



ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

Il progetto del corso di Tecnologie Web - A.A. 2017/18

Fabio Vitali

Mehwish Alam

Angelo Di Iorio

Corsi di laurea in Informatica e
Informatica per il Management

Alma Mater – Università di Bologna

Il progetto di fine corso

- Un sistema VERO, che funziona e fa cose utili
- Realizzabile sia in laboratorio che a casa.
- Enfasi in parte sulla programmazione (approccio procedurale) ma soprattutto sui documenti attivi (approccio dichiarativo)
- Enfasi sul mashup di tecnologie esistenti e sofisticate



Organizzazione dei team

- Ogni persona decide in anticipo se è interessata a sostenere l'esame in estate, autunno, sessione straordinaria o essere ancora indeciso.
- Tutti gli studenti si dividono in team di 3-4 persone. Meno di 3 significa troppo lavoro individuale. Più di 4 significa troppo poco. AL MASSIMO 4. NIENTE ECCEZIONI.
- Ogni team porta il progetto insieme (anche qui, non ci sono eccezioni!). Il team dichiara in anticipo la natura del contributo di ciascun membro oppure accetta che chiunque sia interrogato (e nel dettaglio) su tutto il progetto.
- Apposite strutture dati (sul wiki del corso) vengono tenute aggiornate con il numero, nome e componenti di ogni team, e periodo previsto di esame. Il sottoscritto NON è coinvolto nell'organizzazione dei team.



Ruolo di queste specifiche

- Questo documento contiene le specifiche fondamentali del progetto di fine corso.
- Quanto scritto qui dentro, salvo esplicite eccezioni, deve essere considerato requisito **OBBLIGATORIO** per la consegna.
 - Le frasi scritte in questo colore corrispondono a suggerimenti o scelte opzionali, e sono da considerarsi **non obbligatorie**.
 - Le frasi in questo colore corrispondono a vincoli **obbligatorie** introdotti per le esigenze del progetto universitario, che non sarebbero necessari o opportuni in un prodotto vero per il mercato esterno.
- Se una o più delle specifiche qui introdotte non funzionano, il progetto **NON** è considerato accettabile.
- ***Nel corso dell'anno potrebbero emergere nuovi requisiti o rilassamenti di precedenti requisiti. Un'apposita pagina sul wiki fornirà in maniera aggiornata l'elenco delle modifiche ai requisiti.***





AlphaTube - Un sistema di raccomandazioni per YouTube





ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

Background e Visione

Background: Information overload



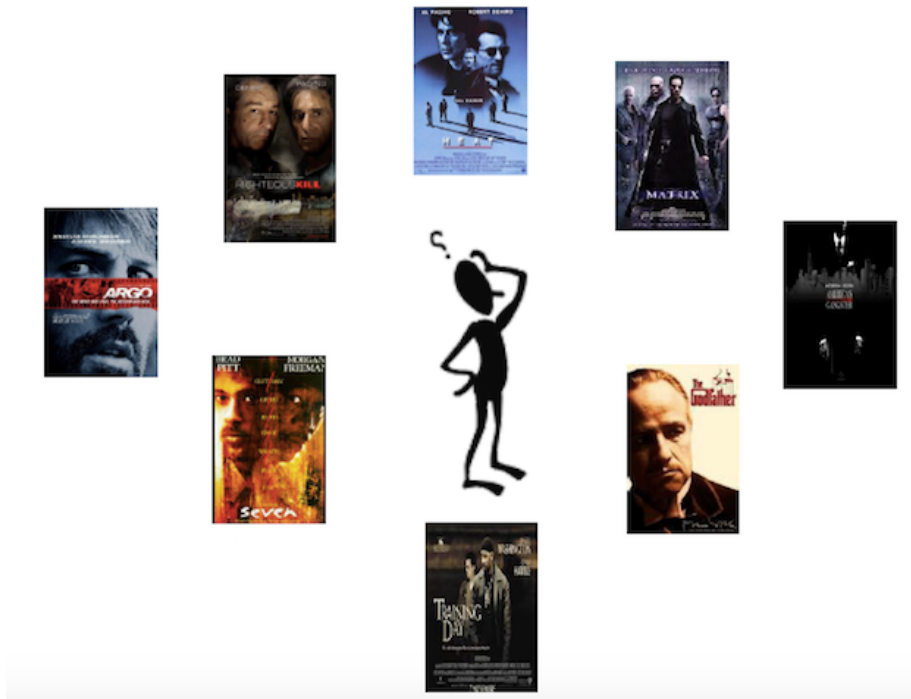
Accesso personalizzato alle informazioni

- Aiutare l'utente a trovare le informazioni a cui è (o potrebbe essere) interessato
- Basarsi sulle preferenze / comportamenti passati
- Escludere (filtrare) informazioni irrilevanti



Recommender System

- Aiuta gli utenti alle prese con sovraccarico di informazioni o di scelte.
- Aiuta gli utenti a trovare gli oggetti di proprio interesse.



Your recent searches

Berlin City Center (Jul 28 - Aug 4)
 1 room, 1 adult

Berlin City Center (Jul 28 - Aug 4)
 1 room, 2 adults

Your Search

Berlin City Center
 7 nights (Jul 28 - Aug 4)
 Business traveler
 1 adult

Change search

What are you looking for?

