Sistemi Collaborativi e di Protezione (SCP)

Corso di Laurea in Ingegneria

Parte 3b – Social Media Technologies and Solutions

Prof. Paolo Nesi

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

http://www.disit.dsi.unifi.it/
nesi@dsi.unifi.it paolo.nesi@unifi.it
http://www.dsi.unifi.it/~nesi, http://www.axmedis.org

Part 3b: *Social Media Technologies and Solutions*

- **Semantic Computing**
  - Descrittori
  - Enrichment
  - Tassonomie
  - SKOS
  - Folksonomies
  - FOAF
- **Suggestion and Clustering**
  - Raccomandazioni / suggerimenti
  - Metrics Similarity Distances
  - Clustering algorithms comparison
  - Performances, Incremental Clustering
  - Suggerimenti U→U an improvement
  - Validazione del modello di suggerimento

- .....
  - .....

Semantic Computing on SN

- **Semantics into:**
  - Different types of Descriptors about
    - Users, content, etc.
  - User contributions:
    - Comments, votes, etc.
- **How semantics processing has supported SN**
  - Indexing and querying
  - Suggestions and recommendations
  - User understanding
- **Tools for Semantic Computing**

Semantic Descriptors

- **Modeling descriptors with formalisms:**
  - XML
  - MPEG-7, metamodel for descriptors and descriptors
  - MPEG-21: item descriptor and/or package
- **Audio, Video, images:**
  - **Low level** fingerprint/descriptors
    - Hash, MD5, etc.
  - **High level** fingerprint/descriptors
    - Genre, rhythms, color, scenes/movements, etc.
    - Evolution of them along the time, along the file
- **Documents**:
  - Keywords extractions, multilingual agnostic, …
  - Summarization
  - Paragraphs modeling and descriptions

Semantic Descriptors and info 1/3

- **user profile descriptions** collected via user registration and dynamically on the basis of user actions, migrated also on the mobile:
  - selected content, performed queries,
  - preferred content, suggested content, etc.;
- **relationships among users/colleagues** (similarly to friendships, group joining) that impact on the user profile and are created via registration, by inviting colleagues, etc.;
- **user groups descriptors** and their related discussion forums and web pages (with taxonomic descriptors and text);

Semantic Descriptors and info 2/3

- **content descriptors** for simple and complex content, web pages, forums, comments, etc.;
- **device capabilities** for formal description of any acceptable content format and parameters, CPU capabilities, memory space, SSD space;
- **votes and comments on contents, forums, web pages**, etc., which are dynamic information related to users;

Semantic Descriptors and info 3/3

- **lists of elements marked as preferred by users,** which are dynamic information related to users;
- **downloads and play/executions** of simple and/or complex content on PC and mobiles, to keep trace of user actions as references to played content, which are dynamic information related to users preferences;
- **uploads and publishing** of user provided content on the portal (only for registered users, and supervised by the administrator of the group). Each Content element has its own static metadata, descriptors and taxonomy; while the related action of upload is a dynamic information associated with the User who performed it. In addition, Content elements can be associated with Groups.

Content Enrichment

- The content, UGC, reaches the Social Network with partial information
- **Content Enrichment is needed** to get enough semantic information for
  - indexing/querying and producing suggestions
- **Content enrichment** may be performed by:
  - Addition/Extraction of semantic descriptors
  - Multilingual translation for metadata
  - Addition of annotations, textual and audiovisual, comments
  - Association of SKOS/taxonomical terms
  - Association of Tags → folksonomy
  - ***Comments, rating, citations, etc.***
  - ***Creation of Aggregations: collection, courses, play lists***

Extraction of semantic descriptors

- **Technical Information**
  - duration, resolution, size, dimension, video rate, sample rate and size, file format, MIME type, number of included files, file extension, etc.
  - libraries or tools can be used to extract information: FFMPEG for video and audio, ImageMagik for images, etc.

- **Context information:**
  - summary and extract keywords;
  - Video processing to: segment in major scenes, understand them, identify objects, colors, etc.
  - audio processing to extract tonality, rhythm, etc.
  - Images processing to extract contained objects, etc.

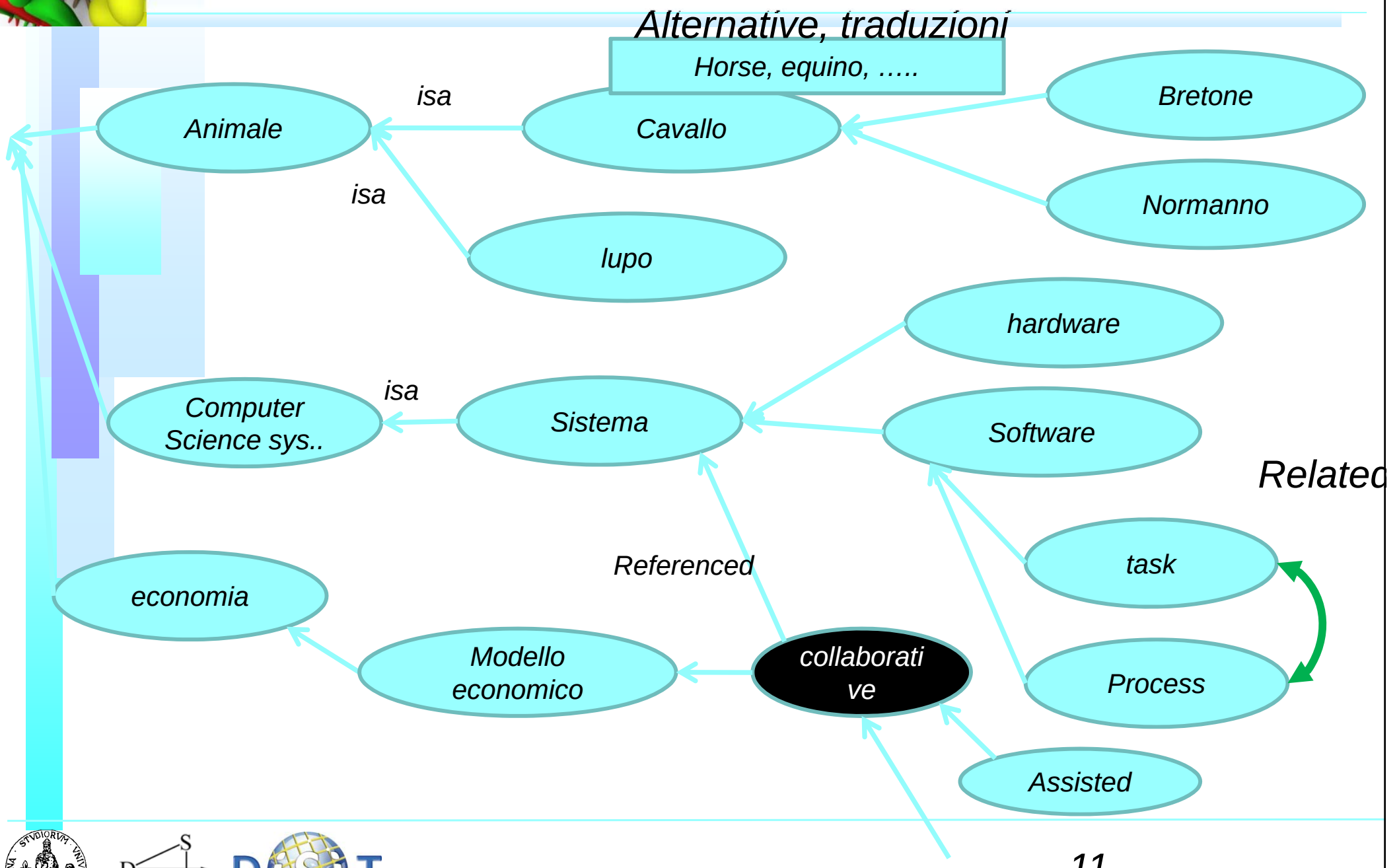
Esempio di Tassonomia

- Classificazione secondo vari assi
- Dominio Specifico
- Multilingua
- Istanze connesse a più nodi
- Preso da: http://mobmed.axmedis.org

- Algoritmi problemi e Tecniche di Rianimazione
- Emergenza ospedaliera
- Emergenza pre-ospedaliera
  - Algoritmi per sintomi di presentazione
  - Generale
    - Attrezzatura disponibile
    - Normative e leggi di riferimento
    - Raduni di massa
    - Trasporto aereo
    - Trasporto in ambulanza
    - Trasporto pediatrico e neonatale
  - Medicina dei disastri
    - Agenti chimici di distruzione di massa
    - Agenti radioattivi
    - Bioterrorismo
  - Trauma pre-ospedaliero
- Sintomi
- Utilità
  - Analgesia, anestesia e sedazione
  - Formule utili
  - Procedure
  - Radiodiagnostica

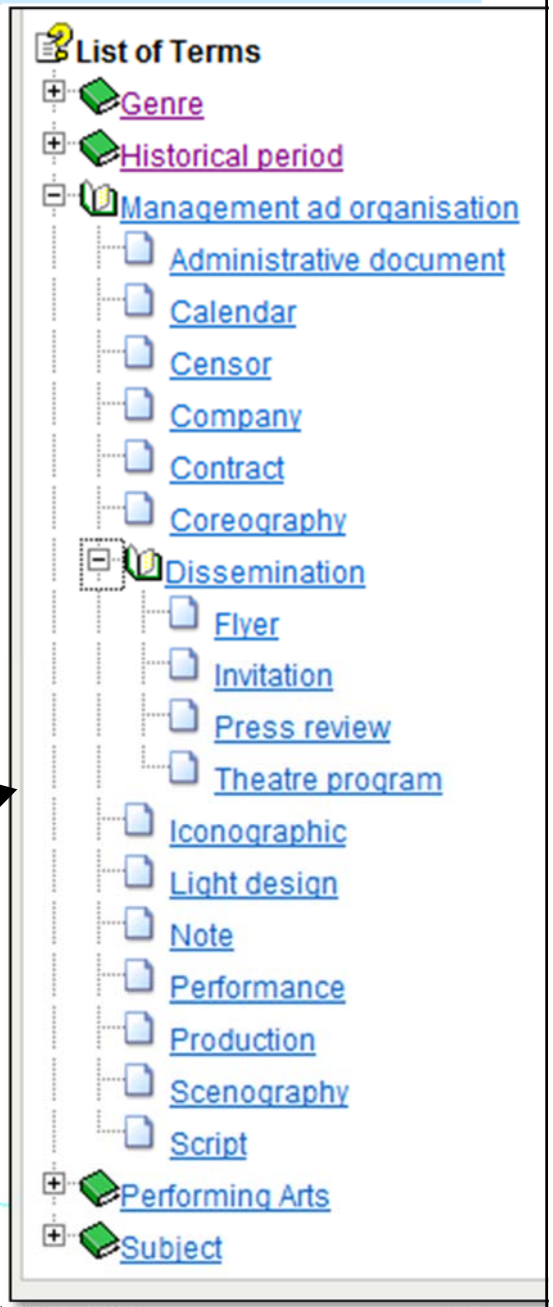
- Algorithms problems and techniques
- Emergency hospital
- Emergency pre-hospital
  - Algorithms for present
  - General
    - Equipment availabl
    - Regulations and la
    - Mass rallies
    - Aviation
    - Ambulance
    - Pediatric and neon
  - Disaster Medicine
    - Chemical agents o
    - Radioactive agents
    - Bioterrorism
  - Pre-hospital trauma
- Symptoms
- Utilities
  - Analgesia, anesthesia
  - Useful formulas
  - Procedures

SKOS simple knowledge organization system



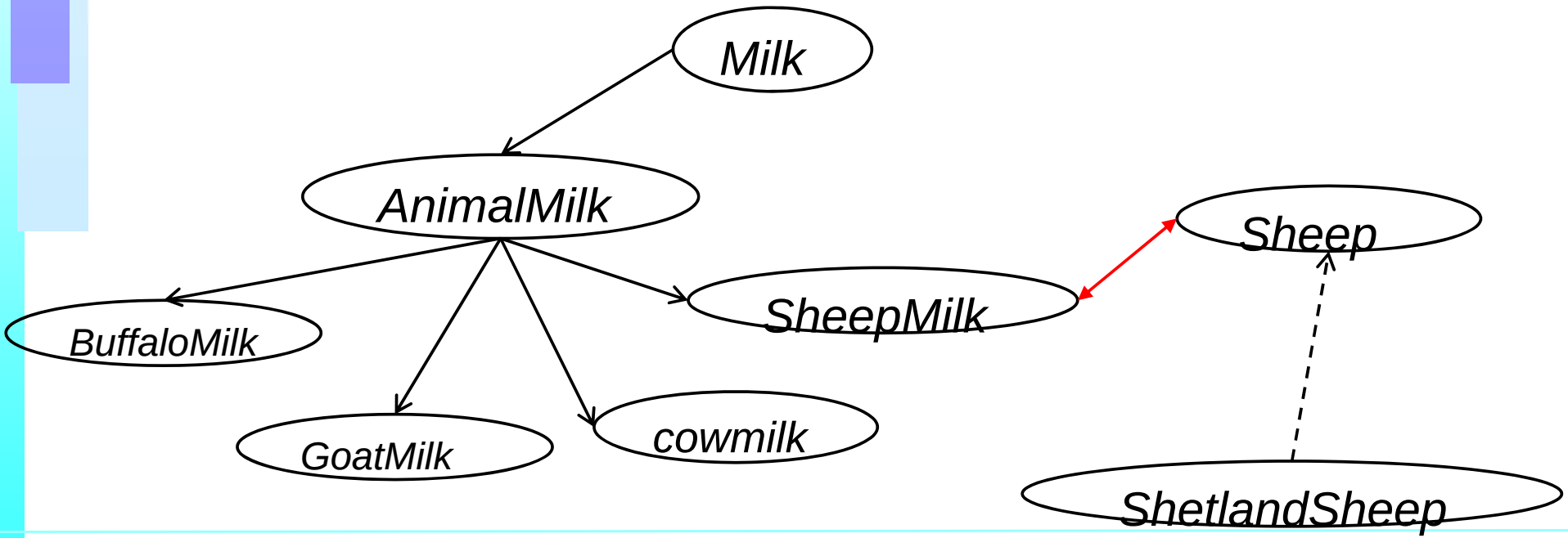
SKOS and taxonomy

- **SKOS: Simple Knowledge Organization System**
  - http://www.w3.org/2004/02/skos/
- One of the simplest models for knowledge organization
  - Hierarchy of specialization plus
  - relationships among elements: “related”
  - Instances associated to nodes of the hierarchy, multiple associations
- Modeling with OWL on RDF stores
- Classical Taxonomical classifications are a subset of SKOS



SKOS

- **SKOS is suitable** for modeling:
  - Thesauri, taxonomies and controlled vocabulary/dictionary
  - Adopted in Europeana for classification of content
- **Relationships**
  - Broader, narrower and related
  - Corresponding to: a generalization of, is a, is related to



Use of Taxonomy/Vocabulary

- In most SN, descriptors are not organized in a taxonomy but only via simple keywords to describe the
  - genre of the content,
  - historical period …..
- **Taxonomical classification** on Social Networks are typically static hierarchical models provided in multilingual
  - Such as in http://www.ECLAP.EU , http://mobmed.axmedis.org
  - The users can associate the content to one or more taxonomy nodes
  - Pros:
    - The queries may infer along the hierarchy and so more general concepts may provide larger results even if the content is not directly associated to them.

