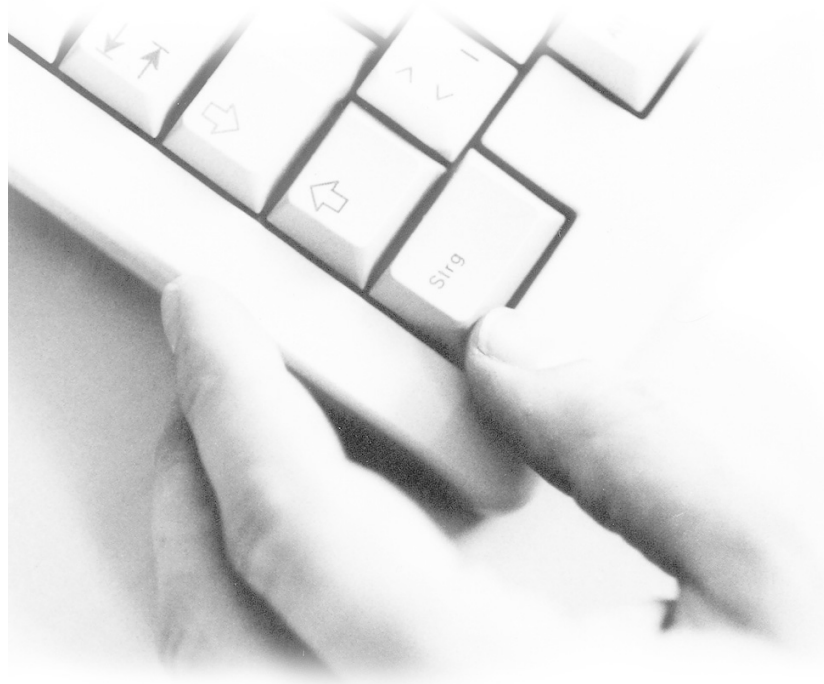


Herzlich Willkommen.



# Effiziente Entwicklung für StarOffice durch UNO Wrapper-Klassen

Volker Weidner  
Mathias Supp

# Über uns...

- Seit 1996 Java Entwicklung
- Fokus: kundenspezifische Projektabwicklung
- 2003/2004: StarOffice Migration Toolkit in Kooperation mit Sun Microsystems GmbH

# Übersicht

- Motivation
- Zugriff auf die StarOffice API
- UNO aus der Java Perspektive
- Verbesserungsidee / Lösungsansatz
- Life Demo
- Einschränkungen und Ausblick
- Noch Fragen?

# Motivation (1)

Wie kann man StarOffice / OpenOffice...

- in andere Anwendungen integrieren?
  - Reporting / PrintEngine
  - Formular- / InPlace-Editor
- erweitern?
- Tools entwickeln?
- Dokumente mit (Makro-)Funktionen versehen?

