Ontology-based Generation of Personalised Data Management Systems: an Application to Experimental Particle Physics

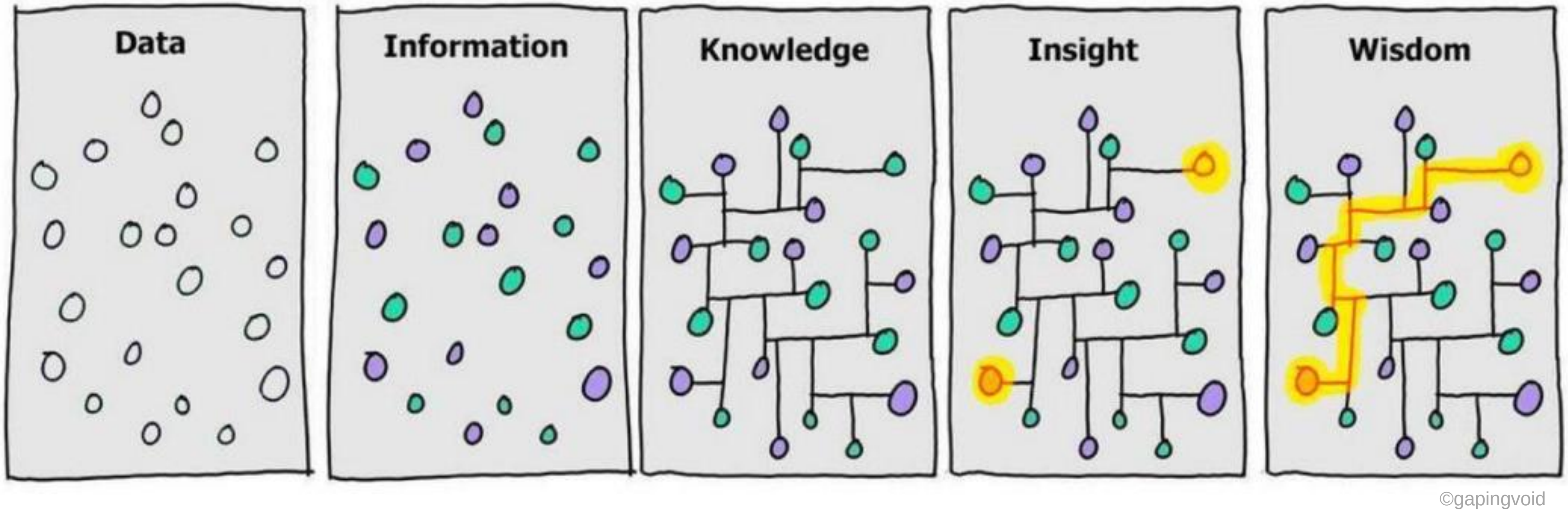
PhD thesis defense of Blerina GKOTSE
Blerina.Gkotse@cern.ch

MINES ParisTech, PSL University, France

25 September 2020

Jury:
Laura GONELLA, Rapporteur
Jean-Baptiste LAMY, Rapporteur
Laurent DUSSEAU, Examinateur
Theodora VARVARIGOU, Examinateur
Pierre JOUVELOT, Directeur de thèse
Federico RAVOTTI, Maître de thèse

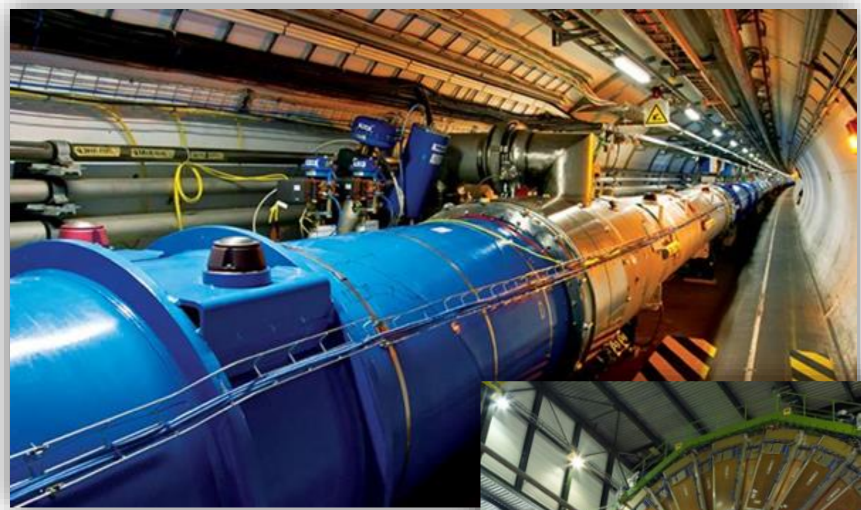
Motivation



From data to wisdom

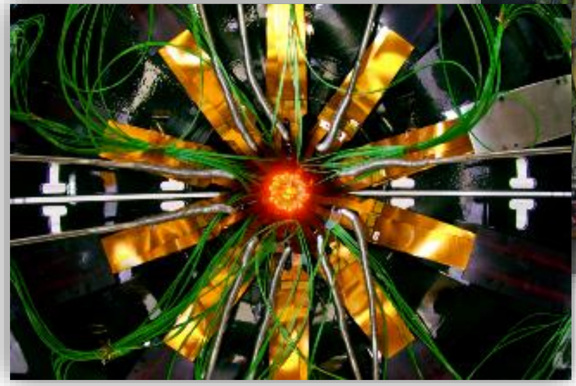
Context

Experimental Particle Physics (EPP)



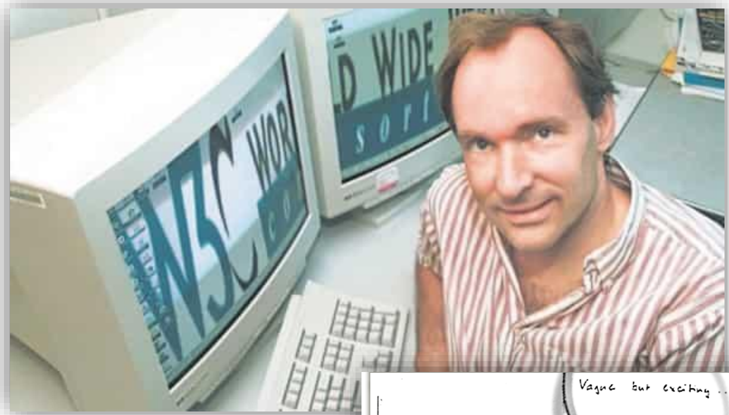
Large Hadron Collider (LHC)

Compact Muon Solenoid (CMS)

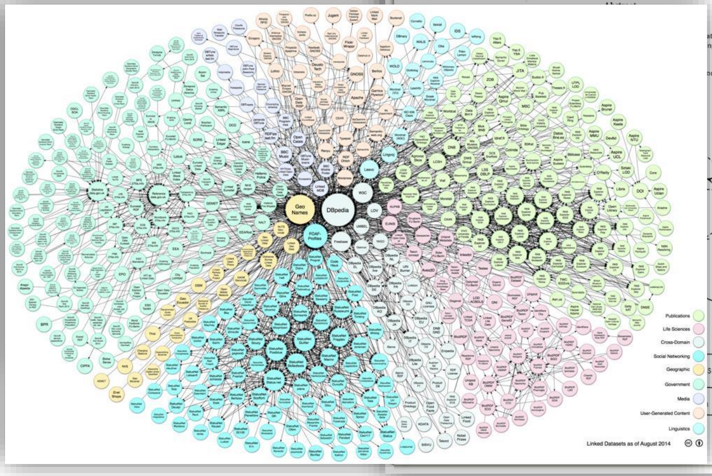


Tracker

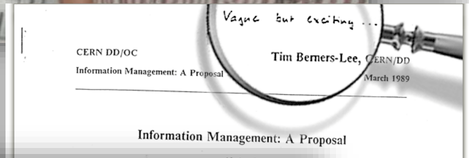
Web Semantics



Tim Berners Lee



Linked data



First Proposal for WWW

Web Semantics

Computers don't understand semantic meaning

E.g., *"My mouse is broken…"*

©publiclibrariesonline.org

©vectorstock.com

©vectorstock.com

Ontology Origin

Deriving from ancient Greek:

Ontology = On (ὄν) + logos (λόγος)

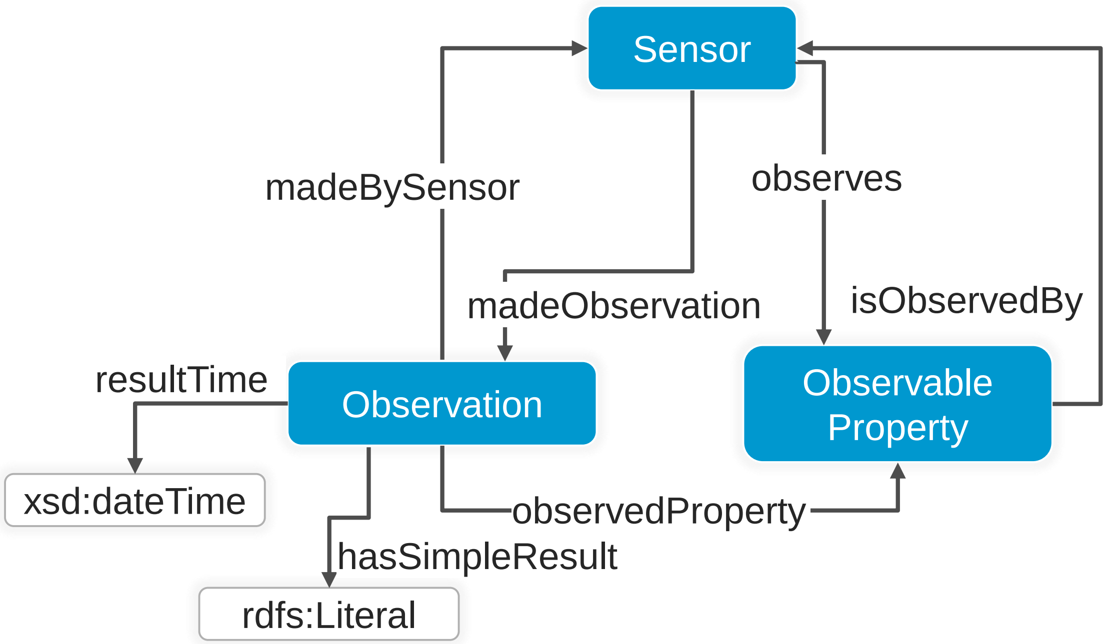
Aristotle

<u>In philosophy</u>:
subject of existence, science of being,
"what exists" in the world

<u>In computer science</u>:
formal description and classification of "what exists",
what can be represented as a knowledge fact

Ontology: Structure

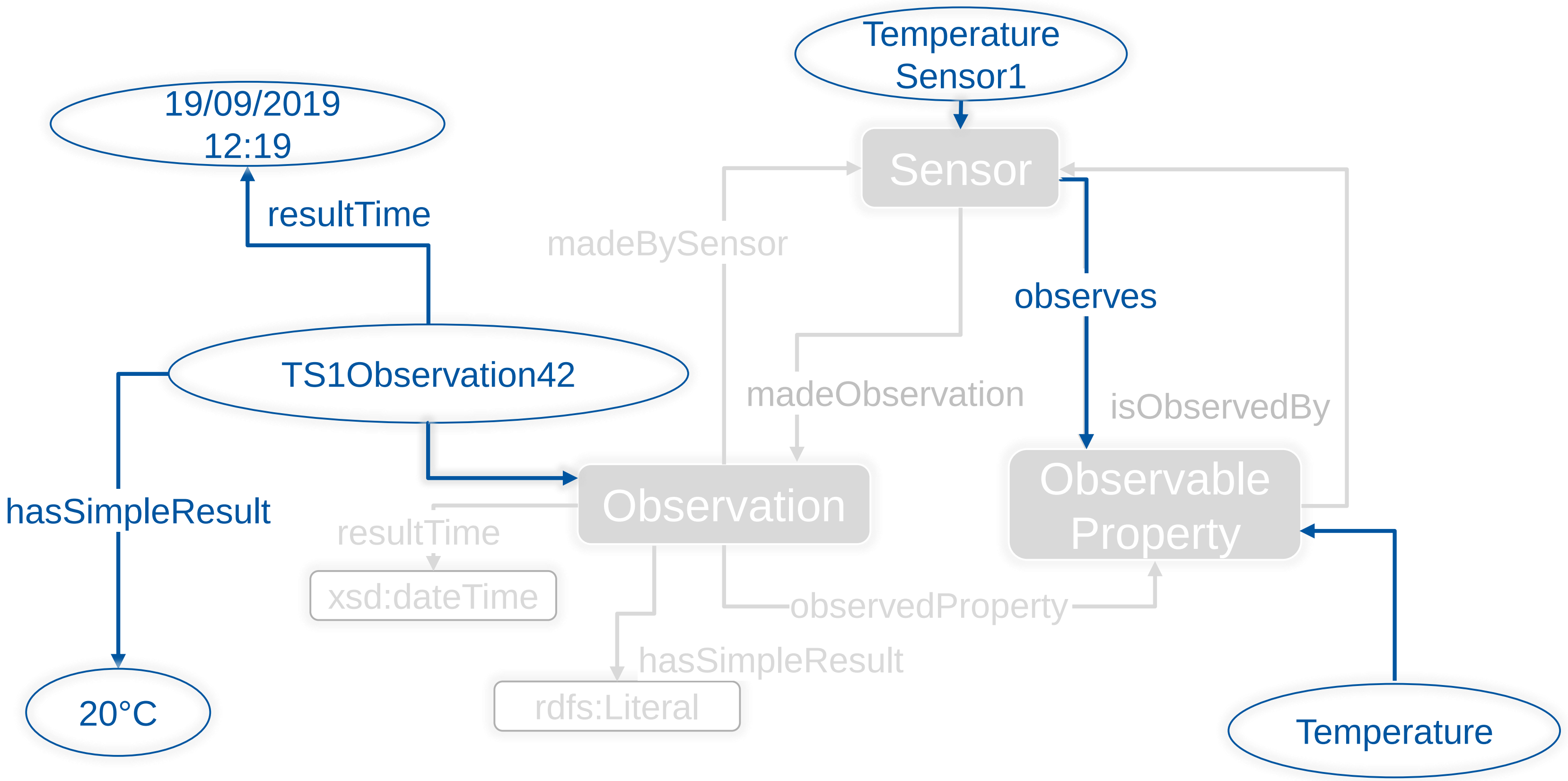
- **Class**: a set of entities of a specific domain of knowledge
- **Relation**: a semantic link among classes, also called **object property**
- **Property**: attribute of specific type, also called **data property**



Excerpt from SOSA (Sensors, Observations, Samples and Actuators) ontology

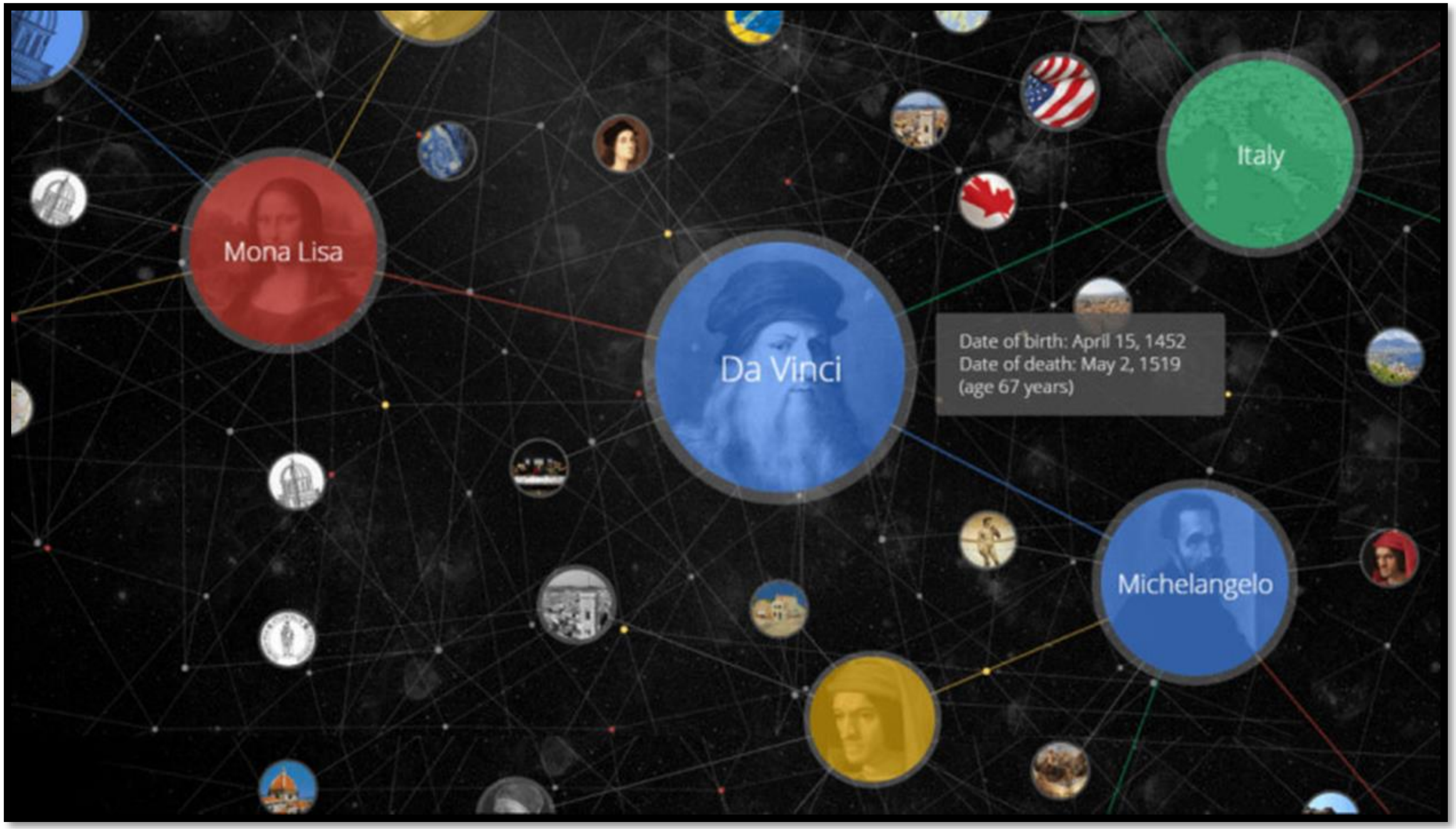
Knowledge Base (KB)