Tags and Folksonomy

- **Folksonomies** are typically
  - produced on the basis of free tags provided by the users in their preferred language
  - simple short statement or keywords
  - used to collect/describe the user point of view of a single content and/or of the whole portal
  - visualized as a tag cloud describing the whole folksonomy, see flickr.
  - not organized in hierarchy
- Tags for the cloud have to be translated to provide them in cloud which can be appreciated by users in their own language

access arrow arts axmedis collection company copyright cultural digital europeana experience experiences guidelines history holte manual mixed only open overview performance performing players reality report requirements royal shakespeare slides theatre

Tags and Folksonomy

- The list of collected tags has to be:
  - Processed to create multilingual representation
  - Cleaned removing stop words
  - Cleaned removing blacklisted words
  - …..

FOAF: Friend of a Friend

- **Friend of a Friend (FOAF):**
  - A format for supporting description of people and their relationships
  - a vocabulary in OWL for sharing personal and social network information on the Semantic Web
  - **Based on AAA principle:**
    - ➔ Anyone can say anything about any topic

- **Modeling Information:**
  - Organization at which people belong
  - Documents that people have created/co-authored
  - Images that depict people
  - Interests/skill of people,…
  - …..

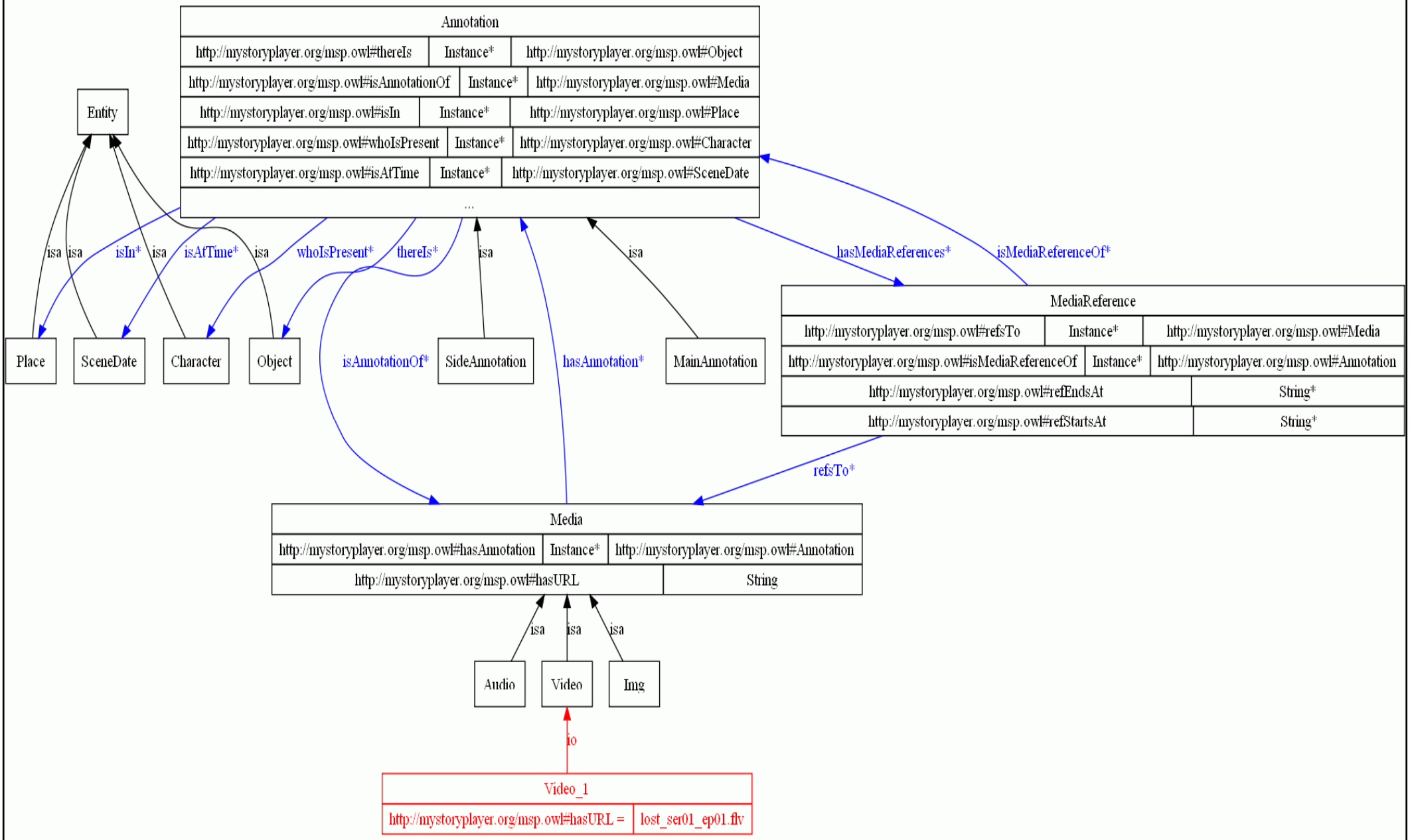
FOAF: Friend of Friend

```
<foaf:Person>
    <foaf:name>Peter Parker</foaf:name>
    <foaf:gender>Male</foaf:gender>
    <foaf:title>Mr</foaf:title>
    <foaf:givenname>Peter</foaf:givenname>
    <foaf:family_name>Parker</foaf:family_name>
    <foaf:mbox_sha1sum>cf2f4bd069302febd8d7c26d803f63fa7f20
      bd82</foaf:mbox_sha1sum>
    <foaf:homepage rdf:resource="http://www.peterparker.com"/>
    <foaf:weblog rdf:resource="http://www.peterparker.com/blog/"/>
    <foaf:knows> <foaf:Person>
      <foaf:name>Aunt May</foaf:name></foaf:Person>
    </foaf:knows>
</foaf:Person>
```

Ontologie

- L'ontologia è una specificazione *formale esplicita* di una *concettualizzazione* di un *dominio*
- Rappresentano:
  - Concetti e oggetti: modelli, categorie, proprietà,..
  - Relazioni fra concetti e fra relazioni
- Idealmente mirano a modellare in modo "*esaustivo*" un dominio

Un Esempio


Annotation
http://mystoryplayer.org/msp.owl#thereIs
Instance*
http://mystoryplayer.org/msp.owl#Object
http://mystoryplayer.org/msp.owl#isAnnotationOf
Instance*
http://mystoryplayer.org/msp.owl#Media
http://mystoryplayer.org/msp.owl#isIn
Instance*
http://mystoryplayer.org/msp.owl#Place
http://mystoryplayer.org/msp.owl#whoIsPresent
Instance*
http://mystoryplayer.org/msp.owl#Character
http://mystoryplayer.org/msp.owl#isAtTime
Instance*
http://mystoryplayer.org/msp.owl#SceneDate
...
Entity
isa
isa
isIn*
isa
isAtTime*
isa
whoIsPresent*
thereIs*
isa
isa
hasMediaReferences*
isMediaReferenceOf*
Place
SceneDate
Character
Object
isAnnotationOf*
SideAnnotation
hasAnnotation*
MainAnnotation
MediaReference
http://mystoryplayer.org/msp.owl#refsTo
Instance*
http://mystoryplayer.org/msp.owl#Media
http://mystoryplayer.org/msp.owl#isMediaReferenceOf
Instance*
http://mystoryplayer.org/msp.owl#Annotation
http://mystoryplayer.org/msp.owl#refEndsAt
String*
http://mystoryplayer.org/msp.owl#refStartsAt
String*
refsTo*
Media
http://mystoryplayer.org/msp.owl#hasAnnotation
Instance*
http://mystoryplayer.org/msp.owl#Annotation
http://mystoryplayer.org/msp.owl#hasURL
String
isa
isa
isa
Audio
Video
Img
io
Video_1
http://mystoryplayer.org/msp.owl#hasURL =
lost_ser01_ep01.flv

Base Ontologica

- Si può formalizzare in OWL (ontology web language), XML (Extensible Markup Language)

- Unifica/Generalizza modelli come:
  - Tassonomie
  - Tesauri
  - Vocabolari
  - SKOS: simple knowledge organization system
  - FOAF: friend of a friend

Base Ontologica

- Le ontologie sono specifiche di un dominio
- Spesso prodotte in team e formalizzate in OWL, vi sono strumenti di:
  - Editing, e.g., Protégé
  - Database semantici, e.g., Sesame in RDF (Resource Description Language)
  - Inferenza su database
  - Query semantiche, per esempio formalizzati in SPARQL (Simple Protocol and RDF Query Language)
  - ….

Part 3b: *Social Media Technologies and Solutions*

- **Semantic Computing**
  - Descrittori
  - Enrichment
  - Tassonomie
  - SKOS
  - Folksonomies
  - FOAF
- **Suggestion and Clustering**
  - Raccomandazioni / suggerimenti
  - Metrics Similarity Distances
  - Clustering algorithms comparison
  - Performances, Incremental Clustering
  - Suggerimenti U→U an improvement
  - Validazione del modello di suggerimento
- .....
  - .....

Recommendations/ Suggestions

- Frequently are considered as interchangeable

- **Our Terminology**
  - **Recommendations**:
    - provided by users to other users
    - provided by the system to users as ads
    - Etc..
  - **Suggestions**:
    - are recommendations produced by the system

Recommendations

- **Different Recommendations/Suggestions**
  - U → U: a user to another user on the basis of user profile
  - O → U: an object to a user on the basis of user profile
  - O → O: an object on the basis of a played object of a user
  - G → U: a group to a user on the basis of user profile
  - Etc…
- **Objects can be:**
  - Advertising, Content, Events, etc.
  - Some of them may have specific descriptors…

Different Recommendations

- **FOR YOU: Suggesting Users to another Users since they**
  - have similar preferences
  - like/prefer what you like/prefer
  - are friends of your friends
  - are in one or more of the your groups
  - are new of the Social Network
  - are the most linked, grouped, active
  - etc.

- **FOR THE SN: Suggesting Users to another Users since they**
  - *are important for the SN and do not have to left alone, the new entry*
  - *are the only contact path for Connecting a remote group, if the path is left a peripheral group will be completely disjoined with respect to the rest of the SN*
  - …

Different Recommendations

- **FOR YOU: Suggested objects/contents/events/groups since they**
  - are the less, most viewed, most played, most played in your group, ..
  - are similar to your highest voted/ranked objects
  - are similar to what you usually play, pay, print, upload, etc.
    - The most played/../voted in absolute
    - The most played/../voted in the last Month/Day, week, etc…
    - The most played/../voted in your area, country, group, etc..
  - are new for the SN
  - belongs to the preferred of your friends, …
  - have been posted/commented by your friends, in your group, …
  - have been recommended by a your friend

- **FOR BUSINESS: Suggested objects/…./groups since they**
  - are new for the SN, and thus are new for the market/business of the SN
  - are commercially proposed and have to be commercially promoted for the business of the SN
  - belong on the log tail of the content distribution/usage

Recommendations

		Recipient of the suggestions		
		User	**Content (played by a user)**	**Group (leader or members)**
Suggested elements	**Users**	Proposing to a user possible colleagues / friends	--no sense--	Proposing at a group responsible possible interested colleagues to be invited
	Contents	Proposing to a user possible interesting contents	Proposing at a play of a content similar content items	Proposing at a group members possible interesting content (not much different with respect to C-C combination)
	Groups	Proposing to a user possible interesting groups	Proposing at a play of a content possible interesting groups in which similar contents are discussed	--no sense--
	Ads	Proposing to a user possible interesting ads	Proposing at a play of a content the possible interesting ads	Proposing at a/all group member/s possible interesting ads

Why to recommend

- **The Social Network owners aim to:**
  - reduce the number of queries to reduce costs
  - push for the long tail content to increment of revenues
  - stimulate the socialization to have more connections
  - get more users,
  - get more value for advertising

- **To create more connected SNs**
  - More cohesion leads to have more resistance to close
  - More connections means more solidity and activity
  - Etc.

Similarity Distance

- The simplest solution for the recommendations/suggestion is to estimate the closest Users or Objects with respect to the reference User/Object
- The estimation of the closest entity between two entities described with multiple symbolic description is an instance of multidomain symbolic similarity distance among their descriptors.
- We can suppose for a while to have the possibility of estimating the similarity distance among descriptors.
- Some indexing tools, such as Lucene/Solr, may help in doing this with a query based on information of the reference user/object.

