

Sistemi Distribuiti

Corso di Laurea in Ingegneria

Prof. Paolo Nesi

Esercitazioni A.A. 2005-2006


Department of Systems and Informatics
University of Florence
Via S. Marta 3, 50139, Firenze, Italy
tel: +39-055-4796523, fax: +39-055-4796363

Lab: DISIT, Sistemi Distribuiti e Tecnologie Internet

nesi@dsi.unifi.it paolo.nesi@unifi.it
<http://www.dsi.unifi.it/~nesi>



Sistemi Distribuiti, Univ. Firenze, Paolo Nesi 2005-2006 1



Esercitazioni derivate da esami passati

- **Caso di studio 1:**
 - ♣ Middleware dinamico per componenti
- **Caso di studio 2:**
 - ♣ Middleware Cooperativo
- **Caso di studio 3:**
 - ♣ Monitoraggio e controllo di segnali audio/video
- **Caso di studio 4:**
 - ♣ Gestione di cartelle cliniche condivise in modalita' controllata





Sistemi Distribuiti, Univ. Firenze, Paolo Nesi 2005-2006 2

Sistemi Distribuiti

Caso di Studio 1: descrizione generale

- Si progetti un sistema distribuito che permetta la costruzione dinamica di applicativi software basati su componenti software.
- I componenti
 - ♣ sono delle istanze di classi, i cui metodi possono avere vari parametri di passaggio e di ritorno;
 - ♣ possono migrare da un applicativo all'altro tramite rete anche mantenendo il loro stato
 - ♣ possono essere acquisiti tramite un canale di rete gestito da un Gestore.
 - ♣ sono ricevuti in forma di codice oggetto eseguibile.
 - ♣ possono essere acquistati e ricercati da ogni Gestore in una directory locale. Il Gestore deve poter effettuare il riconoscimento dei componenti in base alla loro interfaccia descritta in modo formale;
 - ♣ possono comunicare con altri componenti sulla rete tramite il Gestore che permette di effettuare la gestione degli aspetti di middleware (descrivere il ruolo del Gestore);



Sistemi Distribuiti, Univ. Firenze, Paolo Nesi 2005-2006 3

Sistemi Distribuiti

Caso di Studio 1: requisiti

- L'esecuzione di componenti per applicativi remoti e in applicativi locali deve essere completamente dinamica.
 - ♣ Un applicativo può essere costruito anche senza conoscere la loro interfaccia, al momento della compilazione dell'applicativo. Questa, può essere richiesta al momento in cui l'applicazione cerca/acquisisce un certo componente sulla rete.
 - ♣ l'identificazione del componente si utilizza un metodo per la verifica di consistenza fra interfaccia richiesta e quelle accessibili. Per esempio un'Applicazione per l'uso di un database scopre di dover utilizzare un certo componente (driver) e per questo dal sistema operativo gli viene comunicato che deve trovare un componente (driver) che ha da una parte un'interfaccia A e da un'altra un'interfaccia B.
 - Questa applicazione cercherà di ottenere, tramite il Gestore, un componente FR che implementa le interfacce A e B.
- Le applicazioni vengono a conoscenza di che tipo di componenti hanno bisogno poiché possono ottenere la descrizione delle loro interfacce anche solo al momento dell'esecuzione.
 - ♣ Queste possono essere passate/richieste da altri componenti per il loro corretto funzionamento.
 - ♣ Per esempio, il componente FR di prima potrebbe richiedere un ulteriore componente che implementa l'interfaccia C per il suo funzionamento;
- I componenti necessari devono essere ricercati fra quelli accessibili in altri dispositivi/applicazioni (Gestori) localizzabili nella rete. ..





Sistemi Distribuiti, Univ. Firenze, Paolo Nesi 2005-2006 4

Sistemi Distribuiti

C1: Modello di Traccia di Svolgimento

- Identificare e giustificare l'architettura del sistema distribuito:
 - ✦ Client Server, two-tier, n-tier, GRID, P2P, Web Services, CSCW, etc.
 - ✦ Presentare l'architettura proposta con un grafico identificando: applicativi, database, client, server, canali di comunicazione, etc., nelle loro istanze se necessario.
- Identificare e descrivere i sottosistemi
 - ✦ in termini di moduli o packages, presentare le loro relazioni
- Identificare la tecnologia piu' adatta e motivare la scelta:
 - ✦ linguaggio, eventuali tool e librerie, etc.
- Definire i meccanismi di comunicazione fra i sottosistemi e motivare le scelte.
- Identificare le classi principali dell'intero sistema comprensivo di tutti moduli identificati. Definire le loro relazioni.
 - ✦ Riportare un digramma UML con specializzazione, aggregazione, associazione, etc...
- Presentare attributi e metodi delle classi principali del sistema.
- Presentare le strutture delle basi di dati incluse nel progetto.





