



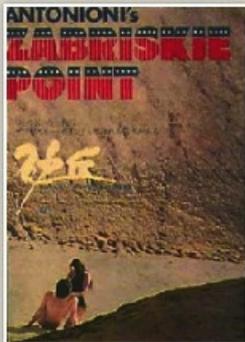
A lightweight model for publishing and sharing Linked Web APIs

Devis Bianchini, Valeria De Antonellis, *Michele Melchiori*
DII –Università di Brescia

**20th Italian Symposium on Advanced Database
Systems (SEBD)**

Talk Outline

- An example
- Linked mashup vision and its conceptual model
- Linking Web APIs
- The I-Mash platform
- Conclusions and future work



Zabriskie Point

Drama, Romance · 1970

An epic portrait of late Sixties America, as seen through the portrayal of two of its children: anthropology student Daria (who's helping a property developer build a village in the Los Angeles desert) and dropout Mark (who's wanted by the authorities for allegedly killing a policeman during a student riot)...

Runtime 1 hour 50 mins
Released 9th February, 1970
Rating R
Director Michelangelo Antonioni
Writer Michelangelo Antonioni

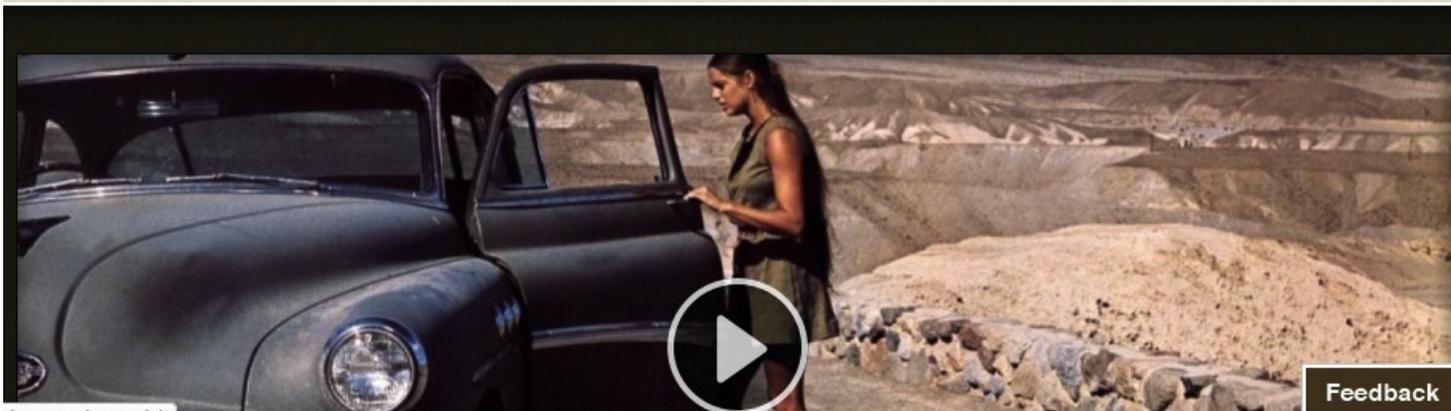
Rating: 65/100

IMDb 6.9   61

Share:

<http://moviegr.am/5560>

0  0 
 Tweet  Like  +1  Submit



The Web designer has to find and integrate Web APIs that:

- Provide information on movies (e.g., Rottentomatoes, `http://api.rottentomatoes.com/api/public/v1.0`)
- Show videos and trailers (e.g., YouTube, `https://developers.google.com/youtube/`)

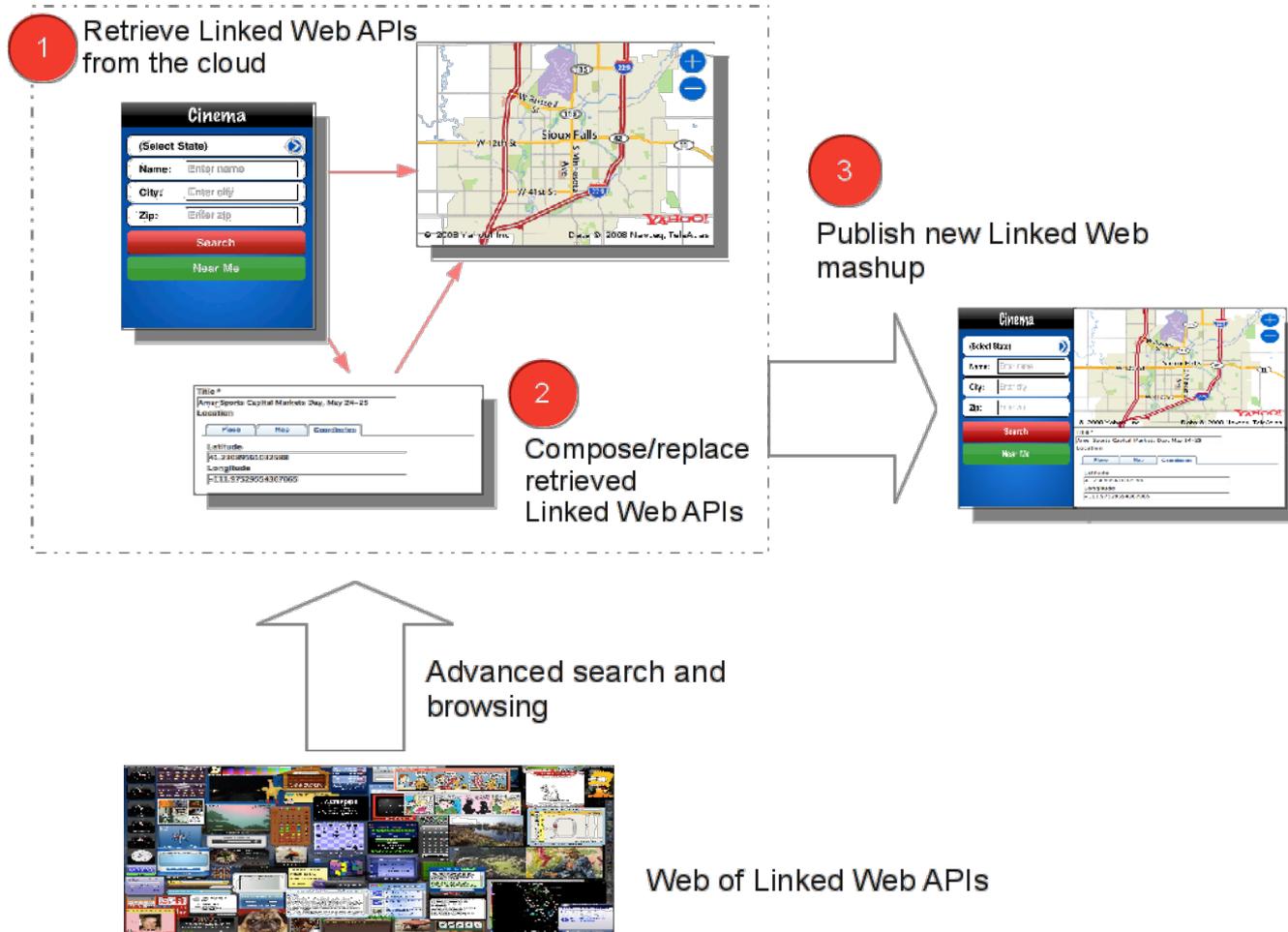
Problems and proposals

- High number of available Web API and heterogeneous documentation makes their selection/integration difficult and time consuming
- We propose a **lightweight model** and architecture for annotating, publishing and sharing Web APIs on the Web of Linked Data
- The purpose is to give support to the Web APIs **producer/consumer** for:
 - **publication** of Web APIs
 - **proactive and interactive selection** of Web APIs and for their integration/composition
 - **reuse** the experience of other developers