- Individual instances of domain's classes
- RDF triples (Resource Description Framework)
- Stored in triple stores



Knowledge Graph (KG)

- Term coined by Google (2012)
- Knowledge bases from a variety of sources
- Used in academia and industry



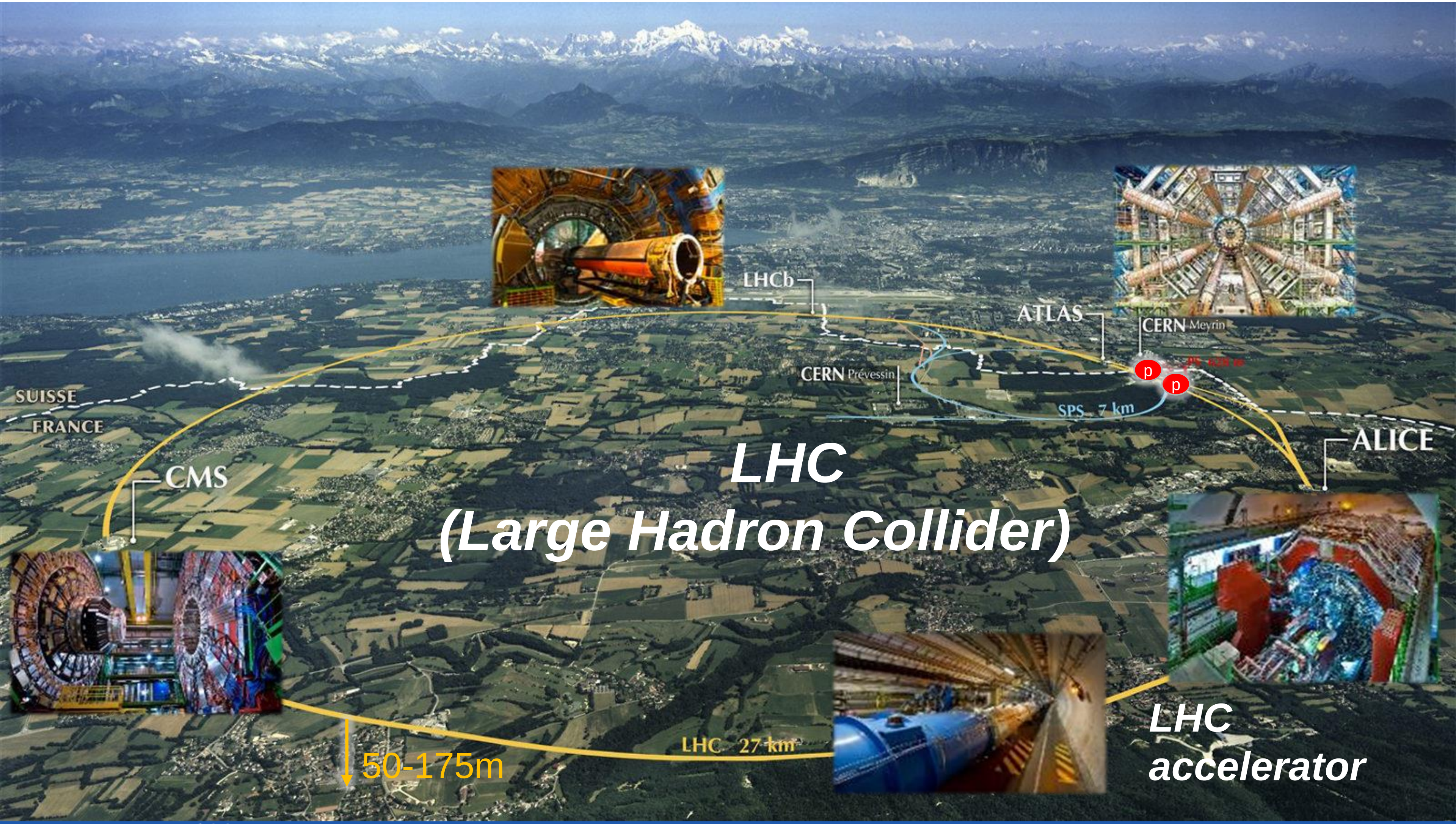
Google Knowledge Graph

©searchengineland.com

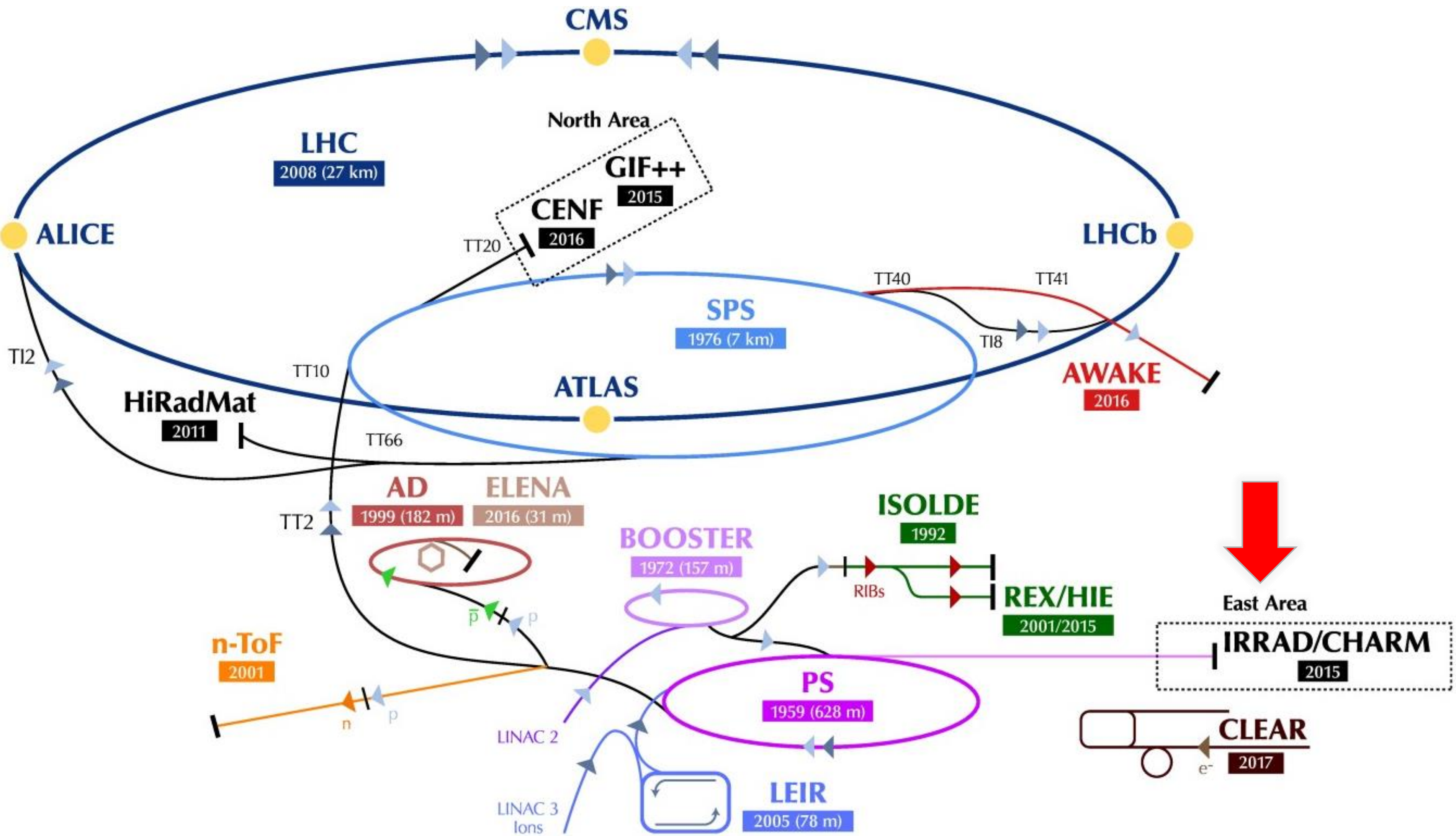
Ontology Goals

Domain Knowledge Sharing	• Common understanding • Interoperability • Inference • Analysis
Natural Language Processing (NLP)	• Query answering and information extraction • Recommender systems
Data Integration	• Communication between databases and datasets

European Laboratory of Particle Physics (CERN)



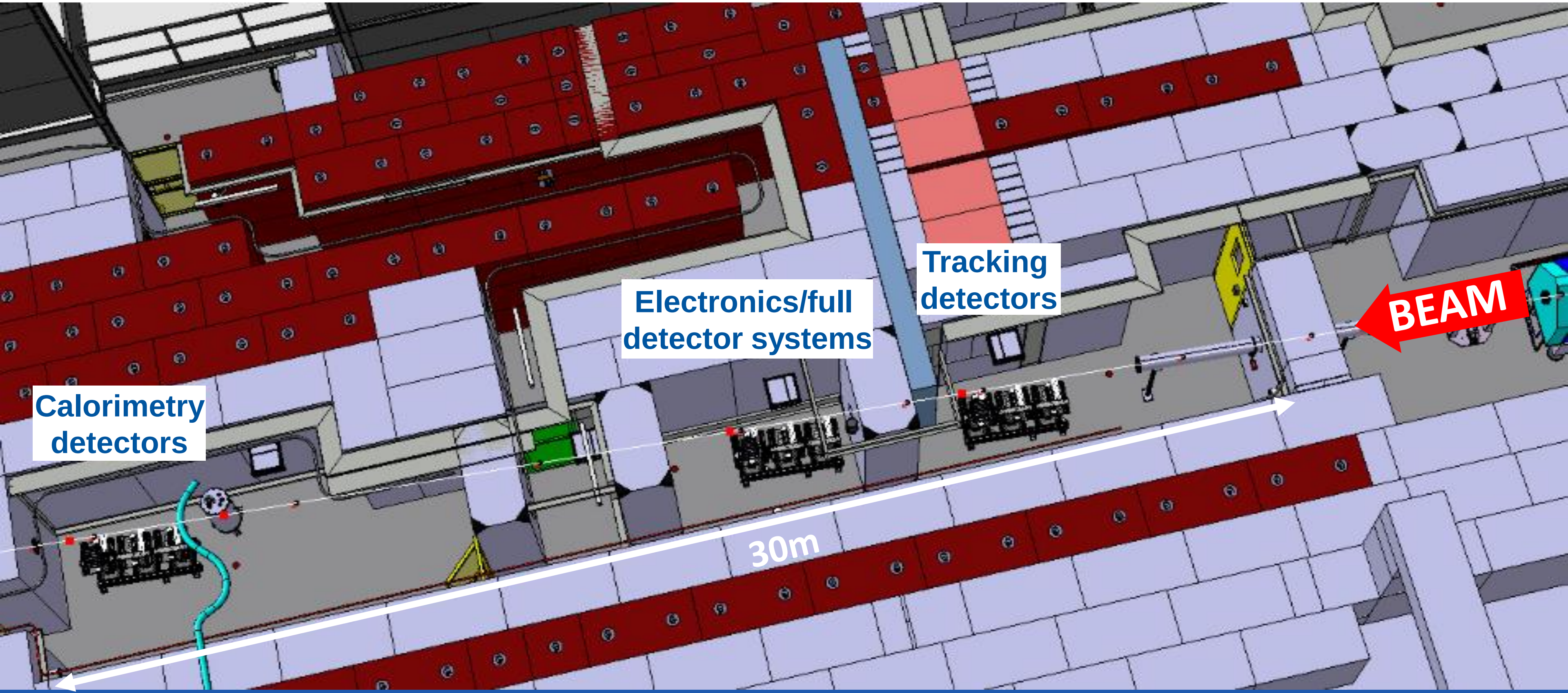
CERN Accelerators Complex



©cds.cern.ch

CERN Proton Irradiation Facility (IRRAD)

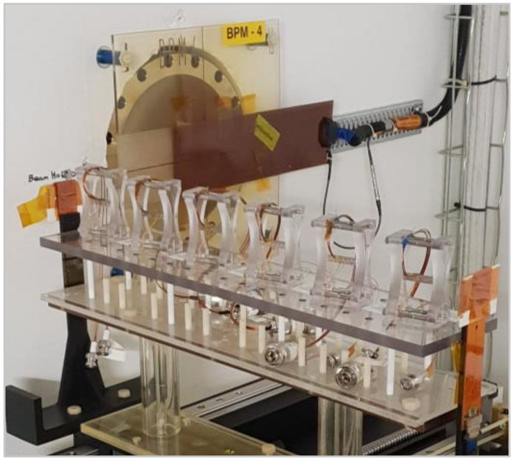
- Reference facility for proton irradiation experiments
- Testing of components of EPP experiments



IRRAD Experiments

Proper data management necessary

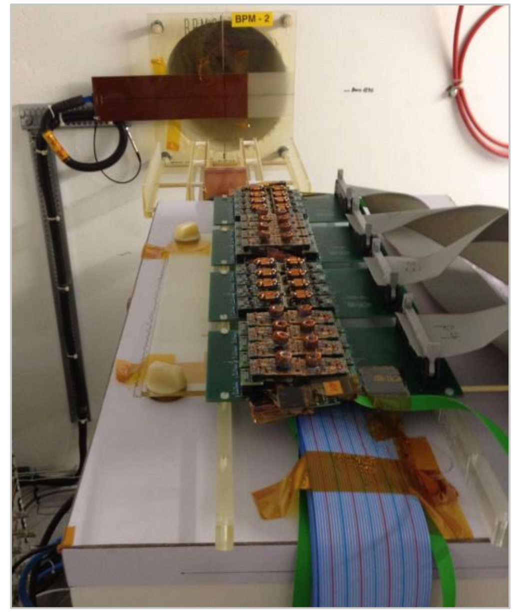
In 2018: **81** irradiation experiments,
97 users, **792** samples, **405** dosimeters,
2056 dosimetry measurements

