



OpenOffice.org Programmability – at a glance

Jürgen Schmidt
OpenOffice.org
Sun Microsystems, Inc.





OpenOffice.org Programmability

– at a glance

- Motivation
- Introduction UNO
- API Concepts & Design Patterns
- Introduction in Extensions
- Service Provider Interfaces
- Q & A



Motivation

... to program with and for OpenOffice.org

- Growing popularity of ODF worldwide
 - > Standardized file format
 - ODF = Open Document Format for Office Applications
 - OASIS and ISO/IEC 26300
 - > Adoption of ODF in more and more public administrations
- Growing popularity of OpenOffice.org
 - > Most popular ODF manipulating office suite > 100 Million downloads
- Multi platform support
 - > Solaris/Open Solaris, Linux, Windows, Mac OS, ...



Motivation

... to program with and for OpenOffice.org

- Demand for
 - > Customization of OpenOffice.org
 - User interface changes
 - Exchange, intercept commands
 - > Extending OpenOffice.org with new functionality
 - Calc Add-ins, Add-ons, Filters, Embedded Java objects, ...
 - > Integration in existing workflows or other applications
 - e.g. OpenOffice.org Bean
 - Create, change, convert ODF documents



Motivation

... to program with and for OpenOffice.org

- Community building
 - > Lower the entrance barrier for developers
 - > Extensions
 - Encapsulated mini projects
 - Program against API's
 - Interface/connector between OpenOffice.org and other applications



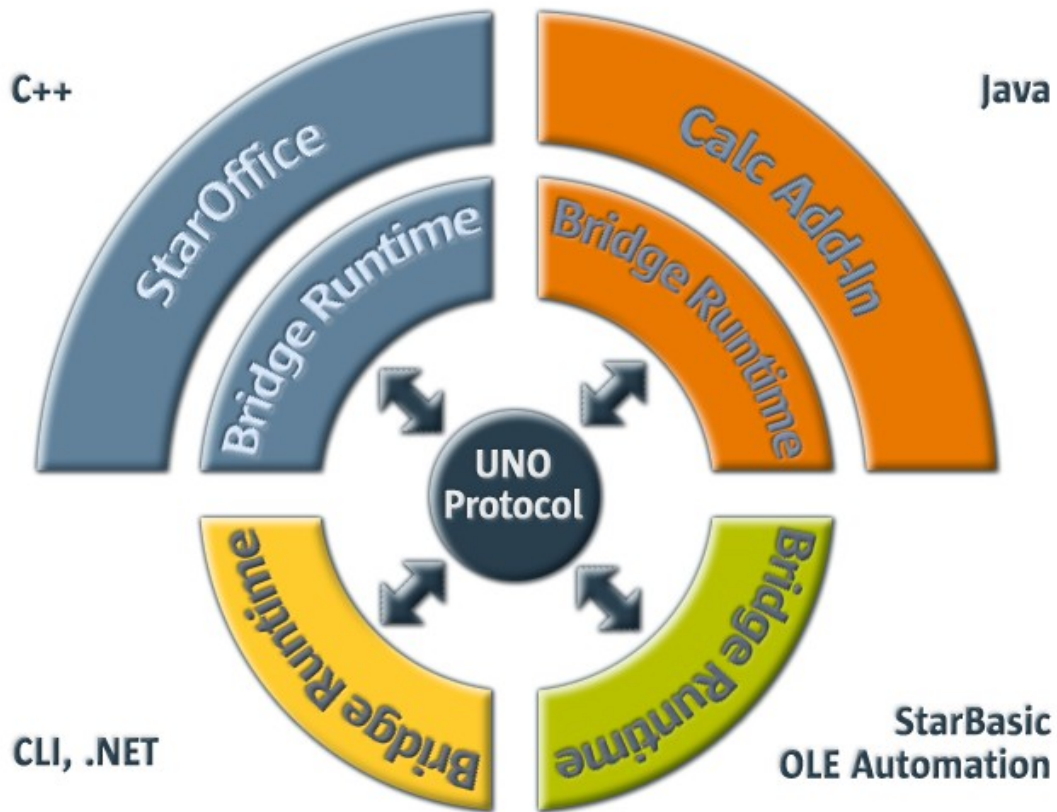
Introduction into UNO

- Universal Network Objects (UNO)
 - > Component technology \approx RMI/DCOM, Corba
 - > Started in 1997
 - At this time no sufficient component technology was available
 - > Language independent
 - API defined in UNOIDL
 - Multi language support (Java, C++, StarBasic, CLI languages, Python, ...)
 - > API calls work in-process, inter-process or remotely
 - > Remote transparency
 - > Independent from the office
 - URE = UNO Runtime Environment