▼ Popular for Business Travelers

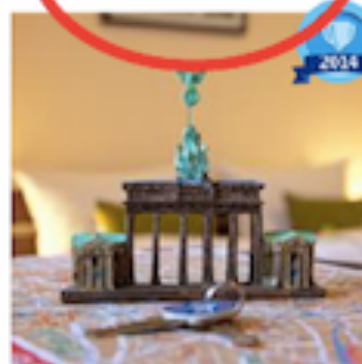
- ☐ Breakfast included (61)
- ☐ Wi-Fi (116)
- ☐ Parking (113)
- ☐ Airport Shuttle (21)
- ☐ Fitness Center (39)

▼ Price (per night)

- ☐ € 0 - € 50 (9)

Berlin City Center: 119 out of 326 properties available

Sort by: **Recommended** Price Stars Review Score



LebensQuelle am Checkpoint Charlie

Good 7.2

Score from 1835 reviews

Value Deal
[Berlin City Center, Berlin](#) – Subway Access
 There are 10 people looking at this hotel.
 Last booked: 5 minutes ago

Twin Room **Breakfast included**

Last chance! This room's popular – only 1 left on our site!

Price for 7 nights

€ 563

Reserve



Hollywood Media Hotel am Kurfürstendamm

Excellent 8.6

Score from 1497 reviews


Value Deal
 120€

[Berlin City Center, Berlin](#) – Subway Access
 There are 8 people looking at this hotel.




ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

Suggested Post



Banggood
 Sponsored ·

ONLY \$9.99. #Waterproof Sport Chest Pouch Bag, Can't miss it ! Shop @Banggood, Hurry!




B: Color Blue

Men Women Waterproof Sport ...



C: Color Red

Men Women Waterproof Sport ...



A: Color Black

Men Women Waterproof Sport ...

26 Likes 1 Comment 1 Share



SUGGESTED GROUPS

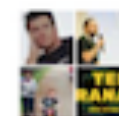
See All



**Singolarità Tecnologica
Technium - TES**

494 members

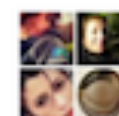
+ Join



**Club degli sviluppatori -
Puglia**

Vincenzo Patruno and 4 other
friends joined

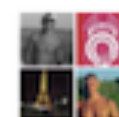
+ Join



**Locali ed Eventi Vegan in
Puglia e Basilicata**

Azzurra Ragone and Arianna
Montemurro joined

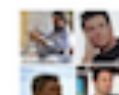
+ Join



Vendo di tutto

5,044 members

+ Join



Open Data Sicilia

Francesco Piersoft Paolicelli and 4
other friends joined



ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

NETFLIX

House of Cards

★★★★★ 2013 TV-MA 1 Season  

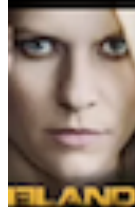
Sharks gliding ominously beneath the surface of the water? They're a lot less menacing than this Congressman.



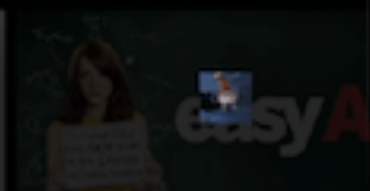
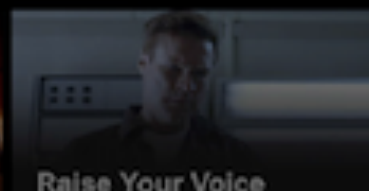
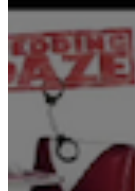
This winner of three Emmys, including Outstanding Directing for David Fincher, stars Kevin Spacey and Robin Wright.