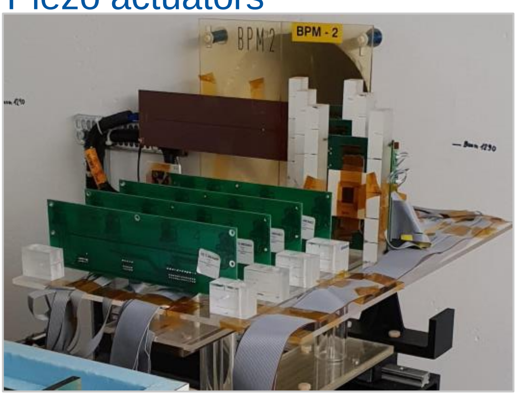


Piezo actuators

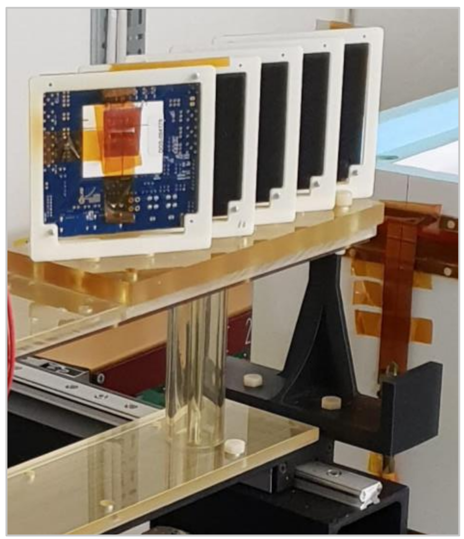
ECAL crystal

DC/DC converters





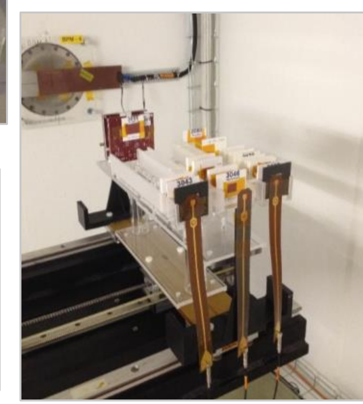
CLARO ASIC



RD53A modules

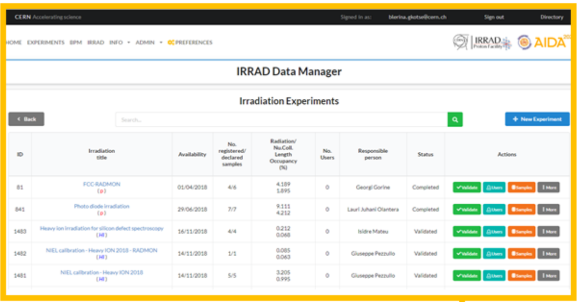
©ps-irrad.cern.ch

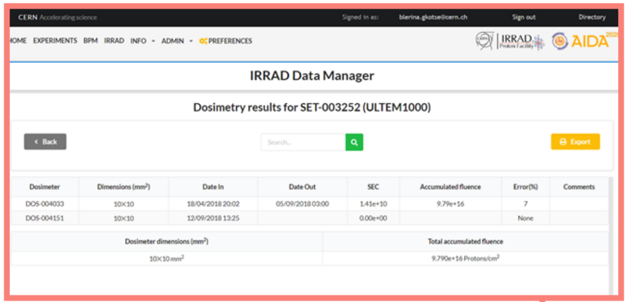
Full-tracking detector module

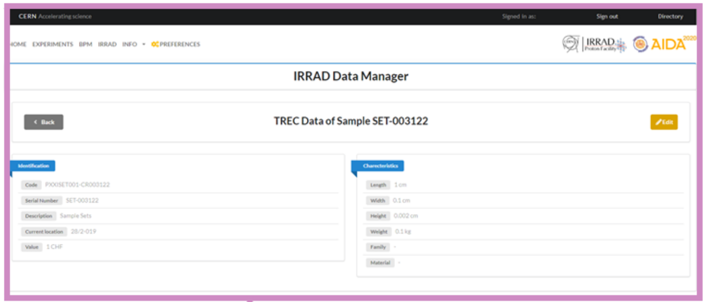


Si pad diode

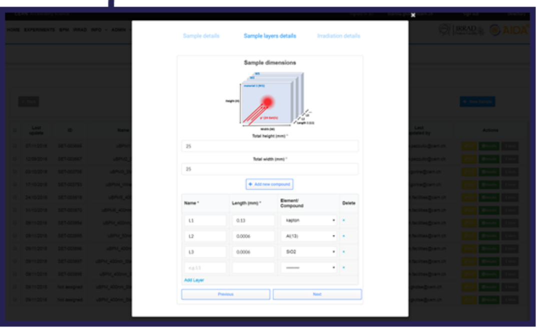
IRRAD Data Manager (IDM)

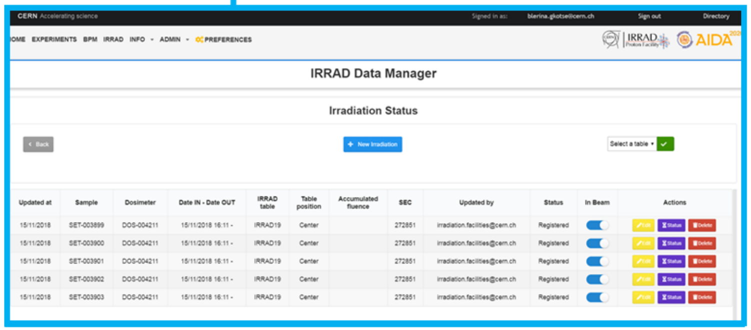


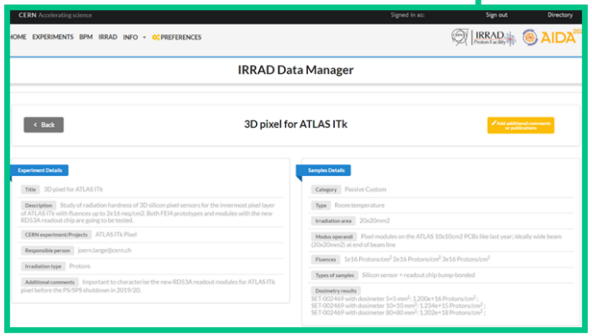




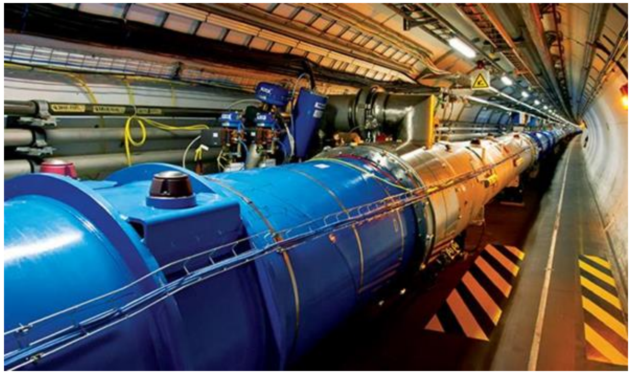
Registration → Planning → Follow-up → Dosimetry Results → Traceability → History





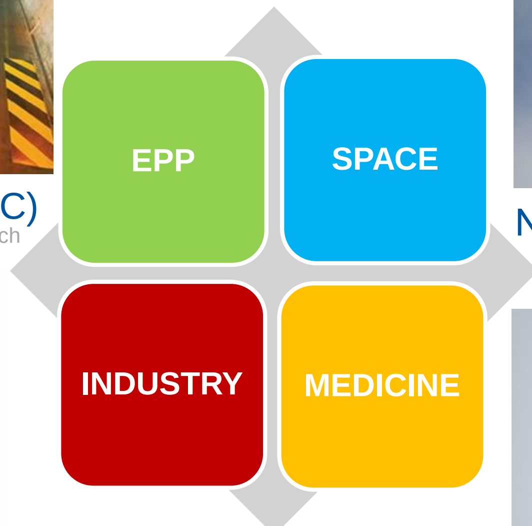


Irradiation Experiments



CERN Large Hadron Collider (LHC)

©cds.cern.ch



NASA’s Orion spacecraft

©arstechnica.com

Bottle sterilization by electron beam irradiation

©Courtesy Frank-Holm Roegner

Clinac® iX System linear accelerator

©medicalexpo.com

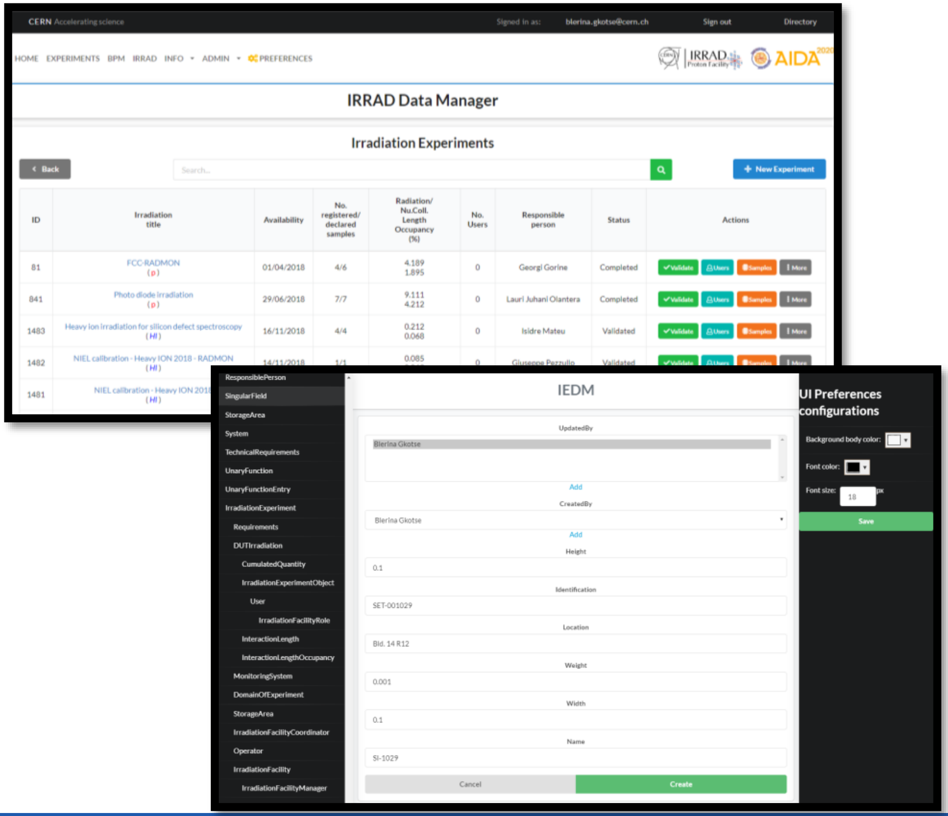
Bridging the Gap in Data Management

- Knowledge sharing among communities
- Small experiments, no strong IT support
- User Experience

✓ Web Semantics

✓ Automatic generation of web applications

✓ UI personalization



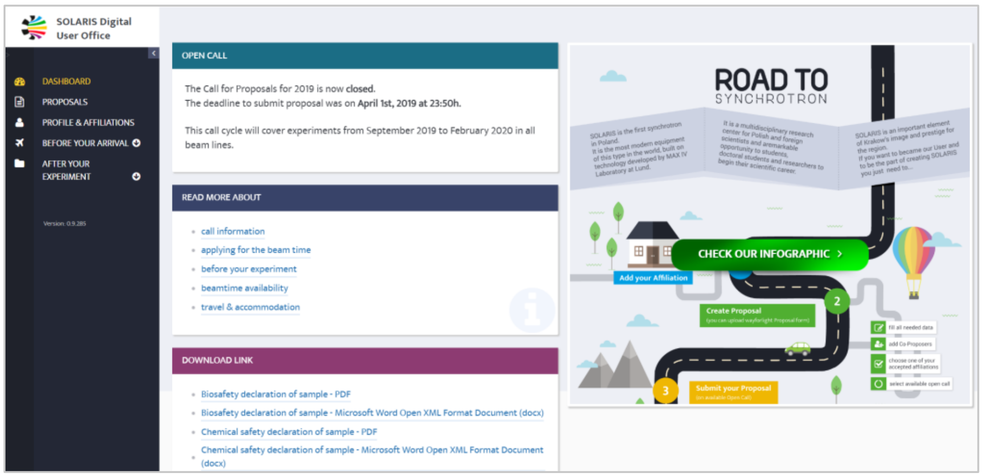
Outline

- Context
  - Web Semantics
  - Experimental Particle Physics
- **Irradiation Experiment Data Management Ontology (IEDM)**
- Automatic Generation of Web Applications from Ontologies
- UI Personalization with Ontology Embeddings

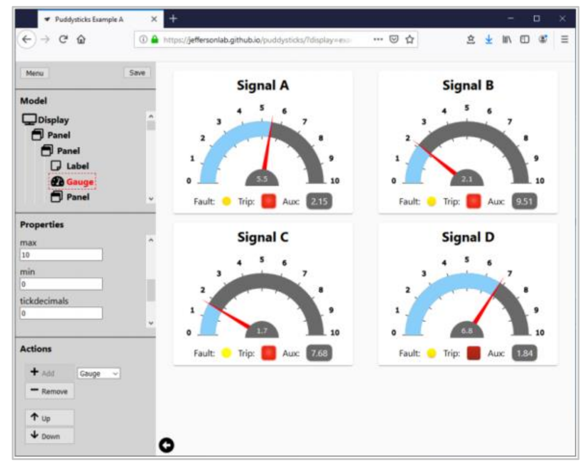
Data Management in EPP

Available tools for:

- Information registration
- Infrastructure availability
- Monitoring and visualization
- Reporting



SOLARIS Digital User Office



Web Extensible Display Manager used at Jefferson Lab for control-systems display

- ***No fully integrated solution for the follow-up of experiments' life cycle***
- ***No use of web semantics***

Domain Ontologies in EPP

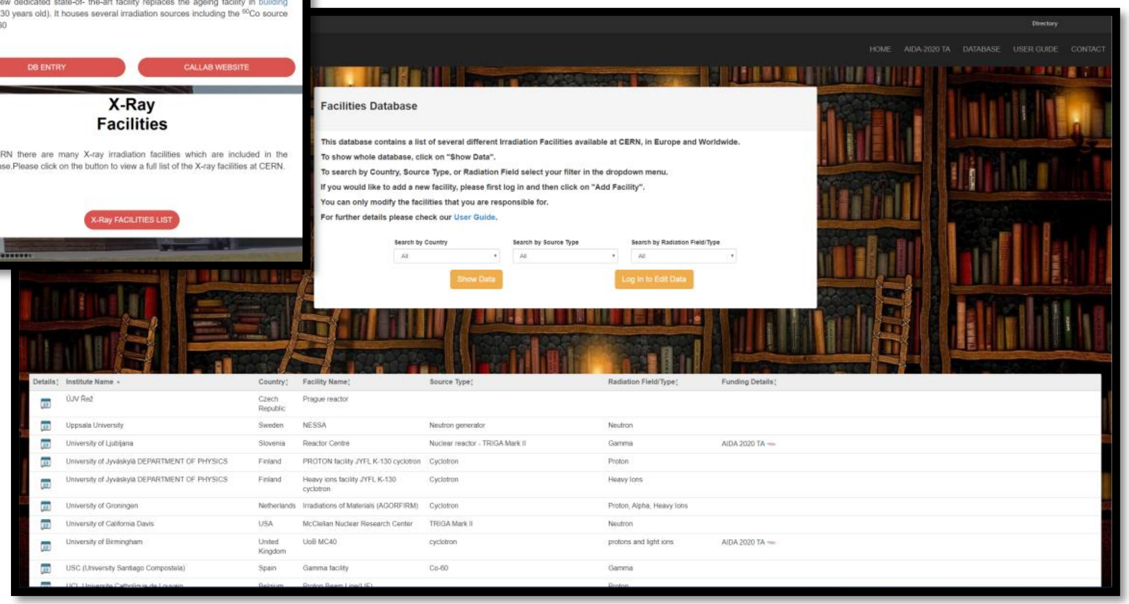
Ontology	Authors	Content	Domain
Web Physics Ontology	V. Vjetkovic	physics equations and relations	Physics
Ontology Design Pattern	D. Carral, *et al.*	analysis of particle physics data	Physics
Synchrotron Ontology	J. Szota-Pachowic	synchrotron components	Physics
Radiation Protection Ontology	C. Barki, *et al.*	irradiation experiments focusing on medicine	Medicine
EXPO – Ontology of Scientific Experiments	L. Soldatova, *et al.*	scientific experiments	General

No ontology for the data management of irradiation experiments

Irradiation Experiments’ Data Research

- Research and compilation of irradiation experiments’ and related facilities’ data
- Domain experts interviewed (~10)

Online DB platform **developed containing the collected data**
(217 facilities up to date)
www.cern.ch/irradiation-facilities

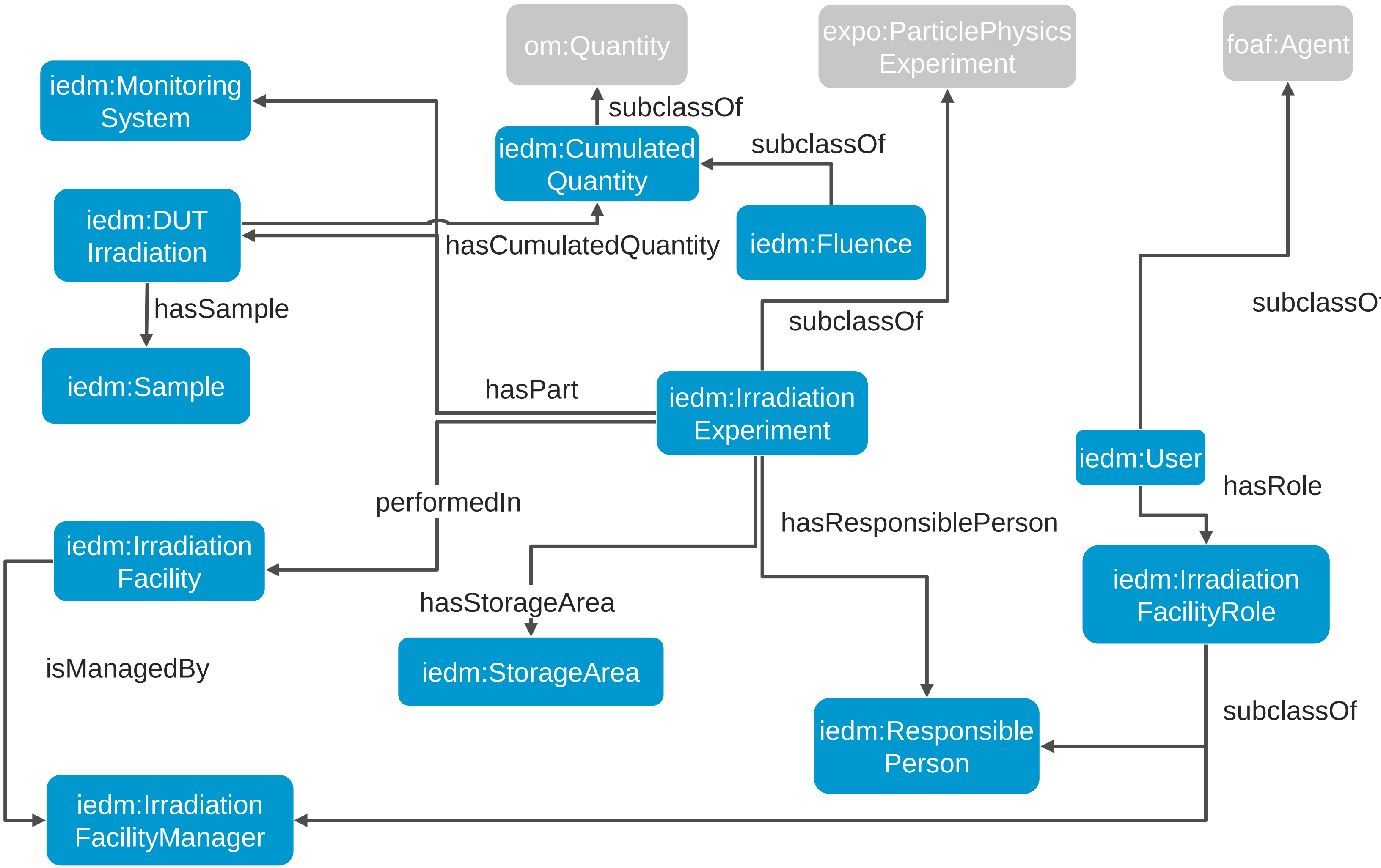


Irradiation Experiment Data Management (IEDM)