Introduction into UNO

UNO get over boundaries





Introduction into UNO

- Universal Network Objects (UNO)
 - > No or minimal code generation
 - Only type definitions
 - e.g. Interfaces: in C++ abstract classes and in Java normal Java interfaces
 - > Implementations exchangeable
 - > Multi threaded
 - > Unicode strings
 - > Exceptions for error handling



API Concepts & Design Patterns

- Design goal “*One API for everything*”
 - > Internally, for better modularization
 - > Macros, remote automation
 - > Exchange/modify components
 - > Extend functionality by new components (extensions)
- Programming against specifications
- Instantiation of new UNO objects
 - > By a factory using a service name or by service constructor methods
 - > Context dependent
 - > Implicitly by accessing subobjects as return value



API Concepts & Design Patterns

- Old style services
 - > Some services are globally available at the service manger
 - e.g. `com.sun.star.bridge.UnoUrlResolver`
 - > Other services are available at specific factories
 - e.g. `com.sun.star.text.TextFieldSupplier` (over the service factory of a `TextDocument`)
 - > Yet other services merely present abstract entities
 - e.g. `com.sun.star.document.OfficeDocument`
 - > A few services are just documentation for sequences of properties
 - e.g. `com.sun.star.document.MediaDescriptor`



API Concepts & Design Patterns

- New style services
 - > Available at the global service manager
 - > Better integrated in the type system
 - 1:1 relation, service corresponds to exactly one interface
 - You obtain service with its specific interface type
 - You pass a service instance into and out of methods by its specific interface type
 - > Instantiation via Service constructor methods
 - Default constructor
 - Parameterized constructors
 - Exception support for service constructors



API Concepts & Design Patterns

- Multiple inheritance interfaces
 - > Group related interfaces together
 - > Resulting super interfaces can be passed around as method parameters etc.
 - > Client code get rid of queryInterface to navigate among related interfaces of objects
 - > Can take over most of the roles service descriptions previously had
 - Attributes replace former properties
 - “get” and/or “set” methods allows easier and type safe access



API Concepts & Design Patterns

Example (UNOIDL)

```
interface XIfc1 {
    void func1();
};
interface XIfc2 {
    void func2();
};
service foo.bar.SomeService {
    interface XIfc1;
    interface XIfc2;
};
```



```
interface XIfc1 {
    void func1();
};
interface XIfc2 {
    void func2();
};
interface XService {
    interface XIfc1;
    interface XIfc2;
};
service foo.bar.SomeService :
    XService {
        create([in] long arg1,
            [in] string arg2);
};
```




API Concepts & Design Patterns

Example code (old style)

```

XComponentContext context =
    com.sun.star.comp.helper.Bootstrap.bootstrap();
XIfc1 xifc1 = (XIfc1)UnoRuntime.queryInterface(
    XIfc1.class,
    context.getServiceManager().
        createInstanceWithArgumentsAndContext(
            "foo.bar.SomeService",
            new Any[] { new Integer(10), "Whatever" },
            context));
xifc1.func1();
XIfc2 xifc2 = (XIfc2)UnoRuntime.queryInterface(
    XIfc2.class, xifc1);
xifc2.func2();
    
```



API Concepts & Design Patterns

Example code (new style)

```
XComponentContext context =  
    com.sun.star.comp.helper.Bootstrap.bootstrap();  
XService xservice = foo.bar.SomeService.create(  
    context, 10, "Whatever");  
xservice.func1();  
xservice.func2();
```



API Concepts & Design Patterns

Common Design Patterns

- Factory
- Global and document centric
- PropertySet, PropertyAccess, ...
- Collection/Containers
- Enumerators/Iterators
- X...Supplier
- Events
- Exceptions for error handling



Introduction into Extensions

- Functional extensions to the existing office core functionality
- **.oxt zip** packages containing UNO components, macro libraries and/or non code data
 - > xcu/xcs files, type libraries, templates, galleries
- Easy deployment via
 - > Extensions Manager
 - > System integration
 - Double click, “Open with ...”
- Versioning
- Dependencies
 - > Minimal version dependency to Office versions