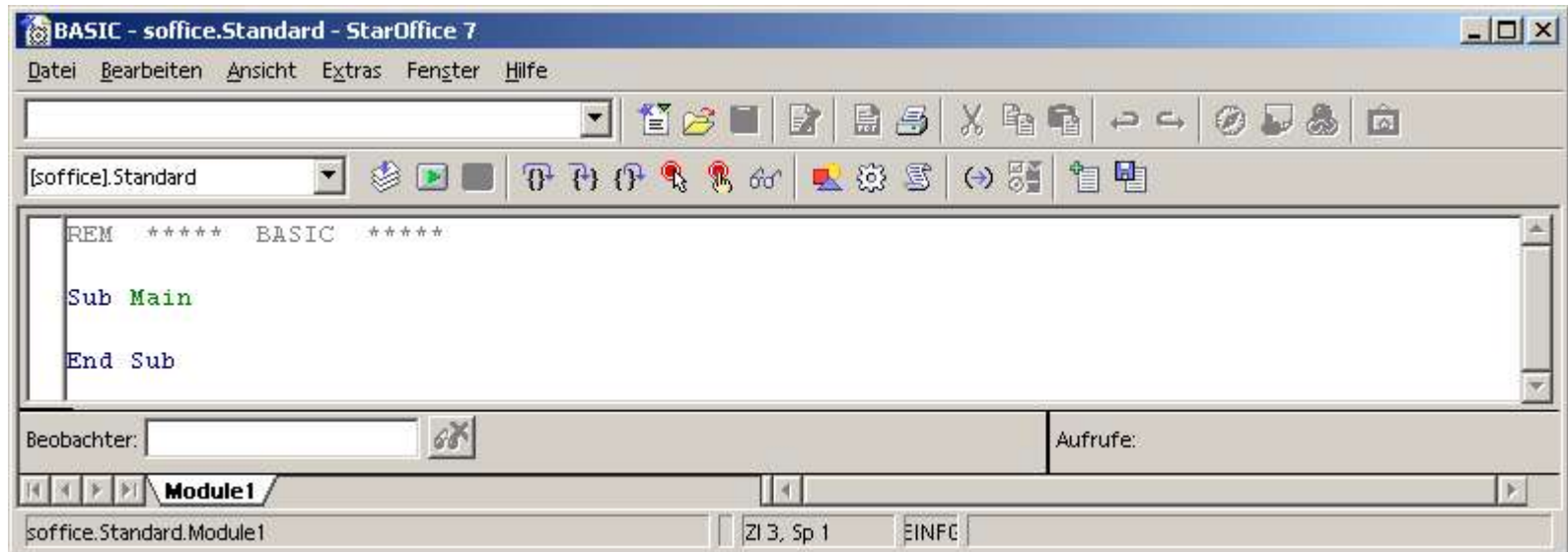
## Motivation (2)

- Steuerung von StarOffice / OpenOffice durch UNO-Schnittstelle möglich
- Generischer Ansatz
  - Sprachunabhängig
  - Spezielle Vorteile einzelner Sprachen nicht genutzt