Sistemi Distribuiti, Univ. Firenze, Paolo Nesi 2005-2006 5

Sistemi Distribuiti

C1: Richieste specifiche

- Descrivere in modo formale il modello di componente
 - ✦ la sua descrizione,
 - ✦ definire una macchina a stati sulla sua evoluzione.
 - ✦ Proporre un modello formale per la descrizione delle interfacce e per la verifica della consistenza fra quelle richieste e quelle accessibili.
- Descrivere il gestore
- Proporre funzionalità aggiuntive che possono interessare le aziende di tali sistemi e le motivazioni. ..
- ..



Sistemi Distribuiti, Univ. Firenze, Paolo Nesi 2005-2006 6



Caso di Studio 1: svolgimento



Sistemi Distribuiti, Univ. Firenze, Paolo Nesi 2005-2006

7

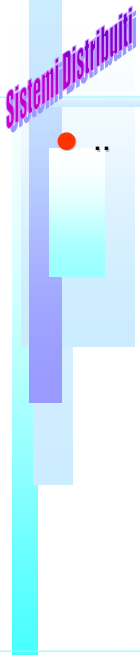


Caso di Studio 1




Sistemi Distribuiti, Univ. Firenze, Paolo Nesi 2005-2006


8



Caso di Studio 1




Sistemi Distribuiti, Univ. Firenze, Paolo Nesi 2005-2006 9



Caso di Studio 2: descrizione generale

- Si consideri un sistema distribuito per la produzione e gestione di informazioni in ambiente cooperativo. Il sistema si basa su di uno strato software (detto MCW, Middleware for Cooperative Work) in grado di supportare un'ampia gamma di applicazioni cooperative diverse.
- Ogni applicazione si puo' basare su di un modello/tipo-di dati diverso (disegni, testo, audio, multimedia, etc.).
- Le applicazioni sono replicate sui vari peer del sistema ma il modello generale e' a dati distribuiti, non concentrati su di un server centrale.
- In questo modo ogni peer del sistema distribuito cooperativo ha una copia delle informazioni che vengono prodotte e processate in modo cooperativo e la mantiene allineata con le quelle degli altri peer.



Sistemi Distribuiti, Univ. Firenze, Paolo Nesi 2005-2006 10

Sistemi Distribuiti

Caso di Studio 2: I requisiti

- **Altri requisiti:**
- Il MCW deve dare un supporto semplice per mantenere allineate le informazioni dei modelli dati delle varie applicazioni in modo indipendente dal modello di dati delle singole applicazioni.
- Il MCW deve inoltre offrire supporto per la rilevazione delle singole Applicazioni e dei computer che le eseguono. I peer/client devono potersi connettere/disconnettere in modo semplice alla rete ed ai vari gruppi di lavoro delle varie applicazioni.
- Il MCW deve poter gestire applicazioni cooperative diverse allo stesso tempo, ognuna delle quali può essere installata una o più volte nei vari peer.
- Il MCW deve poter garantire la consistenza, la completezza e la causalità delle operazioni sui vari modelli dati dei vari peer senza però conoscerne la struttura.
- Le singole applicazioni (i singoli utenti) devono poter iniziare un nuovo gruppo di lavoro cooperativo su un contenuto dati nuovo o vecchio oppure possono connettersi ad un gruppo presente, già iniziato da altri.
- Le singole operazioni che possono essere effettuate sui modelli dati delle applicazioni devono poter essere tracciate in modo da ricostruire l'informazione sui sistemi remoti in modo semplice, ad in modo da sapere chi ha contribuito a cosa nel tempo e da dove. ...



