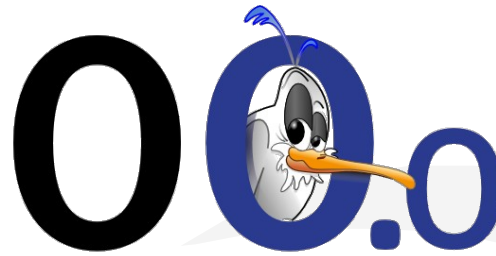


OOoCon 2005



Addons

From production to end-users

Laurent Godard

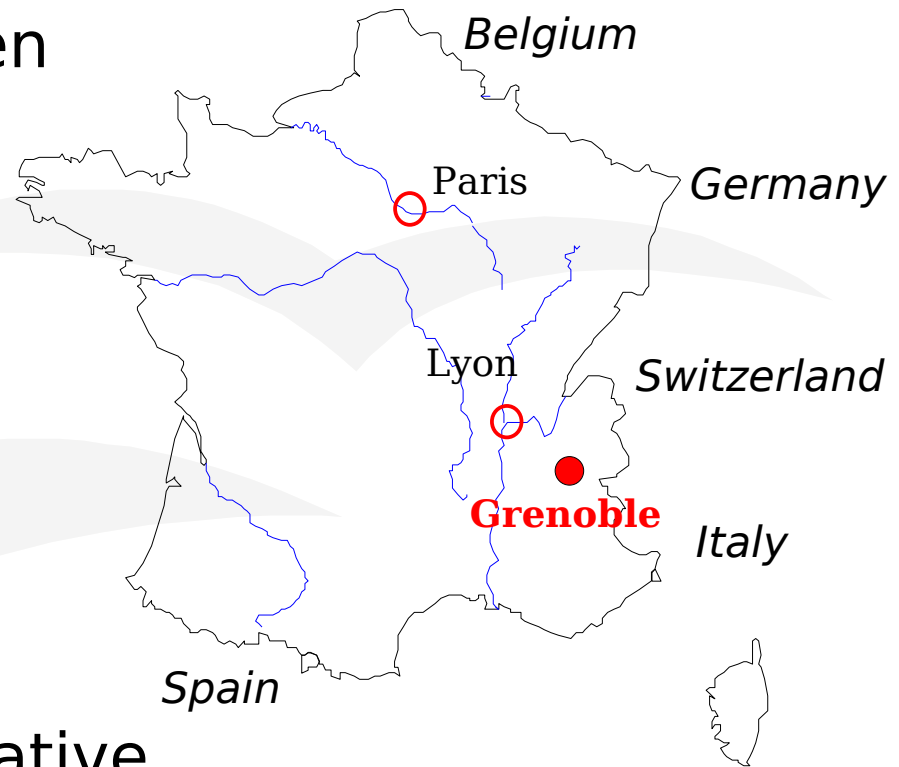


Agenda

- Speaker introduction
- Why this topic ?
- Addons, Extensions ?
- What for ?
- Organizing addons
- Producers needs
- Already existing tools
- Tools for developers
- Policies and guidelines
- Deploy Addons to users
- Native-Lang relay
- Summary
- Discussion - Q & A

Speaker introduction

- 35 years old, married, four children
- Living near Grenoble – France
 - *Grenoble is in french Alpes*
- Discovered OOO 3 years ago
- Involved early in OOO macros
- Author of various tools
 - *OOoConv, DicOOo, FontOOo, OOoWikipedia ...*
- Community contributor representative
- Technical director/software engineer at InDesko/Nuxeo
- In a previous life :
 - *PhD of geomechanics at 3S Laboratory - Grenoble*
« *Robustness and consistency in rheological and numerical modelling in geomechanics* »





Why this topic ?

- New macros developers comes from other office suite as OOo grows in audience
 - *They want to find the toys they had before*
 - *They have some work knowledge they share*
 - *They know nothing about OOo organization and tools*
- OOo needs functionalities
 - *It implies that it at least needs developers What kind of developers*
- OOo 2 is a great step in term of scripting and addon creation
 - *New scripting languages*
 - *Package manager*
- How users are away that new functionalities are available through extensions ?

It is now time to organize the extension creations for OOo



Addons, Extensions ?

- A self-contained Zip file allowing to add functionalities to OOo
 - *The source code*
 - *Some configuration files dealing with GUI integration, parameters*
 - *Other needed resources (icons, images ...)*
- Deployable with OOo standard tools
 - *1.x : using the command line tool pkgchk*
 - *2.x : unopkg command line tool, a GUI is provided*
- Integrated : Fits OOo admin model
 - *For one single user, put in <oooUser>/users directory*
 - *For the whole OOo install (all users), put in <oooInstall>/share directory*
- All is explained in the developer's guide, chapter 4 (4.7.3 for add-ons)
 - <http://api.openoffice.org/docs/DevelopersGuide/Components/Comp>
- See Paolo Montavani presentation



What for ?

