operating systems
+middleware

Realisierung einer verteilten Sprite-Umwelt mit Java und C# unter Verwendung von .NET-Remoting und CORBA

Alexander Großkopf
Alexander Küchler
Robert Schuppenies
Sebastian Steinhauer

kum@sigmadelta.de

06/2004

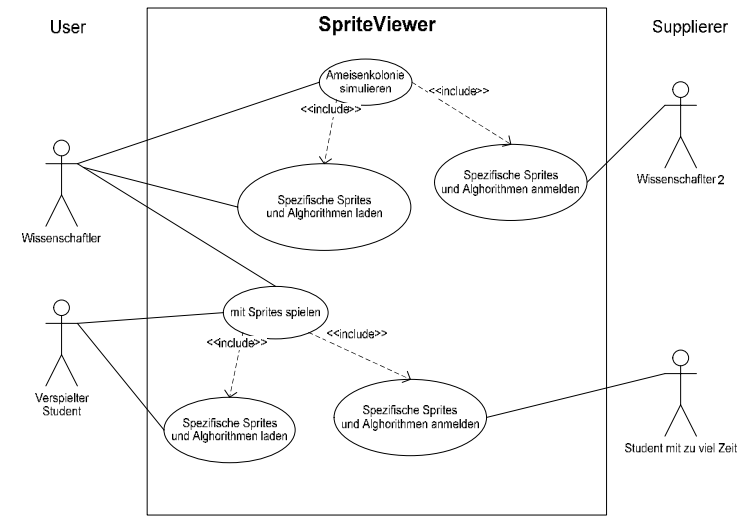
Gliederung

- Die Idee
- Ein Anwendungsbeispiel
- Herangehensweise
- Architektur
- IDL
- CORBA – Implementierung
- .NET Remoting – Implementierung
- Vorführung
- Was nehmen wir mit
- Quellen

Die Idee

- Sprites dargestellt durch DirectX
- Dynamisches Laden und Entfernen von Sprites
- Individuelle Konfiguration von Sprites
- Verteilung von Berechnung
  - ➔ CORBA
- Verteilung von Daten
  - ➔ .NET-Remoting

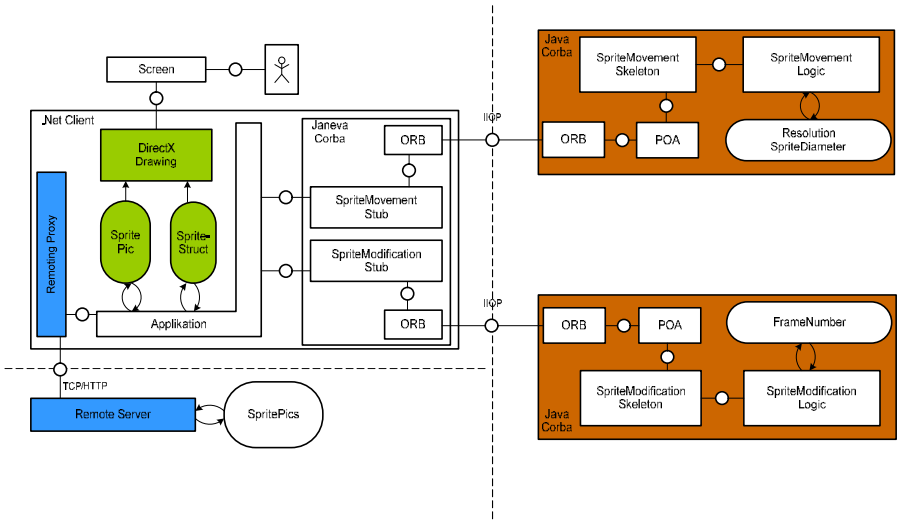
Use Case - Diagramm



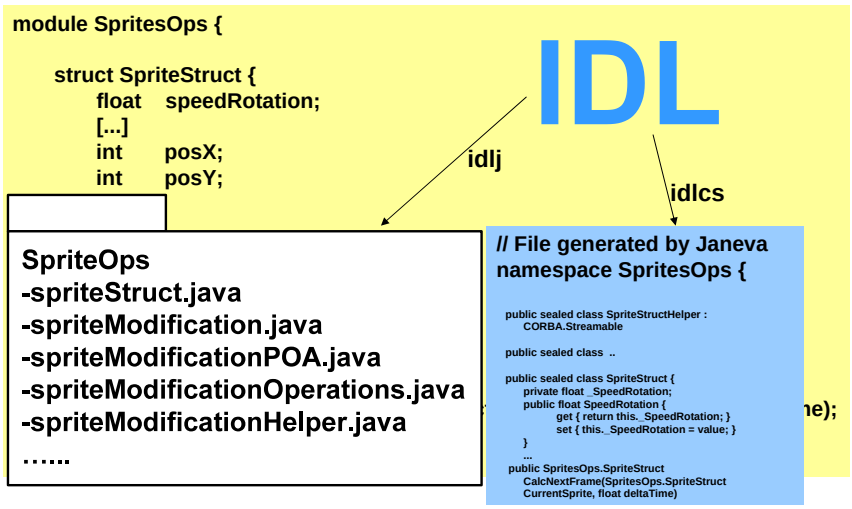
Herangehensweise

- Programmierung gegen ein Interface
- Extreme Programming
- Test mit NUnit & JUnit
- Entwurfsmuster
  - Adapter
  - Proxy
  - Strategy
  - Decorator