Sistemi Distribuiti, Univ. Firenze, Paolo Nesi 2005-2006 11

Sistemi Distribuiti

C2: Modello di Traccia di Svolgimento

- Presentare l'architettura proposta con un grafico identificando: applicativi, database, client, server, canali di comunicazione, etc., nelle loro istanze se necessario.
- Identificare e descrivere i sottosistemi
 - ✦ in termini di moduli o packages, presentare le loro relazioni
- Identificare la tecnologia più adatta e motivare la scelta:
 - ✦ linguaggio, eventuali tool e librerie, etc.
- Definire i meccanismi di comunicazione fra i sottosistemi e motivare le scelte.
- Identificare le classi principali dell'intero sistema comprensivo di tutti i moduli identificati. Definire le loro relazioni.
 - ✦ Riportare un digramma UML con specializzazione, aggregazione, associazione, etc...
- Presentare attributi e metodi delle classi principali del sistema.
- Presentare le strutture delle basi di dati incluse nel progetto.




Sistemi Distribuiti, Univ. Firenze, Paolo Nesi 2005-2006 12

Sistemi Distribuiti

C2: Richieste specifiche

- Discutere le implicazioni dei requisiti in relazione alla scelta dell'architettura per il CSCW.
- Descrivere le caratteristiche, i requisiti, che le applicazioni devono soddisfare per poter essere connesse al MCW e per potersi comportare in modo cooperativo secondo quanto descritto in precedenza. Identificare le interfacce verso il MCW di applicazioni del tipo:
 - ♣ Un editor per produrre disegni
 - ♣ Un editor per modificare testo
 - ♣ Un player/registratore audio e video
 - ♣ Un'applicazione per la lettura/scrittura di segnali digitali da sensori/attuatori..
- Descrivere l'interfaccia utente per ogni operatore
- Proporre funzionalità aggiuntive che possono interessare le aziende di tali sistemi e le motivazioni. ..




Sistemi Distribuiti, Univ. Firenze, Paolo Nesi 2005-2006 13

Sistemi Distribuiti

Caso di Studio 2: svolgimento


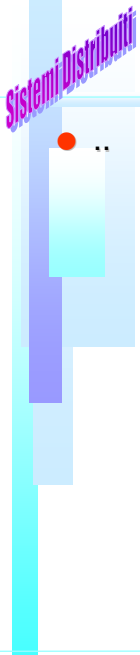
- ..



Sistemi Distribuiti, Univ. Firenze, Paolo Nesi 2005-2006 14

Sistemi Distribuiti

Caso di Studio 2


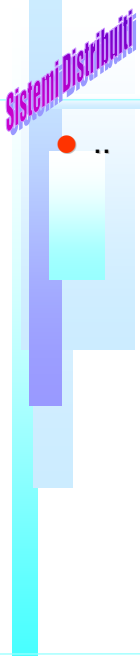


Sistemi Distribuiti, Univ. Firenze, Paolo Nesi 2005-2006

15

Sistemi Distribuiti

Caso di Studio 2



Sistemi Distribuiti, Univ. Firenze, Paolo Nesi 2005-2006

16

Caso di Studio 3: descrizione generale

- Si progetti un sistema distribuito per il monitoraggio (lettura e controllo) di segnali audio e video per l'identificazione di contenuti digitali rispetto ad un database di riferimento.
- Si ipotizzi la possibilità di stimare ogni 3 secondi un vettore di informazioni che possa essere utilizzato per descrivere il contenuto audio e video in via preventiva e in un secondo momento lo stesso vettore possa essere utilizzato per riconoscerlo.
- Si ipotizzi che questa operazione possa essere effettuata sia su segnali audio che video.
- Ogni operazione di ricerca puo' costare circa 1 secondo in un database di 20000 vettori. ..