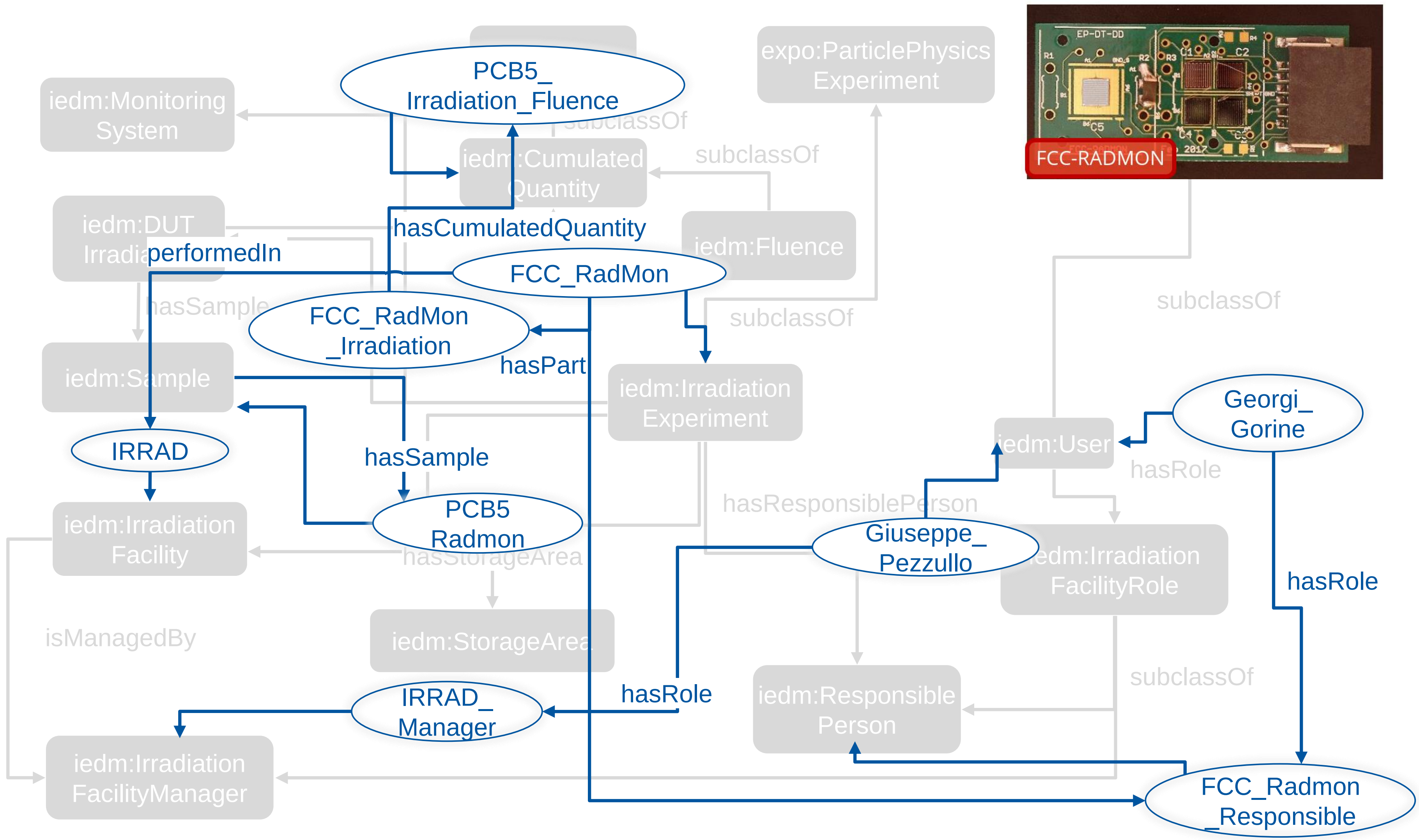
Three foundational ontologies

Name	Authors	Concepts
EXPO – Ontology of Scientific Experiments	L. Soldatova, *et al.*	scientific experiments
OM-2 – Units of Measure ontology	H. Rijgersberg, *et al.*	experimental quantities and physical units
FOAF – Friend Of A Friend Ontology	D. Brickley, *et al.*	networks of people, activities and their relations

IEDM: Core Classes



IEDM: Instances



IEDM Online Documentation

http://cern.ch/iedm

Ontology Specification Draft

language en

Irradiation Experiment Data Management Ontology (IEDM)

Release 2019-03-10

This version:
https://gitlab.cern.ch/bgkotse/iedm/raw/master/iedm.owl

Authors:
B. Gkotse, P. Jouvelot and F. Ravotti

Download serialization:
Format JSON LD | Format RDF/XML | Format N Triples | Format TTL

License:
License license name goes here

Visualization:
Visualize with WebVowl

Cite as:
B. Gkotse, P. Jouvelot and F. Ravotti. Irradiation Experiment Data Management Ontology (IEDM).Revision: 0.1.https://gitlab.cern.ch/bgkotse/iedm/raw/master/iedm.owl

Abstract

Irradiation experiments (IE) are an essential step in the development of High-Energy Physics particle accelerators and detectors. They are used to assess the radiation hardness of experimental devices by simulating, in a short time, the common long-term degradation effects due to high-energy particles. Usually carried out with ionizing radiation, these complex processes require highly specialized infrastructures called "irradiation facilities". Aiming to promote knowledge sharing and digital management of IEs, we introduce IEDM, a new Irradiation Experiment Data Management ontology.

Table of contents

115 classes
941 annotations
24 object properties
16 data properties

Documentation generated using WIDOCO.
D. Garijo, "Widoco: a wizard for documenting ontologies" ISWC 2017

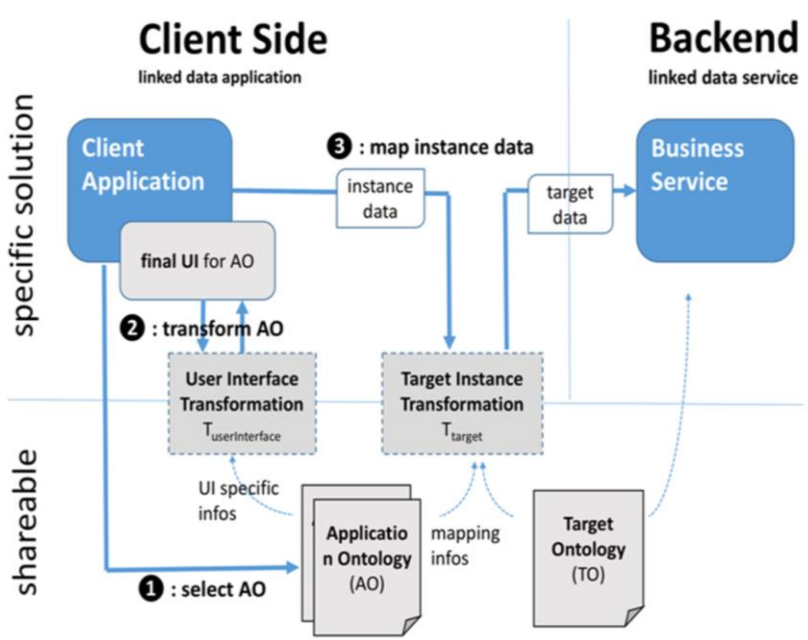
Outline

- Web Semantics
- Experimental Particle Physics
- Irradiation Experiment Data Management Ontology (IEDM)
- **Automatic Generation of Web Applications from Ontologies**
- UI Personalization with Ontology Embeddings

Ontology-based User Interface (UI) Generation

Several related works, more pertinent by Hitz *et al.**:

- User selection of Application ontology and instances
- Automatic transformation to a Target ontology for UI generation
- Proprietary annotations are required, limiting their universality



*M. Hitz, T. Kessel and D. Pfisterer, "Towards Sharable Application Ontologies for the Automatic Generation of UIs for Dialog based Linked Data Applications" 2017.

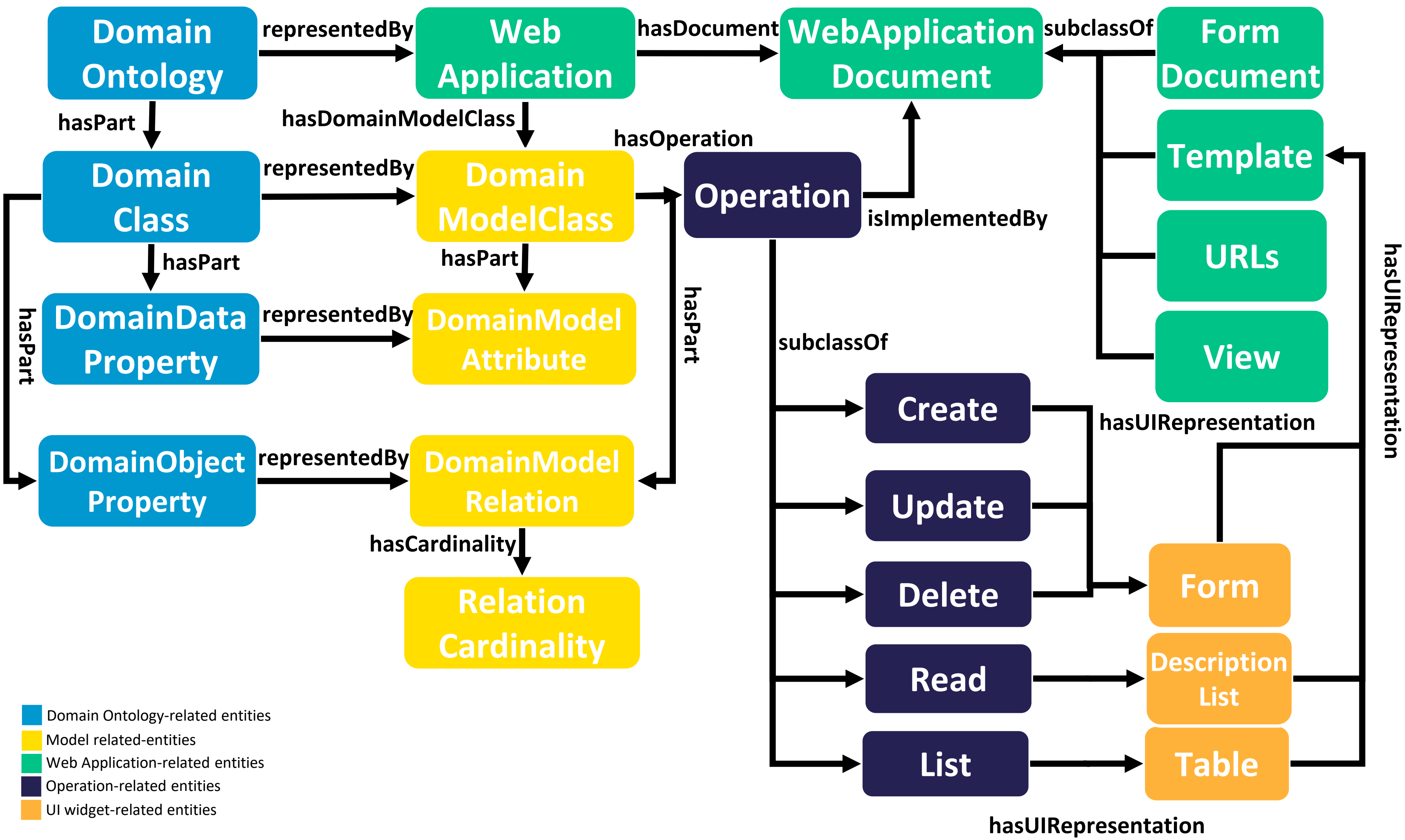
UI Ontologies

Existing ontologies **reused**:

- **Semantic UI** framework
- **LOV UI,** supported by the Linked Open Vocabularies (LOV)
- **RaUL**, RDFa User Interface Language (supported by LOV)
- **HUICA**, Hierarchical User Interface Component Architecture

Ontology	UI elements concepts (Forms, Tables, ...)	Style properties (color, background color, ...)	Data operations	Back-end concepts
Semantic UI	×	-	-	-
LOV UI	×	×	-	-
RaUL	×	-	×	-
HUICA	×	×	-	-

Ontology-based Web Application Ontology (OWAO)



Web Application Generator (GenAppi)

Loading domain and OWAO ontologies (OWL)

Loading Ontologies

Transforming domain classes and object/data properties to Model classes and attributes

Domain Ontology to Model Mapping

Generating Django view files for data operations (CRUD)

Operation Handling

Creating Django Template and URLs files for the UI

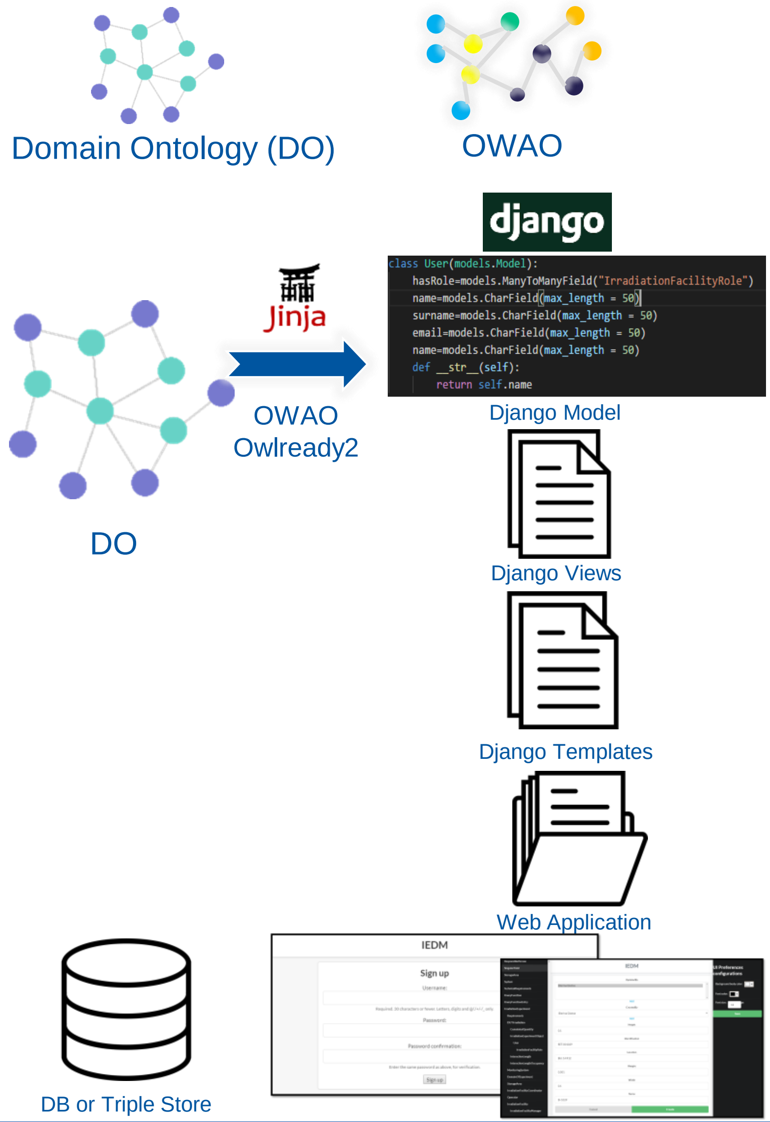
Creating Templates and URLs

Grouping files in specific directories

Web Application Packaging

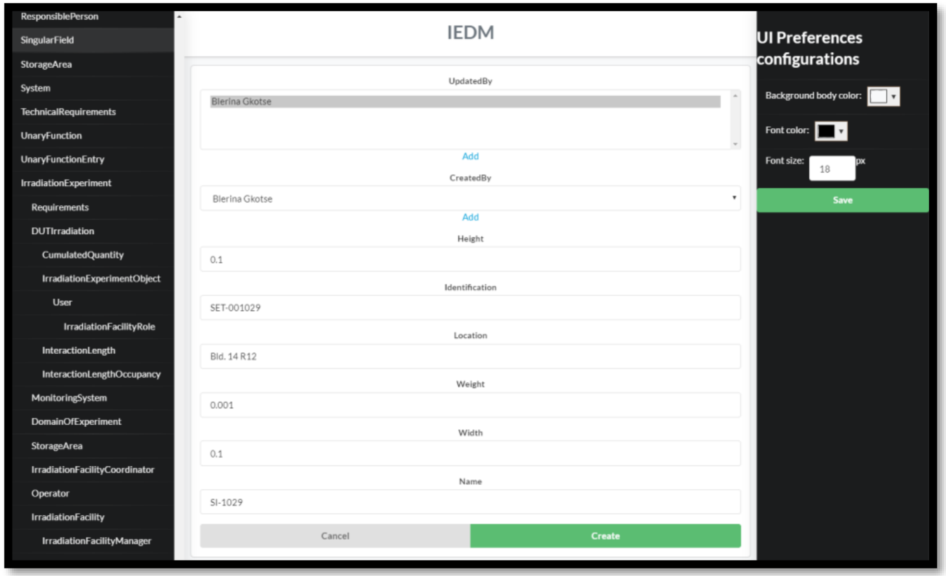
Migrating Model and server ready for start