Complexity of Recommendation 1/2

- **Each day:**
  - N new users reach the SN
- The SN has 1 Million of users: U=10^6

- The SN has to suggest the possible friends to the new N users immediately:
  - Complexity is an O(NU)
  - N*U distances should be estimated in real time/per day
  - If N=10^6 such as on YouTube
  - Thus: 10^12 estimations of 10ms,
    - → 10^10s → which are 317 years !!!

Complexity of Recommendation 2/2

- **Each day:**
  - M new UGC items are uploaded on the SN,
- The SN has
  - 1 Million of content: C=10^6
  - 1 Million of users: U=10^6
- The SN has to estimate the distance of that content with respect to all the other items/objects and users:
  - Complexity is an O(MC+MU)
  - M*C distances to be estimated in real time/per day
  - M*U distances to be estimated in real time/per day
  - If M= 1 Million
  - Thus: 10^12 estimations of 10ms, thus 10^10s, 2*317 years !!!

Technologies for Recommendations

- **Objective:**
  - To provide targeted elements on the basis of the elements descriptors
- **Technical solutions**
  - **create distance matrices** and matching via direct distance or similarities estimations, very unfeasible for millions of elements would be too expensive
  - **making queries on the basis of element profile** to get the most similar. For millions or elements with several aspects or dimensions in descriptors would be very complex
  - **use some clustering to create group of elements**, also based on distances or similarities. If the groups are too many, the precisions can be low while the costs are contained.

Part 3b: *Social Media Technologies and Solutions*

- **Semantic Computing**
  - Descrittori
  - Enrichment
  - Tassonomie
  - SKOS
  - Folksonomies
  - FOAF
- **Suggestion and Clustering**
  - Raccomandazioni / suggerime
  - Metrics Similarity Distances
  - Clustering algorithms comparison
  - Performances, Incremental Clustering
  - Suggerimenti U→U an improvement
  - Validazione del modello di suggerimento
- .....
  - .....

Similarity Distances

		Recipient of the suggestions		
		User	**Content (played by a user)**	**Group (leader or members)**
Suggested element	**Users**	D(U(s,d);U(s,d))	**--no sense--**	D(U(s,d);G(s,d))
	Contents	D(C(s);U(s,d))	D(C(s);C(s))	D(C(s);G(s,d))
	Groups	D(G(s,d);U(s,d))	D(G(s,d);C(s))	**--no sense--**

General Distances Models

- Weighted Models:

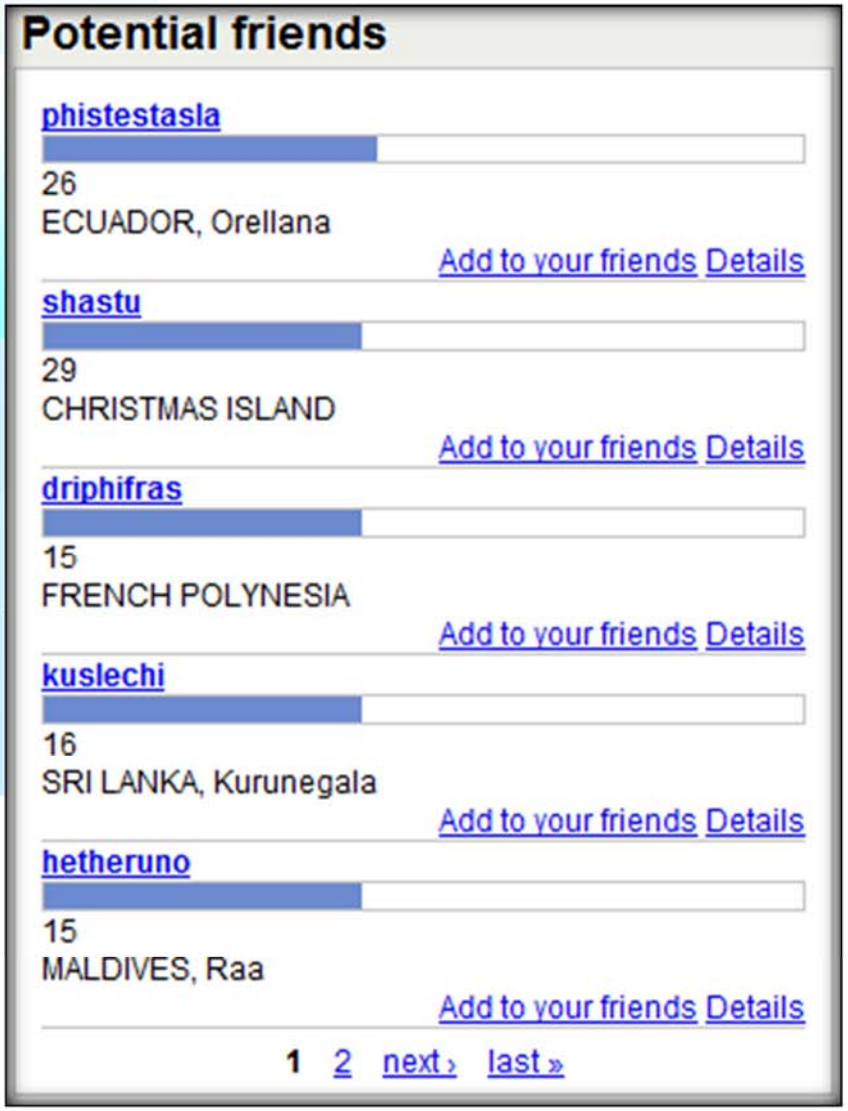
$$D(U1; U1) = k_s \sum_{i=1}^{T_1} x_i Sd_i(U1, U2) + k_d \sum_{i=1}^{T_2} y_i Dd_i(U1, U2),$$

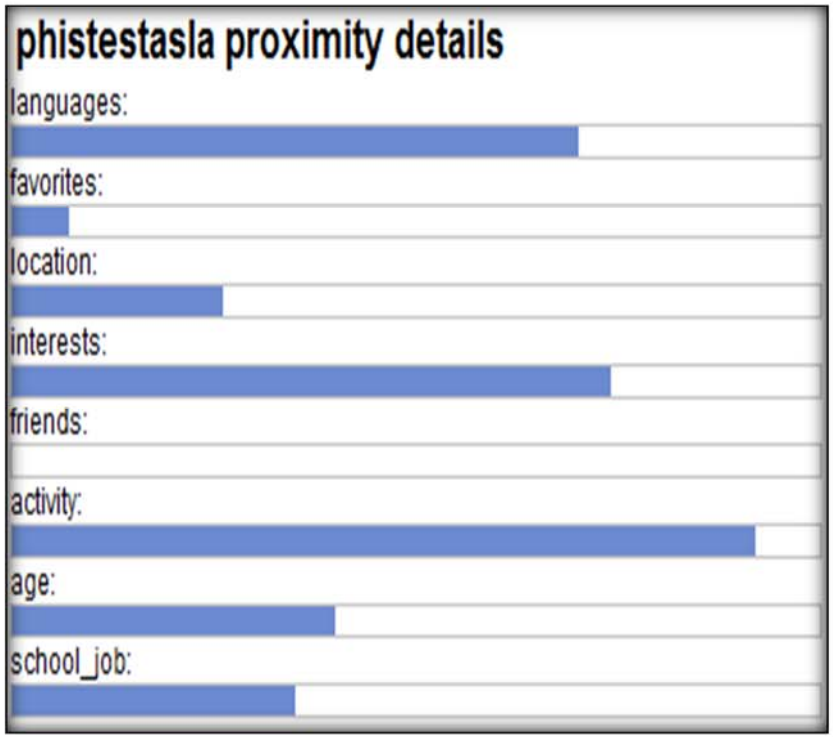
- Vector weighted models:

$$D(U1; U2) = \left\{ \begin{array}{l} K_s(x_1 Sd_1(U1,U2), x_2 Sd_2(U1,U2), \ldots, x_n Sd_{Ts}(U1,U2)), \\ K_d(y_1 Dd_1(U1,U2), y_2 Dd_2(U1,U2), \ldots, y_n D_{Td}(U1,U2)) \end{array} \right\}$$

- The weights can be defined according to the SN goals.
- They can be determined by using multi-linear regressions techniques.

Visualizzazione di Suggerimenti e dist





Static Similarities for User profiles

- **Similarity of User Ages**

$$Sda(User1, User2) = |Age(User1) - Age(User2)| / MAXdelta$$

- **Similarity of user Languages**, Sdl()
  - More than one language
  - Definition of a model
  - Measure of distances

- **Similarity of user nationalities**, Sdn()
  - More than one language
  - ….

Elaborazione dei profili degli utenti

Alcuni dei dati contenuti nei profili degli utenti sono utilizzati per stabilire una metrica fra di essi. Due utenti, A e B, sono confrontati e il risultato è un valore di vicinanza

$$0 \geq v(A,B) \geq 1 \quad \text{e} \quad v(A,B) \neq v(B,A)$$

- *Confronto su 8 categorie: per ognuna il risultato è* $0 \geq v_i(A,B) \geq 1$
  - *Lingue parlate*
  - *Località di provenienza*
  - *Livello scolastico e lavoro*
  - *Età*
  - *Interessi*
  - *Amici*
  - *Attività*
  - *Lista di oggetti preferiti*

- *Periodicamente viene eseguito il confronto di ogni utente con tutti gli altri.*
- *Per ogni utente vengono salvati nel database i 10 utenti con più alto valore di vicinanza.*
- *Il valore di vicinanza è la somma pesata dei risultati nelle 8 categorie.*
- *I pesi sono stati scelti in modo da far valere di più i risultati delle categorie di informazione dinamiche:*
- *Amici, Attività, Preferiti*

$$v(A,B) = \sum_{i \in C} v_i(A,B) \cdot w_i$$

$$C = \{languages, location, interests, schoolJob, age, activity, friends, favorites\}$$

$$0 \leq v_i \leq 1$$

$$\left\{ w_i \geq 0 \quad \sum_{i \in C} w_i = 1 \right\} \quad \text{con } w_i \text{ peso associato alla categoria } i$$

Distance among spoken languages

- a matrix *m[i][j]* is created where each element represents the similarity between the two languages
  - *m[i][j]=m[j][i]*
  - all languages are classified into families (Latin, Anglo-Saxon, Slovenian, Asian, etc.), groups and subgroups;
  - to each leaf, one language or a set of similar languages is assigned;
  - to each branch a numeric weight, , is assigned where *i* is the number of families.
- Thus for the hierarchy, the following property holds:

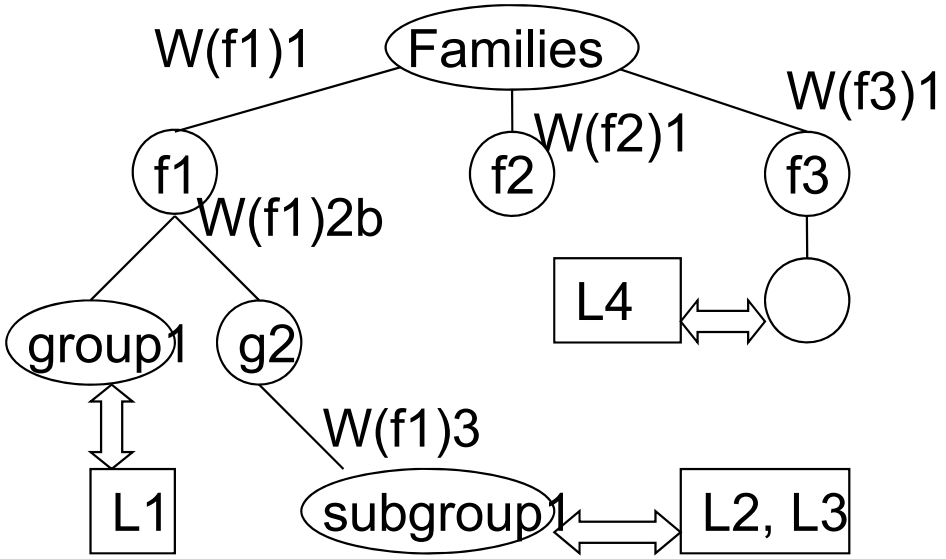
$$0 \le \sum_{j=1}^{BL} w(fi)_j < 1; \quad \text{and} \quad w(fi)_1 < w(fi)_2 < \ldots < w(fi)_{BL}$$

  - *j* is the tree level for language family ; BL is the height to reach the leaf in the tree

Distance among spoken languages

- if User1 selects *(L1, L4)* and User2 selects *(L2, L3, L5):*

$$Sdl(User1, User2) = \max\{m_{12}; m_{13}; m_{15}; m_{42}; m_{43}; m_{45}\}$$



$$\begin{cases} Sdl_1(U1, U2) = \max\{m_{13}; m_{14}; m_{23}; m_{24}\} \\ where: m_{13} = W_{f1\_1}\ , m_{14} = m_{24} = 0, m_{23} = W(f1)_1 + W(f1)_{2b} + W(f1)_3 \end{cases}$$

Static Metric, User specializations

- Each Element *S[i][j]* of the matrix represents the similarity between two specializations

$$Sds(A,B) = \frac{\sum_{i \in P_A} \max_{j \in P_B} (S[i][j])}{AN}$$

- Pa, Pb are the set of specialisation of the Users: A,B
- The measure is not symmetric and bounded 0-1