Caso di Studio 3: requisiti



- il sistema di monitoraggio deve essere scalabile in termini di numero di canali audio e video monitorati.
- L'intero sistema di monitoraggio deve garantire il servizio anche nel caso in cui si debba operare delle operazioni di manutenzione su alcuni componenti. Si possono accettare delle situazioni di guasto con interventi entro 10 minuti.
- Ogni segnale video deve essere monitorato per audio e video, per l'audio viene ovviamente monitorato solo l'audio.
- I segnali video e audio devono essere memorizzati 24 ore su 24.
 - ✦ Per ogni segnale video si puo' immaginare di salvare un frame/immagine (720x480 32 bit per pixel) ogni 2 secondi.
 - ✦ I segnali audio possono essere salvati in formato MPEG3 192 kbit per s.
- Il sistema di monitoraggio deve svolgere in tempo reale la stima del vettore di descrizione e la ricerca di questo nel database in modo da produrre un report verso un server di controllo.
- Il server di controllo deve produrre dei segnali di allarme di vario tipo qualora il sistema riconosca segnali audio o video e deve tenere traccia di tutti gli eventi e dei punti in cui sono avvenuti con i relativi valori.
- Il sistema di memorizzazione deve permettere l'accesso alle registrazioni audio e video degli ultimi 12 mesi in modo rapido ed efficiente.
- Il sistema deve funzionare in continuo pertanto il non si prevedono operazioni di manutenzione straordinaria alla fine dei dodici mesi.



Sistemi Distribuiti

C3: Modello di Traccia di Svolgimento

- Definire l'architettura del sistema distribuito: Client Server, two-tier, n-tier, GRID, P2P, Web Services, CSCW, etc.
- Presentare l'architettura proposta con un grafico identificando: applicativi, database, client, server, canali di comunicazione, etc., nelle loro istanze se necessario.
- Identificare e descrivere i sottosistemi
 - ✦ in termini di moduli/tool o packages, presentare le loro relazioni
- Identificare la tecnologia piu' adatta e motivare la scelta:
 - ✦ linguaggio, eventuali tool e librerie, etc.
- Definire i meccanismi di comunicazione fra i sottosistemi e motivare le scelte.
- Identificare le classi principali dell'intero sistema comprensivo di tutti moduli identificati. Definire le loro relazioni.
 - ✦ Riportare un digramma UML con specializzazione, aggregazione, associazione, etc...
- Presentare digrammi ad oggetti, di interazione e di sequenza fra i principali oggetti attivi inerenti alla negoziazione.
- Presentare attributi e metodi delle classi principali del sistema.
- Presentare le strutture delle basi di dati incluse nel progetto.





Sistemi Distribuiti, Univ. Firenze, Paolo Nesi 2005-2006 19

Sistemi Distribuiti

C3: Richieste specifiche

- Si stimi le dimensioni/capacita dei server di memorizzazione e di riconoscimento. ..
- Descrivere l'interfaccia utente per ogni operatore



Sistemi Distribuiti, Univ. Firenze, Paolo Nesi 2005-2006 20

Sistemi Distribuiti

Caso di Studio 3: svolgimento


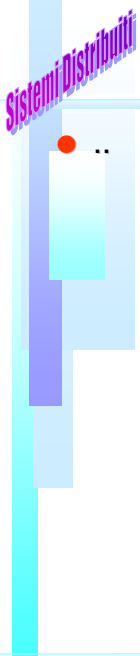


Sistemi Distribuiti, Univ. Firenze, Paolo Nesi 2005-2006

21

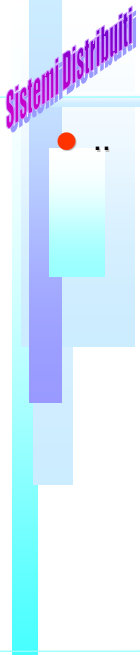
Sistemi Distribuiti

Caso di Studio 3




Sistemi Distribuiti, Univ. Firenze, Paolo Nesi 2005-2006

22

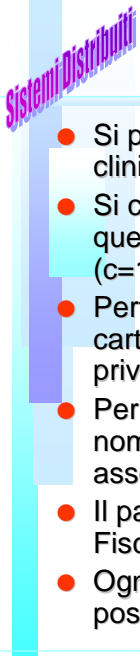


Caso di Studio 3




Sistemi Distribuiti, Univ. Firenze, Paolo Nesi 2005-2006

23



Caso di Studio 4: descrizione generale

- Si progetti un sistema distribuito per la gestione di cartelle cliniche condivise fra ospedali.
- Si consideri di avere M ospedali ($m=1..M$); ognuno di questi ha P pazienti ($p=1..P$) e ognuno di questi ha C ($c=1..C$) cartelle di intervento.
- Pertanto $m.p.c$ e' la cartella c del paziente p in m . Su ogni cartella il Paziente puo' avere inserito delle restrizioni sulla privacy imponendo dei diritti.
- Per esempio la non divulgazione di dati personali quali eta', nome, etc. O la non divulgazione della sua malattia associata al suo nome, etc.
- Il paziente ha un identificativo univoco come il codice Fiscale o il suo numero di registrazione nazionale.
- Ogni ospedale ha anche un numero di Dottori D che possono effettuare dei consulti on demand.