Migration and Server Settings

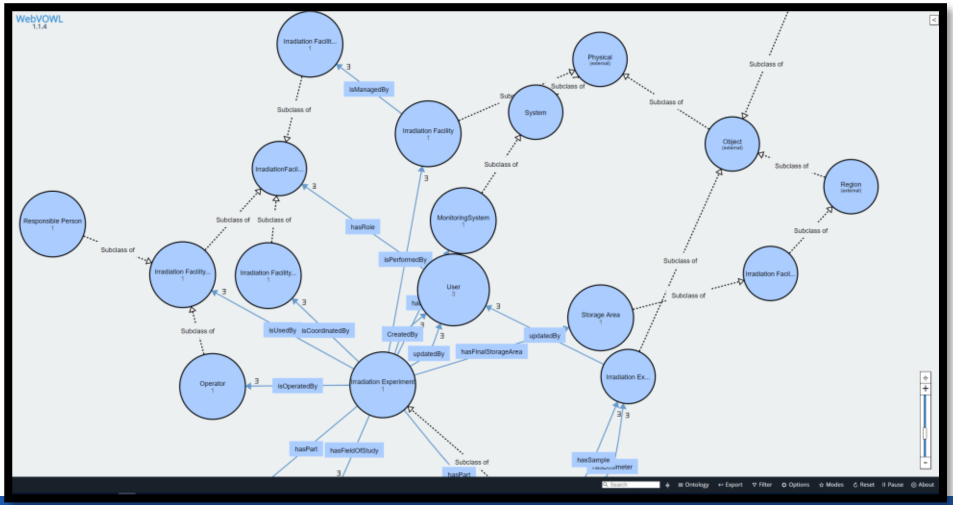


IEDM Use Case

- Input form for each domain-ontology class instance:
  - **Data properties** = **input fields**
  - **Object properties** = **selection fields**, **links** to the corresponding class forms
- ***Django* authentication** and **authorisation**
- **UI adaptable** to user's preferences
- **Visualisation** of domain ontology



Registering an IrradiationExperimentObject instance



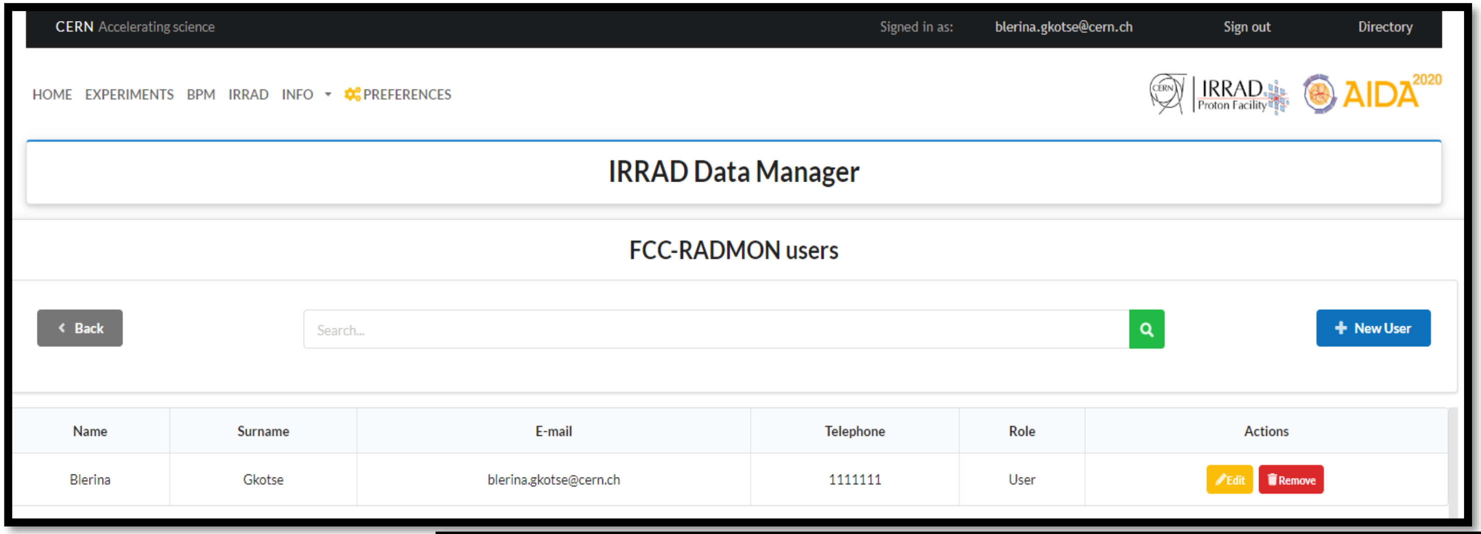
Visualisation using WebVOWL

Generated Application vs. IDM

	IDM	Generated Application
Purpose	IRRAD facility data management	any domain
Software infrastructure	CERN	free and open-source
Storage	Oracle database	any relational database, ontology or triple store
Web Semantic technologies	no	yes
UI Customisation	no	yes
Functionalities	more advanced	CRUD* operations

CRUD* = Create Read Update Delete

User Interfaces Comparison: List Table



IDM list table

IEDM_APP list table

Outline

- Web Semantics
- Experimental Particle Physics
- Irradiation Experiment Data Management Ontology (IEDM)
- Automatic Generation of Web Applications from Ontologies
- **UI Personalization with new Ontology Embeddings**

User Experience

"**User Experience (UX)** refers to a person's entire experience using a particular product, system or service."

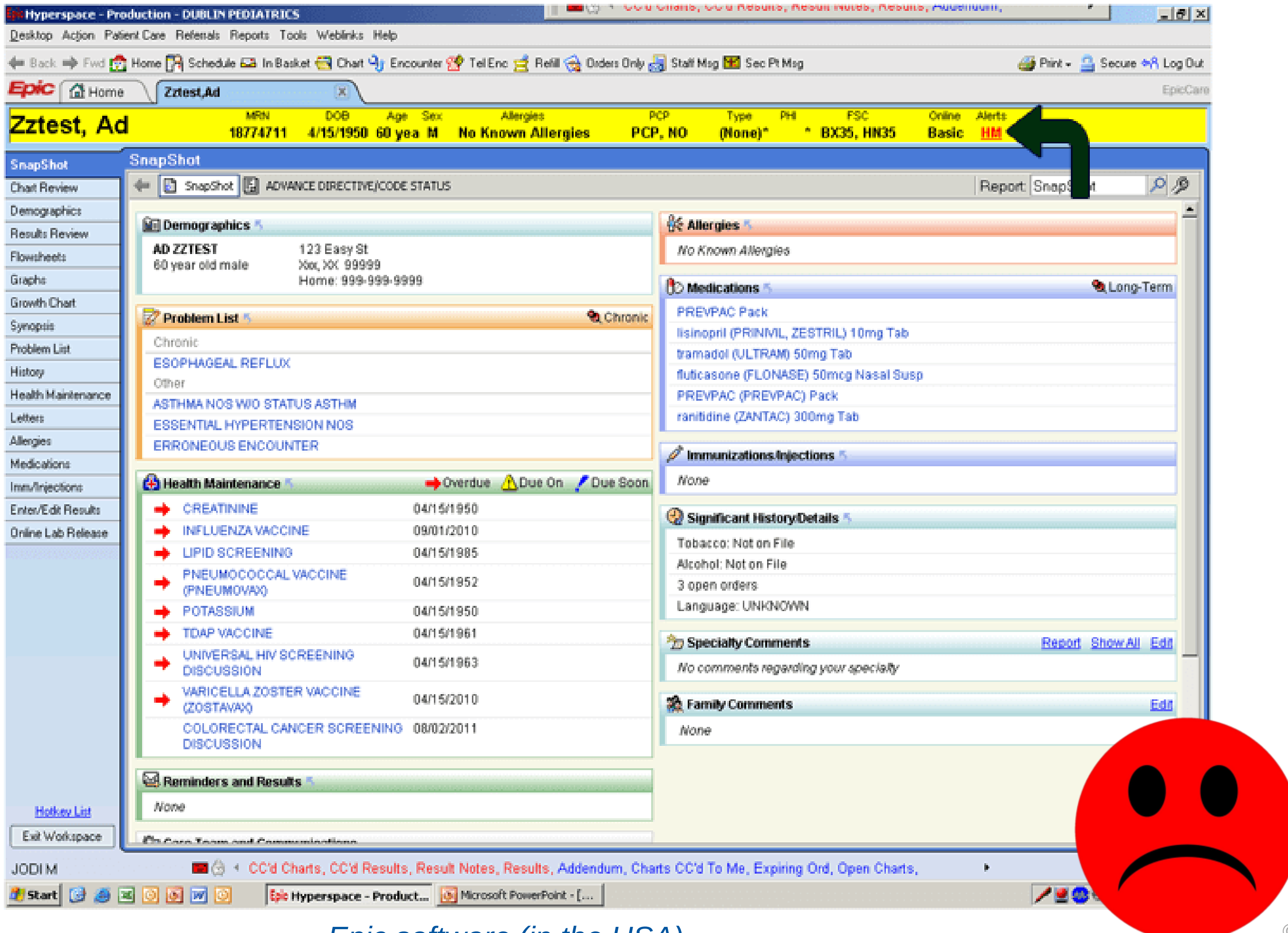
©Interaction Design course UNIL

User Experience Honeycomb

©Peter Morville, "User Experience Design"
http://semanticstudios.com/user_experience_design/

User Experience

Bad UI design can cause serious human errors



Epic software (in the USA)

©https://www.tragicdesign.com/

Personalisation

User Experience factors:

- **UI content**
- **Cultural background**
- **Physiology**
- **Cognitive capability**

Images better displayed on black background

©Tubic Studio | Museu by Ernest Asanov

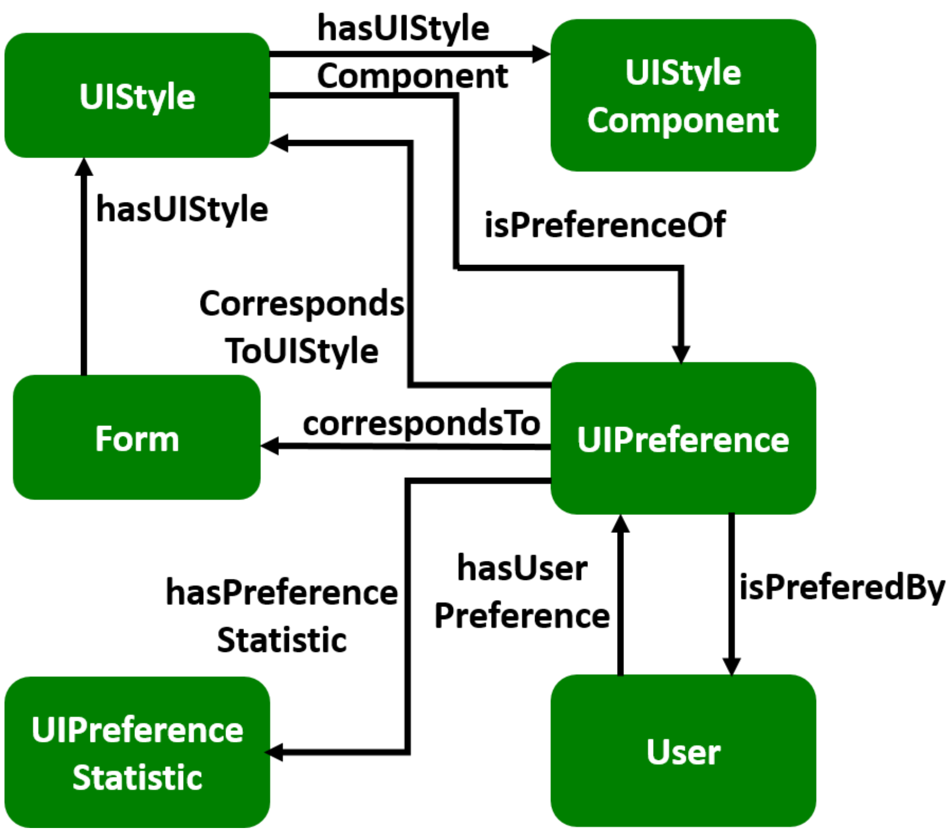
Eye tracking of websites for population of Western countries

©nngroup.com NN/g

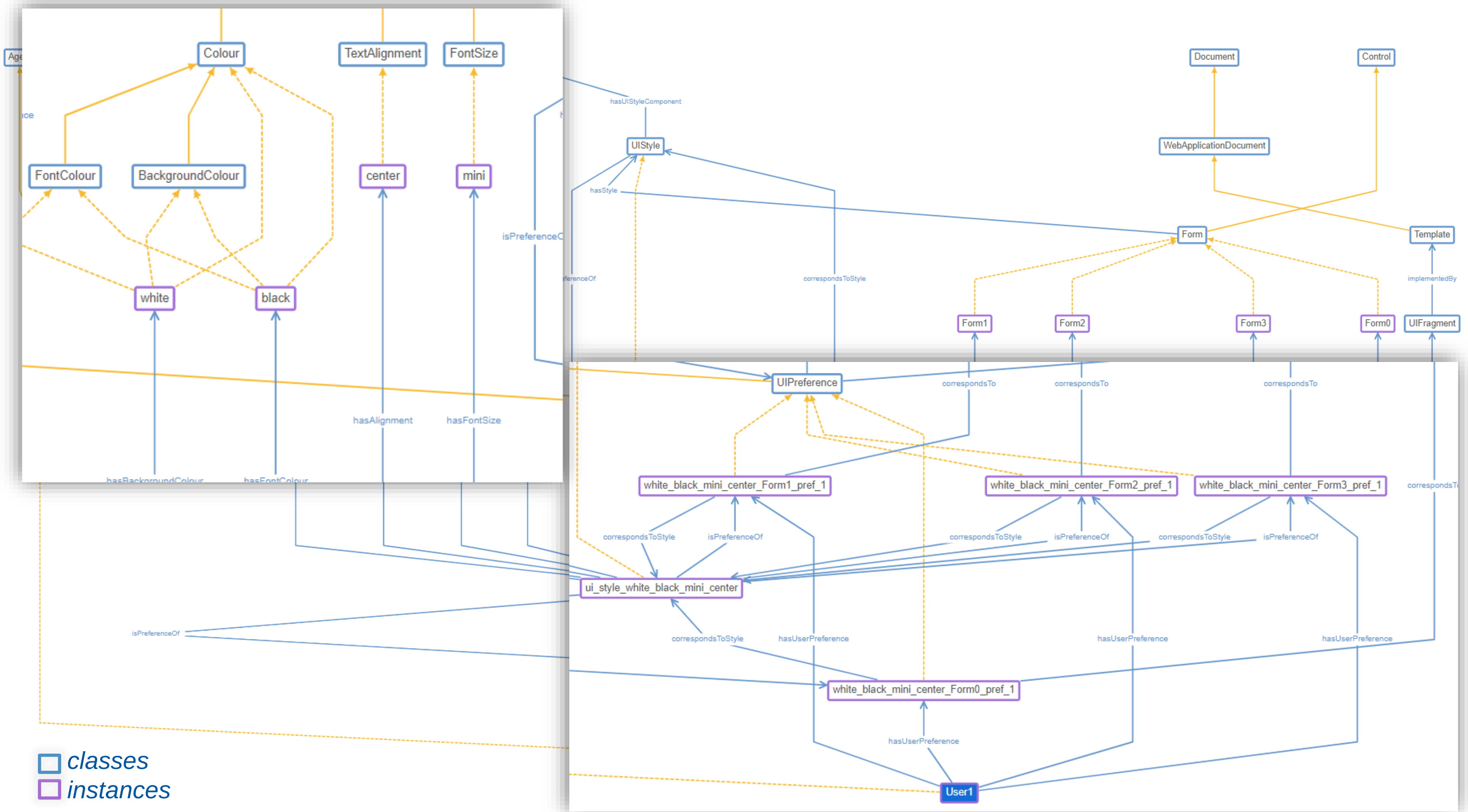
©iconfinder.com

OWAO UI Preferences

- Part of OWAO (Ontology-based Web Application Ontology)
- Used for recommending UI style



