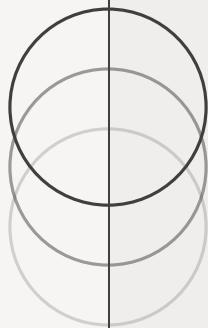




INTRODUCTION TO  
KNOWLEDGE ENGINEERING

# CS RESEARCH KNOWLEDGE GRAPH

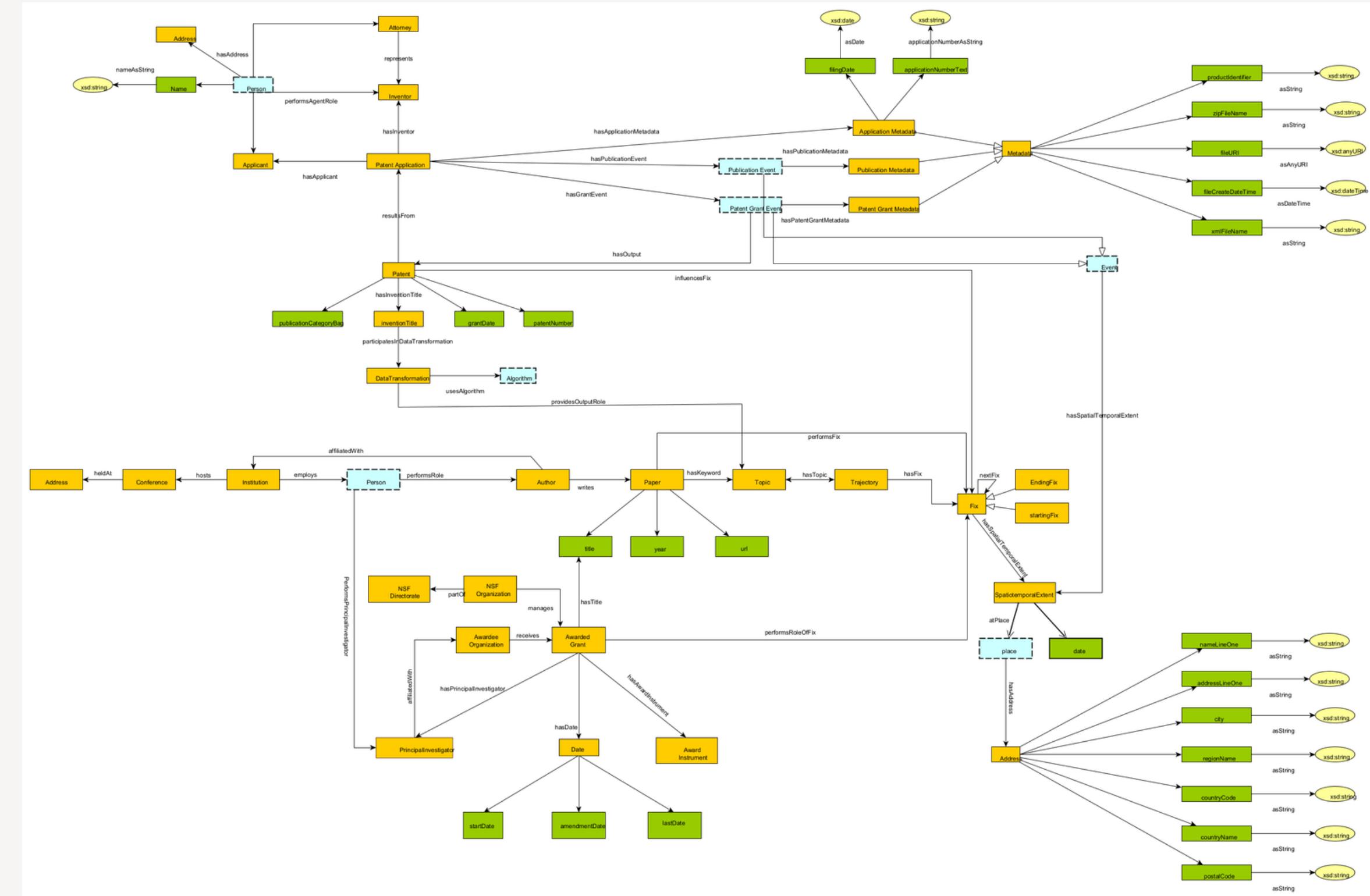
JULIA GRACE M  
MOSES RAJ M  
SUBRITI  
SUMANTH