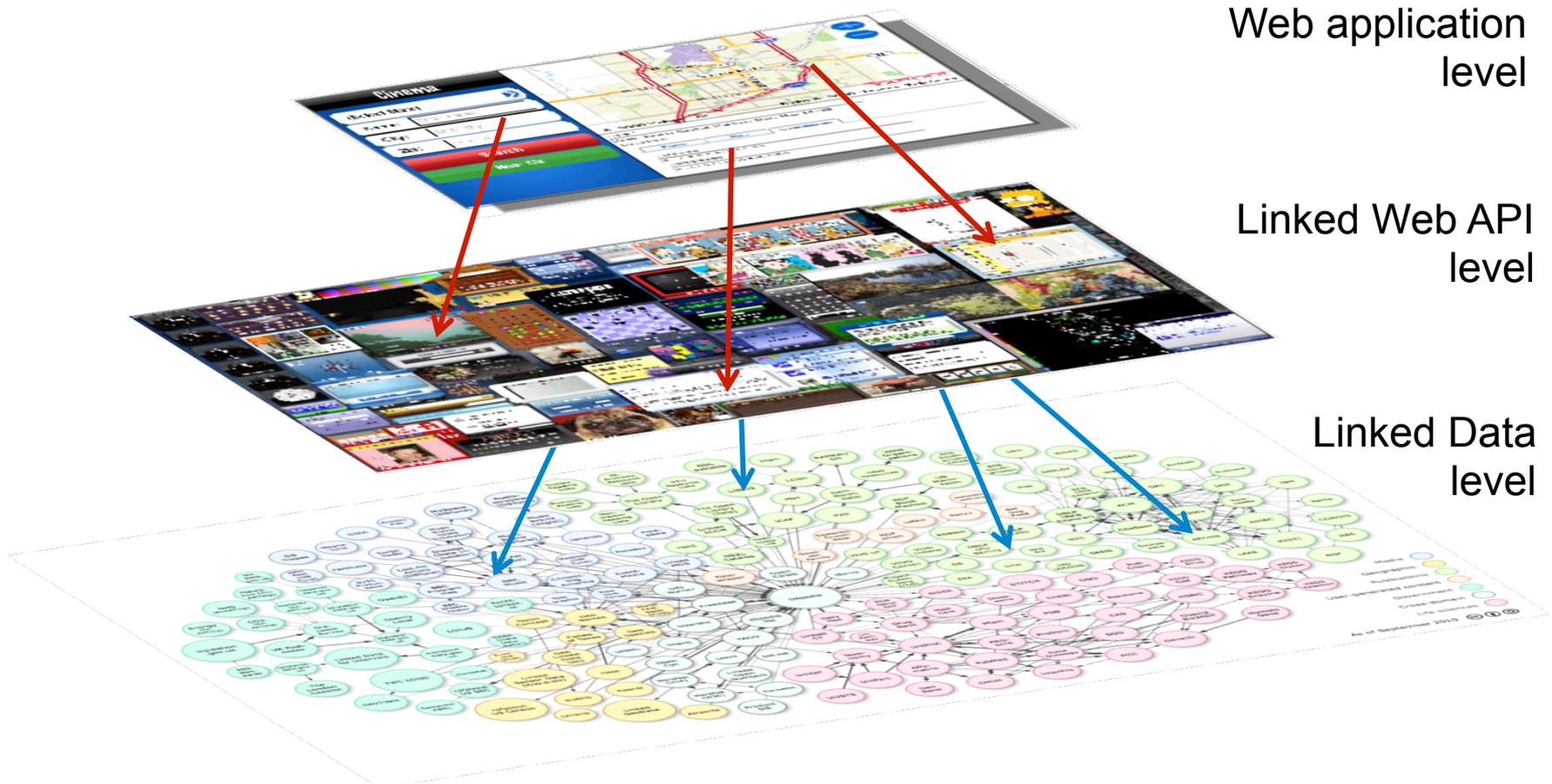
General principles

- The model adheres to the Linked Data principles for publishing resources on the Web in a meaningful way:
 - HTTP URIs are used to identify published Web APIs,
 - useful information (RDF) are provided on the Web API description when someone looks up a Web API through its URI
 - proper links are set to relate Web APIs to each other, thus enabling support to development and sharing of new Web applications