OWAO UI Preferences Example



Excerpt from OWAO ontology showing "User1" instance and UI Preferences (displayed from web Protégé)

Ontology Embedding

Ontology instances → words, text

(e.g. *TemperatureSensor1 observes Temperature*)

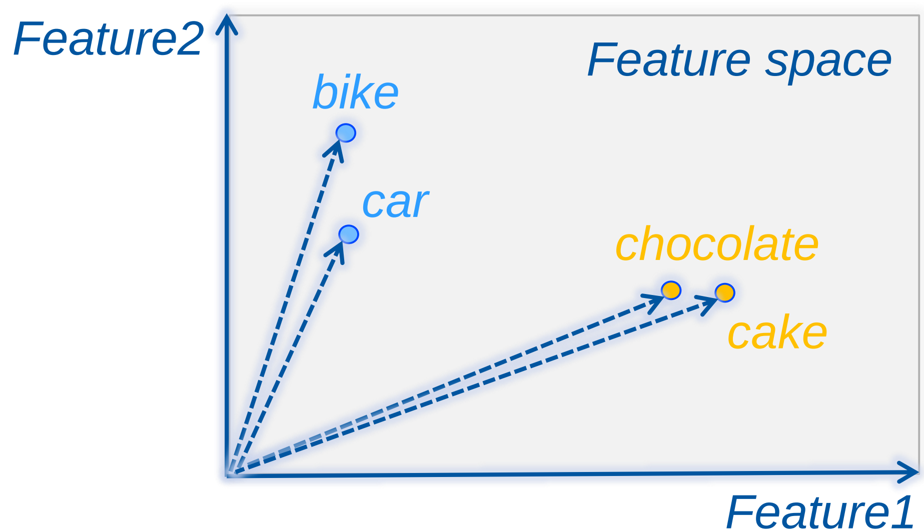
Word representation as feature vectors for tracking

- **similar words** and
- **words appearing in the same context**

Used in **recommender systems** for suggesting similar items

	car	*bike*	*cake*	*chocolate*
Feature1	1	1	6	7
Feature2	3	5	2	2
Feature3	4	4	1	1
Feature4	7	7	2	2

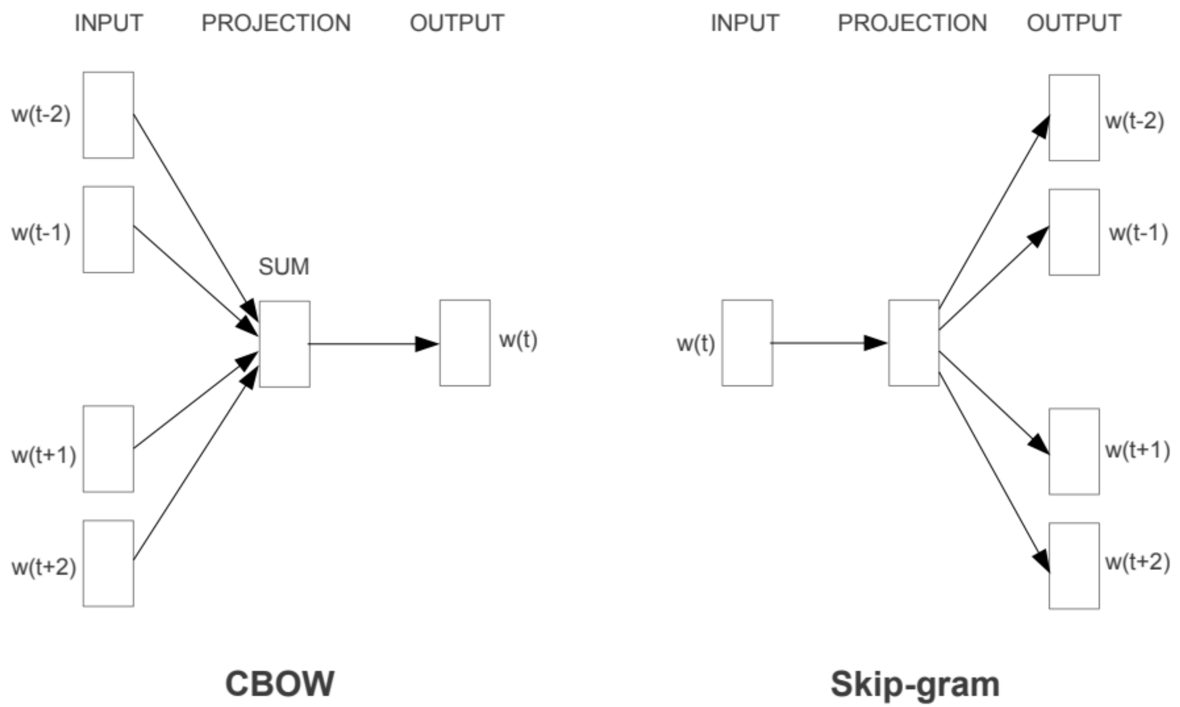
Feature Vectors



word2vec

NLP Model for generating word embeddings*:

- Continuous Bag of Words (CBOW)
- Skip-gram

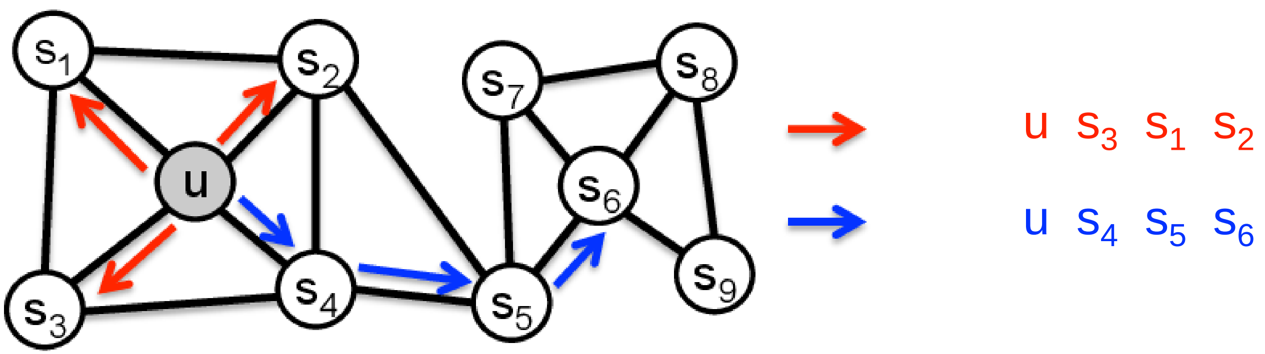


w = word

*A. T. Mikolov et al., "Efficient Estimation of Word Representations in Vector Space"

node2vec / RDF2Vec

- **Ontologies ~ directed graphs**
- **Random walks extraction** to create "**sentences**"
- Two NLP models:
  - ***node2vec*** based on Breadth-first Search (BFS) and Depth-first Search (DFS)[1]
  - ***RDF2Vec*** based on BFS of certain depth and Weisfeiler-Lehman (subtree comparison algorithm)[2]



BFS *and* DFS *paths*

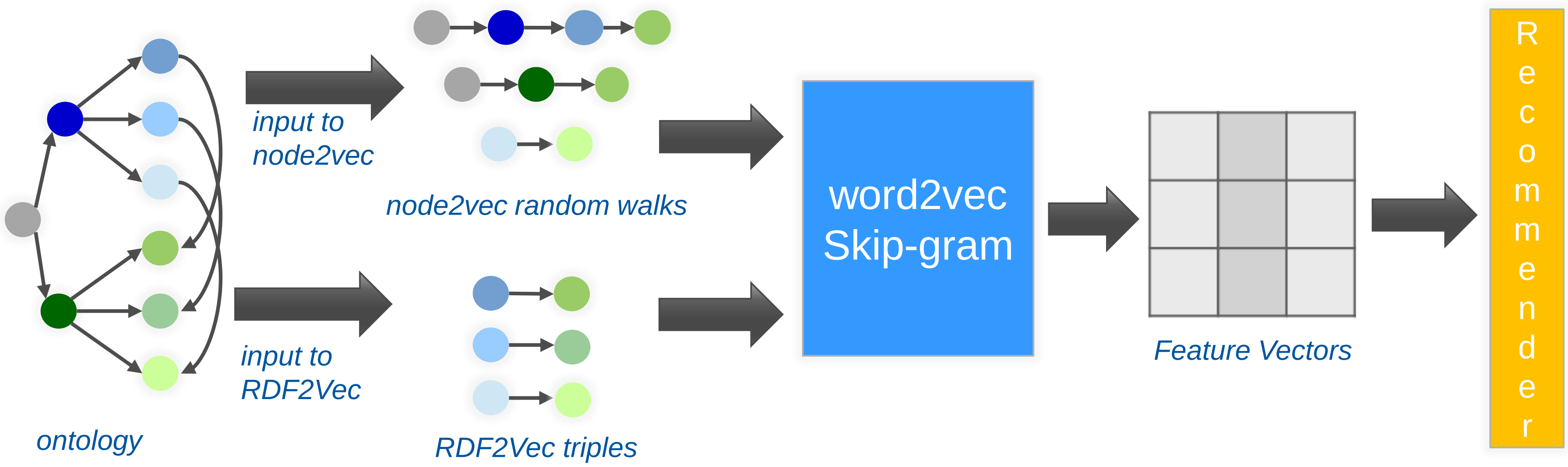
[1]A. Grover and J. Leskovec, "node2vec: Scalable feature learning for networks"
[2]P. Ristoski, et al., "RDF2Vec: RDF graph embeddings and their applications"

New embedding: *ontowalk2vec*

New hybrid NLP model:

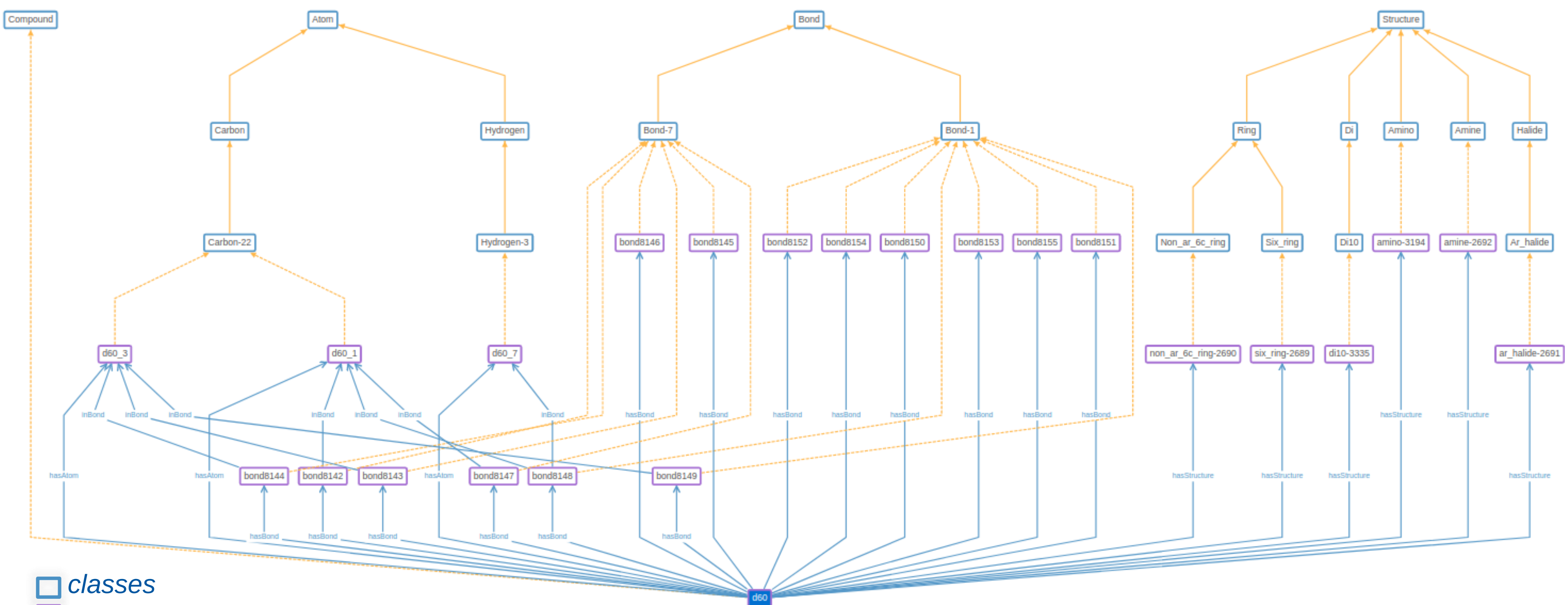
- *node2vec* → explicit structure of the ontology
- *RDF2Vec* → RDF triples

Algorithm:



Evaluation: MUTAG

- KB of 340 complex molecules (carcinogenic, i.e., “MUTAGenic” or not)
- Source ontology for classifying molecules (mutagenic / not-mutagenic)



classes
instances

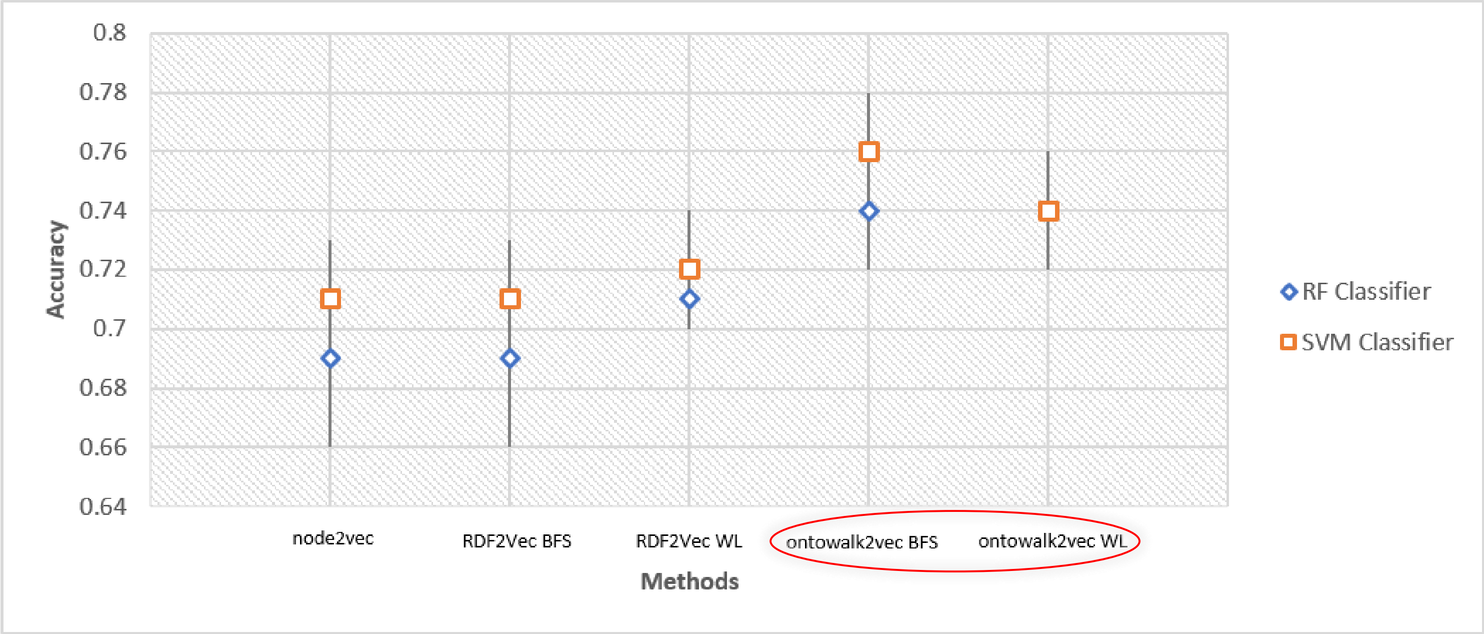
d60 compound

MUTAG Results

Experiments with *RDF2Vec*, *node2vec* and *ontowalk2vec* (run 10 times).