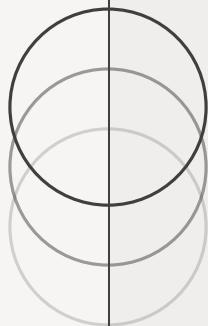


# Introduction

- Research in CS generates thousands of papers, patents, and grants every year.
- All of these are stored in different datasets: DBLP, NSF, USPTO.
- **Challenge:** lack of unified structure for analyzing CS research evolution.
- **Goal:** Create a semantic knowledge graph connecting funding, publications, innovation in the field of CS

# Schema Diagram





# Classes and Relations

- **Core classes:**

Paper, Patent, Grant, Author, Institution, Topic, Trajectory, Fix, SpatiotemporalExtent

- **Key relations:**

writes, performs, manages, hasTopic, hasFix, influences, hasGrantEvent, partOfTrajectory, etc.

# Axiomatization

Axiomatization is the process of defining the core rules and constraints of a knowledge domain so that all other facts can be logically derived.

This screenshot shows the OWLViz interface with the 'Class hierarchy: Paper' tab selected. The left sidebar lists various ontology classes. The main pane displays the inheritance chain for the 'Paper' class, showing it inherits from 'owl:Thing' and numerous other classes like 'Address', 'Algorithm', 'AwardedGrant', etc. A search bar at the top right allows filtering by 'this', 'disjoint', or 'named sub/superclasses'. Below the search bar, a list of 22 asserted uses of the 'Paper' class is shown, including associations with 'Author', 'hasKeyword', and 'performsFix'.

This screenshot shows the OWLViz interface with the 'Annotations: performsFix' tab selected. The left sidebar lists various ontology properties. The main pane displays the asserted annotations for the 'performsFix' property, which is asserted to be a 'owl:topObjectProperty'. It lists various predicates such as 'atPlace', 'employs', 'hasAddress', 'hasAmendmentDate', etc. On the right side, there are tabs for 'Characteristics: perfo' (Functional, Inverse functional, Transitive, Symmetric, Asymmetric, Reflexive, Irreflexive) and 'Description: performsFix' (Equivalent To, SubProperty Of, Inverse Of, Domains (intersection), Ranges (intersection), Disjoint With, SuperProperty Of (Chain)).

This screenshot shows the OWLViz interface with the 'Data property hierarchy: owl:topDataProperty' tab selected. The left sidebar lists various ontology properties. The main pane displays a hierarchical list of data properties under 'owl:topDataProperty', including 'addressLineOne', 'amendmentDate', 'applicationNumber', 'city', 'countryCode', 'countryName', 'date', 'fileCreateDateTime', 'fileURI', 'filingDate', 'grantDate', 'inventionTitle', 'lastDate', 'name', 'nameLineOne', 'patentNumber', 'postalCode', 'productIdentifier', 'publicationCategoryBag', 'regionName', 'startDate', 'title', 'URL', 'xmlFileName', 'year', and 'zipFileName'. Each property is represented by a green square icon.

# Why is Axiomatization important?

Because once these axioms are in place, the reasoner can start doing the heavy lifting.