Other Static Metrics

- Static distance of preferences via Taxonomical classification, where t[i][j] is the matrix of similarity among terms:

$$Sdt(A,B) = \frac{\sum_{i \in T_A} \max_{j \in T_B} (t[i][j])}{AN}$$

  - distance between a couple of taxonomy terms is reported in matrix *t[i][j]*, of *TxT*
- Static distance of preferences via subscription to groups

$$Sdg(A,B) = \frac{card(G_A \cap G_B)}{card(G_A)}$$

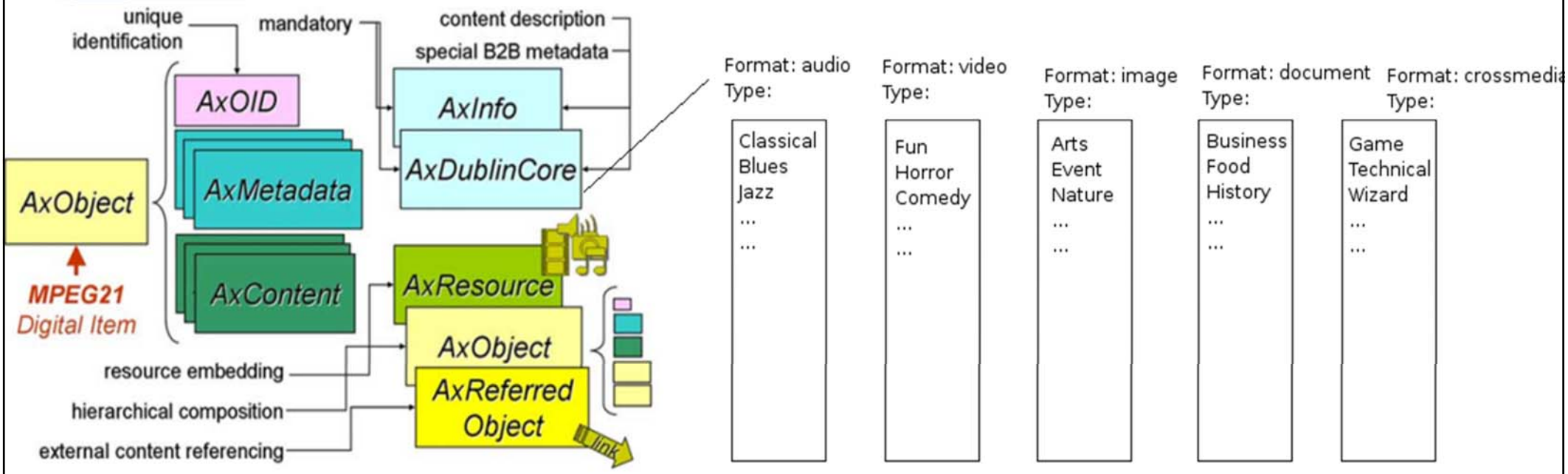
Dynamic Distances

- **Contents**: supposing that acting on similar content may create a similarity between users or from users with content:
  - **positive**: downloads, set as preferred, recommend, publish, play, positive comments, high votes, ..;
  - **negative**: low votes, negative comments, ..;
- **Users**: supposing that acting on other Users is motivated by a similarity with them:
  - **positive**: set as friend, recommend, ...;
  - **negative**: negative comment, unconnect,... ;
- **Groups**: supposing that acting on some Group is due to a similarity between them:
  - **positive**: subscribe, contribute, associate a content with, etc.;
  - **negative**: leave, negative comment …

Dynamic Distances

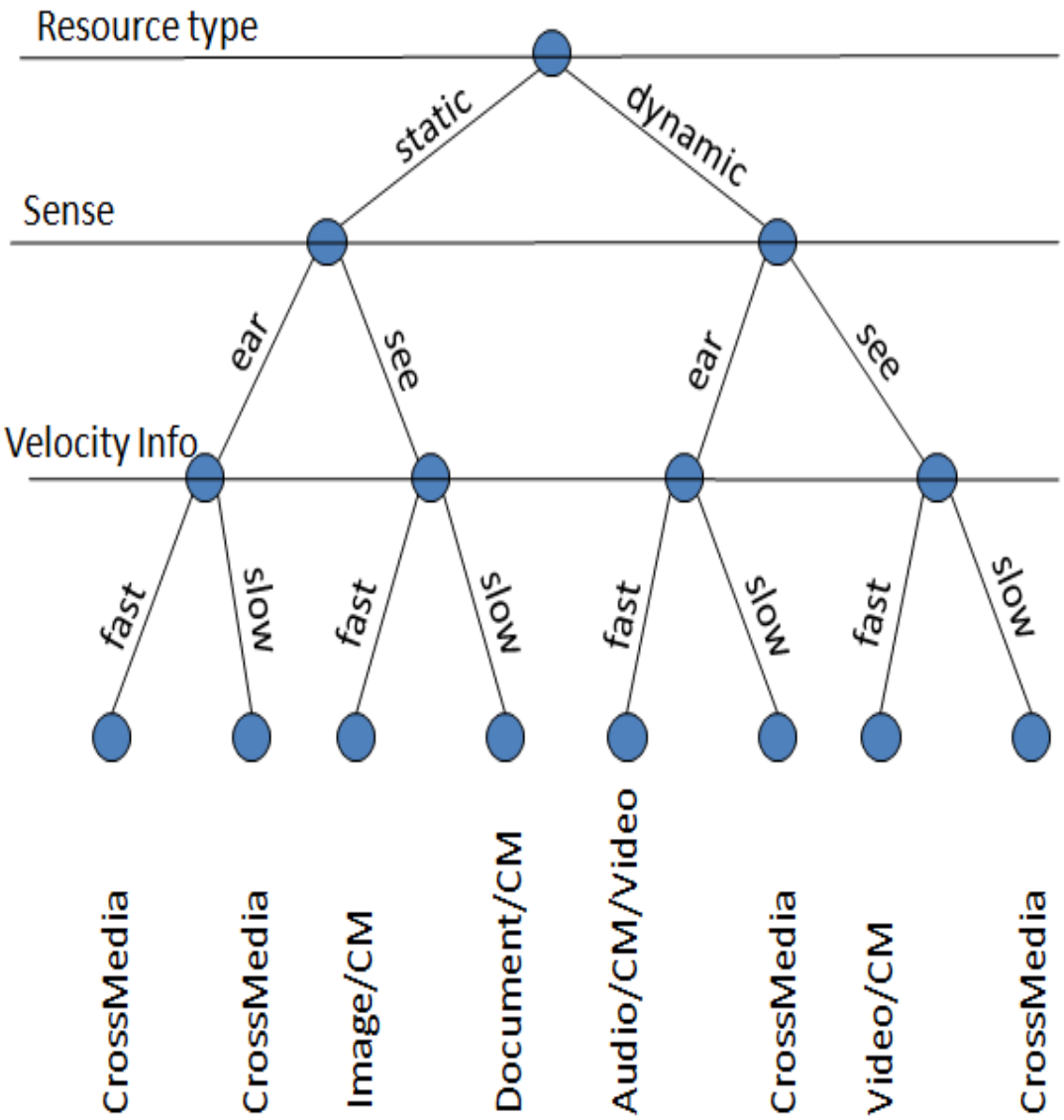
- **Dynamic Metric on User's Interested Taxonomy Topics**
  - Ddt(UA,UB)
  - …
  - Ddt'(C,U): can be used to estimate the similarity distance between a static taxonomy profile and a dynamic taxonomic profile as needed in comparing terms of taxonomy of Content and Users, or of Groups and Content
- **Dynamic Metric on User's Interested Formats**
  - Ddf(UA,UB), Ddf'(C,U)
  - …
- **Dynamic Metric on User's Preferred Content items and colleagues**
  - **….**

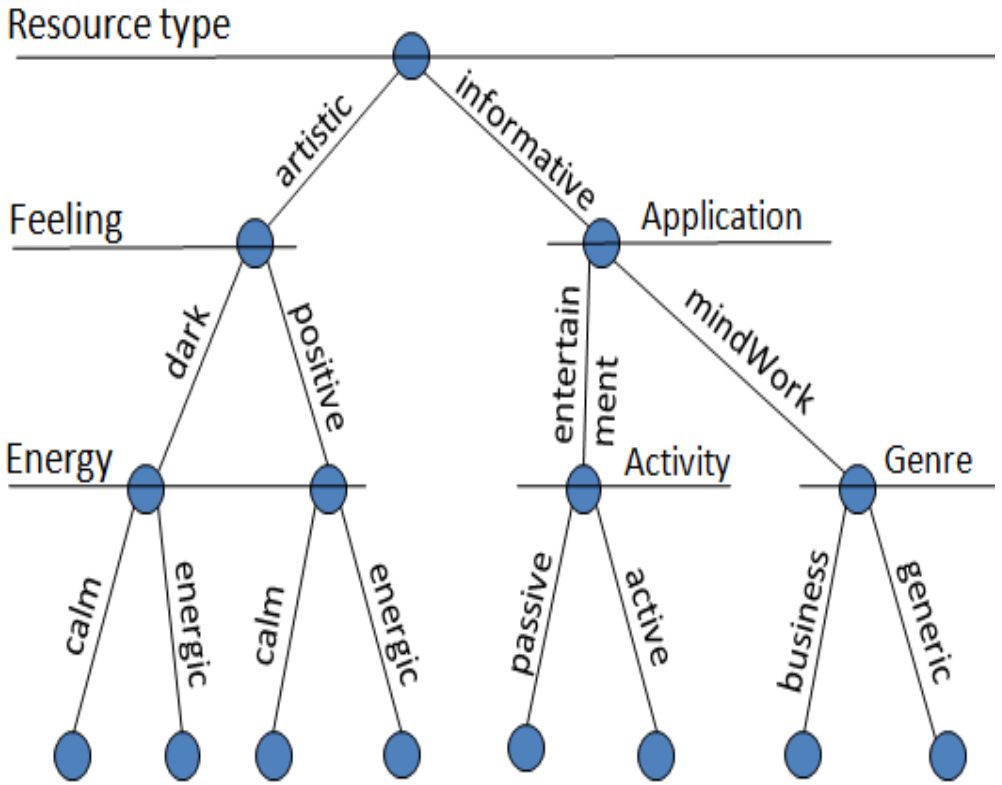
Oggetti Cross Media, Rich Content



- *Metadato Format: indica il formato dell'oggetto. Sono stati scelti 5 possibili valori: "crossmedia", "video", "audio", "image", "document"*
- *Metadato Type: sono stati definiti alcuni possibili valori per questi campi dipendenti dal valore del campo Format. Es. "Classical", "Blues", ecc. per il Format "audio"; "Animation", "Fun", "Horror", ecc per il Format "video".*

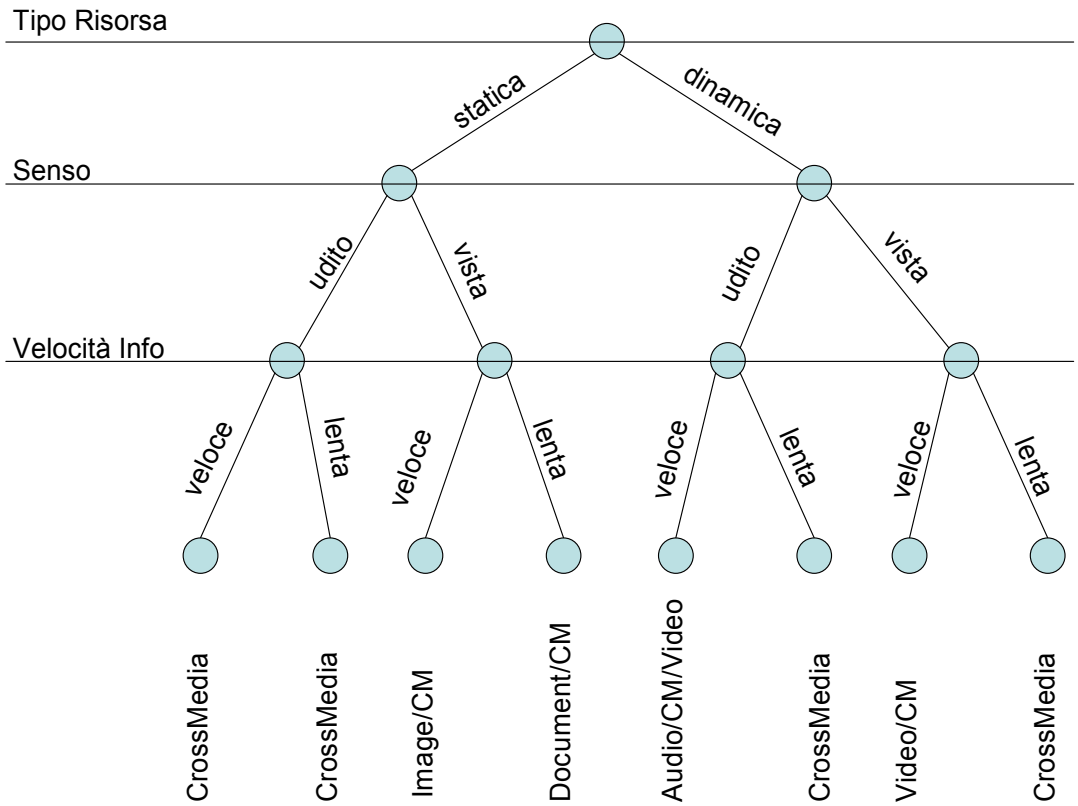
Semantic Distance: Format/Type





Converting Symbols to numbers

- Many aspects of descriptors are symbols: languages, genre, nationality, music classification, video kind, etc..
  - A classification taxonomy or a graph can be created to connect the possible symbolic values assigned to them distances (see the graph in the figure of this slide)
  - A matrix can be created defining distances among all the possible symbolic values (for example among languages)