Popular on Netflix



Top Picks for Jean-François



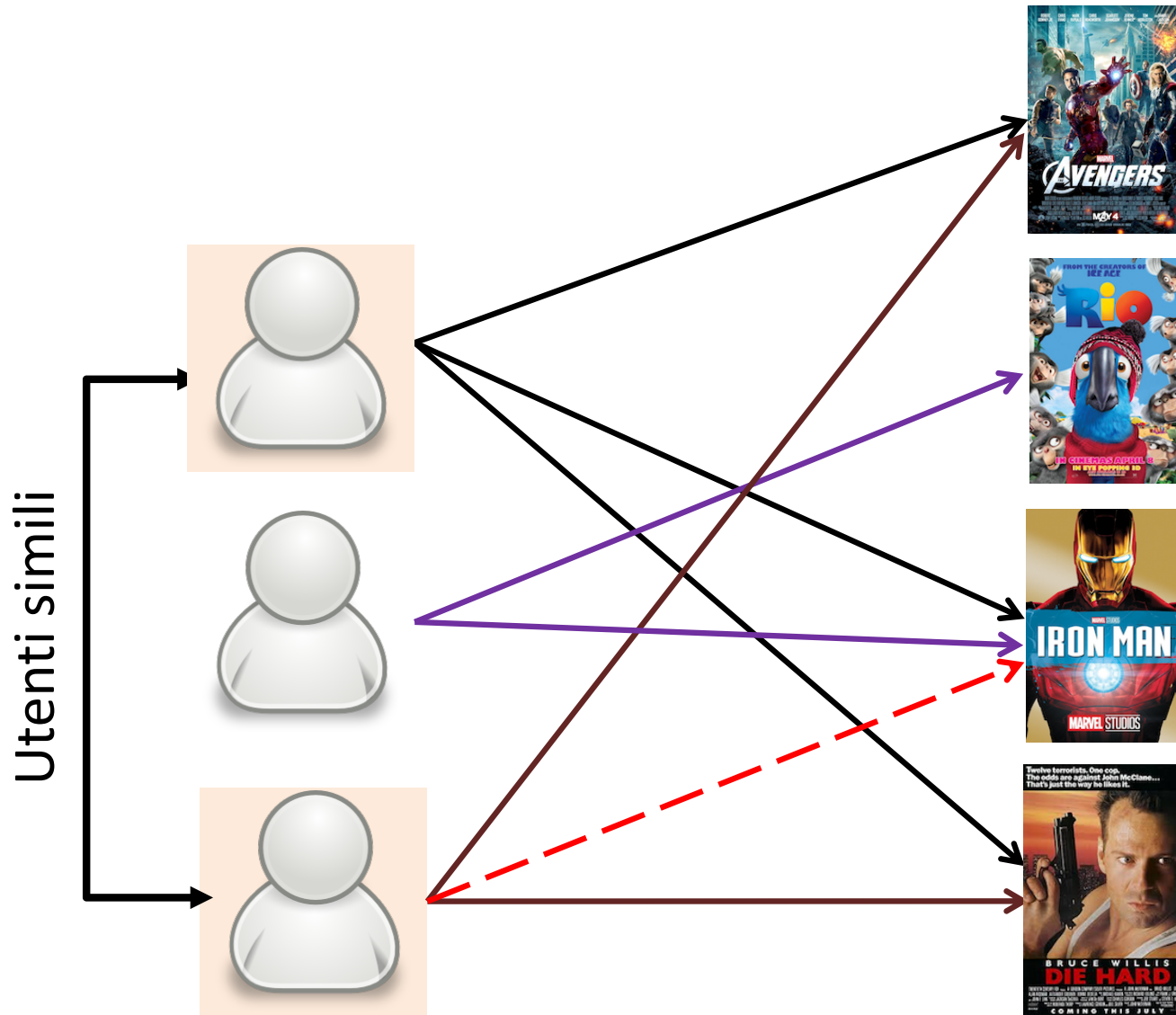
ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

Tecniche di raccomandazione

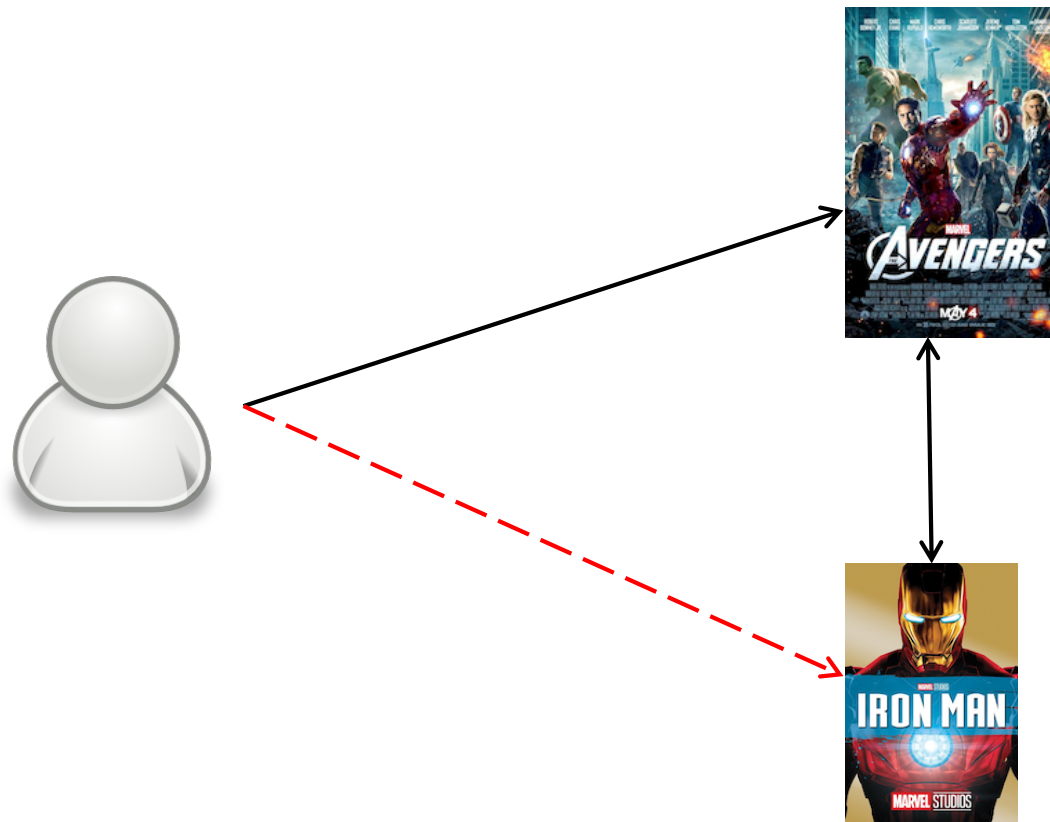
- Content-based
- Filtraggio collaborativo
- Approccio demografico
- Approccio knowledge-based
- Approccio community-based
- Recommender system ibridi