The Linked Web mashup vision

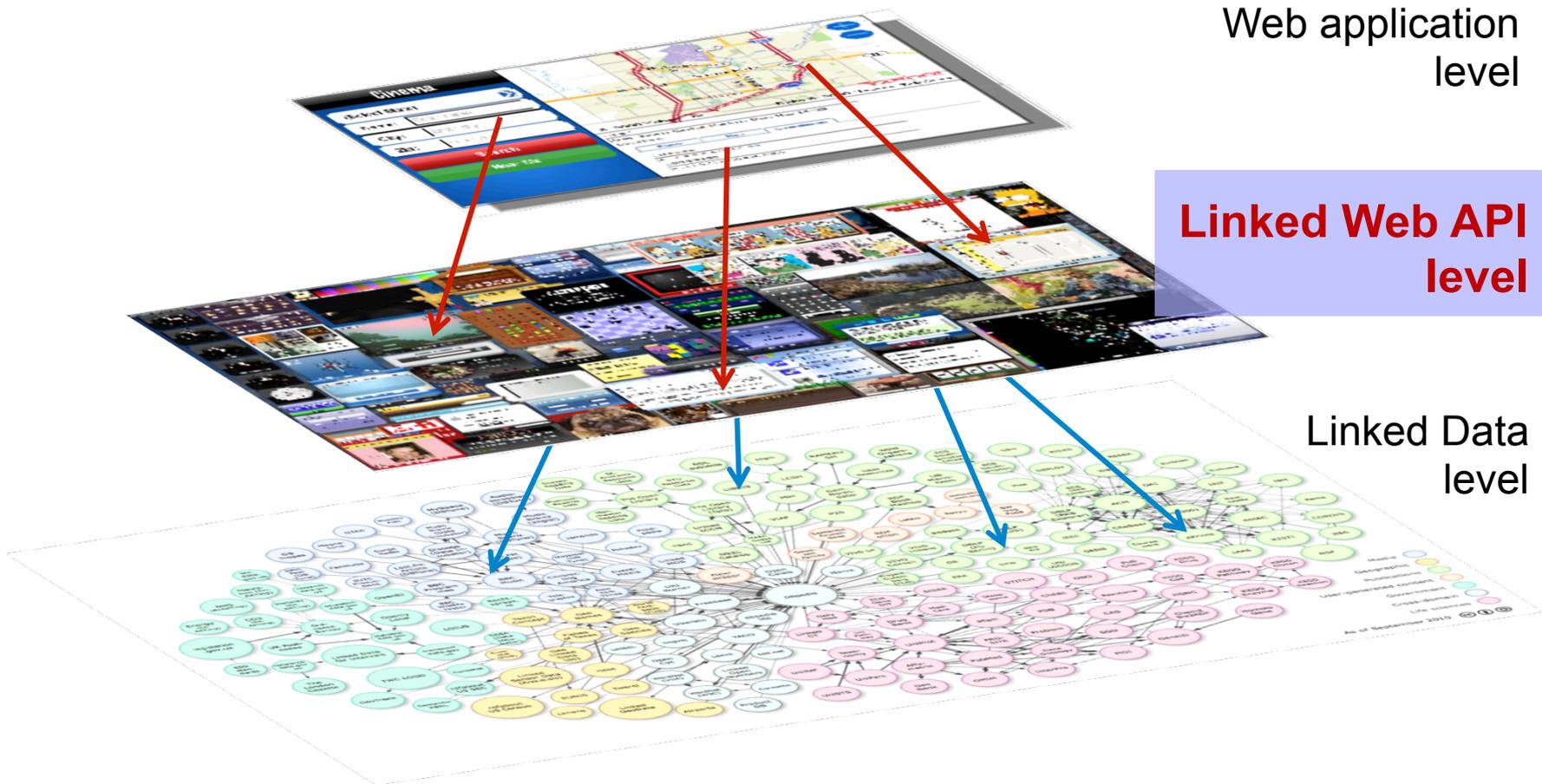


Linked Web API semantic annotation



- M. Maleshkova, C. Pedrinaci, J. Domingue: Semantic Annotation of Web APIs with SWEET. SFSW 2010