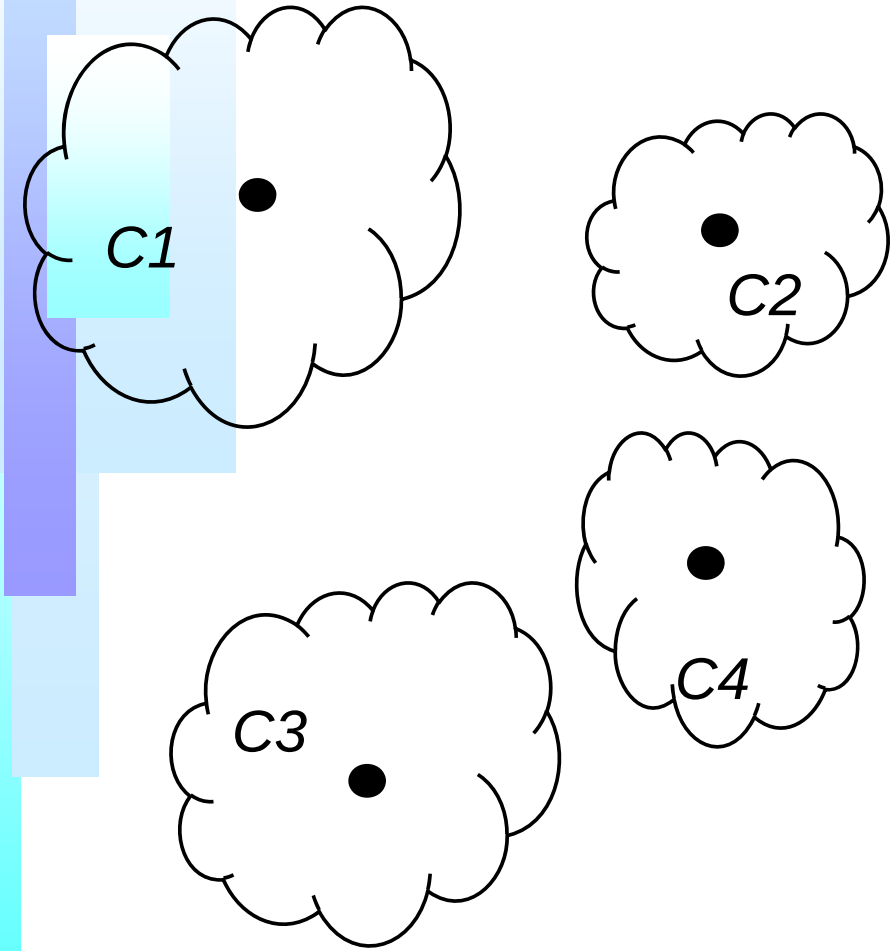
Summary of Similarity Distances

- *D(U,U) = Function of (Sda(), Sdl(), Sdn(), Sds(), Sdg(), Sdt(), Sdf(), Ddt(), Ddf(), Ddp(), Ddc())*
- *D(C,U) = Function of (Sdl(), Ddt'(), Ddf'())*
- *D(G,U) = Function of (Sdl(), Sdt(), Ddt(), Ddf())*
- *D(C,C) = Function of (Sdm(), Sdl(), Sdt(), Sdf())*
- *D(G,C) = Function of (Sdl(), Sdt(), Ddt'(), Ddf'())*
- *D(U,G) = Function of (Sdl(), Sdt(), Ddt(), Ddf(), Ddp())*
- *D(C,G) = Function of (Sdl(), Sdt(), Ddt'(), Ddf'(), Ddp())*

Part 3b: *Social Media Technologies and Solutions*

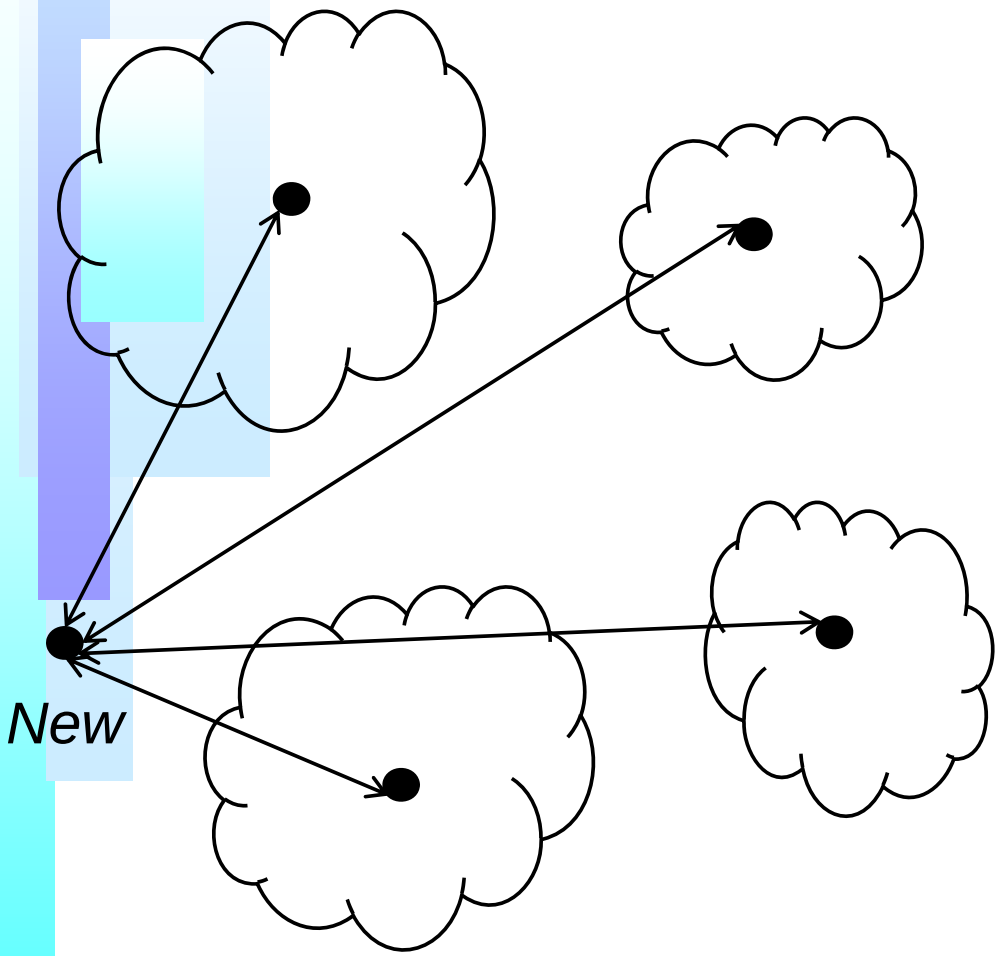
- **Semantic Computing**
  - Descrittori
  - Enrichment
  - Tassonomie
  - SKOS
  - Folksonomies
  - FOAF
- **Suggestion and Clustering**
  - Raccomandazioni / suggerimenti
  - Metrics Similarity Distances
  - Clustering algorithms comparison
  - Performances, Incremental Clustering
  - Suggerimenti U→U an improvement
  - Validazione del modello di suggerimento
- .....
  - .....

Clustering among descriptors



- *K-Means* clustering
  - Based on a multidimensional distance model among each other
  - Define the number of clusters
  - Estimation process to maximize the cohesion among clusters
- Some items can be spare
  - They are classified in any case
- Millions of content items, thousands of clusters, …
- Periodic re-clustering taking into account all the content/objects/users

Clustering among descriptors



- Millions of content items,
- ONLY thousands of clusters
- At each New Object
  - Distance of the new object with respect to cluster Centers
  - Reduction of complexity
- Usable on recommendations:
  - UU, UO, OO, etc.

Clustering k-means

- Good performance in terms of scalability;
  - discovery of clusters with arbitrary shape;
  - ability to deal with noise and outliers;
  - insensitivity to order of input records;
  - support for high dimensionality.
- Complexity of an *O(NKI)*, where *N* is the number of elements, *K* the number of clusters and I the number of iterations.
- k-means has demonstrated the best performances when *N* is largely bigger than *K* and *I* (Everitt, Landau, Leese, 2001).

Clustering k-means

- The K-means assigns each point to the cluster, the centre of which is called centroid and it is the average of all the points in the cluster (it is not a real one).
  - starts by choosing the number of clusters, *K* (which can be determined by using statistical analysis or imposed); randomly generate K clusters centers;
  - assign each point to the nearest possible centroid by means of distance computation;
  - compute again the cluster centers that minimizes the sum of the squared error in associating the points to clusters.
- The convergence is achieved by iterating the last two steps until no or minimal changes are performed on clusters.
- The K-means has been integrated into the AXCP tools by using the Weka implementation (Bellini, Bruno, Fuzier, Nesi, Paolucci, 2009).

K-means problems

- **dependency on** the availability of numerical absolute distance estimations between two numerical values

- **Unfortunatly elements descriptors** are
  - mainly symbolic and in some cases with multiple values,
  - coming from both the semantics and concepts they describe.

K-medoids clustering

- **K-medoids** adopts as a center of the cluster the element which has the minimal average (or the median) distance among the others involved in the cluster.
- This means that the complexity is grounded on $O(K(N-K)^2)$, that for $N>>K$ is an $O(N^2)$.
  - N are the elements
  - K are the medoids/clusters
- initially the clusters centers are some selected elements (Xui & Wunsch, 2009).

K-medoid clustering

- The algorithm is mainly implemented as:
  - random selection of *K* point of *N*;
  - associate each point of the *N* to the closest *medoid* by using some distance metric, may be the Euclidean or others as described in the rest of the paper;
  - For each *medoid m*
    - For each non-medoid *nm:*
      - Swap *m* and *nm* and computer the global costs of the configuration
  - Select the configuration with lowest cost in terms of averaged distance of all the elements in the cluster;
- The last two steps have to be continued until no changes in *medoids* are accepted.

Hierarchical Clustering

- Hierarchical clustering (Xui & Wunsch, 2009) are creating clusters on the basis of the distance among the single elements.
- The process starts by aggregating the closest elements to create smaller clusters of two elements and then aggregating these small clusters with other by following a sort a merging algorithm.
- The aggregation is based on the distance metrics
- Hierarchical algorithms may differ for the mathematical model used for the merging of subclusters: complete linkage, single linkage and averaged.

Cluster Based Recommendations

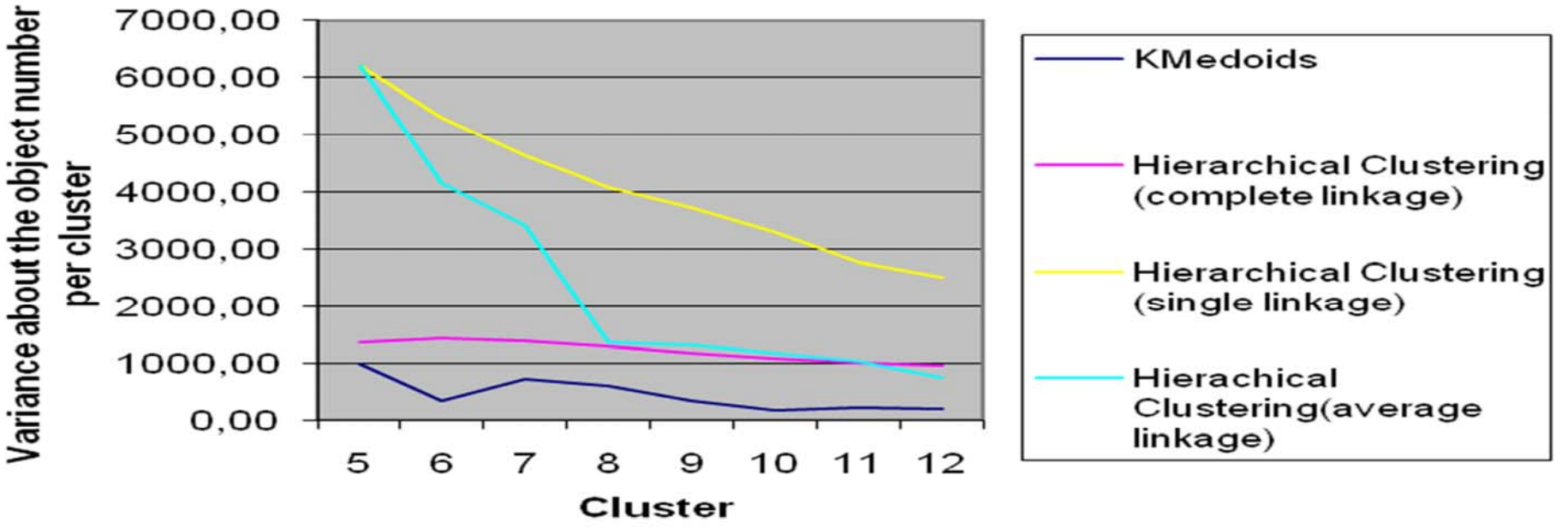
- ***Pros***
  - Lower computational costs, clusters are numerically much less.
  - At each new entry in users or content, the recommendation can be performed on the basis of the centers of clusters, or on cluster description
  - For example in the case of before, with 1.000 clusters, the user recommendation:
    - O(NK), 10^6*10^4*10ms => 3,8 months,
- ***Cons***
  - The proposed users/content/items are not those that are closest and neither the most similar.
  - They are only some (a random selection) of items that are into the cluster that present the high similarity with respect to the item selected (user, content, etc.)