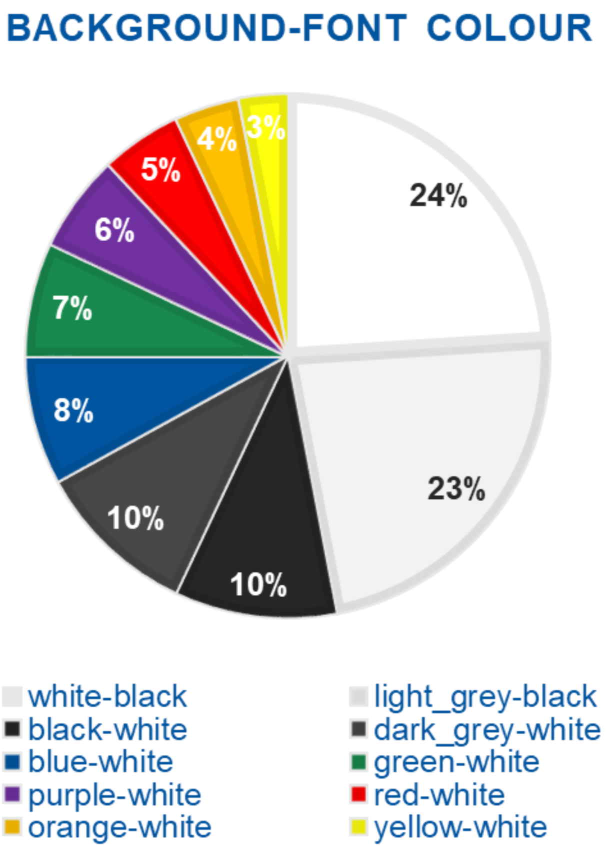
Classification accuracy test:

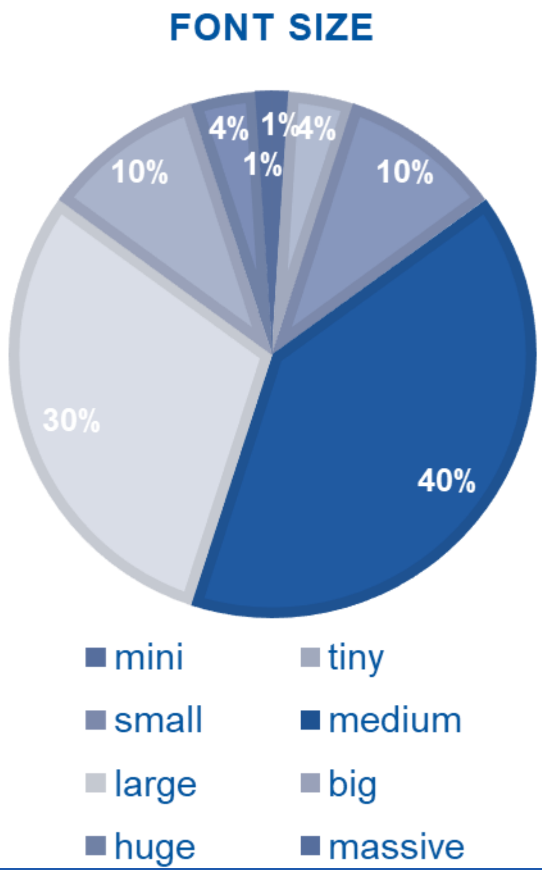
- **Random Forest (RF)**: decision trees on data sub-groups and averaging
- **Support Vector Machine (SVM)**: optimal hyperplane on labeled data

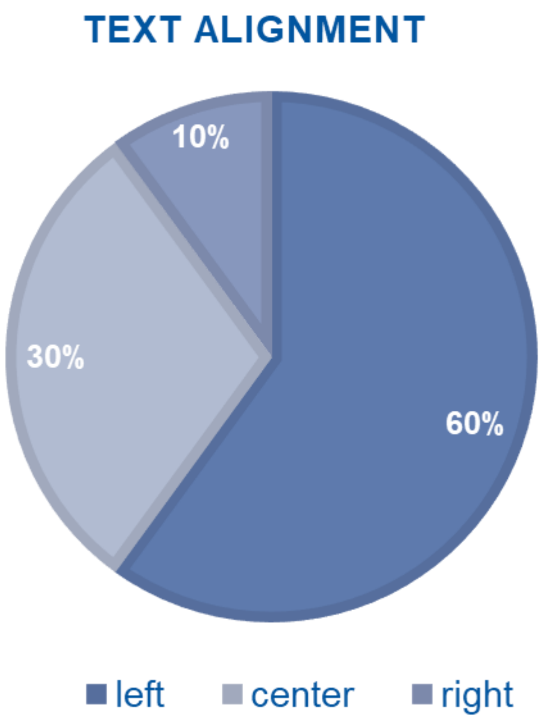


Evaluation: OWAO UI Preferences

- Lack of actual UI datasets → Artificial statistics-based data
- Four UI Style components taken as an example
- Statistics on UI preferences (bibliographic research)







OWAO UI Preferences Results

Classification accuracy:

	Random Forest	SVM
BFS	1.00 (±0.00)	1.00 (±0.00)
WL	1.00 (±0.00)	0.99 (±0.01)

preferred / not preferred

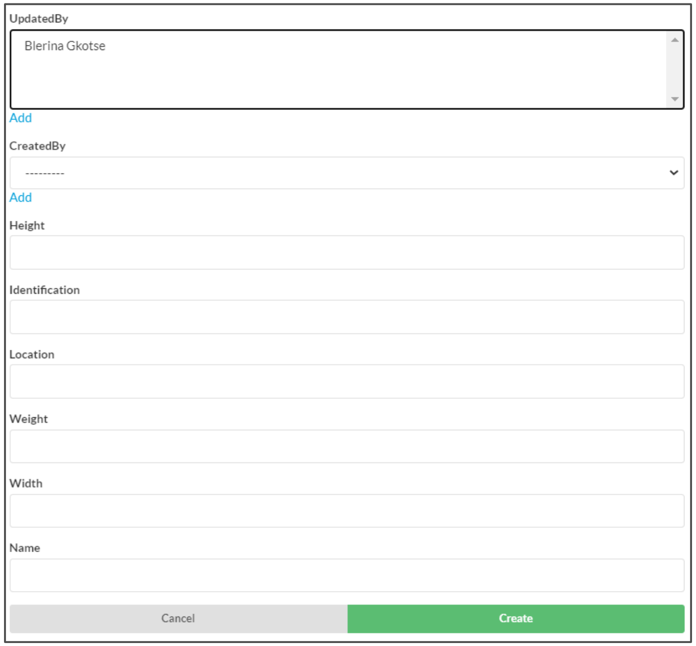
	Random Forest	SVM
BFS	0.990 (±0.001)	0.995 (±0.002)
WL	0.993 (±0.002)	0.995 (±0.001)

popularity zero / low / medium / high

	Random Forest	SVM
BFS	0.83 (±0.04)	0.94 (±0.04)
WL	0.86 (±0.06)	0.92 (±0.05)

popularity low / medium / high

Cosine Similarity Testing

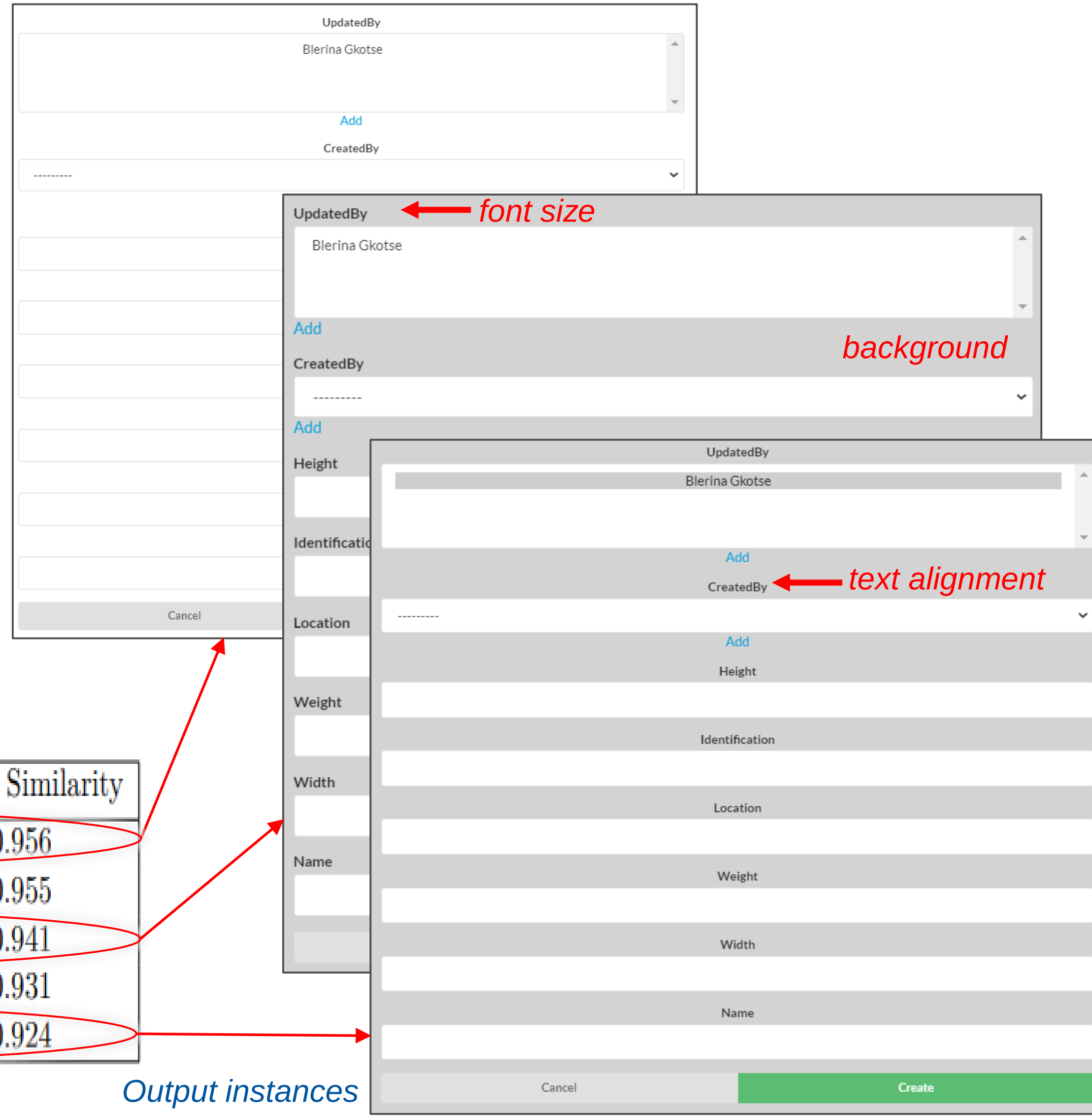


~

Input instance
owao:ui_style_white_black_medium_left

Instances	Cosine Similarity
owao:ui_style_white_black_medium_center	0.956
owao:ui_style_light_grey_black_medium_left	0.955
owao:ui_style_light_grey_black_large_left	0.941
owao:ui_style_white_black_large_left	0.931
owao:ui_style_light_grey_black_medium_center	0.924

Similar instances using ontowalk2vec BFS



Output instances

Conclusion

- **Interdisciplinary work** bridging the gap between Web Semantics and EPP
- **Automatic generation** of web applications **from ontologies** and KGs**:**
  - GenAppi methodology and software tools developed
  - Use case based on the IEDM ontology (and other ontologies)
- Generated web applications:
  - Based on **web semantic technologies**, enabling data integration
  - **Functional** (as is) or **starting point (**for further software development)
- Ontology embedding (ontowalk2vec):
  - Evaluated with two ontologies
  - **Some evidence of better accuracy** than state of the art
  - Generating OWAO **embeddings** and **finding similarities in instances**
  - Used for **recommending personalized UIs**

Perspectives

- IEDM:
  - Further **validation** by the EPP community
  - **Standardisation** of data management for irradiation experiments
  - **Common tools** in the EPP community
- GenAppi methodology:
  - **More UI components** to be integrated in OWAO
- ontowalk2vec:
  - **Collection of data** regarding actual UI preferences
  - Evaluation with **larger datasets**
  - **Extended use** (irradiation experiments similarities – IEDM)

Ontology-based Generation of Personalised Data Management Systems: an Application to Experimental Particle Physics

PhD thesis defense of Blerina GKOTSE
Blerina.Gkotse@cern.ch

MINES ParisTech, PSL University, France

25 September 2020

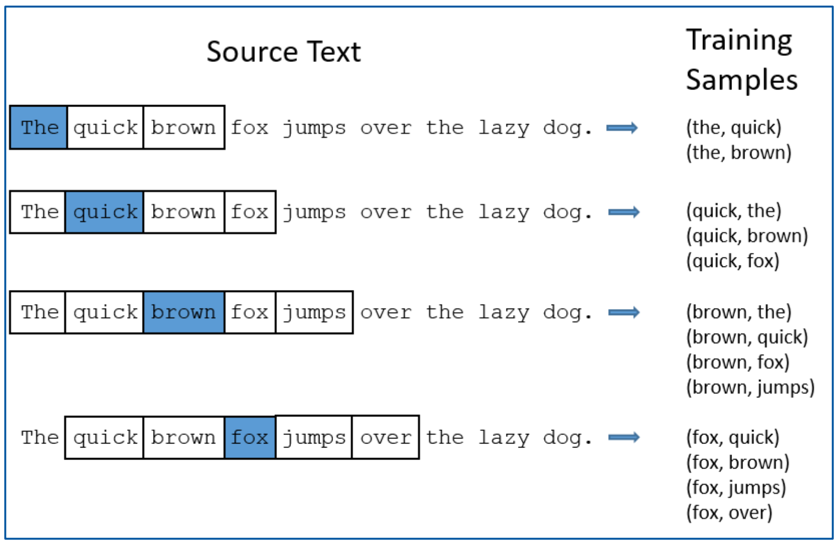
Jury:
Laura GONELLA, Rapporteur
Jean-Baptiste LAMY, Rapporteur
Laurent DUSSEAU, Examinateur
Theodora VARVARIGOU, Examinateur
Pierre JOUVELOT, Directeur de thèse
Federico RAVOTTI, Maître de thèse

This work has received funding from the European Union's Horizon 2020 Research and Innovation program under Grant Agreement no. 654168.

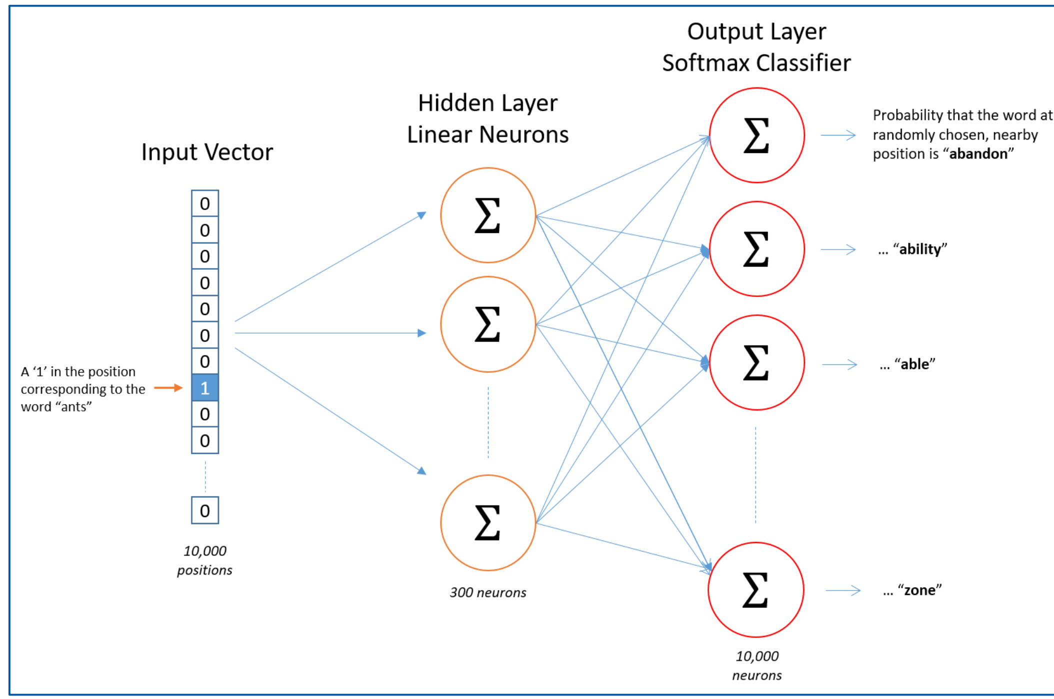
Backup: word2vec - Skip-gram

Given a sequence of training words $w_1, w_2, w_3, \ldots, w_T$, maximize log probability:

$$\frac{1}{T}\sum_{t=1}^{T}\sum_{-c\leq j\leq c, j\neq 0}\log p(w_{t+j}|w_t)$$



*Example for training word sentences**



*Word2Vec Neural Network **

*McCormick, C. (2016, April 19). *Word2Vec Tutorial - The Skip-Gram Model*. Retrieved from http://www.mccormickml.com

Backup: NLP Model node2vec

Ontologies considered as directed graphs → need for extracting **random walks** in the graph to create "sentences".