Linked Web API semantic annotation

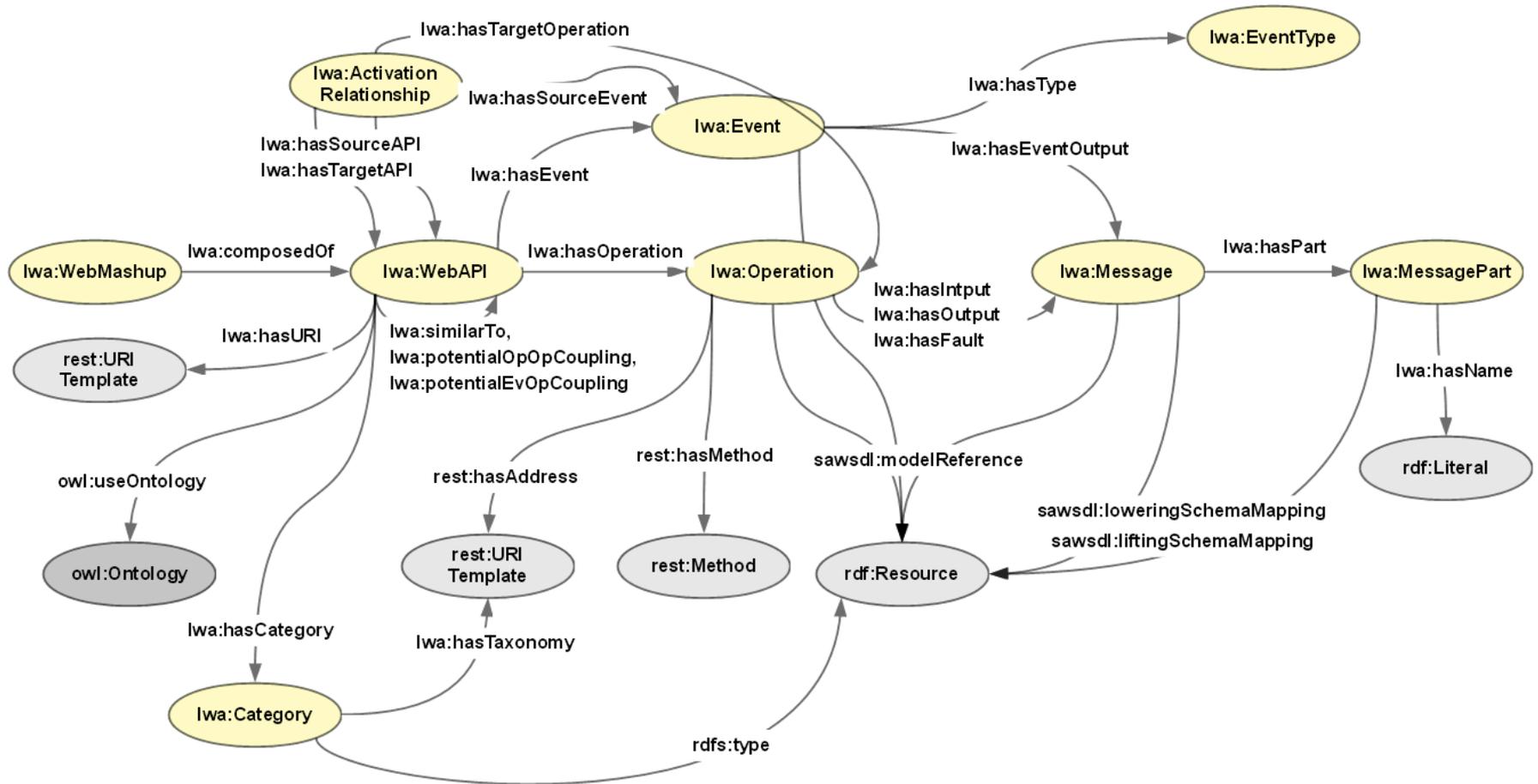


Web application level

Linked Web API level

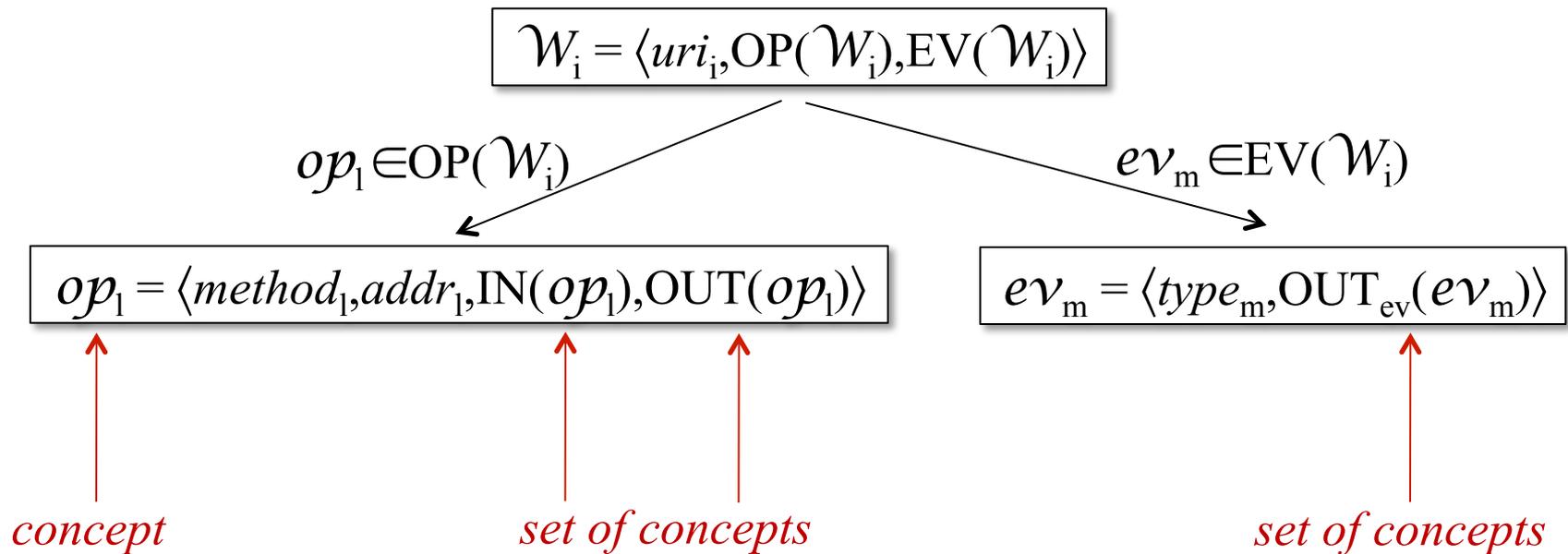
Linked Data level

Linked Web API lightweight model



Linked Web API descriptor

- An abstract representation (API descriptor):



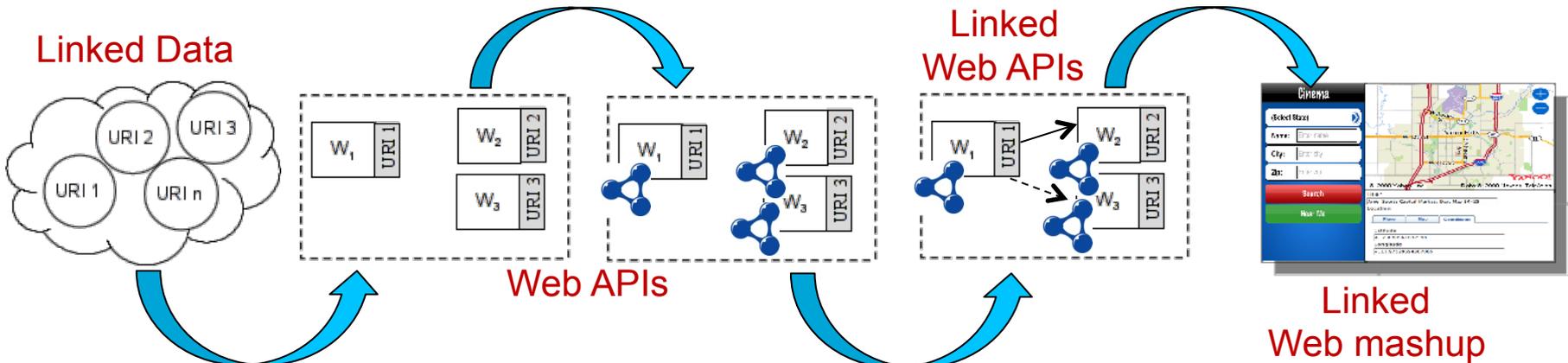
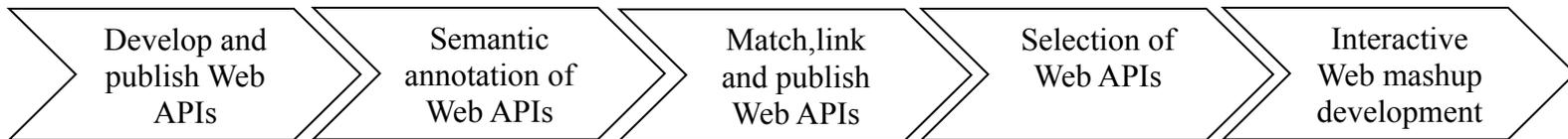
The Linked Web APIs lifecycle



Web API provider



Web mashup designer



Linked Web API publication

Publication of a new semantically annotated Web API

- the Web API provider has to identify related Web APIs that could be linked to the current one through **similarTo** or **potentialCoupling** constructs
 - semi-automatic support-

Publication of a new Web mashup

- the Web designer sets **ActivationRelationships** between Web APIs forming the mashup, so making explicit composition information

Functional similarity between Web APIs

- A Web API descriptor W_i is *functionally similar* to another Web API descriptor W_j if:
 1. W_i and W_j categories are **compatible**
 2. $\text{Sim}_{IO}(W_i, W_j) \geq \delta \in [0..1]$ (**functional similarity degree**)

$$\text{Sim}_{IO}(\mathcal{W}_i, \mathcal{W}_j) = \frac{1}{3} \left[\frac{\sum_{s,t} \text{CAff}(in_s, in_t)}{|IN(\mathcal{W}_j)|} + \frac{\sum_{h,k} \text{CAff}(out_h, out_k)}{|OUT(\mathcal{W}_i)|} + \frac{\sum_{l,m} \text{CAff}(op_l, op_m)}{|OP(\mathcal{W}_i)|} \right]$$

Union of the input sets
of all the W_j operations

Union of the output sets
of all the W_i operations

Concept affinity

Concept affinity evaluation

- Used to evaluate the “semantic affinity” between concepts regardless the ontologies/vocabularies they belong to
- Hybrid approach:
 - a domain-dependent matching based on the ontologies on the Web of Data (**intra-ontology concept affinity**)
 - a terminological (domain-independent) matching based on WordNet (**inter-ontology concept affinity**)

Functional coupling between Web APIs

- A Web API descriptor W_i can be *functionally coupled* with another Web API descriptor W_j if:
 - $Coupl_{IO}(W_i, W_j) \geq \theta \in [0..1]$ (**functional coupling degree**)

$$Coupl_{EvOp}(ev_i, op_j) = \frac{\sum_{h,k} CAff(out_h, in_k)}{|OUT_{ev}(ev_i)|}$$

Union of the output sets
of all the W_i events

$$Coupl_{IO}(\mathcal{W}_i, \mathcal{W}_j) = \frac{\sum_{i,j} Coupl_{EvOp}(ev_i, op_j)}{|EV(\mathcal{W}_i)|}$$

Use cases for Linked Web API mashup (I)

Building a Web mashup from scratch – The Web designer proceeds step by step by selecting a Web API and mapping events with operations

1. The designer specifies **a requested Web API** in terms of categories, operations and/or inputs/outputs
2. The system suggests APIs that **are coupled** with Web APIs already included in the Web mashup under construction