- OOo needs functionalities and developers
- Core development
 - *Requires skills such as C++, Java*
 - *Tools and framework (CVS, building tools, compiler ...)*
 - *Long term involvement, learning curve*
- Prevents some developers to come and bring features to OOo
- As scripting, lot of languages can be used, especially with the coming OOo2.0
 - *User can program in their favourite languages and use OOo API for internal OOo scripting with Java, Python, Basic ...*
- Everyone that have some tools or idea simplifying end-user OOo daily experience can develop an addon and share it



Organizing addons

Goals :

- Even newcomers with minimal knowledge of development tools should be able to program addons and know the rules to comply with
- New addons should be advertised, reviewed and listed in a central repository instead of being spread throughout individual websites.
- End-users should be able to install easily all the addons available in the repository



Organizing addons

- Tools
 - *For developers*
 - *For distributing to end users*
- Policies & Guidelines
 - *What kind of licences ?*
 - *Coding guidelines and best practices*
- Native-lang projects involvements
 - *Helping native-lang addon developers submitting and joining the project*
 - *Translators*



Producers needs

- Easy to access information
 - *Quick start guides*
 - *Mentors, dealing only with the “addon” stuff but forward them to correct resources when dealing with API*
 - *First steps in their native languages if needed*
- Helper tools
 - *Some blank, ready to use, helpers implementing best practices*
 - *Pre-programmed framework for translation, parameters handling, help ...*
- Guidelines and rules to participate to OOo
 - *Coding guidelines for all the supported programming languages*
 - *Somewhere to upload*
- Know that such tools exists and where to find them



Already existing tools

- OOoBasic

- *Xray*
- *Addon builder*
- *Some literature*
- ...

- Python

- *PyXRay (beta)*
- *Helper functions*
- *Very few literature*
- ...

- Java/beanshell

- *SDK*
- *Eclipse plugin*
- *A lot of tools and literature*
- ...

- Javascript

- *Something ???*

- External, parallel web sites

- *OOoMacros*
- *Ian's dokuWiki*
- *OooForum*
- *Code snippets*

Some things already exists
which can be a basis if
federated



Tools for developers

- Guidelines document for addon developers
 - *The developers guide adapted on some scripting languages*
 - *Some guidelines and advices*
- Xray
 - *Mandatory to discover OOo API – Should be included in OOo by default*
 - *Already exists for OOoBasic and Python. Introspection is available at API level*
- Addon creator/packager
 - *Already exists for OOoBasic (in french)*
- Standard model in all the languages supported internally by OOo
 - *A standard empty dialog with a navigation mechanism (eg. Panels using step)*
 - *A common interface to ensure consistency*
 - *Ready to translate : a built-in translation framework*

Tools for developers

The image shows a screenshot of the OpenOffice.org 1.1.4 interface. The main window displays a BASIC script with the following code:

```
REM ***** BASIC *****  
  
sub test  
  
  xray.xray thisComponent  
  
end sub
```

