



Progetto di moduli per lo sviluppo di applicativi client-server basati su Shadow Framework

Candidato: Luigi Pasotti
Relatore: Prof. Alessandro Martinelli

Università degli Studi di Pavia
Facoltà di Ingegneria
Dipartimento di Ingegneria Industriale e dell'Informazione

22 Aprile 2013

- 1 Introduzione
 - Obbiettivo della tesi
 - Web 3D e Tecnologie attuali
 - Shadow Framework
- 2 SF-Remote-Connection
 - Moduli
 - Dataset sostitutivi
- 3 Test
- 4 Risultati e sviluppi futuri

Obbiettivo della tesi

Contesto

- **Shadow Framework**: framework per la realizzazione di applicazioni che fanno uso di **grafica tridimensionale real-time** ideato e sviluppato dall'Ingegnere Alessandro Martinelli
- Caratteristiche:
 - **portabilità**: scritto in Java (desktop, Android), versione C++ (desktop, iOS) in sviluppo
 - **design web-oriented**: in sviluppo una versione javascript
 - **moderno**: massiccio utilizzo delle GPU a pipeline programmabile
 - **estendibile**: uso di design pattern, modulare

Obbiettivo

- Produzione di moduli utili allo sviluppo di applicazioni client-server orientate al web:
 - **estensione del layer dati**
 - **dati grafici trasmessi in rete**

Web 3D

Fruizione tramite il web di contenuti in grafica 3D real-time interattiva

Applicazioni:

- navigazione web
- intrattenimento: giochi, social network
- eCommerce: vetrine interattive

Problematica

- **dimensione dei dati** grafici non trascurabile (GigaByte)

Tecnologie Attuali

Modalità di fruizione usate da framework e applicazioni

- **Plugin per browser:** Flash, Java, Ad hoc
- **WebGL:** javascript
- **Live streaming:** applicazioni native, Java Web Start, ecc.

Modalità di trasmissione dati (e relativi problemi)

- **Approccio stand-alone:** lunghi tempi di caricamento
- **Flusso continuo:** occupazione di banda
- **Trasferimento progressivo:** limitazioni alla qualità dei dati

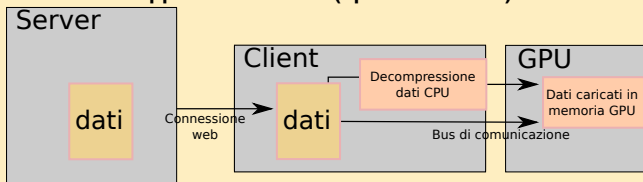
Shadow Framework

Focus sui 3 “momenti” di accesso ai dati grafici:

- **Accesso**
- **Costruzione e Inizializzazione**
- **Visualizzazione**

Pipeline di accesso ai dati

Approccio classico (tipo stand-alone)



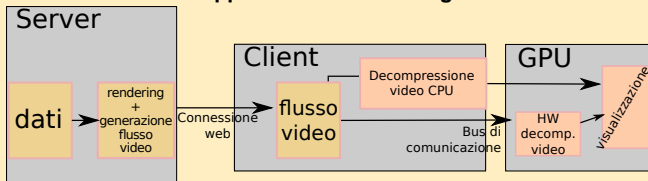
Shadow Framework

Focus sui 3 “momenti” di accesso ai dati grafici:

- **Accesso**
- **Costruzione e Inizializzazione**
- **Visualizzazione**

Pipeline di accesso ai dati

Approccio live streaming

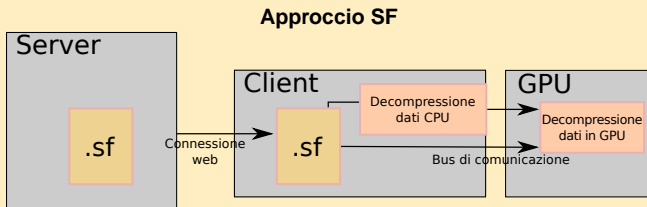


Shadow Framework

Focus sui 3 “momenti” di accesso ai dati grafici:

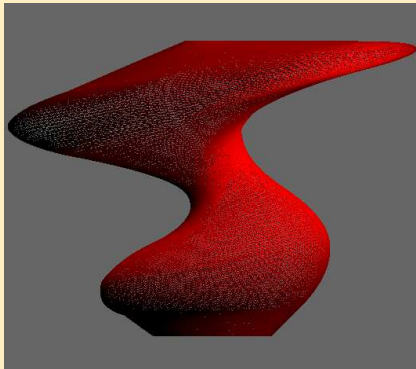
- **Accesso**
- **Costruzione e Inizializzazione**
- **Visualizzazione**