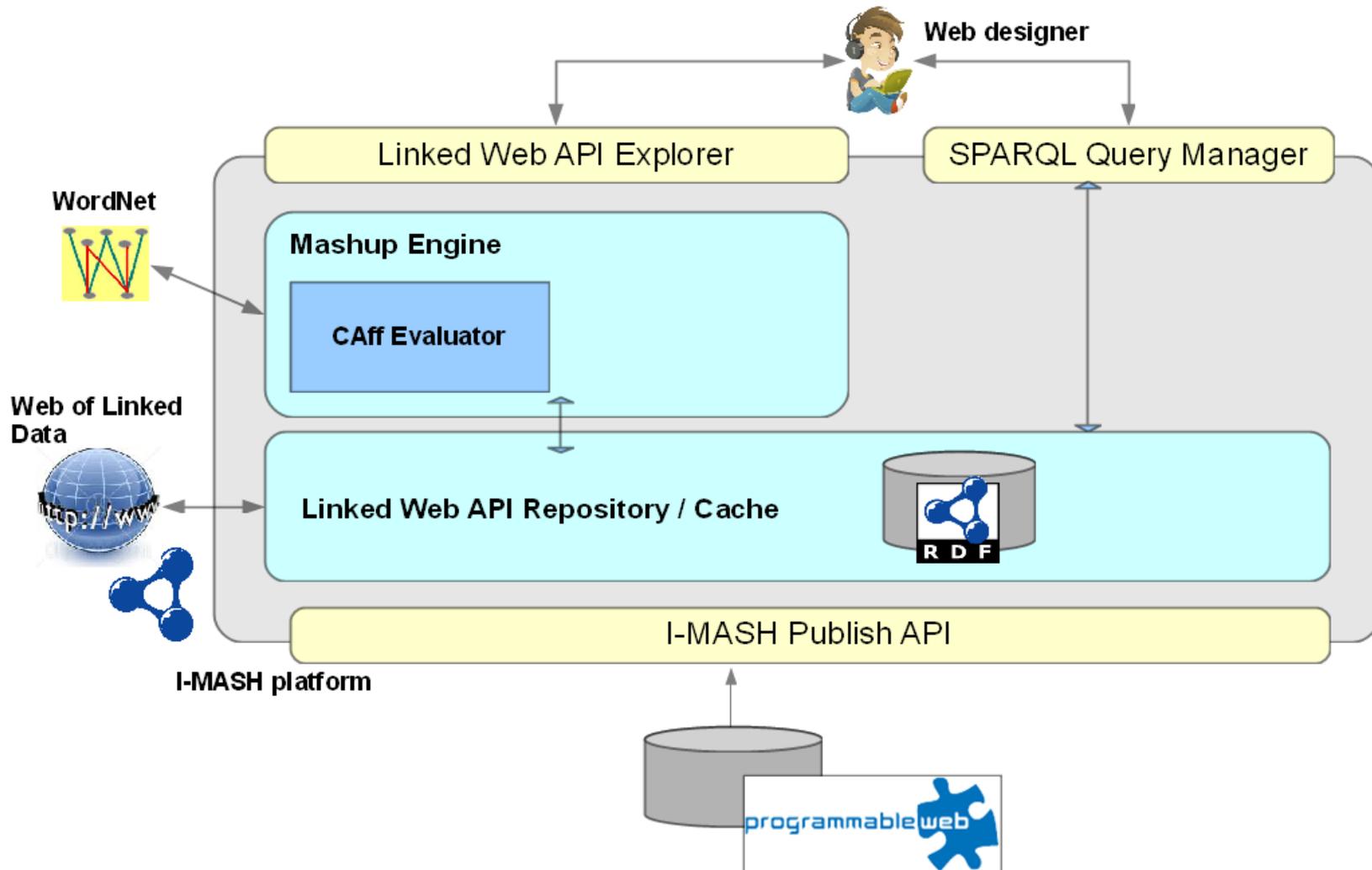
Use cases for Linked Web API mashup (II)

Modifying an existing mashup – The Web designer considers an existing Web mashup on the Web of Data and needs to replace one of the component Web APIs, since

1. they became unavailable
2. application requirements have been changed
3. the designer needs to improve the QoS of the Web mashup

The system suggests APIs that **can substitute** Web APIs already included in the Web mashup under construction

The I-MASH platform



Conclusions and future work

We proposed a lightweight model based on linked data to support both:

- Providers who want to publish Web APIs on the Web of Linked Data
- Designers who aim at exploring/selecting Web APIs for implementing/maintaining Web mashups

Future work:

- Consider other types of relationships (e.g., complementary Web API)
- Evaluate relationships also based on use of Web APIs in existing mashups and on nonfunctional features

A black and white photograph of a hand-drawn message. The words "Thank you" are written in a cursive, handwritten style on a white surface. A grey marker is visible in the upper right corner, with its tip touching the end of the word "you". The background is plain white.

Sweet-style annotation

```
<div class="service" id="svc">
<p>Description of the
  <span class="label">ACME Hotels</span> service:</p>
<div class="operation" id="op1"><p>
  The operation <code class="label">getHotelDetails</code> is
  invoked using the method <span class="method">GET</span>
  at <code class="address">http://example.com/h/fidg</code>,
  with <span class="input">the ID of the particular hotel
  replacing the parameter <code>id</code></span>.
  It returns <span class="output">the hotel details in an
  <code>ex:hotelInformation</code> document.</span>
</p></div></div>
```

Example hRESTS description

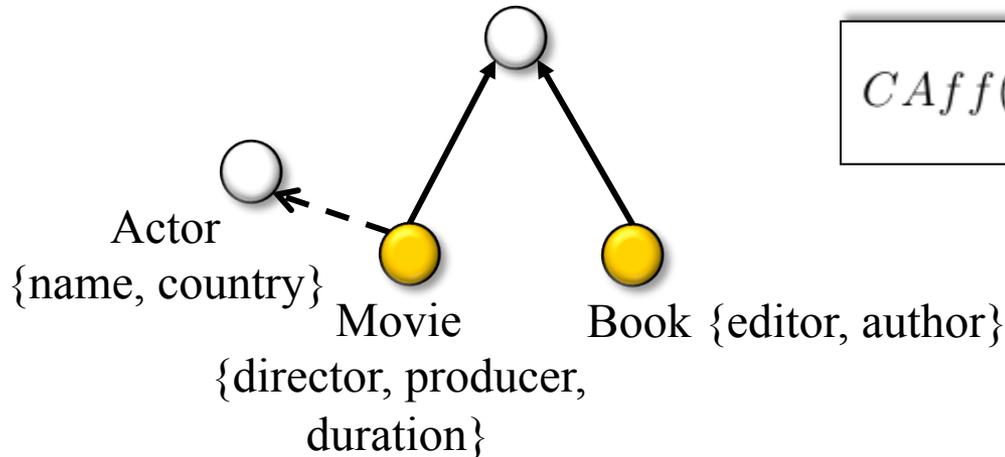
Intra-ontology concept affinity evaluation