Introduction into Extensions

- Online update
 - > currently users must trust the connection and source
 - > HTTPS is coming with OpenOffice.org 2.4
 - > Digital signatures are planned
- License support (simple EULA)
- Extensions repository
 - > <http://extensions.services.openoffice.org>
 - > Free and commercial extensions
 - > Supports the online feature
- Option pages
 - > Integrated in the Office option dialog



Introduction into Extensions

- Future plans
 - > Extendable help (integration in office help system)
 - > Allow semi-automatic update through web site
 - > Automatic notification about updates (combined with office)
 - > Provide information about new extensions (from repository)
 - > Support of https
 - > Digital signatures
 - > GUI redesign
 - > Bundled Extensions
 - > Improve NetBeans OpenOffice.org API Plugin
 - > More UNO AWT controls (grid control)



Introduction into Extensions

Most Popular Extensions | OpenOffice.org repository for Extensions - Mozilla Firefox
File Edit View History Bookmarks Tools Help

User:
Password:
Login
Register

Extensions

Welcome to the home of OpenOffice.org extensions!

Browse extensions

Extension

- My Extensions
- OpenOffice.org Extensions

[Get more extensions here...](#)

Home

Extensions

- All
- By Application
- By Operating System
- Highest Rated
- Most Popular
- Most Recent

Search

Tags

calc Clip Art
document
management
documents Editing
eurooffice free gallery

Downloads initiated: 33745 of version: 1.0.1. The newest version is shown below.

Sun Report Builder

by Sun Microsystems, Inc.

Create with the Sun Report Builder stylish, smart-looking database reports. The flexible report editor can define group and page headers as well as group and page footers and even calculation fields are available to accomplish complex database reports.

Version: 1.0.1
Date: 2007-Oct-05
License: opensource

1 comment [Click here](#) for more information

(9 votes)

Downloads initiated: 26732 of version: 1.0. The newest version is shown below.

Professional Template Pack - English

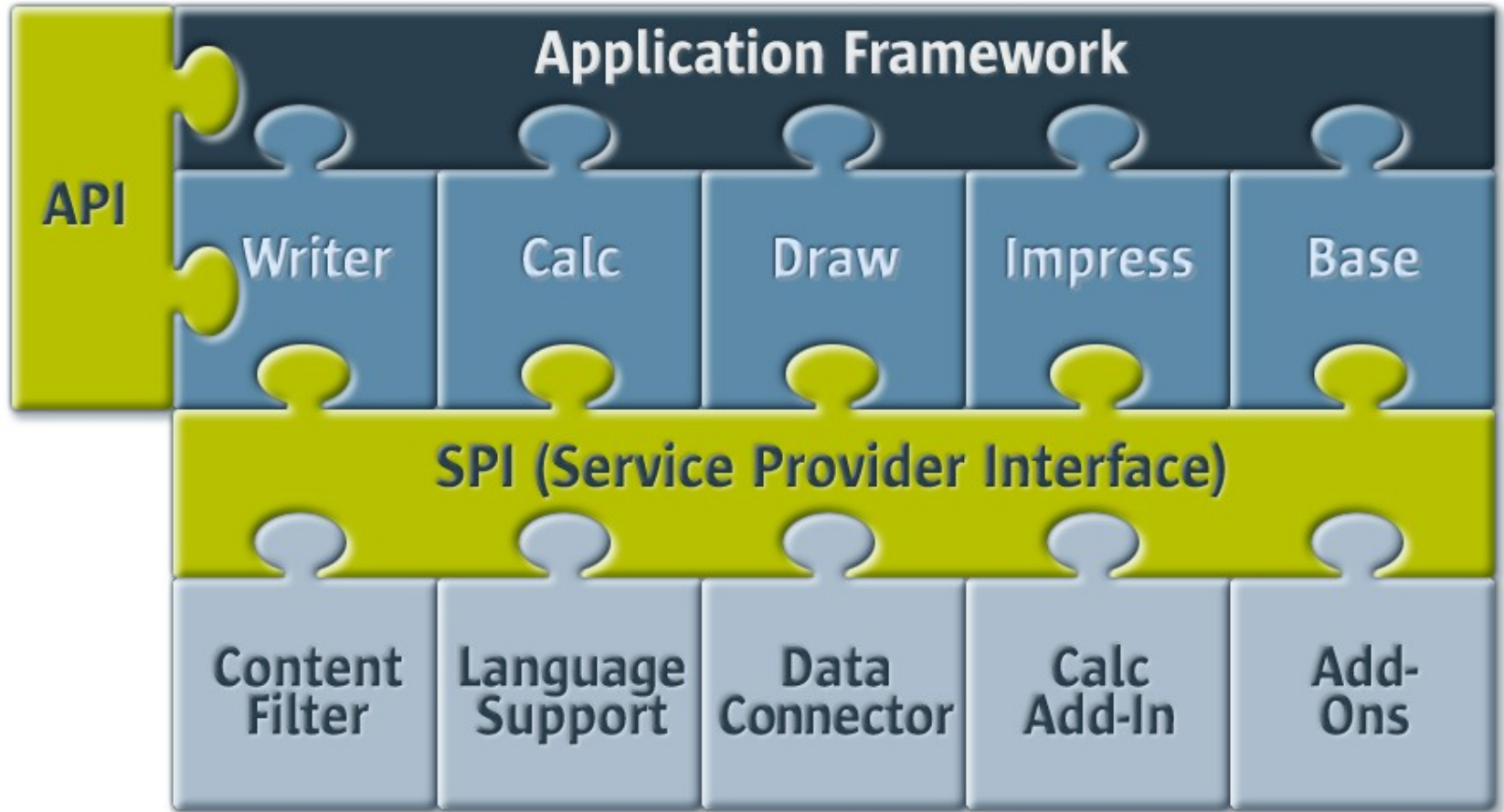
by Sun Microsystems, Inc.