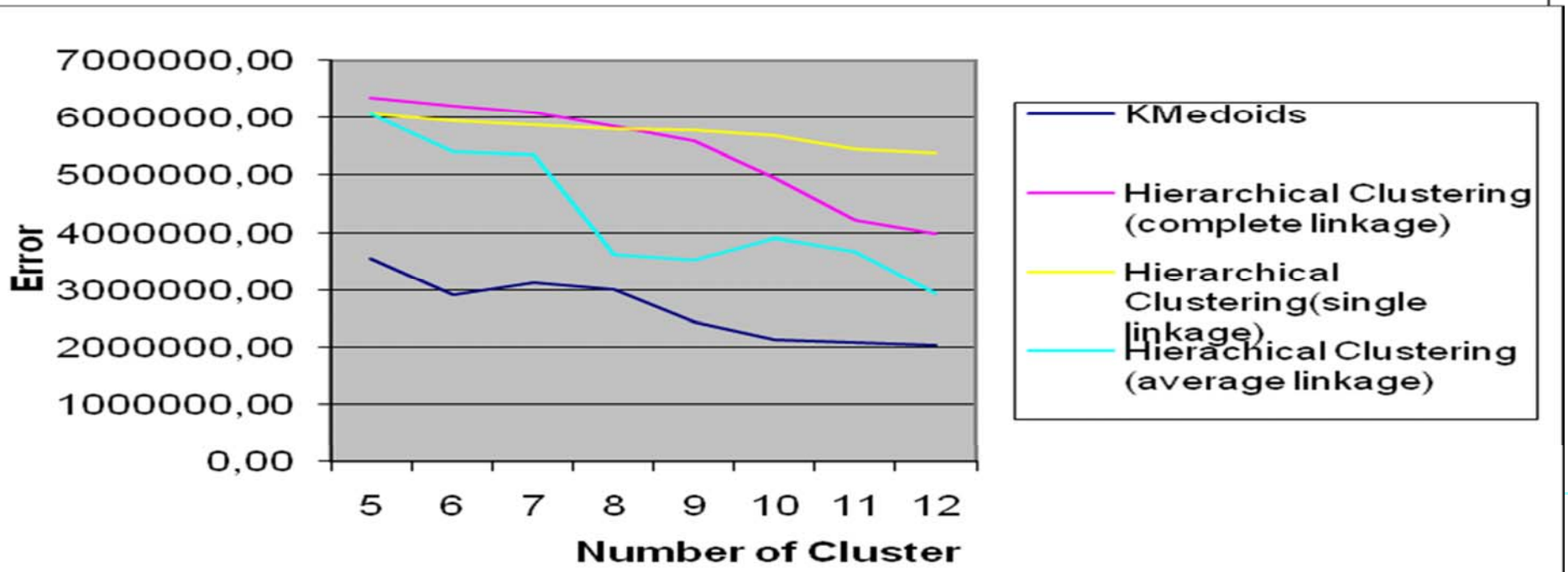
Clustering of Content for C→C

- ***C→U via***
  - *D(Ccs,U) = Function of (Ddt'(Ccs,U), Ddf'(Ccs,U))*
  - *U has a static description for pref. tax and format*
- ***C→C via***
  - *D(Ccs,C) = Function of (Sdt(Ccs,C), Sdf(Ccs,C))*
  - *C has a static description for pref. tax and format*
- ***C→G via***
  - *D(Ccs,G) = Function of (Sdt(Ccs,C), Ddt'(Ccs,G), Ddf'(Ccs,G))*
  - *G has a static description for pref. tax and format*

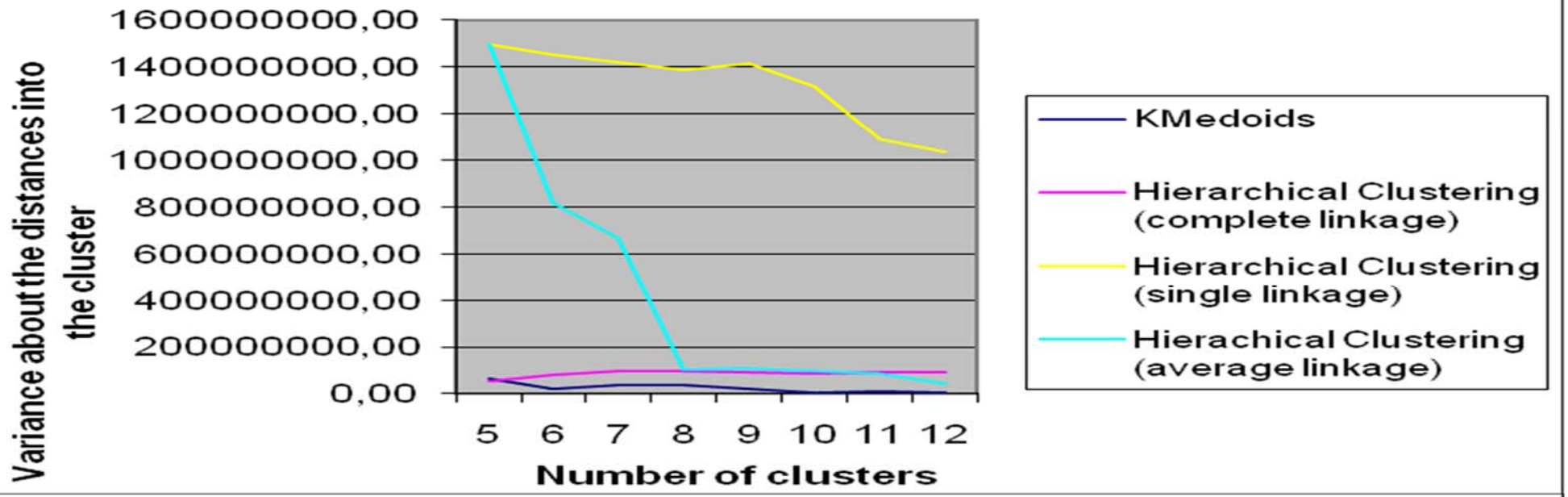
- ***Where**: Ccs: are the cluster centers*
- **And** random selection in the closest cluster

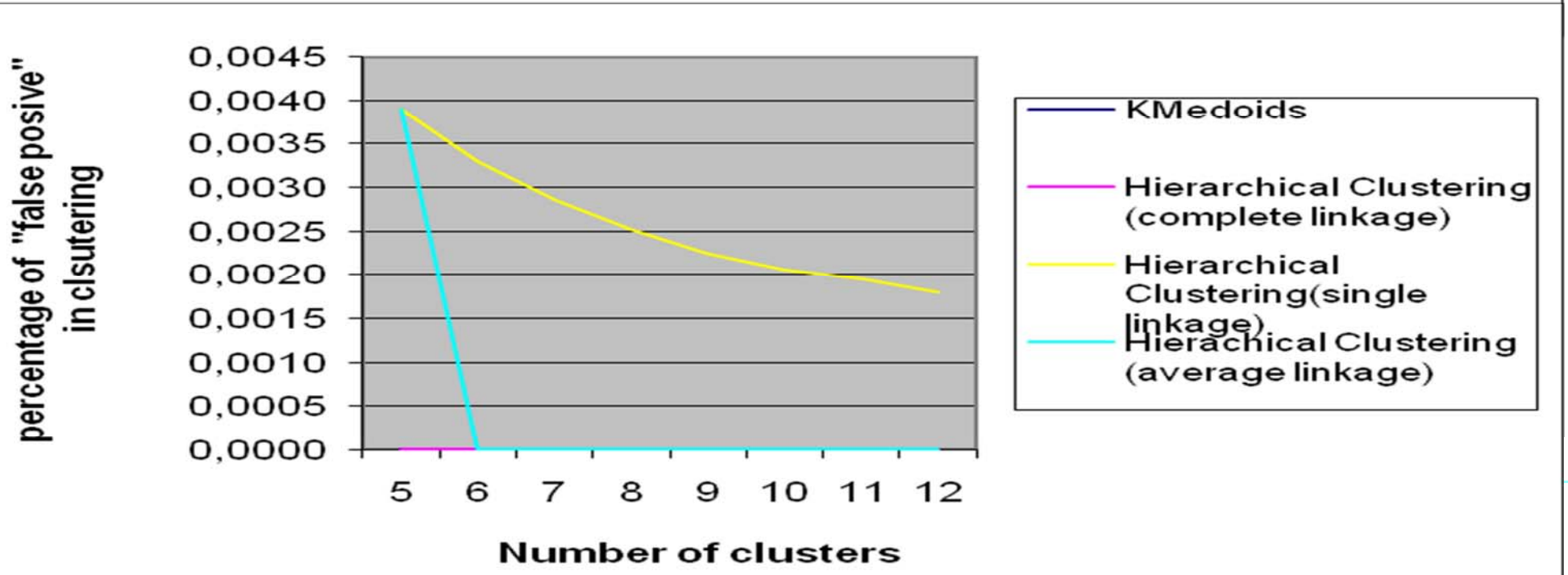
Comparison of Clustering algs





Comparison of Clustering algs



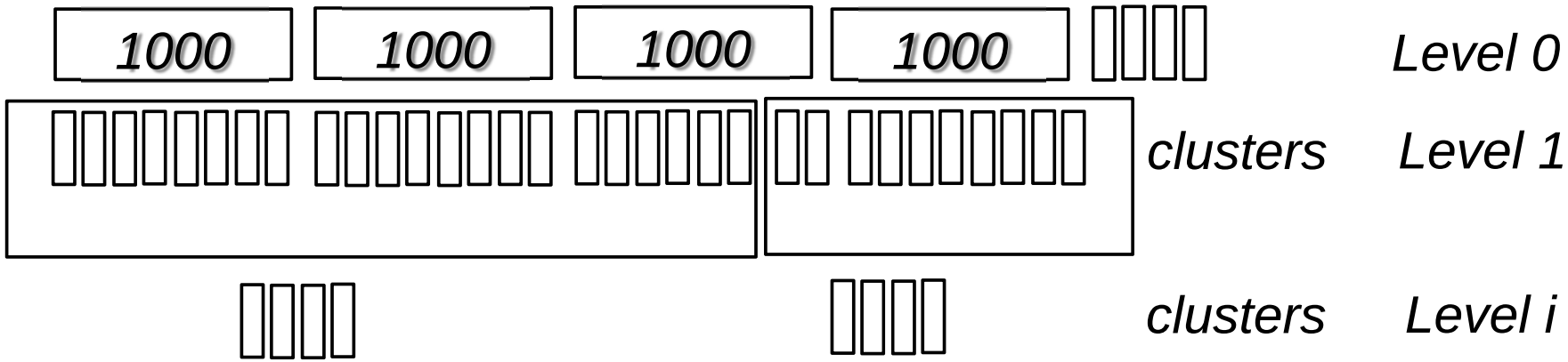


Consideration on Clustering algs

- **K-means** has good performance
  - it works only on absolute distance numbers
  - Not viable for complex domains
- **K-medoids** works on symbolic distances via taxonomies
  - Vector or scalar measures are viable
  - It is quite computationally heavy
  - Excellent in rejecting errors and on classification
- **Hierarchical clustering** with **average linkage**
  - Vector or scalar measures are viable
  - Very efficient in computation
  - Good in rejecting errors and on classification

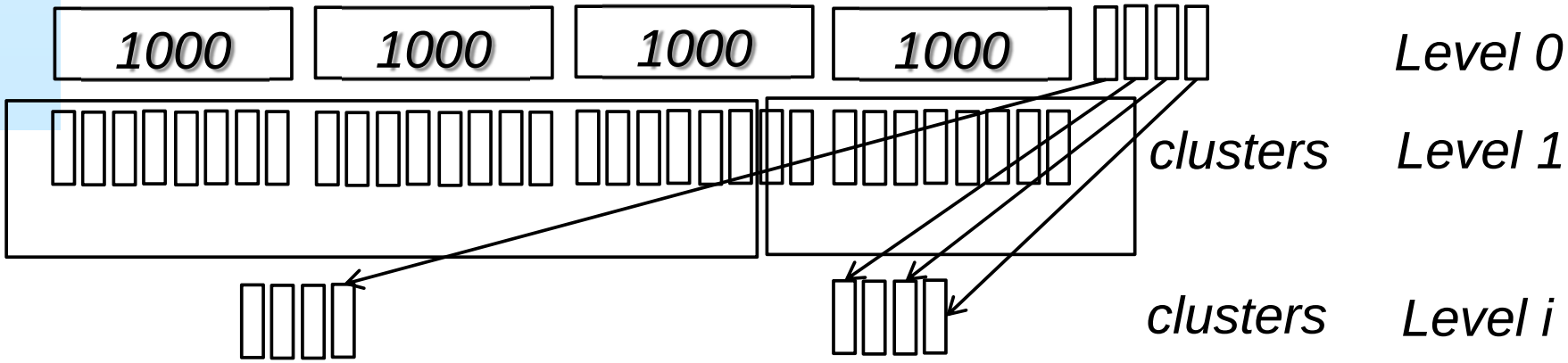
Incremental Clustering

- Segmenting the total number of elements/objects in chunks of 1000
  - estimating *C* clusters for each chunk.
- The resulting centers of clusters can also be segmented in chunks of 1000 elements and clustered in *C* clusters.
- This process may continue until the resulting number of centers clusters is less than the chunk dimension (1000).
- At level 0 of the hierarchy, the clustering has to be performed on *N* elements; at level *i* on $N*(C/1000)^i$.