Properties of c_2 concept

$$CAff(c_1, c_2) = \begin{cases} 1 & \text{if } c_1 \equiv c_2 \\ 1 & \text{if } c_1 \sqsubseteq c_2 \\ \frac{|\mathcal{P}(c_2)|}{|\mathcal{P}(c_1)|} & \text{if } c_2 \sqsubseteq c_1 \\ \frac{2 \cdot |\mathcal{P}(c_3)|}{|\mathcal{P}(c_1)| + |\mathcal{P}(c_2)|} & \text{if } c_3 \equiv LCA(c_1, c_2) \\ 0 & \text{otherwise} \end{cases}$$

Lowest Common Ancestor of c_1 and c_2

EditorialProduct {title, plot, year}

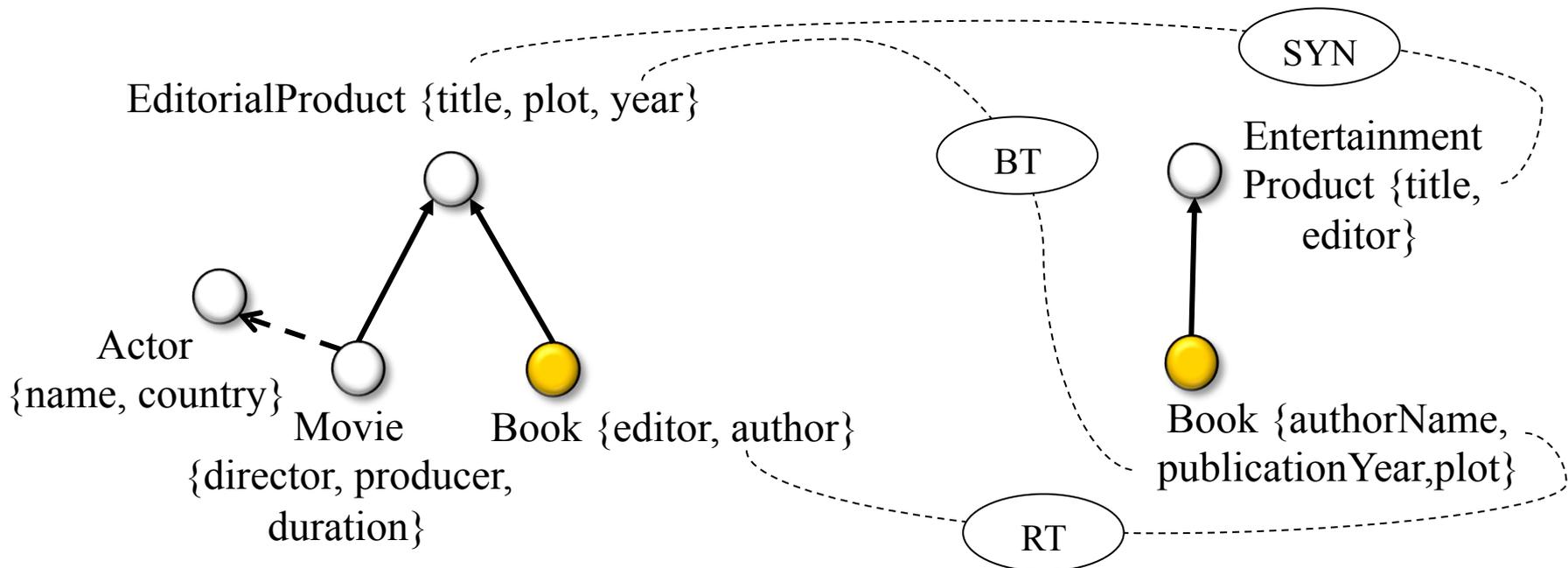


$$CAff(Movie, Book) = \frac{2 \cdot 3}{6 + 5} = 0.545$$

Inter-ontology concept affinity evaluation

Name Affinity

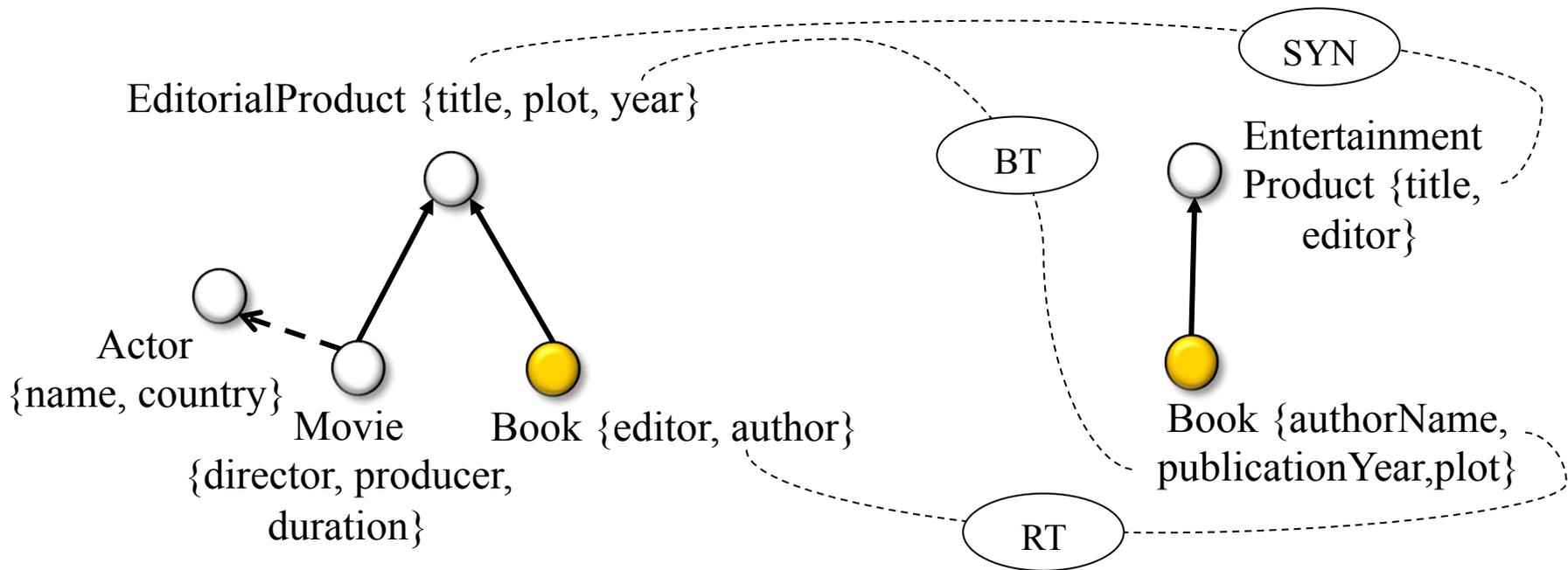
$$CAff(c_1, c_2) = \frac{1}{2} \cdot \left[NAff(n_1, n_2) + \frac{\sum_{p_1, p_2} NAff(p_1, p_2)}{|\mathcal{P}(c_1)|} \right] \in [0, 1]$$