Use the OpenOffice.org template package to save time and professional "look and feel" of templates ensure output is noticed and read. Each template makes creating personalised and individual OpenOffice.org documents a breeze with built-in images and text styles. All you need to do is to add your own content.

More than 80 templates for OpenOffice.org Writer, Calc, Draw and Impress



Service Provider Interfaces





Service Provider Interfaces

Addons

- UI integration via XML configuration
 - > Context dependent (e.g. Writer or Calc only)
 - > Own top level menu and/or toolbar
 - > Merging in existing menus and/or toolbars
- I18n support
- Support by the OpenOffice.org API Plugin for NetBeans
 - > Wizard for high level command, menu and toolbar definition
 - Only new menu and toolbar support, no merging at the moment
 - > Full functional generated code skeleton
 - Build and deployable out of the box



Service Provider Interfaces

Addons

- Job Addons
 - > Special Addons without UI
- Job definition via XML configuration
- Every application or document event can be used to start jobs: *StartApplication*, *OpenFirstWindow*, *NewDocument* etc.
- Jobs can install further components that use other APIs, e.g. EventListeners (PrintListeners, ModifyListeners etc.)



Service Provider Interfaces

Calc Addins

- Built-in functions for the Spreadsheet application
- Restricted subset of UNOIDL types for parameter and return types
 - > Parameter types: *int*, *double*, *string*, *int[][]*, *double[][]*, *string[][]*, *Object (any)*, *Object[][]*, *XCellRange*
 - > *XPropertySet* (implicitly set by the application, only one parameter of this type for each function),
 - > *Object[]* (for varying parameters, allowed as last parameter only, filled with remaining parameters of the function call)
 - > Return types: *int*, *double*, *string*, *int[][]*, *double[][]*, *string[][]*, *Object*, *Object[][]*, *XVolatileResult*