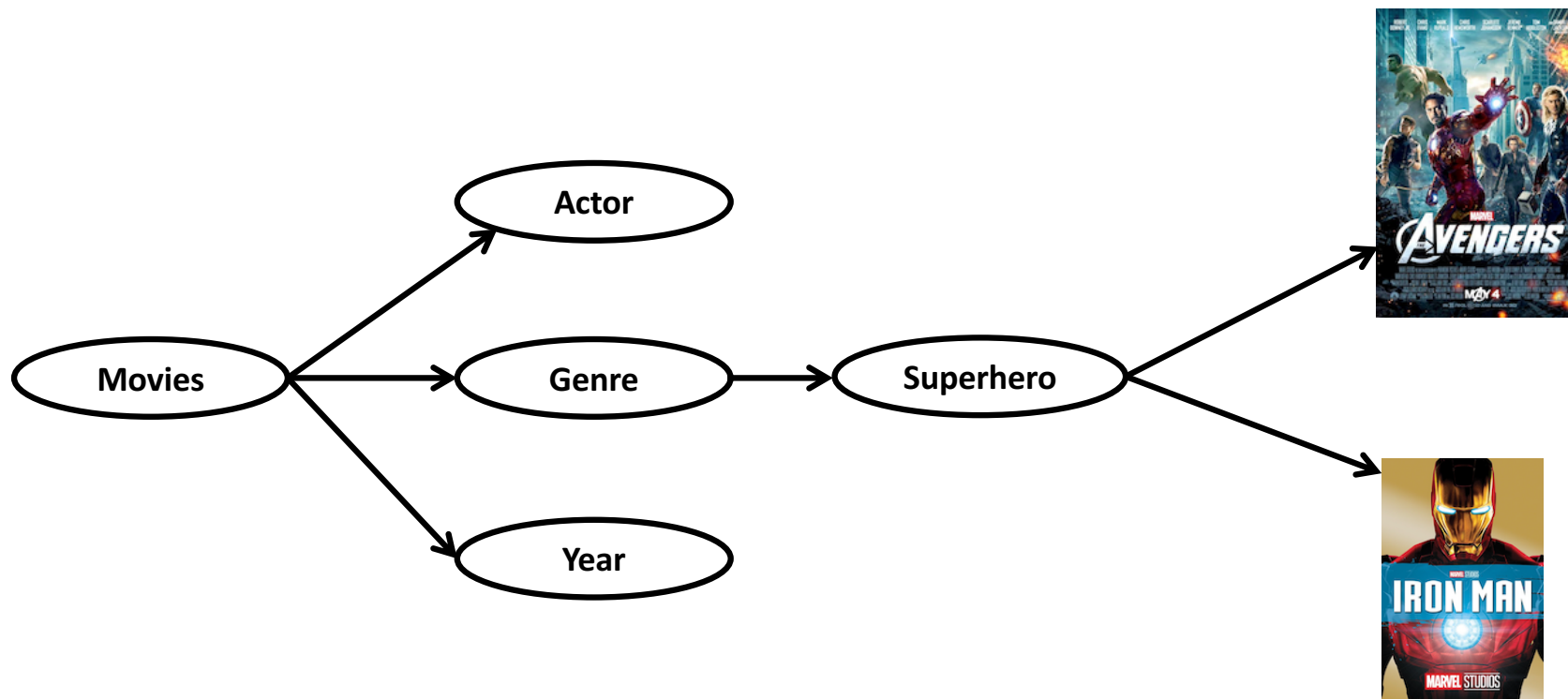
Raccomandazione collaborativa



Raccomandazione content-based



Raccomandazione Knowledge-Based



Altri modi per trovare similarità

- Similarità di grafo
 - Misure per calcolare la similarità di due film basandosi sulle informazioni rappresentate come nodi e archi.
- Similarità testuale
 - Misure per calcolare quanto siano simili due film basandosi sulla descrizione scritta.





ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

Architettura del progetto



Alpha Tube

Alpha Tube è un progetto per la realizzazione di un ambiente complesso in cui accedere mescolando dati di servizi diversi:

Alla base c'è il Visualizer, che permette di visualizzare in maniera interessante video (ad es. musicali) e dati recuperati da Wikipedia su artista, canzone, album visualizzato.

In aggiunta c'è il Recommender, che mostra alcuni video "collegati" a quello in visualizzazione e permette di creare un playlist dei prossimi video. Il Recommender suggerisce i video sulla base di un algoritmo di complessità crescente, basato inizialmente solo su omogeneità di artista, album, titolo.

Il Recommender avanzato utilizza modelli complessi di crowdsourcing, semantic web e social network per creare un sistema di suggerimenti più soddisfacente e personalizzato.

Lo scopo è studiare API di servizi disponibili on line e creare un'applicazione web che acceda ai dati in maniera dinamica, con una rappresentazione visiva in una pagina web.