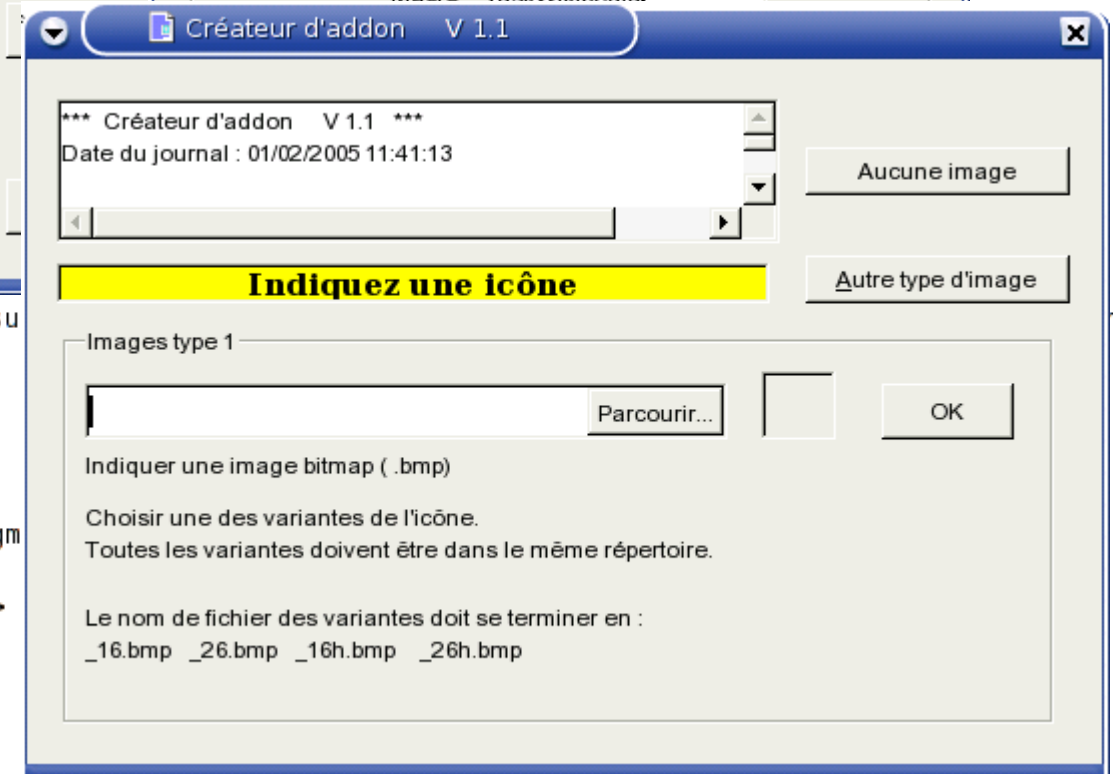
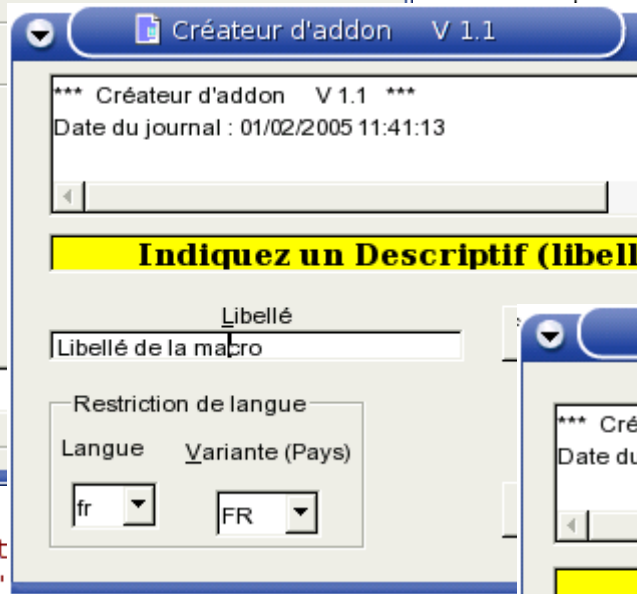
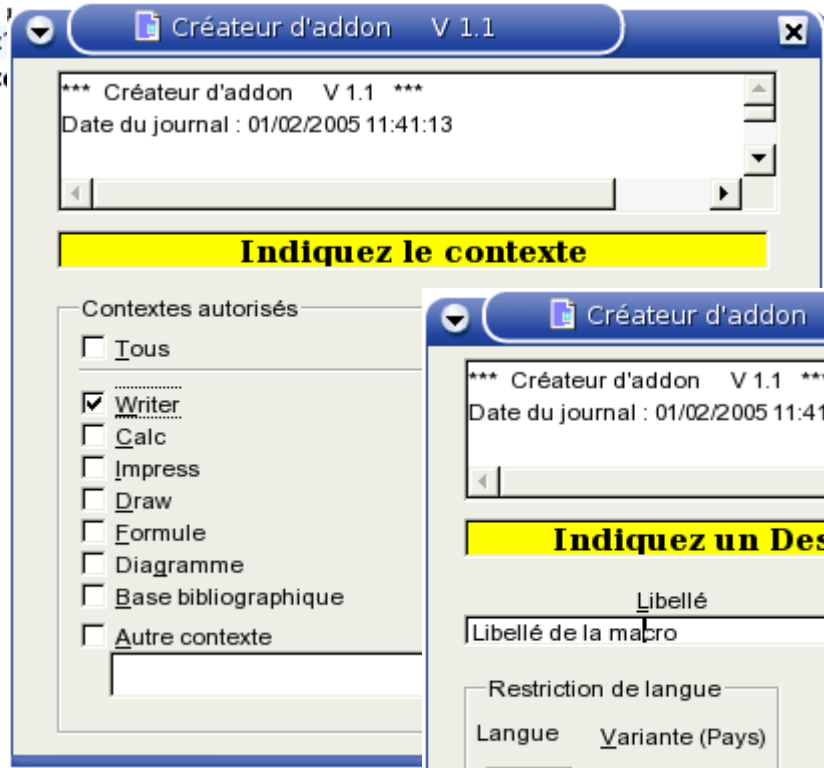
Two Xray tool windows are overlaid on the script. The top Xray window shows the 'Objet d'origine : SwXTextDocument' and the 'Propriétés' tab selected. The bottom Xray window shows the 'Objet d'origine : SwXTextDocument' and the 'Méthodes' tab selected, displaying a list of methods for the 'ThisComponent' object:

Méthodes de l'objet "ThisComponent":		
queryInterface (OBJECT)	as EMPTY	
getViewData (void)	as OBJECT	
setViewData (OBJECT)	as VOID	
getTransferData (OBJECT)	as EMPTY	
getTransferDataFlavors (void)	as ARRAY	
isDataFlavorSupported (OBJECT)	as BOOL	
addPrintJobListener (OBJECT)	as VOID	
removePrintJobListener (OBJECT)	as VOID	
getEvents (void)	as OBJECT	
addCloseListener (OBJECT)	as VOID	
removeCloseListener (OBJECT)	as VOID	
getTypes (void)	as ARRAY	
getImplementationId (void)	as ARRAY	
getParent (void)	as OBJECT	
setParent (OBJECT)	as VOID	
getDocumentInfo (void)	as OBJECT	
disposing (OBJECT)	as VOID	
dispose (void)	as VOID	
addEventListener (OBJECT)	as VOID	

B.Marcelly

Tools for developers

B.Marcelly



```
</node>  
<node oor:name="bt  
<prop oor:name="<br/>  
<value>com.sun.star.text.TextDocument,com.sun<br/>  
</prop>  
<prop oor:name="Title" oor:type="xs:string">  
<value>Incrémenter la taille</value>  
</prop>  
<prop oor:name="URL" oor:type="xs:string">  
<value>macro:///TailleEquations.Module1.Augm<br/>  
</prop>  
<prop oor:name="Target" oor:type="xs:string">  
<value>_self</value>  
</prop>
```



Policies and guidelines

- OOo is a free multilingual multiplatform software
 - *All addons that will be advertised has to be at least under free licence - Do we need to choose among free licences or accept all ?*
 - *Addon has to be multiplatform and verified on supported OOo platforms*
 - *OOo is not english-centric : Addons has to be provided in more languages than english only*
- Addons has to be checked and tested before being supported by OOo
 - *Sources needed*
 - *Signing or trusting source at least (and md5 ?) for delivering to end-users*
 - *Mirrored by OOo to be sure the addon will be available. How ?*



Policies and guidelines

- If very often used and licensing compliant, the addon will be included into OOo
 - *Easy to maintain*
 - *Fits the internal tools (esp. translations, help files, parameters ...)*
- Addons will be used by end-users
 - *The UI has to be consistent : the same way driving users through processes*
 - *No matter the underlying scripting language*
 - *Close to built-in wizards/autopilots philosophy and rendering*
 - *Give tools to easy translate the addons and submit the translation : often users can translate a set of sentences to make an addon they find useful accessible to their native-lang community*



Deploy Addons to users

- Not all development have to be included in OOo core
 - *Only the one that are used by all users*
 - *Prefer uno packages*
- New OOo features for deploying addons
 - *Since 2.0, process is easy for users.*
 - *The goal is to be able to browse "What's new"*
- A repository that can be browsed
 - *A new tool for end user that allow install an addon in "one" click*
 - *Some presentation lines about this addons and metadatas (OOo version, supported languages, addons version, ...)*
 - *Define several levels to qualify addons*
 - *Untrusted (default)*
 - *Verified / Accepted : the main addons (Signed ?)*
 - *Popular / very useful : additional work should perhaps be done for giving it by default into OOo*
 - *Has to be as simple as possible*
 - *Users feedback ? A permanent poll ?*



Native-Lang relay

- New developers need to be welcomed and helped sometimes starting in their own native language
- Even if great tools are provided, newcomers do not know they exist and need to be informed
 - *Native-lang users/developers have to be relays*
- Translate starting guides and documentations
- Translate addons
 - *Some tools have to be provided to ease the translation of addons*
 - *That implies that addons comply some design guidelines*
 - *Translate the description*
 - *Ideally this translation tool should be used by end-users without any development knowledge*



Summary

- Wizards in all “scripting” languages
 - *Navigation mechanism*
 - *Translation*
 - *Parameters*
- Starting guides for all “scripting” languages
- Policies and development guidelines, charter (free software, branding location ...)
- Reviewing team for accepting addons
- A wizard for end users
- A simple tool for translation
- A dedicated project with web resources
 - *let start with dev@scripting.openoffice.org to organize*

Thanks and ...



WE NEED YOU



Discussion

- Ideas
 - Opinions
 - Questions
 - Answers
 - Volunteers ...
- 