Inter-ontology concept affinity evaluation

$$CAff(\mathcal{O}_1.Book, \mathcal{O}_2.Book) = \frac{1}{2} \cdot \left[NAff(\mathcal{O}_1.Book, \mathcal{O}_2.Book) + \frac{NAff(\mathcal{O}_1.Title, \mathcal{O}_2.Title) + \dots}{5} \right]$$

$$= \frac{1}{2} \cdot \left[1.0 + \frac{1.0 + 1.0 + 0.8 + 1.0 + 0.5}{5} \right] = 0.93$$



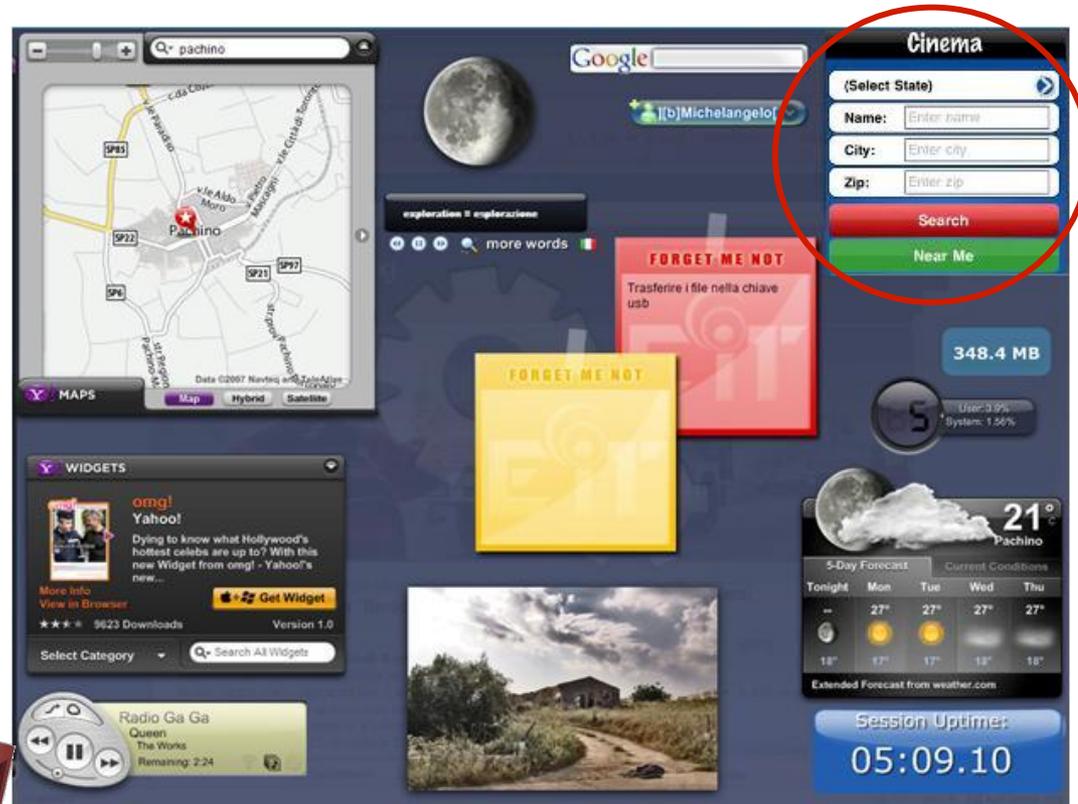
Search selection pattern

I need a Web API to find movie theatres



Search selection pattern

I need a Web API to find movie theatres



Completion selection pattern

What functionalities could be added to my mashup?



Cinema

(Select State) 

Name:

City:

Zip:

Search

Near Me



Completion selection pattern

What functionalities could be added to my mashup?



Cinema

(Select State) 

Name:

City:

Zip:

Search

Near Me



Substitution selection pattern

I need a different Web API to convert coordinates



Cinema

(Select State)

Name:

City:

Zip:

Search

Near Me



Township Range Converter

State:

Principal Meridian:

Township: Fraction: Direction:

Range: Section:

Section Division:

Your Output Information

Output Longitude (Decimal Degrees): -106.54414

Output Latitude (Decimal Degrees): 40.68385

Output Longitude (COMBSS): -106° 32' 38.8924"

Output Latitude (COMBSS): 40° 41' 1.8611"

View on L25S map viewer: [Zoom](#)

Substitution selection pattern

I need a different Web API to convert coordinates



Cinema

(Select State) 

Name:

City:

Zip:

Search

Near Me



Title *

Amer Sports Capital Markets Day, May 24-25

Location

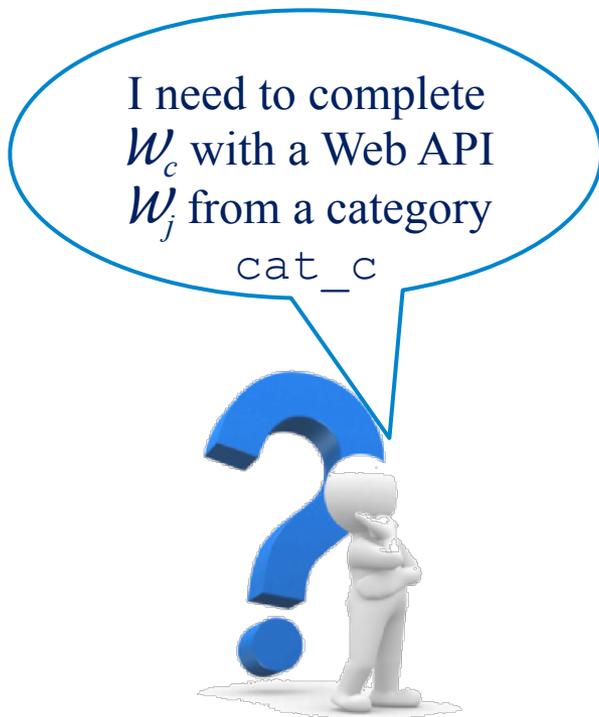
Latitude

41.23089561032588

Longitude

-111.97529554367065

Completion selection pattern



Cinema

(Select State)

Name:

City:

Zip:

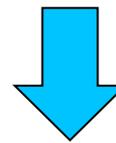
Search

Near Me

\mathcal{W}_c

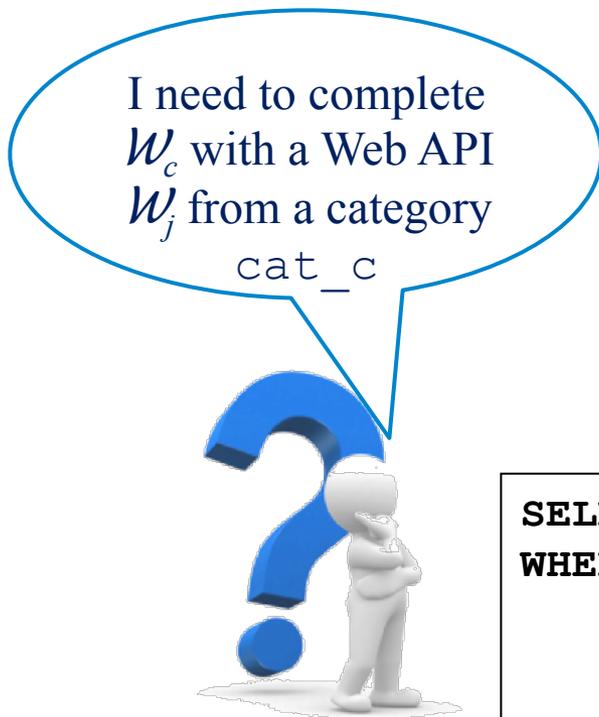


\mathcal{W}_j



$$m_{\tau} = \begin{cases} Coupl_{IO}(\mathcal{W}_c, \mathcal{W}_j) & \text{if } \mathcal{W}_j \in Q^1(\mathcal{W}_j) \\ 0 & \text{otherwise} \end{cases}$$