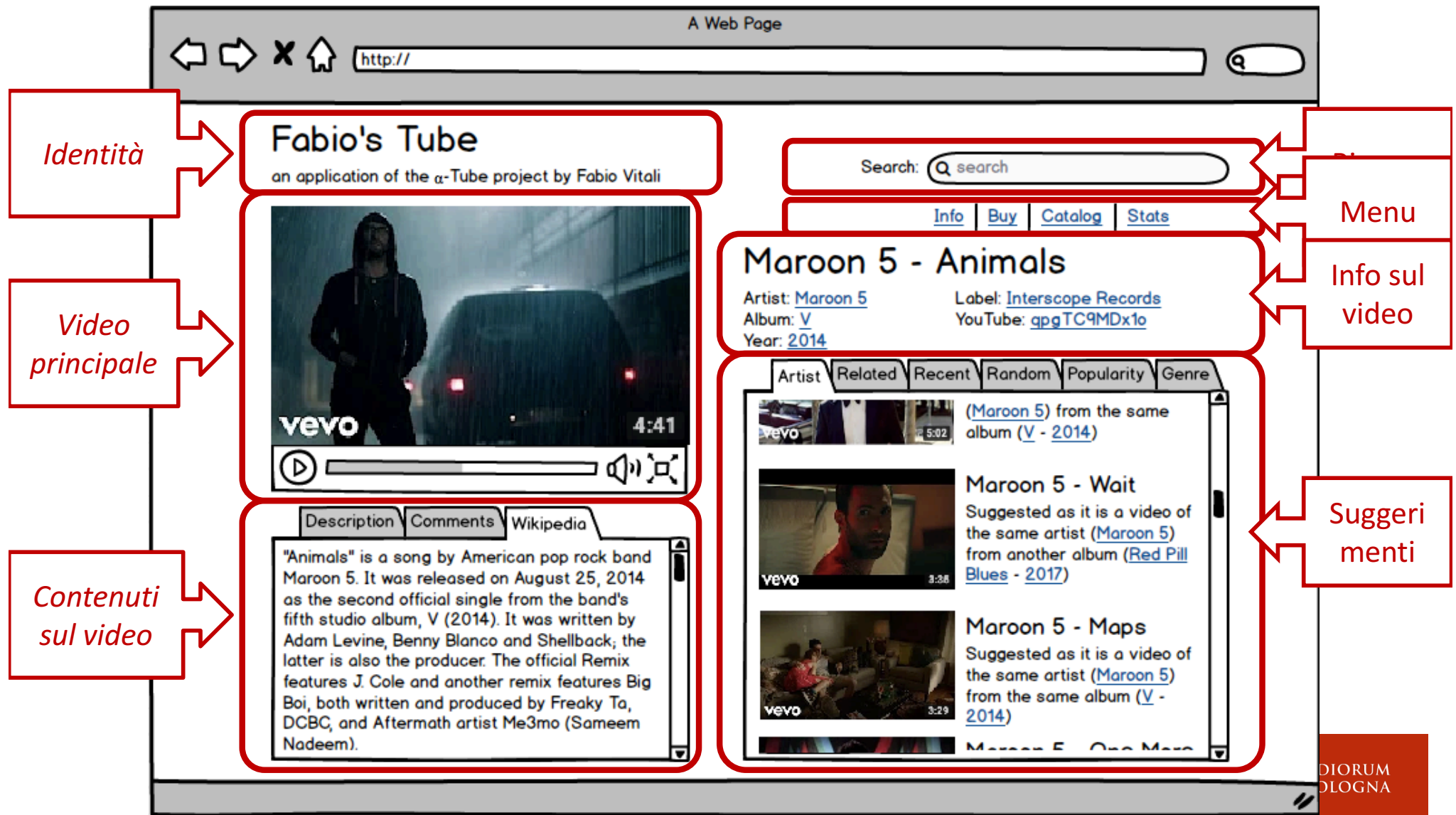
Alpha Tube: il visualizer (1)

- Un'applicazione web per video musicali di YouTube, commenti e informazioni sul video, e suggerimenti di altri video.
- Funzionante su tutti i browser, responsive, mobile-first, probabilmente una single-page application.
- Alcune aree fondamentali, più strumenti secondo necessità ed appropriatezza (molta flessibilità su questo):
 - Visualizzazione del video corrente
 - Visualizzazione di informazioni tecniche su questo video
 - Visualizzazione di contenuti su questo video
 - Descrizione di YouTube
 - Commenti di YouTube
 - Descrizione su Wikipedia (DBPedia)
 - Visualizzazione di altri video suggeriti
 - Secondo algoritmi diversi di suggerimento





Il visualizer (2)





Il visualizer (3)

- Il layout mostrato è ***solo un esempio***.
 - Sentitevi MOLTO PIÙ LIBERI di inventare.
- Video player con controlli soliti.
- Search
 - E' possibile cercare un video per titolo, nome della canzone, nome dell'artista, codice YouTube
- Barra del menu
 - Tutti i servizi che desiderate (quelli mostrati sono solo un esempio) + lista di partenza dei video + info sul team.
- Info sul video (tutti cliccabili)
 - Nome canzone, nome artista, nome album, anno, codice YouTube, membri della band, genere musicale, durata e qualità del video, ecc.
 - Queste informazioni non sono necessariamente presenti o marcati come tali nei dati di YouTube, vanno cercati su DBPedia sulla base di quanto disponibile nei dati YouTube.
 - Non bisogna essere perfetti, ma bisogna essere ***abbastanza precisi***





Il visualizer (4)

- Area contenuti
 - Descrizione del video, commenti di YouTube, contenuti di DBPedia sul brano, **sull'album, sull'artista, sul genere, tweet che menzionano il brano o l'artista, ecc..**
 - Possono essere visualizzati alternativamente, oppure tutti insieme, separati oppure mescolati, purché sia esplicita ed evidente l'origine e la natura di ciascun contenuto.
 - Visualizzare solo parte del contenuto e permettere di scrollare, secondo la dimensione della pagina.
- Area suggerimenti
 - Liste multiple di video prodotte da algoritmi diversi di raccomandazione. Per ogni video viene mostrato un'immagine, qualche informazione tecnica (ad esempio il titolo) e il motivo per cui viene raccomandato.
 - Le liste possono essere visualizzate alternativamente, oppure tutte insieme, separate oppure mescolate, purché sia esplicita ed evidente l'origine e la natura di ciascun contenuto.
 - Visualizzare solo parte della lista e permettere di scrollare, secondo la dimensione della pagina.