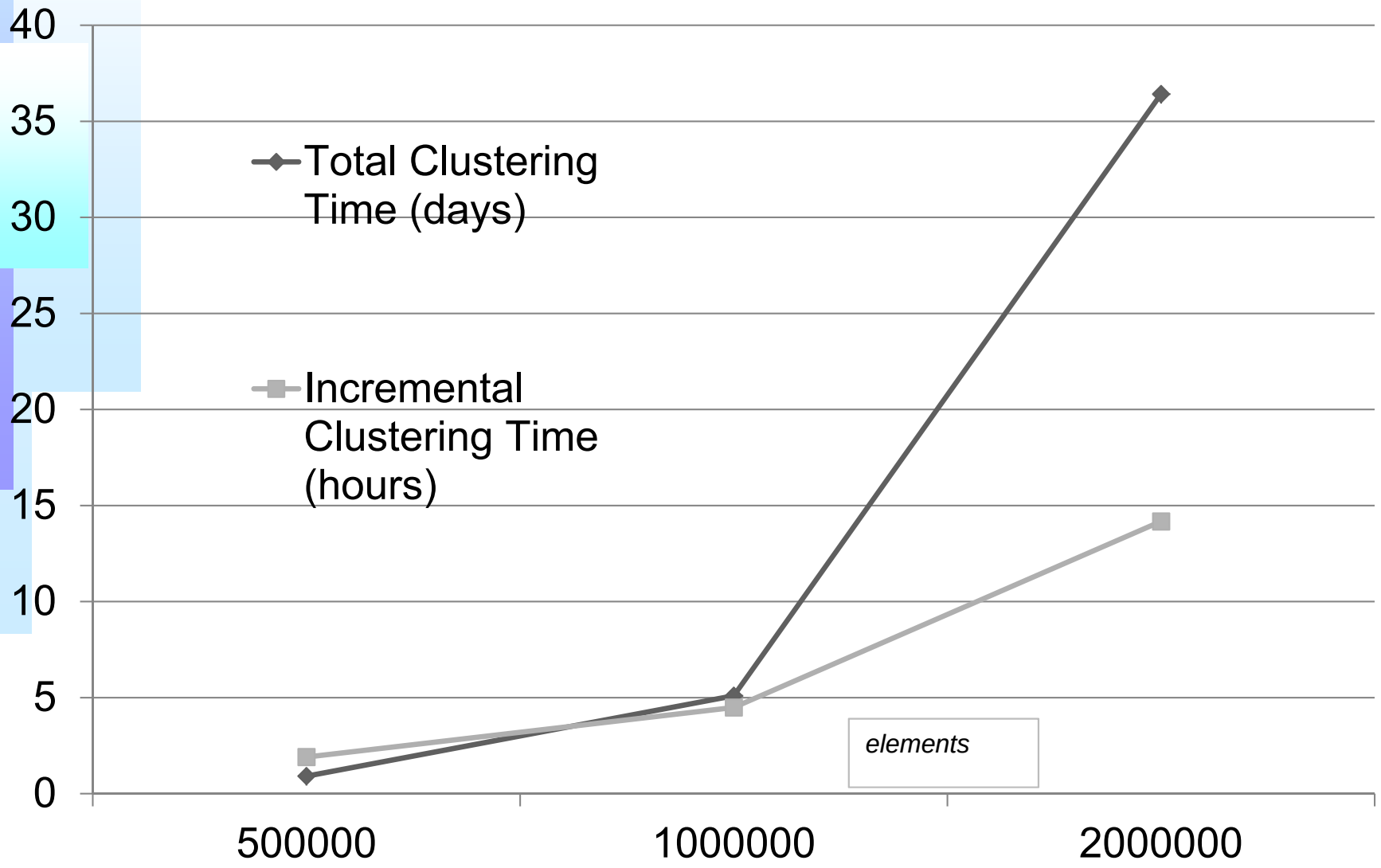


Incremental Clustering

- According to this model, every time a new element is added it can be temporary classified in the closest cluster according to its similarity with the last level *C* centers.
- Once 1000 new elements are obtained, they are removed from temporary status and the incremental clustering can restart re-clustering from level 1 to last (since the selected elements for each level is random, this allows to regularize the new solution).



Incremental Clustering



Incremental Clustering

- **1 millions of elements**
  - total clustering time:
    - in the order of 5.1 days (including data access),
  - incremental clustering of the last 1000 elements is performed in 4.49 hours (including data access),
    - resulting time saving is of 27 times.

- **2 millions of elements**
  - resulting time saving is of about 61 times.

Part 3b: *Social Media Technologies and Solutions*

- **Semantic Computing**
  - Descrittori
  - Enrichment
  - Tassonomie
  - SKOS
  - Folksonomies
  - FOAF
- **Suggestion and Clustering**
  - Raccomandazioni / suggerimenti
  - Metrics Similarity Distances
  - Clustering algorithms comparison
  - Performances, Incremental Clustering
  - Suggerimenti U→U an improvement
  - Validazione del modello di suggerimento
- .....
  - .....

Raccomandazione Utente ⇔ Utente

- La raccomandazione sulla base della similarità tra utenti viene generata utilizzando due set di informazioni, le statiche e le dinamiche
  - ✓ Informazioni statiche: età, lingue parlate, tipi di oggetti di interesse…
  - ✓ Informazioni dinamiche: amici, preferiti, caratteristiche oggetti visti …
- Calcolo similarità mediante confronto Utente ⇔ utente
  - ✓ Effettuato su dati eterogenei (lingue parlate, nazionalità…)
  - ✓ Dipendente dalle classificazioni tassonomiche degli oggetti visti

Informazione	Tipo	Metrica	Esempio
Località	Lista di elementi in formato ISO-3166	$v_{location}(A,B)$	IT
Lingua	Lista di elementi in formato ISO-369	$v_{language}(A,B)$	It-it
Età	Numero (Unix Timestamp)	$v_{age}(A,B)$	633420000
Gruppi di appartenenza	Lista di gruppi	$v_{groups}(A,B)$	[Digital Meets Culture, Italy, …]
Connessioni comuni	Lista di utenti	$v_{friends}(A,B)$	[ivanb, …]
Categorie di contenuti di interesse	Lista di categorie	$v_{interest}(A,B)$	«Gestione e organizzazione, …»
Percentuali di visualizzazione delle categorie di contenuti	Array di numeri reali	$v_{taxonomy}(A,B)$	[1:0,4; 2:0,5; …]

Le raccomandazioni Utente ⇔ Utente precedenti

La soluzione precedente:

- *Non rappresenta propriamente le preferenze degli utenti.*
  - ✓ *Col tempo ogni contenuto fruito incide sempre meno nella rappresentazione del profilo degli utenti. All'aumentare delle fruizioni, la singola influisce sempre meno sul calcolo della vicinanza perché dispersa in un insieme dalla cardinalità monotonicamente crescente.*
  - ✓ *Classificazione del contenuto fruito dall'utente rende la similarità stimata su base tassonomica poco accurata*
  - ✓ *Non avevano influenza nel calcolo le classificazioni dei contenuti promossi e dei contenuti preferiti, ma venivano considerati solo quelli visti e scaricati in un'unica categoria*
- *Non propone altro che utenti con gusti e preferenze simili nei contenuti.*
  - ✓ *Mancanza di metriche di valutazione di vicinanza diverse dalla valutazione dei gusti*

Obiettivi e percorso del progetto

- Rendere maggiormente dinamiche e accurate le raccomandazioni proposte agli utenti del social network ECLAP
- Studiare e implementare nuove politiche di raccomandazione utente-utente
- Verificare
  - ✓ l'efficacia delle tipologie di suggerimento
  - ✓ quali tipi di informazioni spingono gli utenti a stringere amicizia

La ristrutturazione

La struttura attuale:

- *Migliorata la valutazione delle preferenze degli utenti*
- *Migliorato e riadattato l'algoritmo di calcolo per la similarità sulla base dei profili statici*
- *Introdotte altre tipologie di suggerimento diverse da quelle basate solo sulla similarità tra i profili degli utenti*
- *Aumentata scalabilità considerando nel calcolo solo gli utenti attivi del sito*

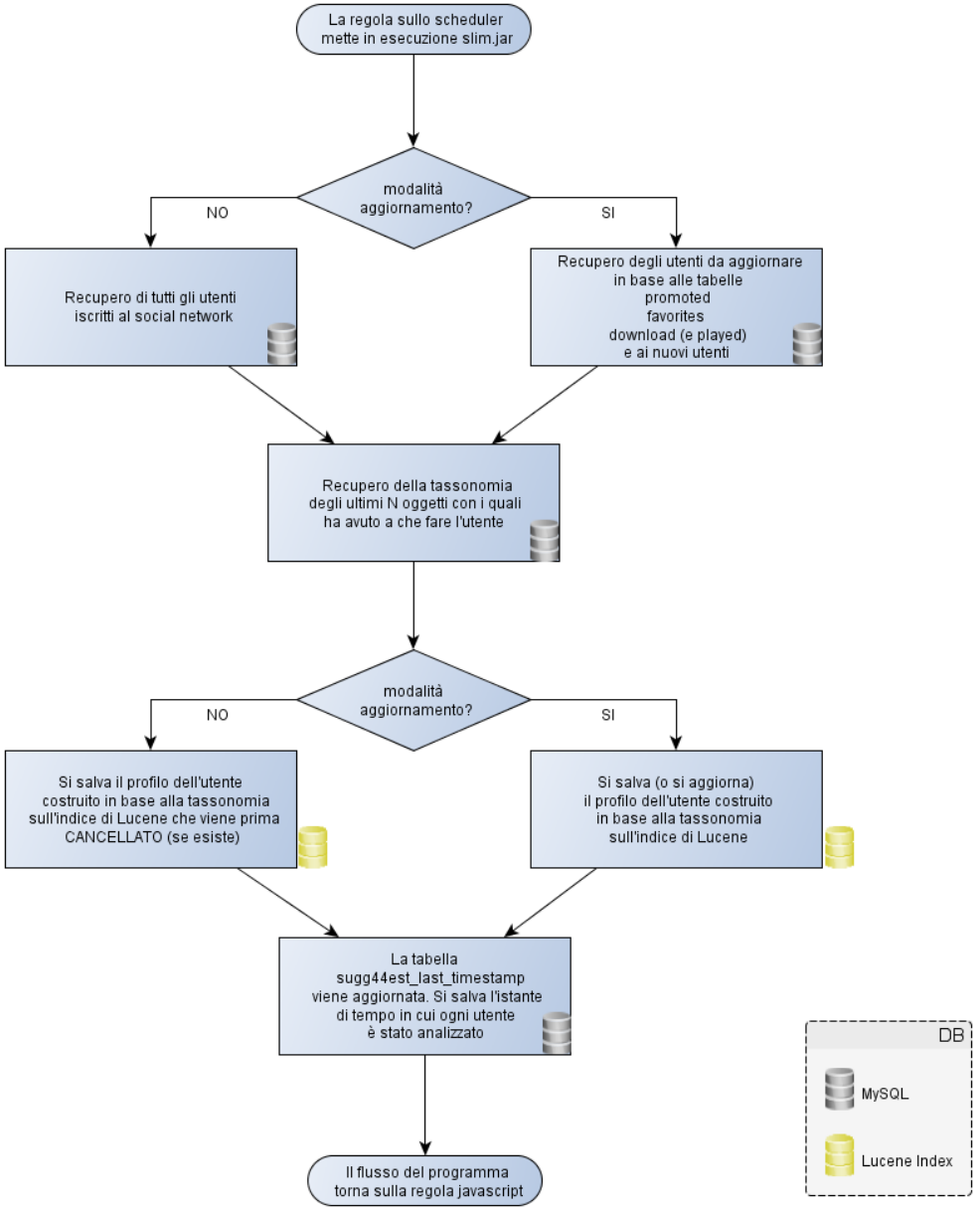
Informazione	Tipo	Metrica	Esempio
Percentuali di visualizzazione delle categorie di contenuti	Array di numeri reali	$v_{taxonomy}(A, B)$	[1:0,4; 2:0,5; ...]

trasformata in

Informazione	Tipo	Metrica	Esempio
Viste	Stringa composta da termini della tassonomia	$proximity_{dynamic}(A, B)$	«Performing Art Danza Balletto, ...»
Scaricati			«Performing Art Danza Balletto, ...»
Aggiunti ai preferiti			«Gestione e organizzazione, ...»
Promossi			«Performing Art Musica Blues, ...»

Dinamicizzazione

- *Per il calcolo del profilo dinamico degli utenti :*
  - ✓ *Si verifica stato del sistema per differenziazione delle modalità di calcolo (tutti gli tenti considerati, solo quelli attivi)*
  - ✓ *Per ogni utente in elaborazione*
    - ✓ *vengono recuperate le stringhe testuali contenenti le classificazioni tassonomiche degli ultimi N oggetti fruiti in tutte le lingue presenti su ECLAP*
    - ✓ *Tali classificazioni vengono raggruppate a seconda del tipo di interazione (visualizzazione, download, promozione, aggiunta ai favoriti), concatenate l'una all'altra e salvate in un documento che viene memorizzato all'interno di una base dati indicizzata tramite il motore Lucene.*
- *Una prima specifica dell'importanza della classificazione tassonomica relativa al tipo di interazione con i contenuti è applicabile a questo livello dell'algoritmo per la generazione delle raccomandazioni.*



Calcolo Prossimità

- *Per trovare gli utenti simili in termini di preferenze viene sfruttata una somiglianza testuale fra i documenti grazie al motore Lucene. Viene usato il documento relativo ad un utente come chiave di ricerca.*
  - ✓ *Se due utenti hanno preferenze simili i documenti memorizzati dalla procedura, descritta prima, relativi ad entrambi gli utenti hanno dei termini al loro interno in comune.*
  - ✓ *Viene sfruttata la misura di similarità TF-IDF nella versione implementata all'interno del motore Lucene.*

$$score(q,d) = coordfactor(q,d) \cdot querynorm(q) \cdot \sum_{t \in q} (tf\ (t \in d) \cdot idf(t)^2 \cdot boost(t) \cdot norm(t,d))$$

  - ✓ *d è il documento;*
  - ✓ *q è l'interrogazione;*
  - ✓ *$tf(t \in d)$ è la frequenza del termine nel documento;*
  - ✓ *$idf(t)$ è la frequenza inversa del documento;*
  - ✓ *$coord(q,d)$ è un fattore basato su quanti dei termini della query sono trovati nel documento d.*
  - ✓ *$queryNorm(q)$ è un fattore di normalizzazione.*
  - ✓ *$boost(t)$ è un fattore che pesa al momento della ricerca il termine t.*
  - ✓ *$norm(t,d)$ è il prodotto del moltiplicatore del singolo documento, che viene impostato al momento dell'inserimento nell'indice, il moltiplicatore del termine, che viene impostato prima di aggiungere il termine al documento e la lunghezza (in token) del campo dove compare il termine all'interno del documento.*

$$proximity_{dynamic}(A,B) = \frac{LuceneScore(doc_A, doc_B)}{\max_{X \in queryResult} LuceneScore(doc_A, doc_X)}$$

- *La metrica introdotta per il calcolo della prossimità in base alle preferenze viene normalizzata rispetto al massimo dei punteggi ottenuti mediante la query che restituisce i documenti relativi agli utenti simili.*
  - ✓ *Questo perché deve essere confrontabile con i valori di prossimità statica ottenuti tramite la vecchia procedura riadattata*

Modello di calcolo

$$proximity_{static}(A,B) = F(v_{language}(A,B), v_{location}(A,B), v_{friends}(A,B), v_{groups}(A,B), v_{age}(A,B), v_{taxonomy}(A,B))$$

