServiceManager [$localContext ServiceManager]

set unoResolver [$localServiceManager createInstanceWithContext
    "com.sun.star.bridge.UnoUrlResolver" $localContext]

set connect_str "uno:socket,host=localhost,port=2002;urp;StarOffice.ComponentContext"

set remoteContext [$unoResolver resolve $connect_str]

set desktop [$remoteServiceManager createInstanceWithContext "com.sun.star.frame.Desktop"
    $remoteContext]
```

- ... and you have another empty desktop (without a document)

a second example

- set text to Impress document

```
set factory [$document queryInterface [$document ::tcluno::getTclunoType {TypeClass
    com.sun.star.lang.XMultiServiceFactory INTERFACE}]]

set xObj [$factory createInstance $shapeType]

set point [tcluno_soffice::createPoint [concat [lrange $rectangle 0 1] [lindex $rectangle 4]]]

set size [tcluno_soffice::createSize [lrange $rectangle 2 4]]

set xShape [$xObj queryInterface [$xObj ::tcluno::getTclunoType {TypeClass
    com.sun.star.drawing.XShape INTERFACE}]]

$xShape setPosition $point;    $xShape setSize $size;     $page add $xShape

set xText [$xShape queryInterface [$xShape ::tcluno::getTclunoType {TypeClass com.sun.star.text.XText
    INTERFACE}]]

$xText setString $text
```

- ... and a text within a rectangle is created

a first example (with itcluno)

- no equivalent code available
- ... but

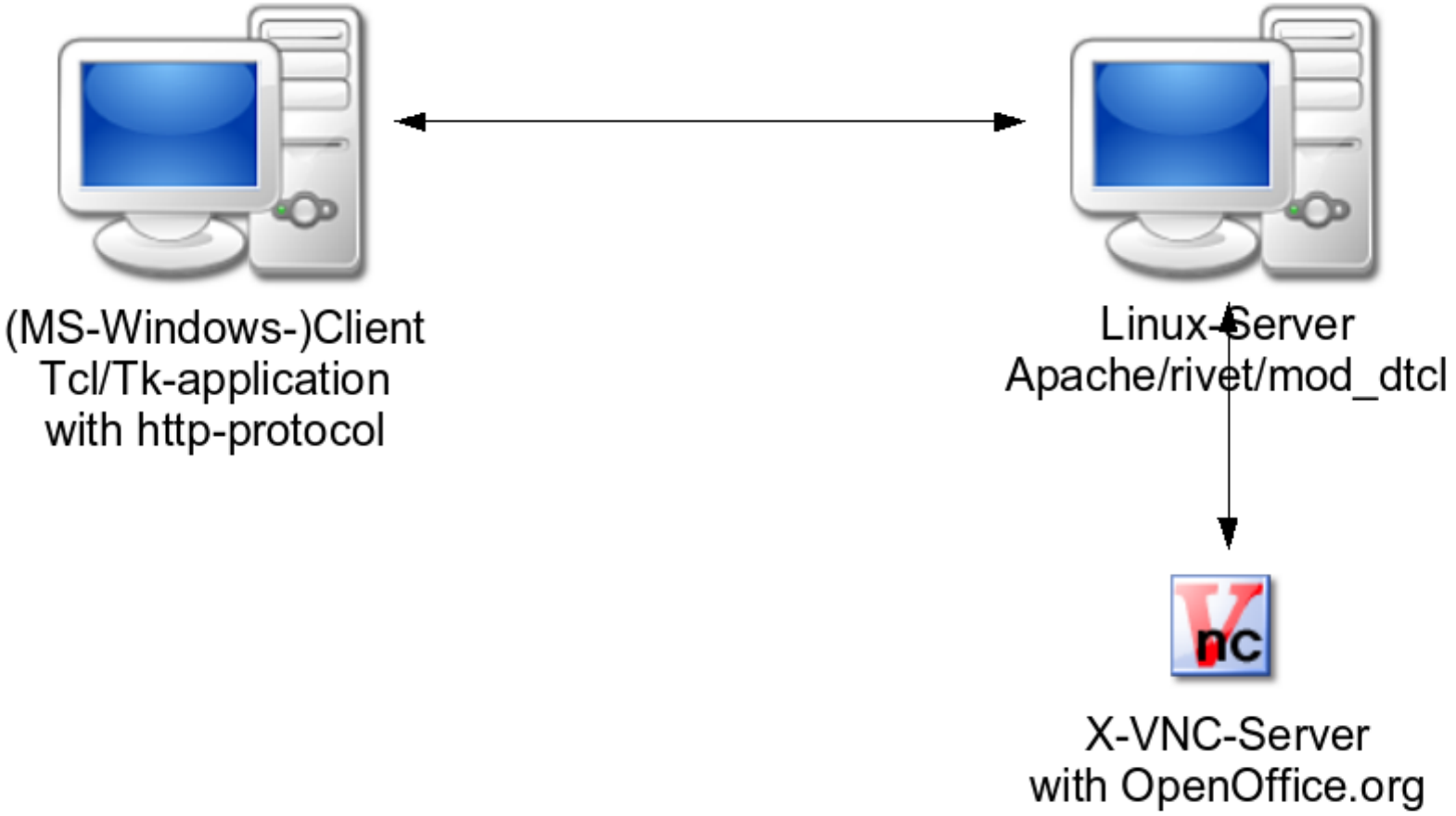
```
set document [itcluno::Presentation document -filename "TclEurope2006.odp]"
```

- ... connects to the desktop and opens the appropriate document

a second example (with itcluno)

- $document setText $page [list 1 2 3 4 cm] $text

Usage of tcluno



why creating unospection?

- All classes and methods are documented in the OpenOffice.org's documentation
- but ...
  - which classes are available?
  - which methods and properties does an object have?
- they all are listed in alphabetical order
- some are shown in the developer's guide
- I can't remember all of them

how do I use unospection?

- unospection consists of
  - a list of all known objects created so far
  - a table of properties
  - a table of methods
  - a list of services
  - a list of interfaces
  - an interface to show the documentation
  - an interface to search through the registry

where to get tcluno and friends?

- tcluno is hosted at sourceforge.net
  - http://sourceforge.net/projects/tcluno
  - single files are kept within the subversion repository (don't browse CVS, it is out of date)

which packages do I need?

- at least you need
  - tclurtp
  - tcluno
- if you don't want to go the hard way
  - itcluno
- if you want to introspect the classes
  - unospection
  - tclunodocs

are there requirements for using tcluno?

- OpenOffice.org has to be started with network support
- edit the appropriate Setup.xcu file
  - in the node <node oor:name="Office">

```
<prop oor:name="ooSetupConnectionURL" oor:type="xs:string">
  <value>socket,host=localhost,port=2002;urp</value>
</prop>
```

- or
  - start OpenOffice.org with parameters:

```
soffice "-accept=socket,host=localhost,port=2002;urp;" "" &
```

what is the status of the documentation?

- the normal way to use tcluno is to use the itcluno classes
- all of the classes and methods in itcluno are documented (well) as html page
- if you want to use the low level tcluno and tclurtp procedures
  - you have to go the hard way
  - read the source code

where can I get some examples?

- itcluno includes some small examples within the html documentation
  - which can be copied and pasted
  - which can be started out of the box in itcluno's demos directory with the RunExamples.tcl script
  - ...

here are some examples

- $:itcluno::library/demos/RunExamples.tcl

Any Questions?