Il recommender

- Un modulo che genera liste di n video, $10 \leq n \leq 30$. Usa molteplici algoritmi. L'output è uguale per tutti i tipi di algoritmo:
 - Thumbnail, titolo, codice YouTube, spiegazione della raccomandazione, **artista, titolo, album, anno, genere.**
- Svvariati tipi di algoritmi:
 - *Random*: lista di video casuale
 - *Search*: video che fanno match con la ricerca dell'utente (ok se $n < 10$)
 - *Related*: lista di video collegati secondo YouTube
 - *Recent*: lista di video visualizzati recentemente dall'utente
 - *Fvitali*: lista di video proposti da un'API di Fabio Vitali
 - *Popularity*: raccomandazione per popolarità assoluta, relativa locale **e relativa globale (di più a breve)**
 - *Similarity* (artista, genere, **membri della band, ecc.**): lista di video dello stesso artista, dello stesso genere (altri artisti simili), di band in cui suonano alcuni membri della stessa band, ecc.
- In tutti i casi è importante essere in grado di fornire una spiegazione della presenza del video nella lista





Recommender *random*

- Una lista di n video scelti a caso da YouTube
- Scollegato dal video dell'area principale.
- Ad ogni caricamento la lista cambia.
- La lista contiene sempre e solo video musicali





Recommender *search*

- L'utente inserisce testo nel campo di search
- Scollegato dal video dell'area principale.
- Il sistema usa l'API di YouTube per cercare alcuni video associabili a quel testo
- Se l'utente inserisce il codice di un video, viene trovato solo quel video
- I risultati del recommender *search* vengono visualizzati ***solo se l'utente l'ha usato per cercare un video***, altrimenti no.





Recommender *related*

- Il sistema usa l'API di YouTube per cercare video collegati a quello visualizzato nell'area principale.

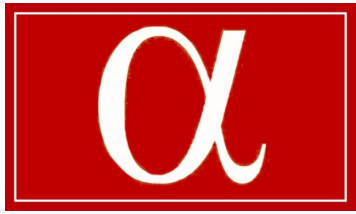




Recommender *recent*

- Il sistema ricorda quali video sono stati visualizzati recentemente dall'utente.
- Scollegato dal video dell'area principale.
- La lista è inizialmente vuota, poi si popola progressivamente di video.
- Ordinato temporalmente.
- Se lo stesso video viene visto più volte, compare una volta sola nella posizione più recente.
- Un video viene considerato visto solo se il player lo ha mostrato per più di 15 secondi.





La lista di partenza

- In un sistema reale tutti i video assumono naturalmente una storia di visualizzazioni sulla base della quale basare la raccomandazione.
- Noi siamo pochi e abbiamo poco tempo, e per dimostrare che il sistema funziona dobbiamo creare una situazione fittizia di video molto commentati.
- La classe intera deve quindi selezionare 40-50 video musicali, da specificare su un'apposita pagina del wiki, su cui si concentreranno gli sforzi iniziali del progetto.
- Suggerimento: scegliete brani musicali molto diversi tra loro per genere, artista ed epoca storica (includere musica classica, jazz, musiche etniche?)
- L'utente può selezionare qualunque video di YouTube, ma in prima pagina l'esaminatore può accedere ad uno dei 40-50 video di partenza scegliendolo da un'apposita voce del menù e mostrare i suggerimenti su questi come partenza.



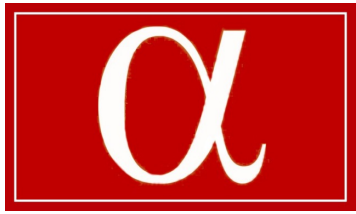


Recommender *popularity*

- Quattro recommender per raccomandare i video più popolari sulla base di quattro criteri diversi:

<i>Popolarità</i>	<i>Locale</i>	<i>Globale</i>
<i>Assoluta</i>	I video più visti dagli utenti del progetto	I video più visti dagli utenti di tutti i progetti del corso
<i>Relativa</i>	I video più scelti dopo quello mostrato nell'area principale dagli utenti del progetto	I video più scelti dopo quello mostrato nell'area principale dagli utenti di tutti i progetti del corso





Popolarità locale

- Il sistema tiene un contatore delle visualizzazioni dei video da parte di tutti gli utenti, ed è in grado di elencarne i più visti.
- Un video si definisce *visto* se il player ha funzionato per più di 15 secondi. Se l'utente riguarda ancora lo stesso video, viene contato due volte.
- Si può anche considerare la parte effettivamente visualizzata di video (numero di secondi, oppure percentuale della durata) ed usarla creativamente per le classifiche di popolarità .