It can infer new relationships, detect errors, and verify whether our model is logically sound—almost like having a built-in quality check for the ontology.

```
### http://www.semanticweb.org/sumanthjampani/ontologies/2025/10/Project/manages
:manages rdf:type owl:ObjectProperty ;
    rdfs:domain :NSFOrganization ;
    rdfs:range :AwardedGrant .

### http://www.semanticweb.org/sumanthjampani/ontologies/2025/10/Project/nextFix
:nextFix rdf:type owl:ObjectProperty ;
    rdfs:domain :Fix ;
    rdfs:range :Fix .

### http://www.semanticweb.org/sumanthjampani/ontologies/2025/10/Project/partOf
:partOf rdf:type owl:ObjectProperty ;
    rdfs:domain :NSFOrganization ;
    rdfs:range :NSFDirectorate .

### http://www.semanticweb.org/sumanthjampani/ontologies/2025/10/Project/participatesInDataTransformation
:participatesInDataTransformation rdf:type owl:ObjectProperty ;
    rdfs:domain :InventionTitle ;
    rdfs:range :DataTransformation .

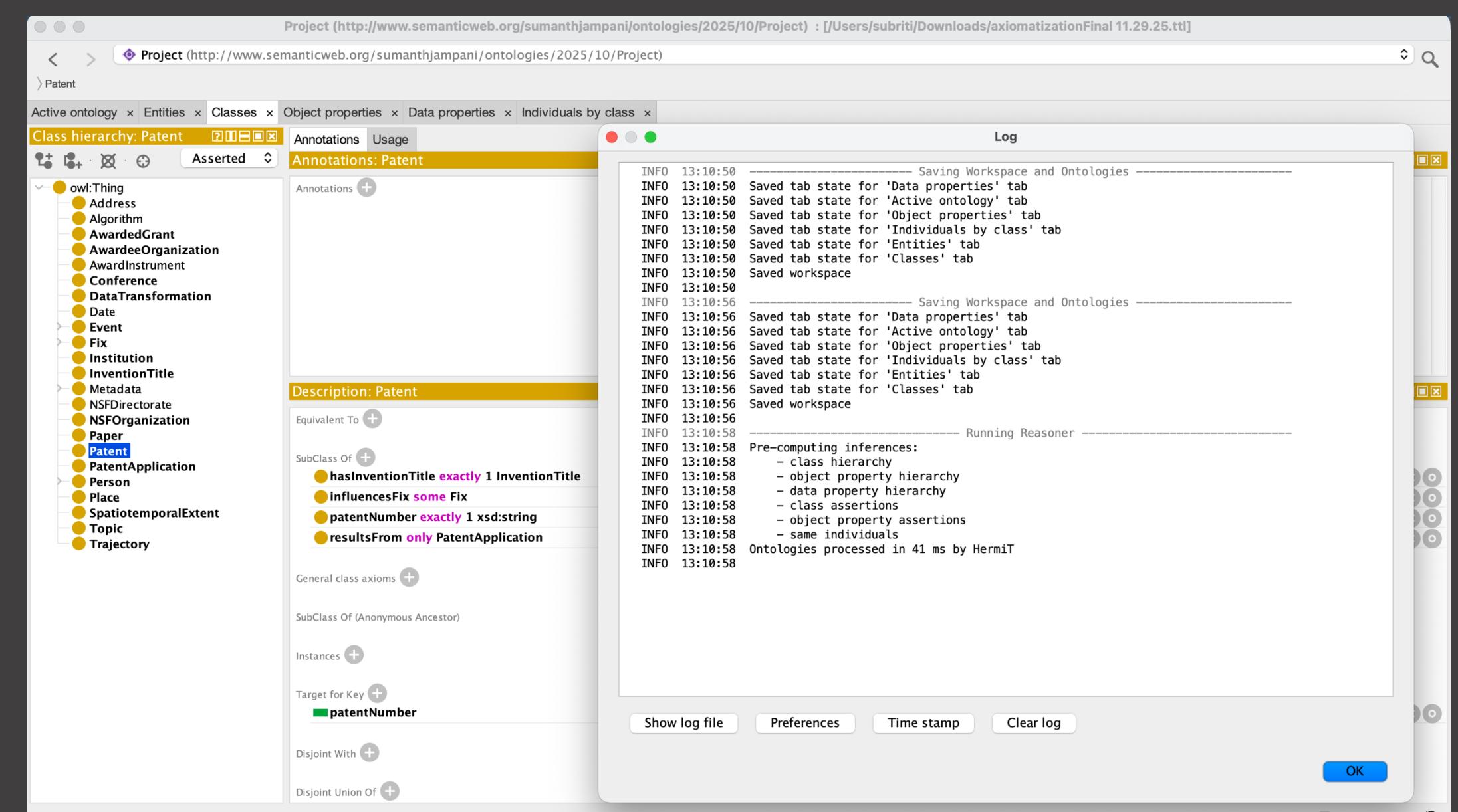
### http://www.semanticweb.org/sumanthjampani/ontologies/2025/10/Project/performsAgentRole
:performsAgentRole rdf:type owl:ObjectProperty ;
    rdfs:domain :Person ;
    rdfs:range :Inventor .

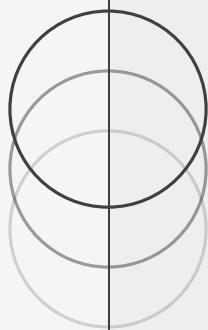
### http://www.semanticweb.org/sumanthjampani/ontologies/2025/10/Project/performFix
:performsFix rdf:type owl:ObjectProperty ;
    rdfs:domain :Paper ;
    rdfs:range :Fix .

### http://www.semanticweb.org/sumanthjampani/ontologies/2025/10/Project/permformsRole
:performsRole rdf:type owl:ObjectProperty ;
    rdfs:domain :Person ;
    rdfs:range :Author .

### http://www.semanticweb.org/sumanthjampani/ontologies/2025/10/Project/permformsRoleOfFix
:performsRoleOfFix rdf:type owl:ObjectProperty ;
    rdfs:domain :AwardedGrant ;
    rdfs:range :Fix .

### http://www.semanticweb.org/sumanthjampani/ontologies/2025/10/Project/providesOutputRole
:providesOutputRole rdf:type owl:ObjectProperty ;
    rdfs:domain :DataTransformation ;
    rdfs:range :Topic .
```