Idee: Spezialisierte Lösung für Java anbieten

# Zugriff auf die StarOffice API (Intern) 1

StarBasic ⇨ eingebaute IDE in StarOffice



# Zugriff auf die StarOffice API (Intern) 2

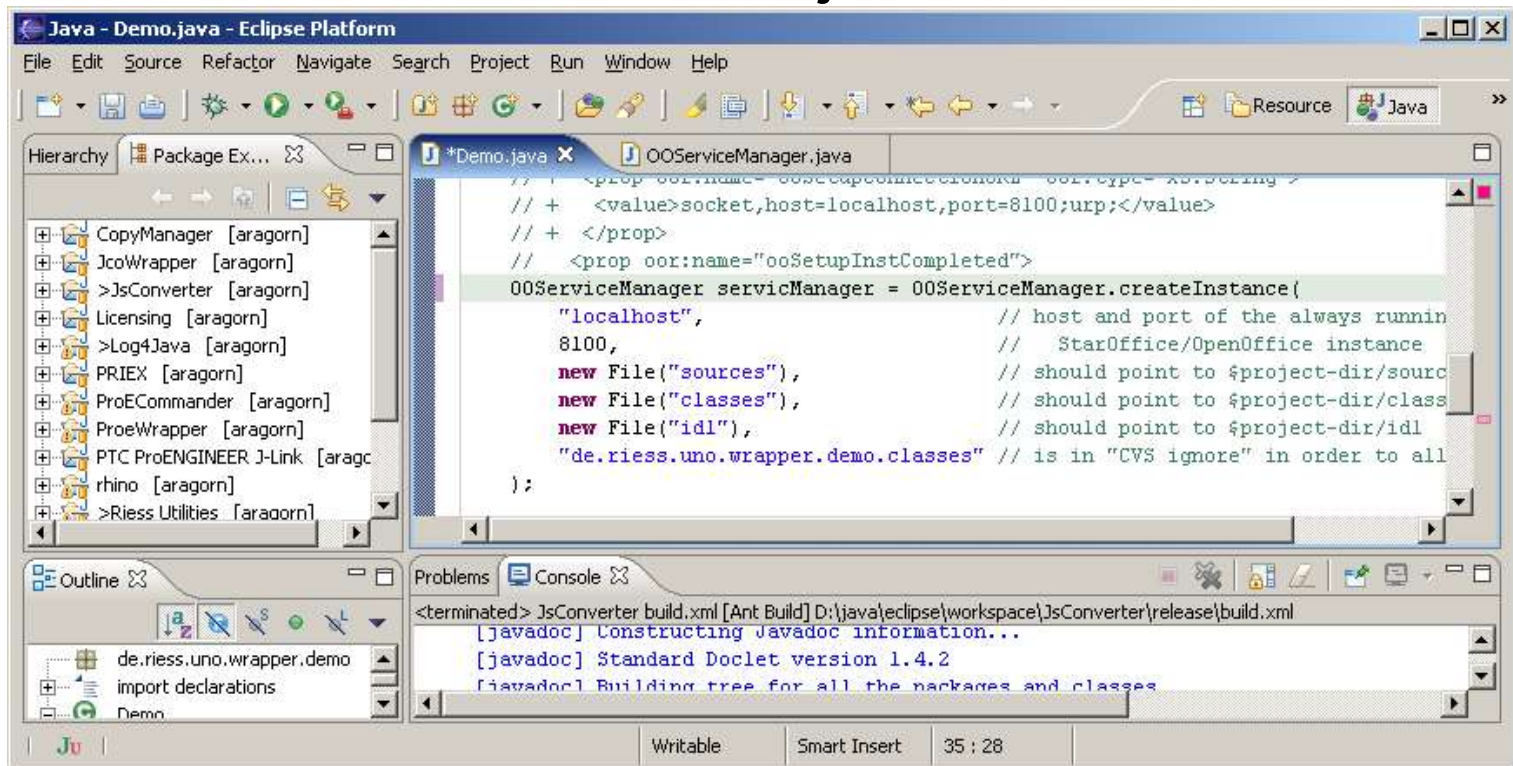
## StarBasic

- Vorteile:
  - Eingebaut – aufrufbar über Menü
  - Einfacherer Zugriff auf StarOffice API
- Nachteile:
  - Code aus Macro-Recorder verwendet nur Slot-Kommandos
  - IDE / Formularelemente nicht zeitgemäß (Keine Command-Completion)



# Zugriff auf die StarOffice API (extern) 1

UNO = Universal Network Objects



# Zugriff auf die StarOffice API (extern) 2

UNO Nutzung beispielsweise von Java aus.