Popolarità relativa

- Il tipo più importante di analisi della recommendation: non il successo intrinseco di un video, ma la relazione tra il video raccomandante e il video raccomandato
 - L'utente visualizza il video A, ed insieme a questo fornisce alcune liste di suggerimenti.
 - Se l'utente sceglie il video B dalla lista k di suggerimenti, e lo guarda, allora abbiamo creato una relazione di successo tra il video A e il video B secondo la raccomandazione k: $A \sim_k B$
- Poiché serve la visualizzazione e non basta la navigazione, posso creare una relazione solo alla fine della visualizzazione di B.
- Inoltre, se da A vado a B, da B torno ad A, e da A (anche senza visualizzarlo nuovamente) vado a C, allora la relazione esiste tra A e C.





API di popolarità globale (1)

- La popolarità globale richiede che io usi le informazioni di tutti i progetti del corso. Quindi è indispensabile accedere alle informazioni di navigazione di tutti.
- Viene quindi proposta un'API per accedere liberamente alle informazioni di popolarità del proprio sistema.
- Ogni team prima di consegnare il progetto inserisce il proprio indirizzo in fondo ad una lista in una pagina del wiki.
- Ogni gruppo che consegna dopo **DEVE** utilizzare i dati di popolarità di tutti i gruppi presenti nella lista prima del proprio indirizzo.
- Sono più fortunati i primi progetti, più sfortunati gli ultimi.





API di popolarità globale (2)

Un progetto disponibile presso siteXXXX.tw.cs.unibo.it **deve** mettere a disposizione un servizio all'indirizzo:

<https://siteXXXX.tw.cs.unibo.it/globpop?id=YYYYYY>

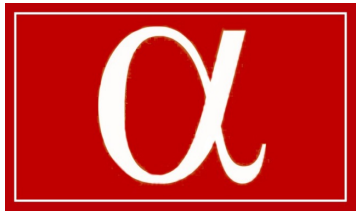
tale che:

- Il servizio è accessibile da qualunque indirizzo in GET
- YYYYYY è l'ID di un video di YouTube
- La risposta è un documento JSON come il seguente:

```
{
  "site": "siteXXXX.tw.cs.unibo.it",
  "recommender": "YYYYYY",
  "lastWatched": "Mon, 09 Apr 2018 09:03:39 GMT",
  "recommended": [{
    "videoId": "ZZZZZZ",
    "timesWatched": 7,
    "prevalentReason": "ArtistSimilarity",
    "lastSelected": "Mon, 09 Apr 2018 11:05:27 GMT",
  }, {
    "videoId": "WWWWWW",
    "timesWatched": 3,
    "prevalentReason": "GlobalPopularity",
    "lastSelected": "Mon, 07 Apr 2018 19:21:13 GMT",
  }, ...]
}
```

Uno di:

- *Random*
- *Search*
- *Related*
- *Recent*
- *Fvitali*
- *LocalPopularity*
- *GlobalPopularity*
- *ArtistSimilarity*
- *GenreSimilarity*
- *BandSimilarity*



Recommender *fvitali*

- Serve per permettere ai primi gruppi di avere qualche dato, e ai successivi per fare debugging senza caricare troppo gli altri gruppi.
- E' un'API identica a quella della popolarità globale, disponibile al primo indirizzo della lista dei gruppi pubblicata sul wiki.
- Sarà messa a disposizione appena completata la lista iniziale di video.
- Le informazioni di questa lista ***non vanno mescolate*** a quelle degli altri recommender (sarà probabilmente una lista statica).





Recommender *similarity*

- Due (tre) recommender diversi basati sul video mostrato nell'area principale
 - *ArtistSimilarity*: una lista di video dello stesso artista del video principale, **ordinati per album e anno di pubblicazione**.
 - *GenreSimilarity*: una lista di video di artisti ed album simili a quelli del video principale, **ordinati per vicinanza di genere**. Sbagliato includere video dello stesso artista.
 - *BandSimilarity*: (solo per video di band) una lista di video di band in cui suonano membri della band del video principale, **ordinati per vicinanza di anno con il video principale**. Sbagliato includere video della stessa band.





La troppa similarità

- Tutti i recommender per similarità ordinano dal più simile al meno simile.
- Data la ricchezza di materiale di YouTube, molti video sono MOLTO simili tra loro. Ad esempio posso avere copie multiple dello stesso video.
- Evitare la troppa similarità. Dovete suggerire dei percorsi interessanti, e non classificare dei video.





I pulsanti back e forward

- L'utente può e deve usare i pulsanti back e forward per andare avanti e indietro nella navigazione.
- In una single page application, back e forward vanno catturati e trasformati in navigazione interna alla applicazione stessa.
- Solo il play del video (per almeno 15 secondi) caratterizza la raccomandazione.
- Se l'utente fa back o forward, evitate di ricaricare ogni volta tutti i dati del video che abbiamo visto pochi secondi fa.
- Usate cache e memoria locale del browser in maniera appropriata.



Knowledge-Based Recommendations

- La raccomandazione similarity usa knowledge-based recommendation
- Usate approcci esistenti basati su conoscenza a proposito di musica e video
- Usate ontologie specializzate (ad es. Music Ontology)





The Music Ontology

The Music Ontology provides a model for publishing structured music-related data on your web site or through your API.

If you're unsure about why you should publish music-related data in this way, please read [our FAQ](#).

Getting started

Learn how to start publishing music-related data in your web pages or API, using The Music Ontology.