- *La vecchia procedura è stata privata delle metriche per il calcolo della prossimità dinamica.*
  - ✓ *Adesso si ha una funzione che stima la similarità esclusivamente sulla base delle informazioni statiche ed attribuisce ad ogni metrica un peso P_i, questa si va a combinare con la metrica per la prossimità dinamica per il calcolo della prossimità tra due utenti.*

$$proximity(A,B) = proximity_{dynamic}(A,B) \times \gamma_d + proximity_{static}(A,B) \times \gamma_s$$

- *È possibile assegnare pesi diversi alle diverse prossimità in modo da dare più rilevanza ai dati relativi alle preferenze o ai profili statici.*
  - ✓ *Talvolta accade che i profili statici siano meno accurati perché gli utenti tendono a non riempirli. In questo caso la prossimità dinamica è usata come supporto.*

- *L'offerta dei suggerimenti è stata ampliata includendo la tipologia di raccomandazioni strategiche e la tipologia serendipity.*
  - ✓ ***Raccomandazioni strategiche**: agli utenti che hanno pochi colleghi vengono suggeriti utenti con molti colleghi e viceversa. Questo per cercare di aiutare i nuovi utenti a socializzare e per "recuperare" gli utenti che non accedono al sito da molto tempo e hanno perso contatti con i nuovi utenti e/o interesse nei confronti del portale.*
  - ✓ ***Raccomandazioni serendipity**: suggerendo un amico in maniera casuale (magari con interessi completamente diversi dai propri), l'utente, spinto dalla curiosità di nuovi contenuti, può creare contatti con il nuovo amico, ampliando la lista dei suoi interessi.*

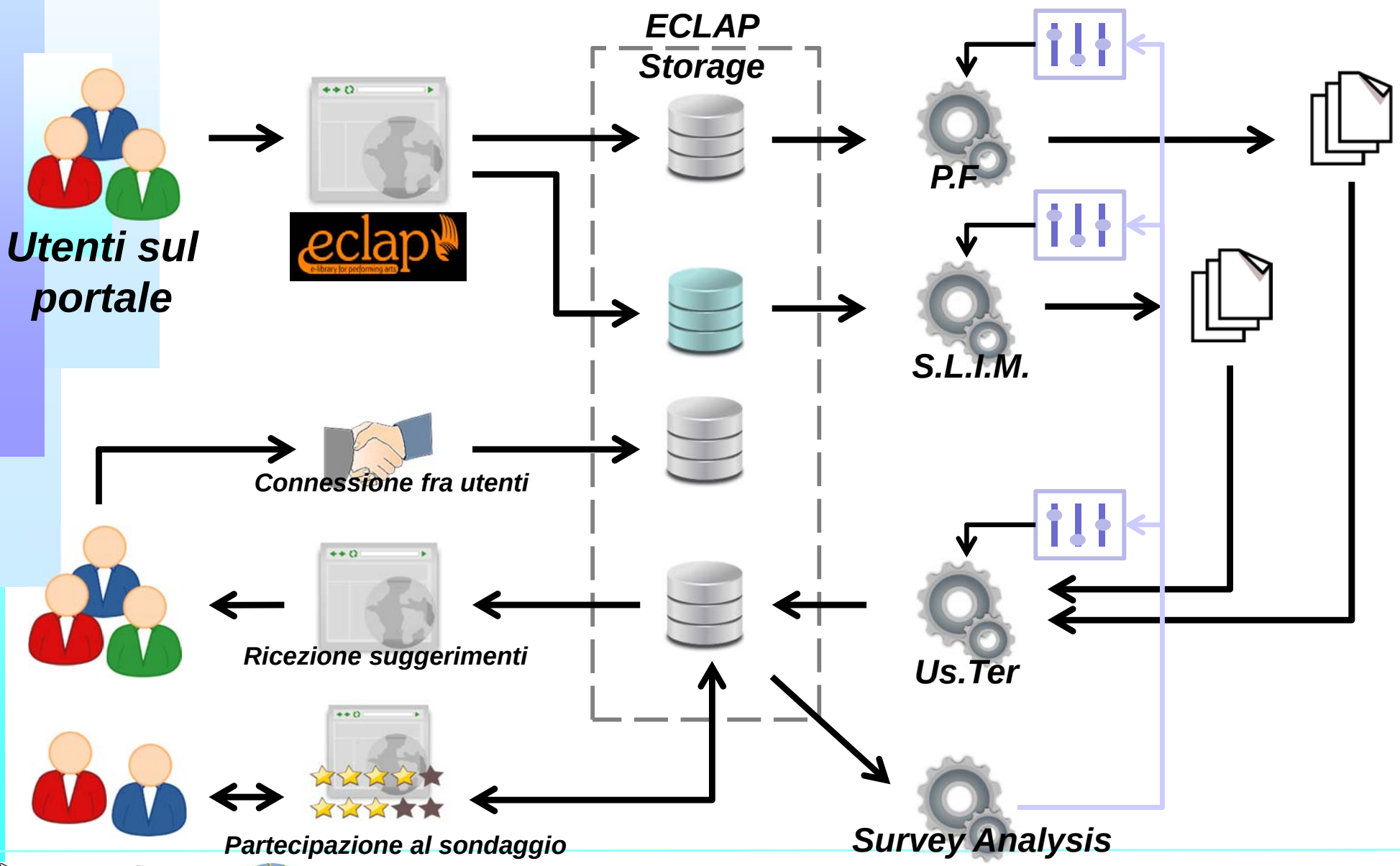
Architettura del sistema

- *Il calcolo della prossimità viene effettuato tramite delle procedure JAVASCRIPT che sfruttano l'architettura grid dell'ECLAP BackOffice.*
  - ✓ *Due procedure valutano rispettivamente la prossimità statica e dinamica degli utenti e generando delle matrici di prossimità e l'archivio con i documenti inerenti le preferenze degli utenti.*
  - ✓ *Una terza procedura utilizza i risultati delle due sopra per generare le raccomandazioni secondo tutte le politiche di suggerimento implementate e le memorizza sul database tracciando ogni dato relativo all'elaborazione*

- *È possibile elaborare i profili o le raccomandazioni di gruppi di utenti in parallelo*
  - ✓ *Ogni istanza di esecuzione delle procedure determina su quali utenti può lavorare e li «prenota» in maniera transazionale in modo che non ci siano altre istanze che trattano gli stessi utenti.*
  - ✓ *Le procedure sono in grado di risolvere problemi derivanti dal crash di un nodo elaborativo del grid, che porterebbero gli utenti prenotati dalle istanze non più attive a non essere mai processati a causa delle prenotazioni.*
  - ✓ *L'elaborazione è una attività pianificata per essere eseguita ad intervalli di tempo regolari ma le procedure sono predisposte anche per essere istanziate on-demand.*

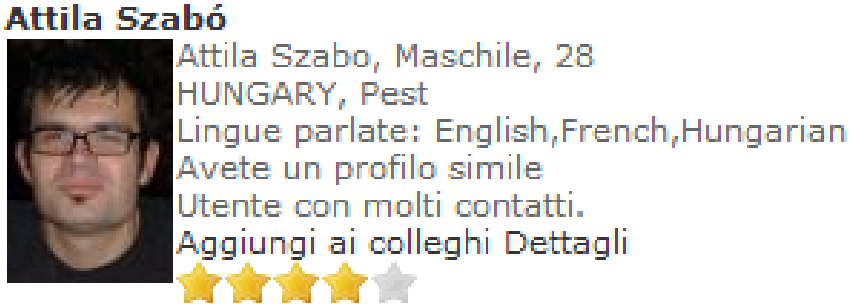
- *Il flusso di esecuzione di alcune procedure JAVASCRIPT si sposta su programmi scritti in JAVA per l'interfacciamento con il motore di Lucene.*
  - ✓ *I programmi JAVA lavorano puntualmente, senza concorrere sugli stessi dati, e permettono alle procedure JAVASCRIPT il rilevamento di eventuali errori al momento dell'esecuzione e la notifica tramite email.*

Architettura del sistema



La validazione

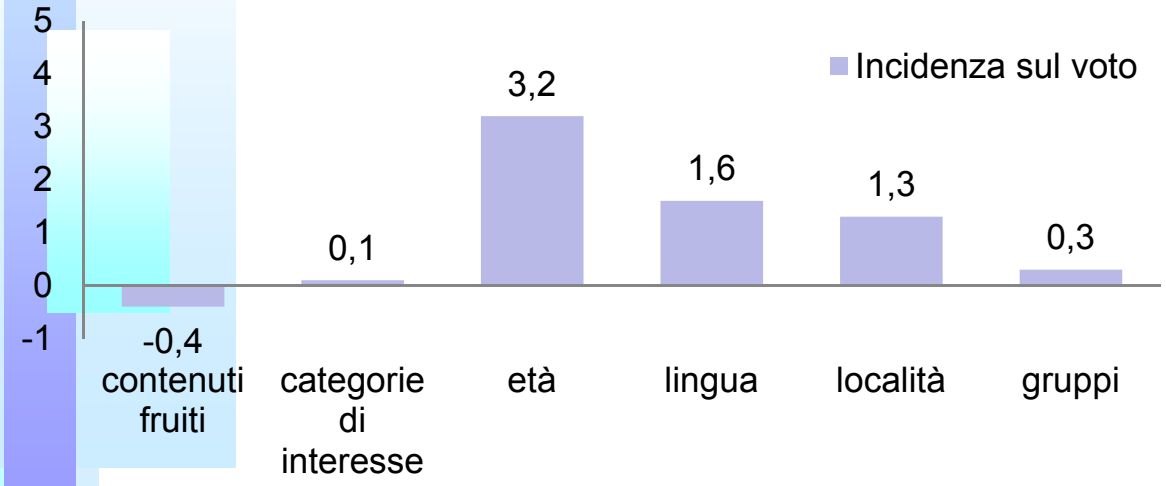
- I suggerimenti proposti nel sondaggio sono un sottoinsieme di quelli elaborati dal sistema e presentano una serie di informazioni relative agli utenti.
  - ✓ Viene chiesto di votare quanto un suggerimento è ritenuto interessante.
  - ✓ In questo modo non si valida la qualità delle metriche ma l'efficacia che hanno i suggerimenti sulla base delle informazioni che vengono fornite.



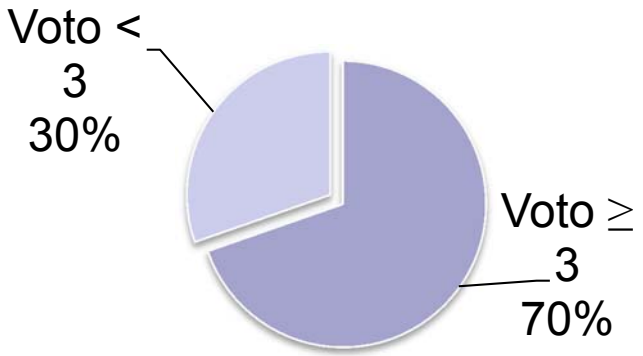
- *I parametri relativi alla generazione di una raccomandazione sono tutti tracciati e tramite i valori delle metriche e il voto lasciato dagli utenti si stima l'efficacia che ha mostrare un dettaglio relativo ad un utente o meno.*
  - ✓ *Questo è possibile solo perché i dati sulla similarità sono stati calcolati tramite le procedure precedenti.*
- *È possibile indagare quali sono le tipologie di raccomandazione che vengono gradite maggiormente.*
- *L'analisi dei dati viene effettuata tramite regressione multilineare per ottenere un modello nella forma*

$$y(A,B) = v_{language}(A,B) \cdot P_{language} \cdot \gamma_{language} + v_{location}(A,B) \cdot P_{location} \cdot \gamma_{location} + v_{friends}(A,B) \cdot P_{friends} \cdot \gamma_{friends} + v_{groups}(A,B) \cdot P_{groups} \cdot \gamma_{groups} + v_{age}(A,B) \cdot P_{age} \cdot \gamma_{age} + v_{taxonomy}(A,B) \cdot P_{taxonomy} \cdot \gamma_{taxonomy} + proximity_{dynamic}(A,B) \cdot P_{proximity_{dynamic}} \cdot \gamma_{proximity_{dynamic}}$$

La validazione

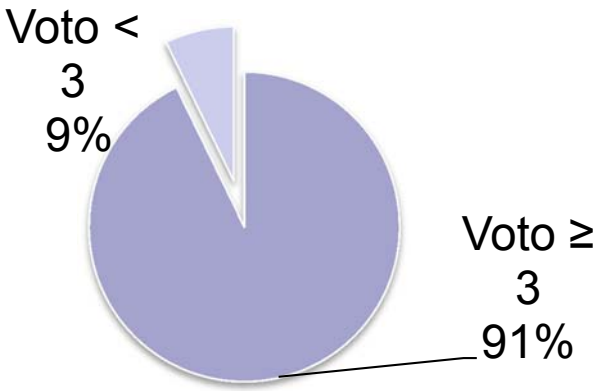


Statistica della regressione	
R multiplo	0, 9624
F - Value	131,7795
Significatività di F	2,3389E-33



Tipologia Serendipity

- ✓ *Competenze*
- ✓ *Gruppi di appartenza*



Tipologia Strategici

- ✓ *Popolarità*

Sunto

- *È stato ripreso e migliorato il sistema di raccomandazioni utente-utente interno al social network ECLAP negli aspetti di*
  - ✓ *Scalabilità*
  - ✓ *Flessibilità*
  - ✓ *Performance*
  - ✓ *Offerta dei suggerimenti (nuove tipologie)*
- *È stato implementato un meccanismo per la validazione dell'efficacia delle informazioni mostrate a fianco delle raccomandazioni che prevede la partecipazione degli utenti ad un sondaggio.*
- *Dall'analisi delle risposte degli utenti alla domanda «Quanto trovi interessante questo suggerimento?», corredata dalle informazioni sul profilo dell'utente suggerito, è emerso che le principali motivazioni che spingono l'utente a stringere amicizia sono l'età, la lingua, la località, la popolarità e la competenza.*
- *Tramite la nuova rappresentazione delle preferenze degli utenti è possibile migliorare la proposta di altri tipi di raccomandazioni, come ad esempio quelle relative ai contenuti.*