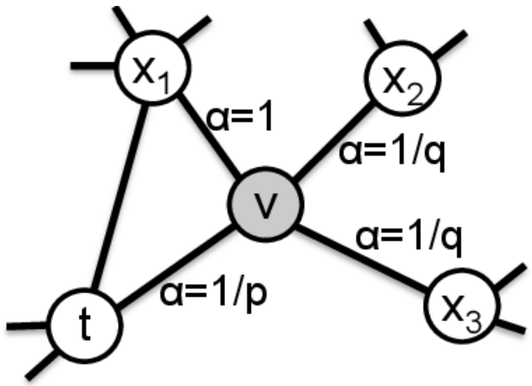
- Two algorithms for traversing graphs:
  - Breadth-first Search (BFS)
  - Depth-first Search (DFS)
- Based on hyperparameters:
  - p: controlling the probability that a node in the walk is revisited
  - q: controlling the choice whether a BFS or DFS approach should be employed
- word2vec for generating the final embeddings

$$P(c_i = x \mid c_{i-1} = v) = \begin{cases} \frac{\pi_{vx}}{Z} & \text{if } (v,x) \in E \\ 0 & \text{otherwise} \end{cases}$$

π_{vx} is the unnormalized transition probability between nodes v and x, and Z is the normalizing constant.

$$\alpha_{pq}(t,x) = \begin{cases} \frac{1}{p} & \text{if } d_{tx} = 0 \\ 1 & \text{if } d_{tx} = 1 \\ \frac{1}{q} & \text{if } d_{tx} = 2 \end{cases}$$

$\pi_{vx} = \alpha_{pq}(t, x) \cdot w_{vx}$
d_{tx} = shortest path distance between t and x



The walk transitions from t to v and is evaluating its next possible steps out of node v

*A. Grover and J. Leskovec, "node2vec: Scalable feature learning for networks"

Backup: Hyperparameters

Hyperparameter	Explanation	Default	Range	Comments
alpha	The initial learning rate	0.025	[0.1, 0.05,0.025,0.01]	Too big learning rate may not converge, too small may get to a local minimum
iteration	Number of iterations (epochs) over the corpus	5	[1,5,20,50]	The more iteration, the better the model is trained
vector size	Dimensionality of the word vectors	100	[20,100, 200, 500]	Bigger vector space the more features are learned for each word
window	Maximum distance between the current and predicted word within a sentence	5	[3,5,7]	since we are using RDF triples it would be wise to keep a window of min 3 words, and we add 2 more levels about the preferences and styles
min_count	Ignores all words with total frequency lower than this.	5	[0,1]	Since there are not many data and repeating instances, we don't want to remove too many data
negative / hierarchical softmax	If negative > 0, negative sampling will be used, the int for negative specifies how many "noise words" should be drawn (usually between 5-20). If set to 0, no negative sampling is used. If negative = 0, hierarchical softmax is used	5	[0,1,5,10]	Since we don't have too many noisy words, we try to run our algorithm with low negative sampling
depth	maximum level traversing the graph	1	[1,2,3]	After 3, too many and long random walks generated and the training becomes too heavy

Backup: Hyperparameters

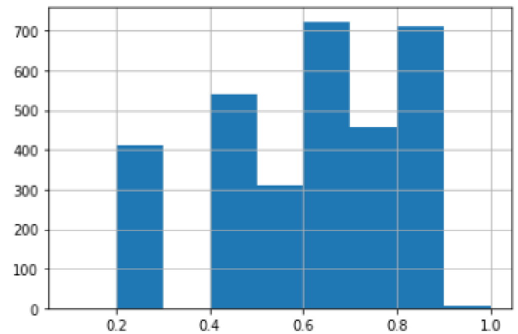


Figure 6.12: BFS F1 score histogram

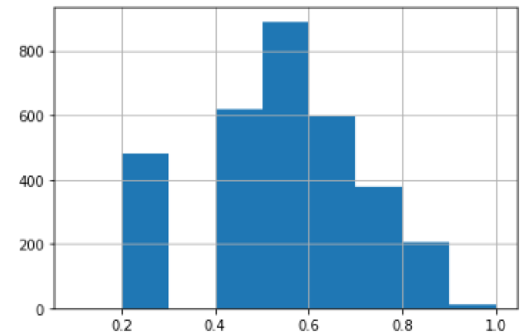


Figure 6.13: WL F1 score histogram

Hyperparameters								Evaluation metrics		
win	vs	hs	lr	iter	neg	cnt	dep	rec	prec	F1
3	200	1	0.010	1	0	1	1	1	0.85	0.923
7	100	0	0.025	1	5	0	2	1	0.857	0.923
5	500	0	0.025	1	20	0	2	1	0.857	0.923
7	**500**	**0**	**0.025**	**1**	**10**	**0**	**2**	1	0.857	0.923
5	200	1	0.010	1	0	1	1	1	0.857	0.923
5	100	10	0.025	1	5	0	2	1	0.857	0.923
7	500	0	0.010	5	5	1	1	1	0.857	0.923
7	20	0	0.025	50	1	0	2	0.833	1	0.909

Table 6.19: Top F1 score (> 0.9) for BFS

Hyperparameters								Evaluation metrics		
win	vs	hs	lr	iter	neg	cnt	dep	rec	prec	F1
7	100	0	0.010	5	5	1	1	1	0.857	0.923
7	**500**	**0**	**0.025**	**1**	**10**	**0**	**2**	1	0.857	0.923
3	500	0	0.050	1	5	0	1	1	0.857	0.923
5	500	0	0.025	1	20	1	1	1	0.857	0.923
5	20	0	0.025	1	20	1	1	1	0.857	0.923
7	20	0	0.025	1	10	1	1	1	0.857	0.923
7	100	0	0.025	1	10	0	2	1	0.857	0.923
3	100	0	0.050	1	5	0	1	1	0.857	0.923
7	20	0	0.010	5	5	1	1	1	0.857	0.923
3	20	0	0.050	1	5	1	1	1	0.857	0.923

Table 6.20: Top F1 score (> 0.9) for WL

Backup: Metrics

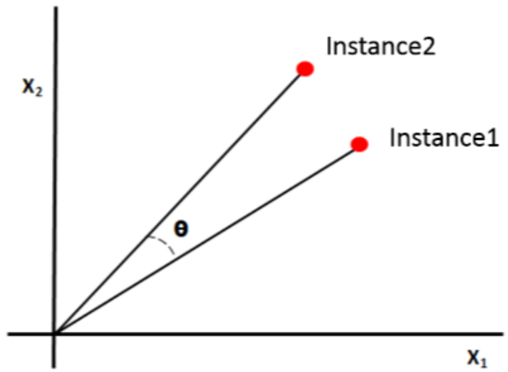
		Actual	
		Positive	Negative
Predicted	Positive	True Positive	False Positive
	Negative	False Negative	True Negative

$$\text{Precision} = \frac{||\text{True Positives}||}{||\text{True Positives}|| + ||\textit{False Negatives}||}$$

$$\text{Recall} = \frac{||\text{True Positives}||}{||\text{True Positives}|| + ||\textit{False Positives}||}$$

$$\text{F1 score} = 2 * \frac{Precision * Recall}{Precision + Recall}$$

Backup: Cosine Similarities Example

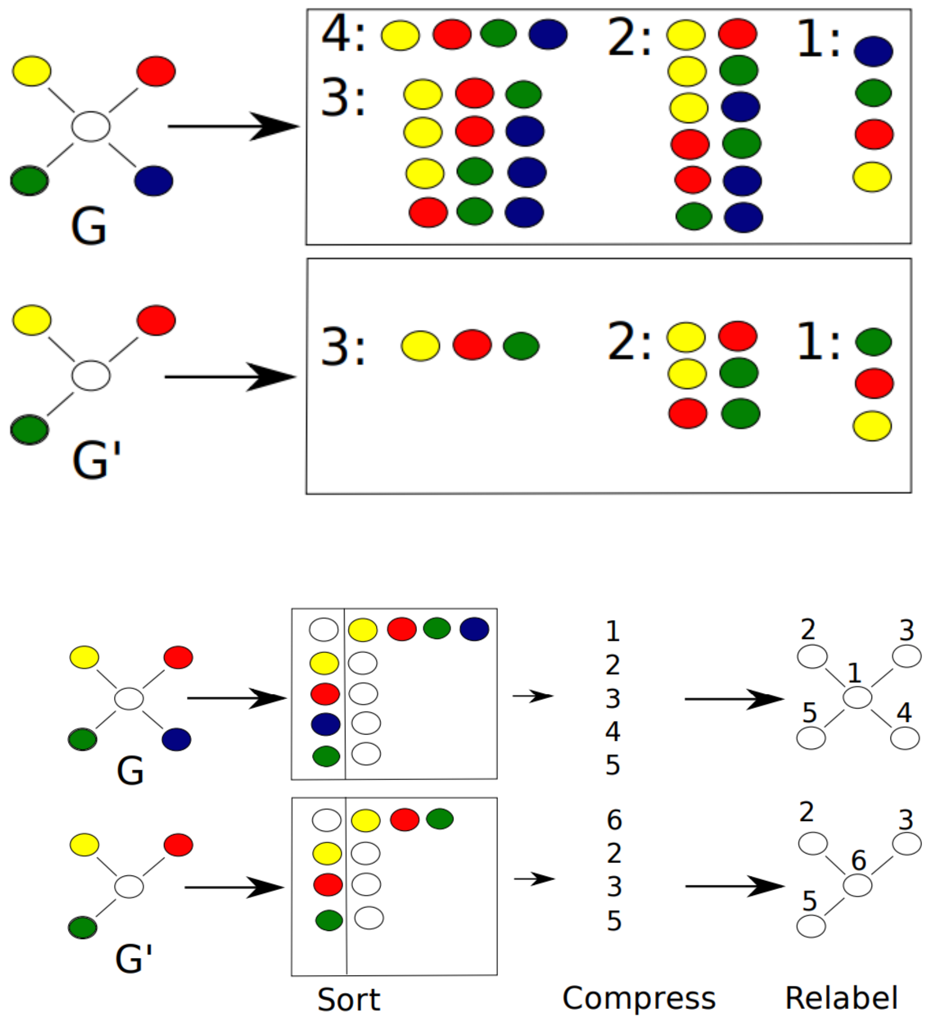


$$cos(\theta) = \frac{A.B}{\|A\|\|B\|}$$

```
[[1.   0.75 0.75 ... 0.25 0.   0.  ]
 [0.75 1.   0.75 ... 0.   0.25 0.  ]
 [0.75 0.75 1.   ... 0.   0.   0.25]
 ...
 [0.25 0.   0.   ... 1.   0.75 0.75]
 [0.   0.25 0.   ... 0.75 1.   0.75]
 [0.   0.   0.25 ... 0.75 0.75 1.  ]]
```

Cosine similarity table

Backup: Weisfeiler Lehman Algorithm



Backup: Recommenders

- Content-based methods:

A model for user-item interactions where items or users' explicit features are provided

+ Advantages: User independence, transparency, no cold start
- Disadvantages: Overspecialisation

- Collaborative filtering methods:
  - Model-based: A model for user-item interactions where users and items representations are learned from interaction matrix
  - Memory-based: Depending on similarities between users or items in terms of observed interactions

+ Advantages: It may suggest more different items
- Disadvantages: It requires a lot of interactions, users and item ratings

Backup: *GenAppi* Algorithm

Algorithm 1 GenAppi

Input: *domain_ontology*

Output: client code and server

```
load domain_ontology, owao
load Jinja_templates
map_domain_ontology_to_model(domain_ontology, owao)
create URL_file
for op in owao.Operations() do
    create view_file(op)
    get template(op) from Jinja_templates
    for cl in domain_ontology.classes() do
        adjust template(op) to cl
        view_file(op).append(cl.template(op))
        create cl.URL(op)
        URL_file.append(cl.URL(op))
    end for
end for
WebVOWL_JSON_file ← owl2vowl(domain_ontology)
assemble files in directory
migrate Model
start server
```

Backup: *ontowalk2vec* Algorithm

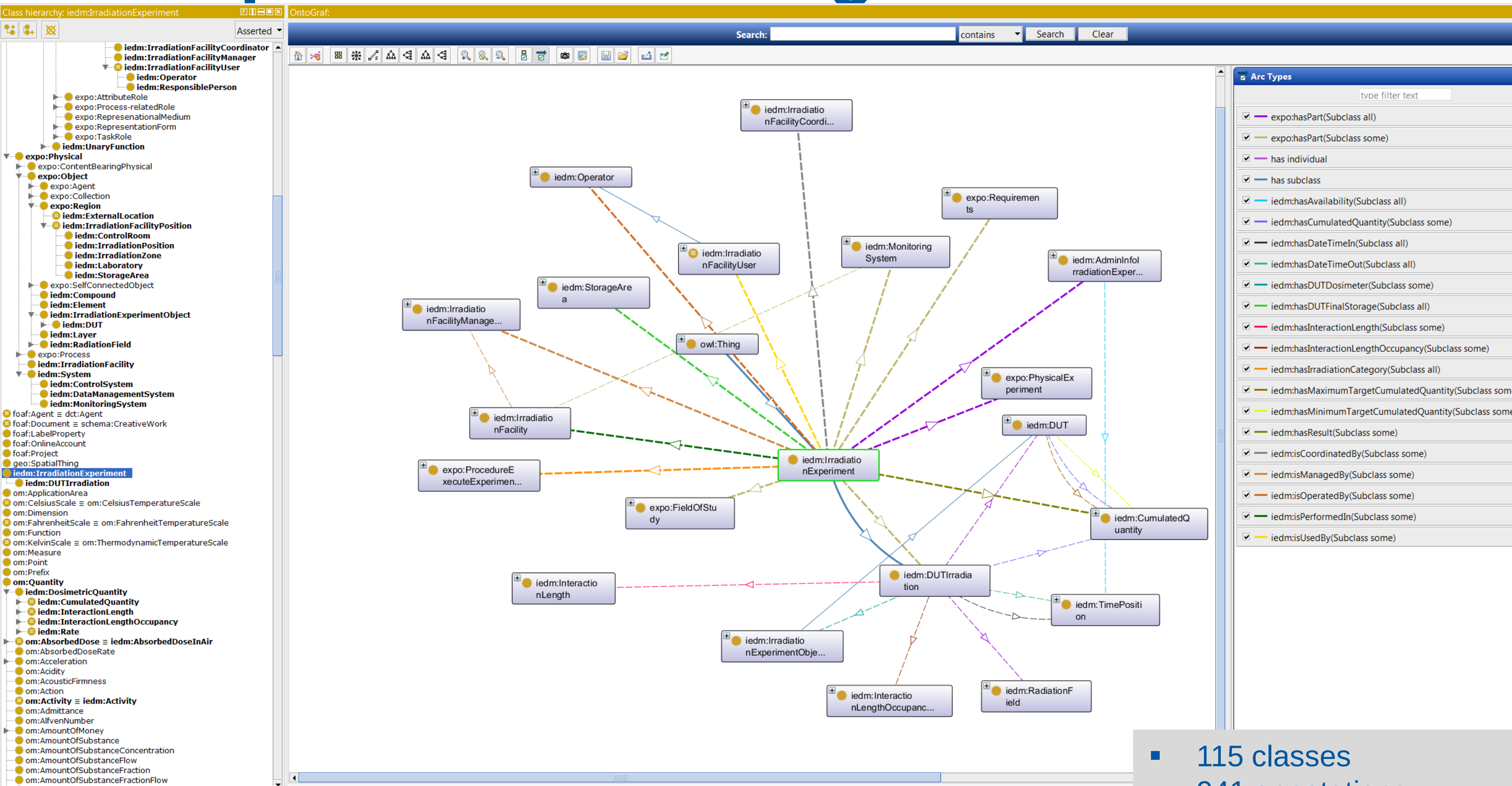
Algorithm 2 ontowalk2vec and classification

Input: *ontology*, *training_data*, *test_data*

Output: ontology embeddings and classification analysis data

```
Read ontology, training_data, test_data
Separate training_data to training_members and training_labels
Separate test_data to test_members and test_labels
graph ← ontology_to_graph_conversion(ontology)
node2vec_graph ← node2vec.graph(graph, directed, p, q)
node2vec_graph.preprocess_transition_probabilities()
node2vec_walks ← node2vec_graph.simulate_walks()
bfs_walks ← RDF2VecBFS(graph, training_members ∪ test_members)
bfs_embeddings ← word2vec(bfs_walks ∪ node2vec_walks)
wl_walks ← RDF2VecWL(graph, training_members ∪ test_members)
wl_embeddings ← word2vec(wl_walks ∪ node2vec_walks)
for embeddings in [bfs_embeddings, wl_embeddings] do
    Separate embeddings to training_embeddings and test_embeddings
    rf ← RandomForestClassifier()
    svm ← SVM()
    for classifier in [rf, svm] do
        classifier.fit(training_embeddings, training_labels)
        classifier.predictions ← classifier.predict(test_embeddings)
        classifier.accuracy ←
                classifier.accuracy_score(test_labelspredictions)
        classifier.confusion_matrix ←
                confusion_matrix(test_labels, classifier.predictions)
    end for
    tsne ← TSNE().fit_transform(embeddings)
    Plot tsne
    top_similarities ← embeddings.most_similar(instance_1)
    similarity_value ← embeddings.similarity(instance1, instance_2)
end for
```

Backup: IEDM in Protégé



- 115 classes
- 941 annotations
- 24 object properties
- 16 data properties