Learn more

Read the Music Ontology specification in full, describing all the classes and properties that can be used.

About

Get more information on The Music Ontology, its potential and its uses, by reading our frequently asked questions.

Site designed and built by [Rory Pickering](#)

Photograph is a derivative of work by Charles Thompson, licensed under Creative Commons.



Usare Knowledge Graphs per suggerire video

Usare grafi di DBpedia

- La pagina DBpedia dei Maroon 5 (http://dbpedia.org/page/Maroon_5) contiene una lista di album
- La pagina DBpedia dell'album “V” ([http://dbpedia.org/page/V \(Maroon 5 album\)](http://dbpedia.org/page/V_(Maroon_5_album))) contiene una liste delle canzoni con dettagli come autori, suonatori, ecc.
- Usare query SPARQL per accedere a queste informazioni (<https://dbpedia.org/sparql>)
- Oppure anche usare Music Ontology (<http://musicontology.com>)





ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

Valutazione del progetto

Criteri di valutazione (1)

Criteri di valutazione saranno:

1. la generalità del tool:

- quanto le soluzioni per la compatibilità sono forzate e quanto sono frutto di scelte ottimali per framework, organizzazione del codice e uso corretto delle tecnologie disponibili

2. la flessibilità:

- quanto le soluzioni tecniche adottate sono solide, strutturate, facilmente comprensibili, facilmente estendibili, facilmente adattabili a nuovi device / browser / sistemi operativi / modelli di dati / modelli di annotazione



Criteri di valutazione (2)

3. l'usabilità:

- Quanta attenzione è data alle esigenze di utenti (sia lettori, sia annotatori) che non conoscono i dettagli né del semantic web, né delle sintassi utilizzate, né del modello concettuali prescelto

4. la sofisticazione grafica

- Quanta attenzione viene data alla presentazione delle informazioni, al rapporto tra dimensioni delle maschere e dimensioni dei dati da rappresentare, al rapporto tra label comprensibili e dati formalizzati, alla corretta differenziazione nei tipi di dati e di annotazioni.



Il contributo individuale

- Ogni membro di ogni team deve dimostrare di aver contribuito in maniera determinante alla realizzazione del progetto.
- Ad inizio della presentazione ogni membro dichiara che cosa ha realizzato, e il docente, in totale autonomia, decide se questo contributo è o non è sufficiente.
- Realizzare solo HTML e CSS non è sufficiente.
- Realizzare parti marginali del codice (login, logout, lettura delle preferenze, ecc.) non è sufficiente
- Un candidato ideale si è occupato sia della parte HTML/CSS, sia della parte di programmazione, sia client sia server.
- La distribuzione ideale dei compiti è funzionale e non architetturale.



Il lavoro di team

- Tutti i membri dei team sono tenuti a lavorare e lavorare insieme.
- E' meglio essere parte attiva di un progetto mediocre che passiva di un progetto meraviglioso.
- Non saranno tollerati i portatori di pizze
- Mi riservo all'esame di scoprire il contributo individuale di ciascuno, indipendentemente dalla bontà del progetto consegnato.



Suggerimenti per l'esame

- Venite alla presentazione con il progetto che funziona. Se non va io vi faccio tornare. Per questo preferisco
 - vedervi una settimana dopo l'appello con il progetto che funziona
- piuttosto che
 - perdere tempo con la presentazione di un progetto che non va,
 - mandarvi via in lacrime, e
 - vedervi una settimana dopo l'appello con il progetto che funziona
- Venite allo scritto avendolo preparato. Non c'è niente di più irritante di vedere ragazzi svegli e competenti (vi si riconosce) che prendono 8 o 10 allo scritto perché ci hanno solo provato.
- Lo scritto non è difficile per chi ha studiato, è impossibile per chi non l'ha fatto.



Flessibilità del corso

- Prova scritta e prova di progetto sono indipendenti.
 - Il progetto è sempre di gruppo
 - Lo scritto è sempre individuale
- Potete provare lo scritto tutte le volte che volete
 - Il voto precedente verrà cancellato solo se consegnate un nuovo scritto
 - Gli scritti sono solo alle date degli appelli ufficiali
- Potete presentare il progetto tutte le volte che volete
 - Solo se lo decide consensualmente TUTTO IL GRUPPO
 - Potete ritirarvi dalla presentazione del progetto in qualunque momento e tornare una settimana o due dopo con le correzioni che ritenete opportune.



Rigidità del corso

(nessuna eccezione per nessun motivo)

- Il progetto deve funzionare. Completamente ed esattamente.
- Il progetto deve risiedere su una macchina del dipartimento.
- Se volete installare librerie e SW speciali per il progetto, verificate prima che questo sia possibile sulle macchine e sui sistemi operativi offerti dal dipartimento
- Il progetto deve venire presentato da tutto il gruppo insieme
In via eccezionale è accettabile che una persona sia collegata via skype. In nessun caso è accettabile che si presenti in una data una parte del gruppo e in una data diversa gli altri.
- Non usate tecnologie che richiedano compilazioni o procedure complesse di aggiornamento del codice eseguito.
Deve essere possibile aprire un file sul server e modificarlo al volo durante la prova.





ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

Fabio Vitali, Mehwish Alam

Dipartimento di Informatica – Scienze e Ingegneria
Alma mater – Università di Bologna

fabio.vitali@unibo.it
mehwish.alam@unibo.it

www.unibo.it