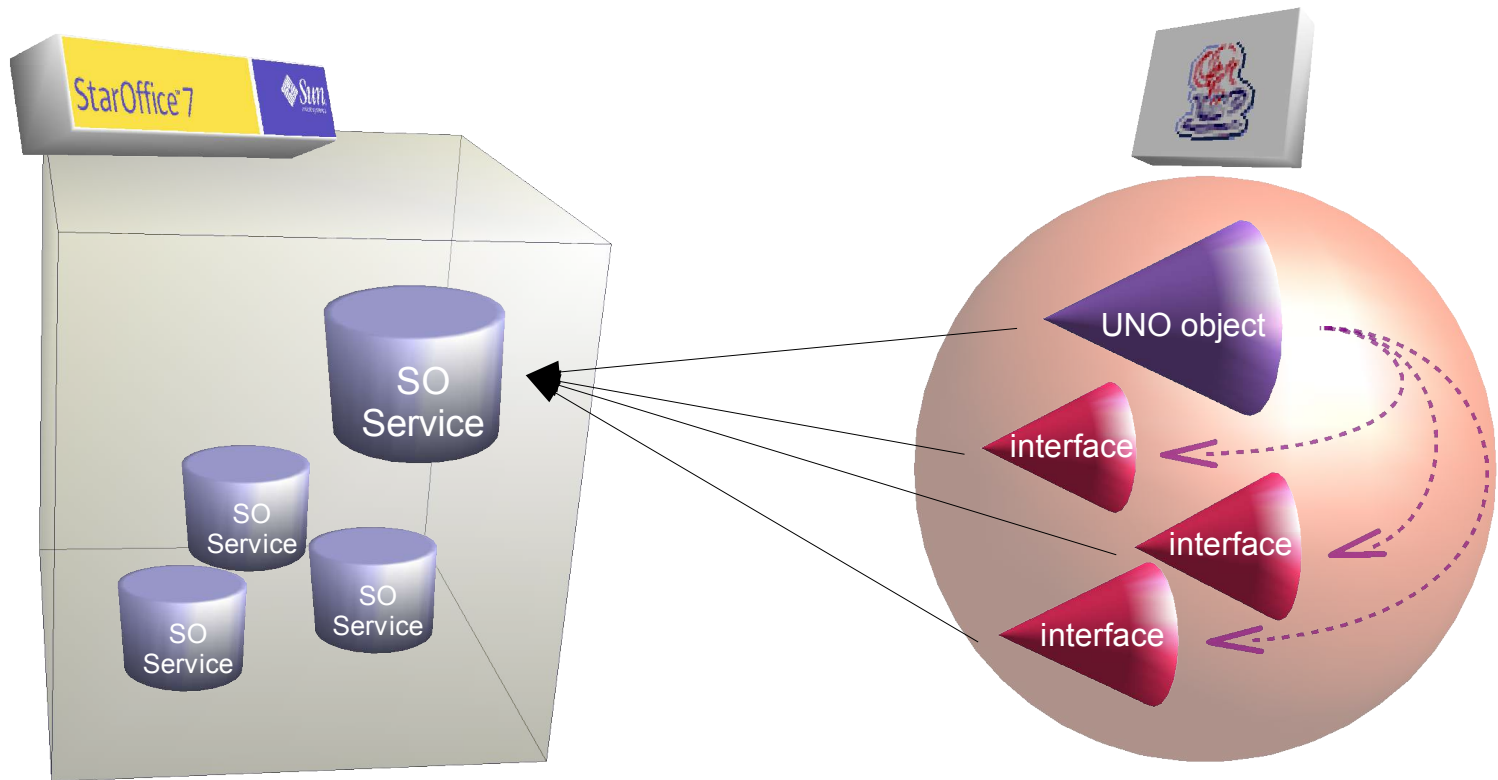
- Vorteile:

- Standard IDE kann eingesetzt werden (Refactoring / Code Completion)
- Objektorientierte Umgebung

- Nachteile:

- Zugriff auf StarOffice API via UNO unhandlich
- Keine passende Dokumentation für Java Entwicklungen

# UNO aus der Java Perspektive (1)



## UNO aus der Java Perspektive (2)

- SO Service implementiert
  - Interface 1
  - Interface 2
  - ...
- UNO Service kann per `queryInterface(Typ)` Stubs für Interfaces liefern
- Property Zugriff namentlich über untypisierten `getter/setter`
- Erzeugung von Objekten?
- Filter Zugriff?

# Live Demo

## Java Uno Wrapper @ Work

IDE = Eclipse

präsentiert durch  
Mathias Supp

# Java-Uno Wrapper (1)

- Leistungsmerkmale
  - Kapselung der Socket-Verbindung
  - Kapselung bei Zugriff auf ComponentFactory und Desktop
  - automatische Erzeugung neuer Wrapper Klassen incl. Javadoc Kommentare
  - Wrapper Klassen enthält sichtbar alle Services, Methoden, Factories und Properties des zugehörigen Objektes
  - streng typisierte Signaturen der Property-Zugriffe

# Java-Uno Wrapper (2)

- Vorteile
  - Command Completion
  - Typsichere Property Zugriffe
  - Keine queryInterface Aufrufe mehr
  - Wrapperklassen mit Javadoc kommentiert
  - Factory Methoden zur Instanziierung von Service Objekten

# Kontaktinformation

.riess applications gmbh  
Draisstraße 10  
76307 Karlsbad

Telefon: +49 7202 707 - 0  
Email: [volker.weidner@riess.de](mailto:volker.weidner@riess.de)  
Email: [mathias.supp@riess.de](mailto:mathias.supp@riess.de)  
Web: [www.riess.de](http://www.riess.de)



# Noch Fragen?

Jetzt ist Ihre Chance...

# Vielen Dank für Ihre Aufmerksamkeit