# Data Transformation

## Common Entities

- **Authors**

DBLP → authors of papers

Patents → inventors

NSF → Principal Investigators (PI/Co-PI)

- **Research Topics / Keywords**

DBLP → extracted from paper titles/keywords

Patents → extracted from abstracts/claims

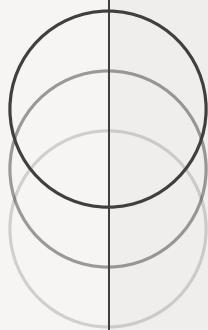
NSF → topics from award abstracts/keywords

- **Temporal Information**

DBLP → publication year

NSF → award year & duration

Patents → filing date, grant date



# Data Transformation

## Data Cleaning

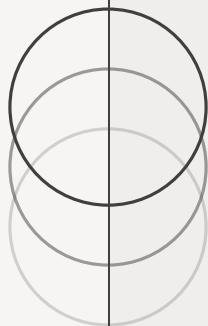
- Convert all dates to a single format
- Standardize author/institution names
  - Normalize topics/keywords

## Topic Extraction

- TF-IDF keyword extraction
- Contextual embeddings (BERT)
  - Topic modeling

## Clustering to Find Topic Groups

- Group similar papers, patents, and NSF awards together
  - Creates unified topic clusters



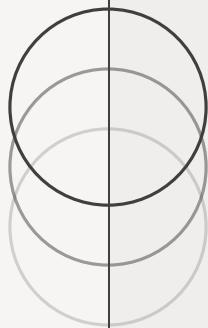
# Competency Questions

**1Q.** Which institutions have received the most NSF grants related to AI research?

**Ans)** Grants are identified through the hasKeyword relation connecting a Paper to a Topic.

Each Grant is then linked to an Institution via the awardedTo relation.

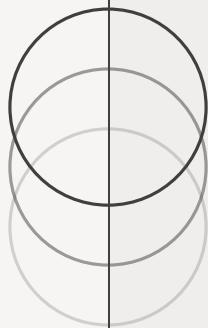
Counting the number of grants for a particular Topic gives the answer of how much funding is going into one Research Topic.



## 2Q. How has CS topics evolved over time?

**Ans)** Each Topic has a Trajectory, made up of time-ordered Fixes, and each Fix corresponds to a Paper with a publication date.

By aggregating these fixes along the trajectory, the ontology provides a timeline showing how a CS topic has evolved over time.

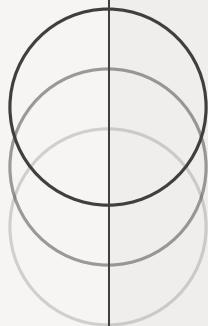


**3Q.** Which NSF projects produced the highest innovation velocity?

**Ans)** The number of papers and patents that are being released in a particular Topic after a grant is issued is the innovation velocity

Using the awardDate of the grant and the hasDate values of resulting papers/patents, the ontology measures how quickly new outputs appear.

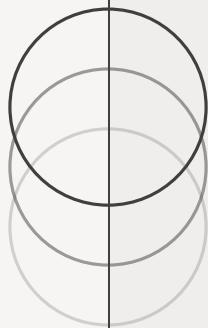
This allows computation of innovation velocity.



**4Q.** Which authors frequently publish in both academic venues (conferences/journals) and also file patents?

**Ans)** The schema links an Author to Paper and Inventors to their Patents.

Person who appear in both sets—those who have writes → Paper (Venue) and files → Patent (hasInventor)—are identified as publishing academically while also generating patents.

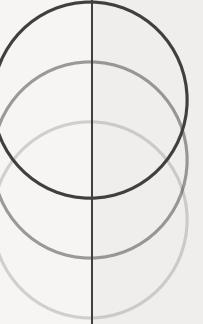


**5Q.** Which NSF-funded areas have produced patents most rapidly (shortest time lag)?

**Ans)** Each Grant is linked to one or more Topics and to resulting Patents.

By comparing the awardDate of a grant with the hasDate of patents connected to it, the ontology can compute the shortest time lag.

This shows which NSF-funded areas generate patents the fastest.



**THANKYOU**