Die Architektur



IDL



CORBA-Implementierung

- Verbindung zu einem entfernten Objekt mit CORBA

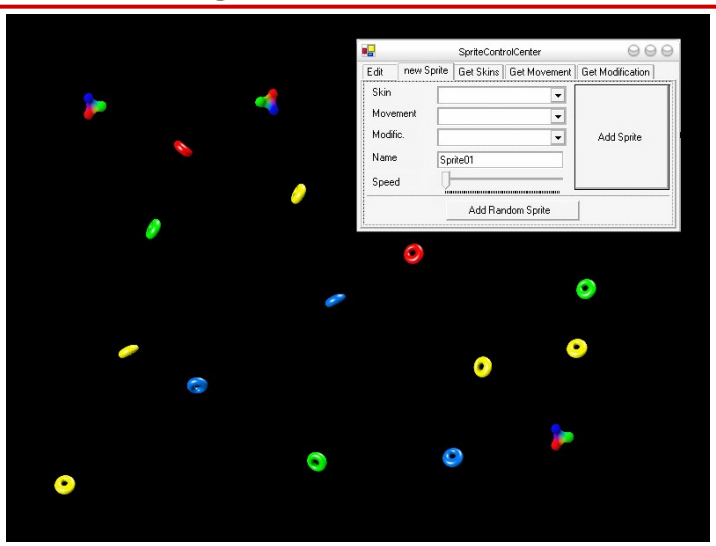
```
CORBA.ORB orb = CORBA.ORB.Init(args);
String ior = IOR;
CORBA.Object NameServiceObj =
  orb.StringToObject(ior);
CosNaming.NamingContextExt NameService =
  CosNaming.NamingContextExtHelper.Narrow(NameServic
  eObj);
CORBA.Object CORBAMovObj =
  NameService.Resolve(NameService.ToName(Name));
SpritesOps.spriteMovement MovObject =
  SpritesOps.spriteMovementHelper.Narrow(CORBAMovObj
  );
return MovObject.CalcNextPos(CurrentSprite,
  deltaTime);
```

.NET Remoting-Implementierung

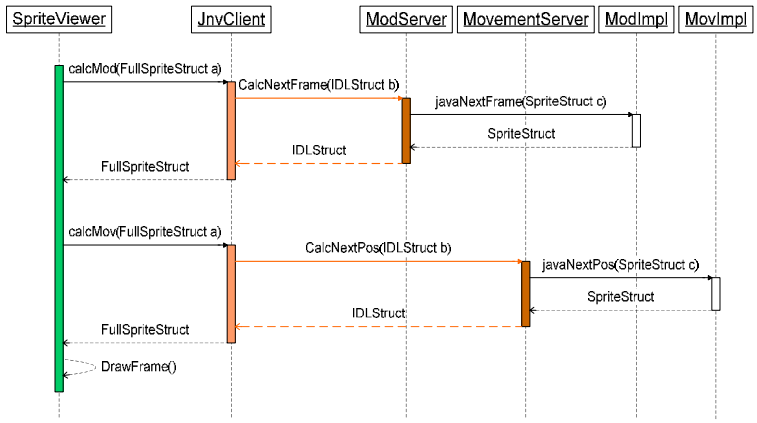
Verbindung mit .NET Remoting

```
TcpCh = new TcpChannel(0);
ChannelServices.RegisterChannel(TcpCh);
SkinListObject tmpRemoteObject =
  (SkinListObject)Activator.GetObject(
     typeof(SkinListObject),
     „tcp"+"://"+URL+":"+Port+"/"+"SkinListObject");
return tmpRemoteObject.GetSkinNames();
```


Vorführung



Ein Sequenzdiagramm



Janeva & Java Namensdienst

- testen von Janeva über „localhost“ hinweg wäre mit ~3000€ Lizenz (per CPU) verbunden
- Janeva konnte den JavaCorbaNamensdienst nicht finden
  - ior als File übergeben
  - Lösungen über Rechnergrenzen hinweg
    - ftp skripten mit Bulletproof und UltraFXP
    - SpriteNS.java (schreibt IOR in txt)
    - email, mailsslots, remoting…..

Was haben wir gelernt?

- XP funktioniert gut
- CVS wäre wichtig gewesen
- Aufwandsschätzungen oft x2
- Entwurfsmuster sind hilfreich
- Teamarbeit kann sehr effektiv sein
- Infrastruktur ist wichtig
- Interoperabilität (eingeschränkt) möglich

Ausblick

- Alternative Verteilungsplattformen
- Alternative Corba - Implementierungen

- *Janeva kaufen*
- Kritische Betrachtung von Corba in diesem Fall

- Sinnvolle Inhalte für die Sprites
- Debuggen

Quellen/Literaturhinweise

- [1] Borland Software Corporation: Janeva Developer‘s Guide. www.borland.com Stand: 16.06.2004
- [2] Sun Microsystems, Inc.: CORBA Programming with JSE 1.4. http://java.sun.com/developer/technicalArticles/releases/corba/ Stand: 16.06.2004
- [3] Sun Microsystems, Inc.: „Java IDL: Interoperable Naming Service (INS) Example „. http://java.sun.com/j2se/1.4.2/docs/guide/idl/INStutorial.html Stand: 16.06.2004
- [4] Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides „Design Pattern – Elements of Reuseable Object-Oriented Software“, 1997, Addison-Wesley Professional
- [5] Robert Orfali, Dan Harkey, „Client/server programming with Java and CORBA“,1997, John Wiley & Sons Inc
- [6] Andreas Vogel, Keith Duddy,„Java programming with CORBA – advanced techniques for building distributed Applications [..]“,1998, John Wiley & Sons Inc