Pipeline di accesso ai dati

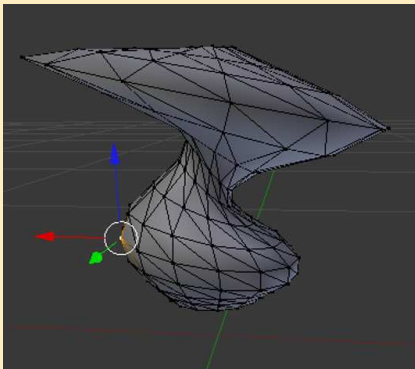


Confronto rappresentazione dati

Rendering della stessa geometria in **formato SF** (sinistra) ed in forma di **mesh poligonale** (destra)



- dimensione xml: **3 kB**
- dimensione file .sf: **263 Byte**
- dimensione file .sf (con sola geometria): **64 Byte**

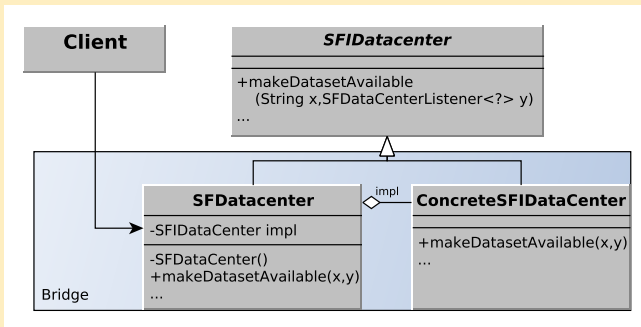


- dimensione Collada: **55,3 kB** (standard xml riconosciuto)
- dimensione obj: **34,3 kB**
- dimensione file .3ds: **8,6 kB**

Layer Dati Shadow Framework

Dati gestiti tramite astrazione

- **SFDataset**: unità base di ogni dato del framework
- **SFDataCenter**: gestore centralizzato dei dati
 - *Singleton*
 - *Bridge*



SF-Remote-Connection

Moduli del progetto

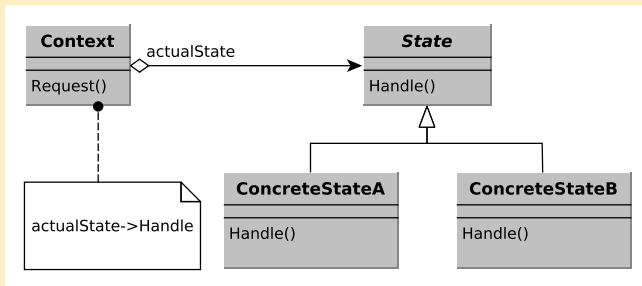
- **Base Communication**
 - Libreria per comunicazione via TCP
- **RemoteDataCenter Tool**
 - Implementazione di *SFIDataCenter* che usa la comunicazione di rete
- **Client**
 - Libreria per applicazioni client
- **Server**
 - Libreria per applicazioni server

Base Communication

Libreria per comunicazione via TCP

Caratteristiche salienti

- Gestione connessioni client e server
- Scambio messaggi testuali via TCP
- **Configurabilità** protocollo di comunicazione attraverso il **pattern State**

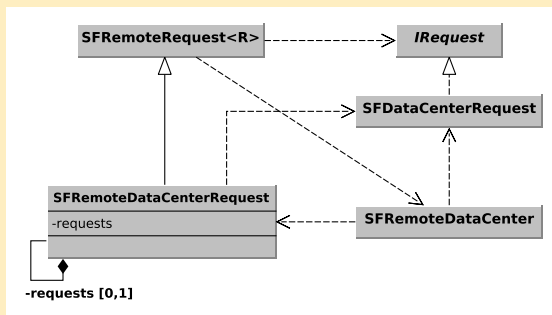


RemoteDataCenter Tool

Implementazione di *SFDataCenter* che usa la comunicazione di rete

Caratteristiche salienti

- Coda di richieste
- Meccanismo a **Dataset sostitutivi**

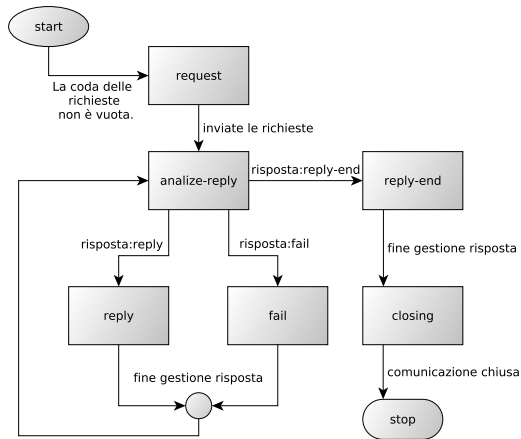


Client

Libreria per applicazioni client

Caratteristiche salienti

- Implementazione del reperimento dati
 - **Bufferizzazione** delle richieste
 - **Multithread**
- Macchina a stati per la gestione del protocollo lato client
 - **Espandibilità**
 - **Test-oriented**