Service Provider Interfaces

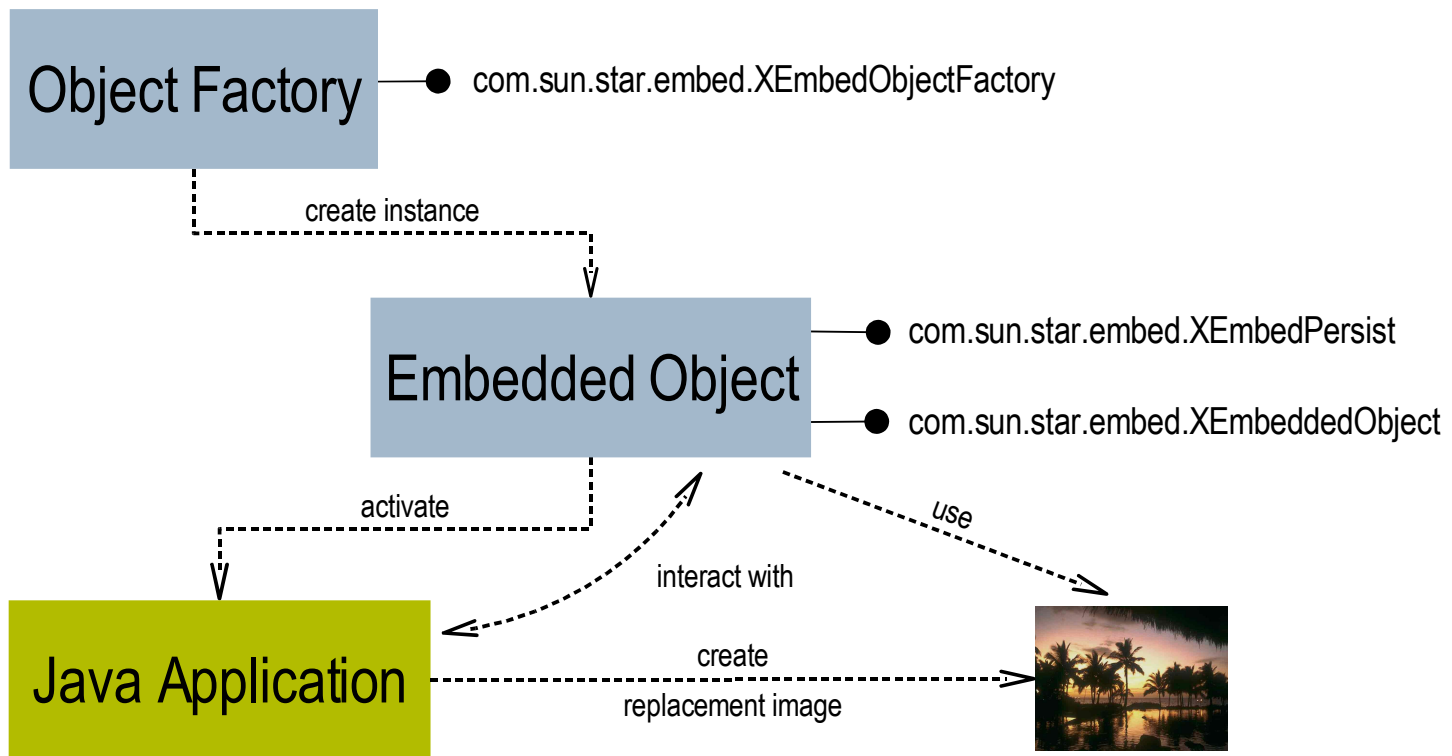
Calc Addins

- Smart integration in built-in function dialog
 - > Preselection of function category
- Localization support
 - > e.g. display name, human readable description
- Support by the OpenOffice.org API Plugin for NetBeans
 - > Wizard for high level function definition
 - > Full functional generated code skeleton
 - Build and deployable out of the box



Service Provider Interfaces

Java Embedded Objects





Service Provider Interfaces

Smart Tags

- Smart tags have been introduced with MS Office to add contextual information to office documents
- Core functionality in OpenOffice.org 2.3
- Smart Tags are provided by Smart Tag libraries
 - > Ideally deployed as extension
- Basic functionality of a Smart Tag library consists of
 - > Smart Tag recognizer component
 - `com.sun.star.smarttag.SmartTagRecognizer`
 - > Smart Tag action component
 - `com.sun.star.smarttag.SmartTagAction`