Completion selection pattern



Cinema

(Select State)

Name:

City:

Zip:

Search

Near Me

W_c



W_j

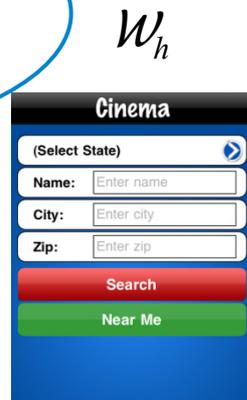


```

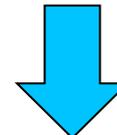
SELECT ?uri
WHERE {
    ?api lwa:hasURI ?uri ;
        lwa:hasCategory ?cat .
    Wc_uri lwa:potentialCouplingWith ?api .
    {
        {?cat owl:equivalentTo cat_c .} UNION
        {cat_c rdfs:subClassOf ?cat .}
    }
}
    
```

Q^I

Substitution selection pattern



W_j



$$m_{\tau} = \begin{cases} Sim_{IO}(\mathcal{W}_s, \mathcal{W}_j) + \sum_k Coupl_{IO}(\mathcal{W}_j, \mathcal{W}_k) + \sum_h Coupl_{IO}(\mathcal{W}_h, \mathcal{W}_j) & \text{if } \mathcal{W}_j \in Q^2(\mathcal{W}_j) \\ 0 & \text{otherwise} \end{cases}$$

Substitution selection pattern

I need to substitute a
Web API W in a
category
a new W

```

SELECT ?uri
WHERE {
    ?api lwa:hasURI ?uri ;
        lwa:hasCategory ?cat ;
        lwa:similarTo Ws_uri ;
        lwa:potentialCouplingWith ?Wk_uri .
    ...
    Wh_uri lwa:potentialCouplingWith ?api .
    ...
    Ws_uri lwa:hasCategory ?cat_s .
    {
        {?cat owl:equivalentTo cat_c .} UNION
        {cat_c rdfs:subClassOf ?cat .}
    }
}
    
```

Q^2

W_k



$$m_{\tau} = \begin{cases} \text{Sim} \\ 0 \end{cases}$$

$W_j \in Q^2(W_j)$
otherwise

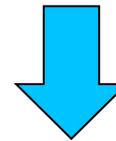
Search selection pattern

I need a Web API \mathcal{W}_r from a category $CAT(\mathcal{W}_r)=cat_r$



$$\mathcal{W}_r = \langle CAT(\mathcal{W}_r), opt(\mathcal{W}_r) \rangle$$

$$opt(\mathcal{W}_r) = \langle OP(\mathcal{W}_r), IN(\mathcal{W}_r), OUT(\mathcal{W}_r) \rangle$$



$$m_{\tau} = \begin{cases} Sim_{IO}(\mathcal{W}_r, \mathcal{W}_j) & \text{if } \mathcal{W}_j \in Q^3(\mathcal{W}_j) \text{ AND } opt(\mathcal{W}_r) \neq \langle \emptyset, \emptyset, \emptyset \rangle \\ 1 & \text{if } \mathcal{W}_j \in Q^3(\mathcal{W}_j) \text{ AND } opt(\mathcal{W}_r) = \langle \emptyset, \emptyset, \emptyset \rangle \\ 0 & \text{otherwise} \end{cases}$$



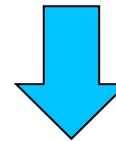
Search selection pattern

I need a Web API \mathcal{W}_r from a category $CAT(\mathcal{W}_r)=cat_r$



$$\mathcal{W}_r = \langle CAT(\mathcal{W}_r), opt(\mathcal{W}_r) \rangle$$

$$opt(\mathcal{W}_r) = \langle OP(\mathcal{W}_r), IN(\mathcal{W}_r), OUT(\mathcal{W}_r) \rangle$$



$\{ Sim_{IO}(\mathcal{W}_r, \mathcal{W}_j) \text{ if } \mathcal{W}_j \in Q^3(\mathcal{W}_r) \text{ AND } opt(\mathcal{W}_r) \neq \langle \emptyset, \emptyset, \emptyset \rangle$

```
SELECT ?uri  $Q^3$ 
WHERE {
  ?api lwa:hasURI ?uri ;
        lwa:hasCategory ?cat .
  {
    {?cat owl:equivalentTo cat_r .} UNION
    {cat_r rdfs:subClassOf ?cat .}
  }
}
```



Exploring Linked Web APIs

Search for Web APIs
Category (required): Operations: Inputs: Outputs:

Substitution selection pattern

```
graph LR; A((MOVIESANDTHEATRES)) ---|0.897| B((FINDMOVIETHEATRES)); B ---|0.625| C((COMINGSOON))
```

InternetVideoArchive
selectedMovie selected Review
findMovie findReview

GoogleMaps
selected Coordinates
center show

FindMovieTheatres
selected Theatres
showFilms showTheatres

Completion selection pattern

Highweight model for partitioning and sharing Linked Web APIs

Exploring Linked Web APIs

Search for Web APIs
Category (required): Operations: Inputs: Outputs:

Substitution selection pattern Add activation:

Completion selection pattern

```
graph LR; A{{MAPVIEWER}} ---|0.734| B{{FINDMOVIETHEATRES}}; B ---|0.8165| C{{GOOGLEMAPS}}
```

InternetVideoArchive

selectedMovie	selected Review
findMovie	findReview

FindMovieTheatres

selected Theatres	
showFilms	showTheatres

InternetVideoArchive - FindMovieTheatres

- selectedMovie → showFilms
- selectedMovie → showTheatres
- selectedReview → showFilms
- selectedReview → showTheatres



A lightweight model for publishing and sharing Linked Web APIs

Devis Bianchini, Valeria De Antonellis, *Michele Melchiori*
DII –Università di Brescia

PROBLEM: High number of available Web API (abt 6500 on programmableWeb.com) and heterogeneous documentation makes it difficult their selection/integration to define applications (Web mashups)

- We propose a **lightweight model** for semantic annotating, publishing and sharing Web APIs on the Web of Linked Data
- The purpose is to give support to the Web APIs **producer/consumer** for:
 - **publication** of services and Web APIs
 - **proactive and interactive selection** of Web APIs and their integration/composition
 - **reuse** the experience of other developers

The Linked Web mashup vision