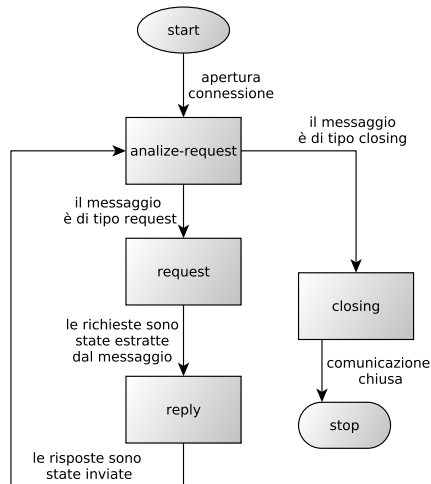


Server

Libreria per applicazioni server

Caratteristiche salienti

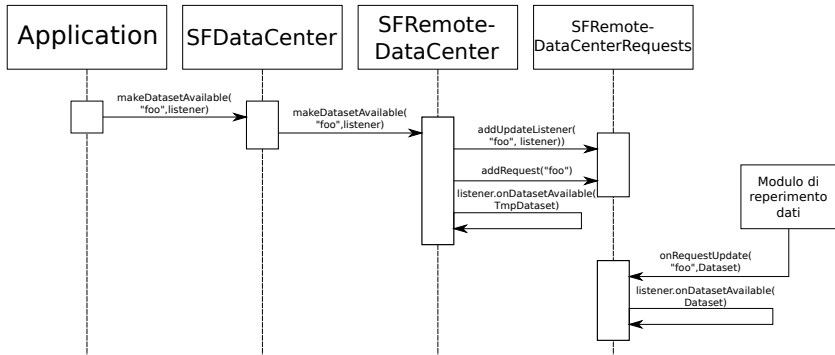
- **No stack grafico**, solo gestione dei dati
- **Multithread**
- Macchina a stati per la gestione del protocollo lato server
 - **Espandibilità**
 - **Test-oriented**



Dataset sostitutivi

Motivazioni

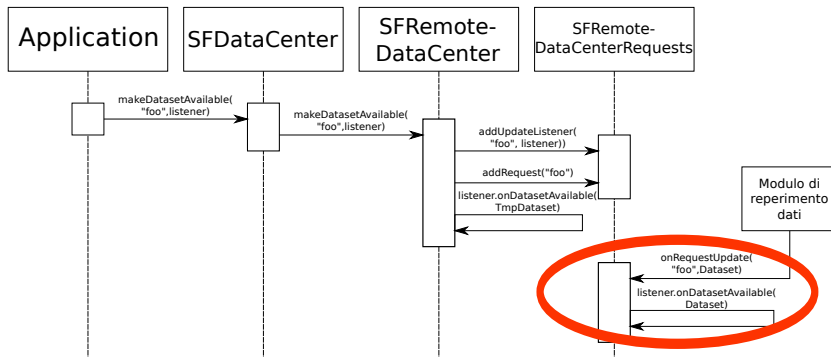
- Reperimento dati non bloccante
- Possibilità di definire dei “segna-posto” per gli elementi grafici



Dataset sostitutivi

Motivazioni

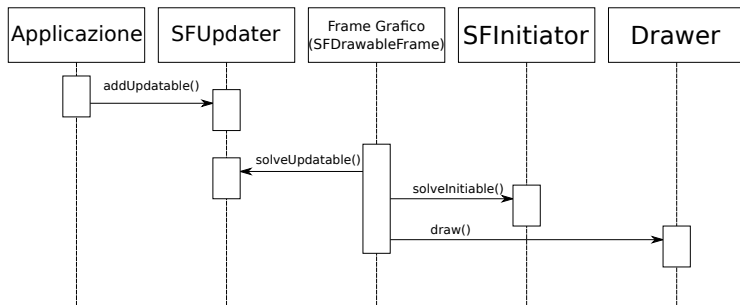
- Reperimento dati non bloccante
- Possibilità di definire dei “segna-posto” per gli elementi grafici



Dataset sostitutivi

Update dei dati

- Deve affrontare la **costruzione e inizializzazione** dei dati ricevuti
- Deve essere sincronizzato con il processo di rendering:
 - meccanismo interno al framework che usa un Updater e un Initiator.



Test

Test dal vivo

Buona visione

Conclusioni

Conclusioni

- I moduli:
 - rispecchiano le caratteristiche volute;
 - sono facilmente **estendibili**.
- Acquisita familiarità con pratiche e tecniche riconosciute:
 - uso di **Design Pattern**;
 - metodi di **sviluppo agile**;
 - strumenti di **controllo di versione concorrente**.
- Importante **feedback** sulle componenti interne dello Shadow Framework

Sviluppi futuri

- Strumenti per una comunicazione più complessa
- Tool per la definizione dei Dataset sostitutivi
- Invio di Dataset aggregati in librerie



Progetto di moduli per lo sviluppo di applicativi client-server basati su Shadow Framework

Candidato: Luigi Pasotti
Relatore: Prof. Alessandro Martinelli

Università degli Studi di Pavia
Facoltà di Ingegneria
Dipartimento di Ingegneria Industriale e dell'Informazione

22 Aprile 2013