Service Provider Interfaces

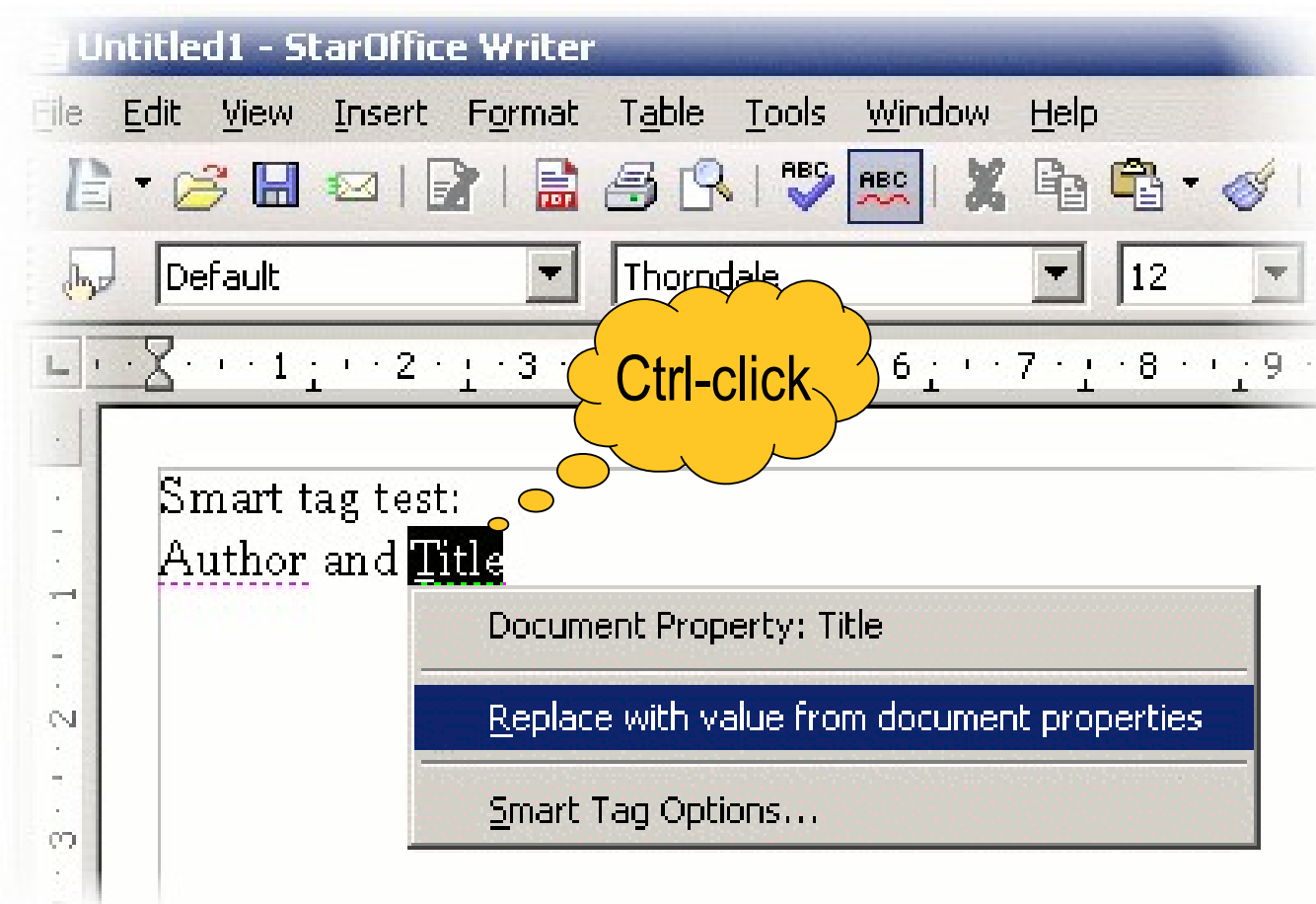
Smart Tags

- Localization support via the Smart Tag interfaces
- Recognized Smart Tags are underlined violet
- Ctrl-click on recognized Smart Tag opens Smart Tag menu
 - > Here comes the Smart Tag libraries into the game
 - > Captions of available actions are shown here
- Smart Tag library can support several recognizer with corresponding actions



Service Provider Interfaces

Smart Tags





Summary

- OpenOffice.org is programmable in different languages ✓
- Fine grained API for nearly everything ✓
 - > You miss something? Please submit an API feature request
- SPI's for specific functional areas ✓
- Office functionality usable in own applications ✓
- Smart deployment of Office extensions as „oxt” packages ✓
- Growing tools support ✓
 - > Lower entry and fast feeling of success
 - > Automation of recurring tasks
 - > Speedup development
 - > Reduce development costs



More information

- API project
 - > home page: api.openoffice.org mailing list: dev@api.openoffice.org
 - > IRC (freenode): #ooo-api for all API relevant topics
- Extensions project
 - > home page: extensions.openoffice.org
 - > mailing list: dev@extensions.openoffice.org
 - > IRC (freenode): #ooo-ext for general extension topics
- OpenOffice.org Wiki
 - > OpenOffice.org API plugin for NetBeans
 - http://wiki.services.openoffice.org/wiki/OpenOffice_NetBeans_Integration



Q & A



OpenOffice.org Programmability – at a glance

Jürgen Schmidt
juergen.schmidt@sun.com