Sistemi Distribuiti, Univ. Firenze, Paolo Nesi 2005-2006

24

Sistemi Distribuiti

Caso di Studio 4: requisiti

- il sistema deve permettere ad ogni singolo ospedale di
 - ♣ effettuare la visione di Cartelle dei pazienti con altri ospedali sulla base dei diritti, direttamente operando delle ricerche sulla base di informazioni contenute nelle cartelle o del nome/identificativo del paziente.
 - ♣ richiedere un Consulto cooperativo su un certo caso ad un certo numero di altri Dottori.
- La negoziazione avviene fra due ospedali in accordo ai diritti e regole di privacy firmate dal paziente sulla relativa cartella.
- La realizzazione del Consulto cooperativo deve avvenire in modo distribuito e concorrente. Ogni ospedale azienda deve poter aprire un numero elevato di consulti allo stesso tempo, con uno o piu' ospedali relativamente ad uno o piu' cartelle di uno o piu' pazienti.
- In ogni ospedale vi possono essere piu' individui e terminali/applicazioni che aprono Consulto, le seguono nelle loro fasi alternandosi al loro controllo ed evoluzione, fino alla conclusione, etc.
- Il Consulto deve
 - ♣ avvenire tramite un canale di comunicazione protetto.
 - ♣ tenere conto del profilo del Dottore e della sua identificazione e autenticazione.
 - ♣ si sviluppa secondo una macchina a stati. Per esempio: APERTO, INCORSO,, ABORTITA, CONCLUSA, etc. (il candidato, puo' definire una macchina a stati diversa, che deve essere giustificata e descritta). La macchina a stati deve poter essere definita dall'ospedale, come le motivazioni per la transizione di stato. Per ogni stato esiste uno stato aggiuntivo che definisce se il Consulto si trova in attesa di un'azione da parte dell'ospedale che la richiede.
- Quando lo stato di un Consulto cambia il richiedente deve essere informato in modo da poter procedere quanto prima allo sfruttamento del Consulto, riducendo i tempi di attesa.
- Ogni ospedale deve poter tenere traccia dei Consulto in corso con le altri ospedali, riaprirle, conoscere lo stato, eventualmente procedere secondo le indicazioni della controparte e ritornare in attesa, verificare quali delle transazioni si sono concluse con successo o meno...



Sistemi Distribuiti, Univ. Firenze, Paolo Nesi 2005-2006 25

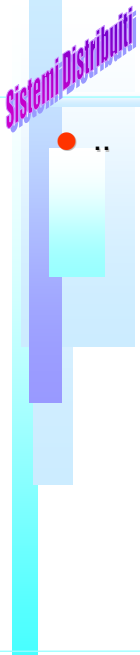
Sistemi Distribuiti

Caso di Studio 4: svolgimento


- ..



Sistemi Distribuiti, Univ. Firenze, Paolo Nesi 2005-2006 26

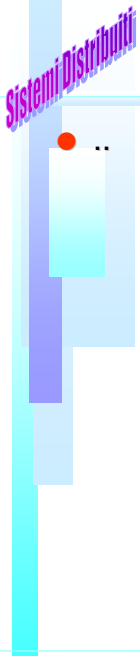


Caso di Studio 4




Sistemi Distribuiti, Univ. Firenze, Paolo Nesi 2005-2006

27



Caso di Studio 4



Sistemi Distribuiti, Univ. Firenze, Paolo Nesi 2005-2006

28



The slide features a decorative header in the top-left corner with the text "Sistemi Distribuiti" in a purple, slanted font. Below this, a vertical cyan bar is partially visible. A horizontal cyan line spans the width of the slide. In the bottom-left corner, there is a circular logo of the University of Florence and a small diagram consisting of a square with an 'S' and a triangle. The footer contains the text "Sistemi Distribuiti, Univ. Firenze, Paolo Nesi 2005-2006" and the page number "29".