



# Building Interoperable Knowledge Graphs with X3ML Framework







Yannis Marketakis

Institute of Computer Science, Foundation for Research and Technology – Hellas, Heraklion, Greece  
Computer Science Department, University of Crete, Heraklion, Greece

Tutorial presented in the 14<sup>th</sup> International Joint Conference on Knowledge Graphs (IJCKG 2025)

## Outline



	Introduction	4-7	5'
	X3ML Framework	8-18	10'
	X3ML Software	19-44	30'
	Looking to the Future	45-54	7'
	Conclusion	55-58	3'
	Q&A Session	59	5'

# Supplementary Resources

[Description](#) [Learning Objectives](#) [Format and Schedule](#) [Audience](#) [Presenter](#) [Material](#)

## Building Interoperable Knowledge Graphs with X3ML Framework

Tutorial at IJCKG 2025, October 16, 2025

### Description

This tutorial introduces participants to the X3ML Framework, a suite of specifications and tools for constructing knowledge graphs from structured data. The tutorial will provide both conceptual foundations and hands-on demonstrations, guiding attendees through the workflow of defining schema mappings and transforming source data into RDF aligned with domain ontologies. At the core of the framework is a declarative, technology-agnostic mapping language that supports collaborative and maintainable schema mapping practices. Participants will learn how X3ML facilitates not only the creation of semantic mappings but also the exploration and verification of the resulting knowledge graphs to ensure correctness and completeness. The session will further highlight domain-independent applications of X3ML across fields such as cultural heritage and biodiversity, while also showcasing emerging methods that integrate Large Language Models (LLMs) to accelerate and simplify schema mapping tasks.

### Learning Objectives


By the end of this tutorial participants will be able to:

- understand the principles and challenges of transforming heterogeneous source data into RDF knowledge graphs
- apply the X3ML mapping language and associated tools to design, implement and validate schema mappings
- gain hands-on experience in generating semantically rich knowledge graphs and verifying their quality
- explore innovative techniques leveraging Large Language Models (LLMs) to reduce manual effort and enhance the schema mapping process

Equipped with this knowledge, participants will be able to effectively use X3ML framework in real-world scenarios to build interoperable, semantically robust knowledge graphs across diverse application domains.


### Format and Schedule

The tutorial will be held on **October 16, 2025** as part of the **IJCKG 2025 conference**, and it will be structured to balance conceptual understanding with practical application. It will begin with a slide-based presentation introducing the core concepts of knowledge graph construction and the X3ML framework. This will be followed by a live online demonstration, showcasing the workflow of schema mapping and RDF generation using the framework tools and resources. Finally, participants will engage in guided hands-on exercises to practice defining schema mappings and transforming data collections themselves. The entire session will last one hour, ensuring an engaging and focused learning experience that combines theory, demonstration and active participation within the conference program.

 **X3ML-Tutorial** Public


[Pin](#) [Watch 0](#) [Fork 0](#) [Star 0](#)

[main](#) [1 Branch](#) [0 Tags](#)  [Add file](#) [Code](#)

 **ymark** Update README.md ✓ bd39f0a · last week 44 Commits

assets	initial import	3 weeks ago
exercises	Update README.md	last week
images	initial import	3 weeks ago
LICENSE.txt	initial import	3 weeks ago
README.md	Update README.md	last week
index.html	changed header title	2 weeks ago

[README](#) [License](#)

 **X3ML-Tutorial**

This repository contains resources that accompany the **tutorial on knowledge graph construction using the X3ML Framework**, presented at **IJCKG 2025**. It contains exercises that guide users getting familiar with X3ML framework.

### Learning Objectives

Through these [exercises](#), participants will learn how to:

- Design schema mappings using the **X3ML language**
- Use the **3M mapping editor** to define correspondences between source data and ontology concepts
- Apply mappings with the **X3ML Engine** to generate **RDF** aligned with domain ontologies
- Explore the transformed RDF data in a user-friendly manner

### About

A tutorial about X3ML framework

- [Readme](#)
- [View license](#)
- [Activity](#)
- [0 stars](#)
- [0 watching](#)
- [0 forks](#)


### Releases

No releases published  
[Create a new release](#)

### Packages





No packages published  
[Publish your first package](#)

### Deployments 39

 **github-pages** last week

[+ 38 deployments](#)

### Languages

 <b>CSS</b> 42.0%	 <b>SCSS</b> 37.0%
 <b>JavaScript</b> 13.9%	 <b>HTML</b> 7.1%

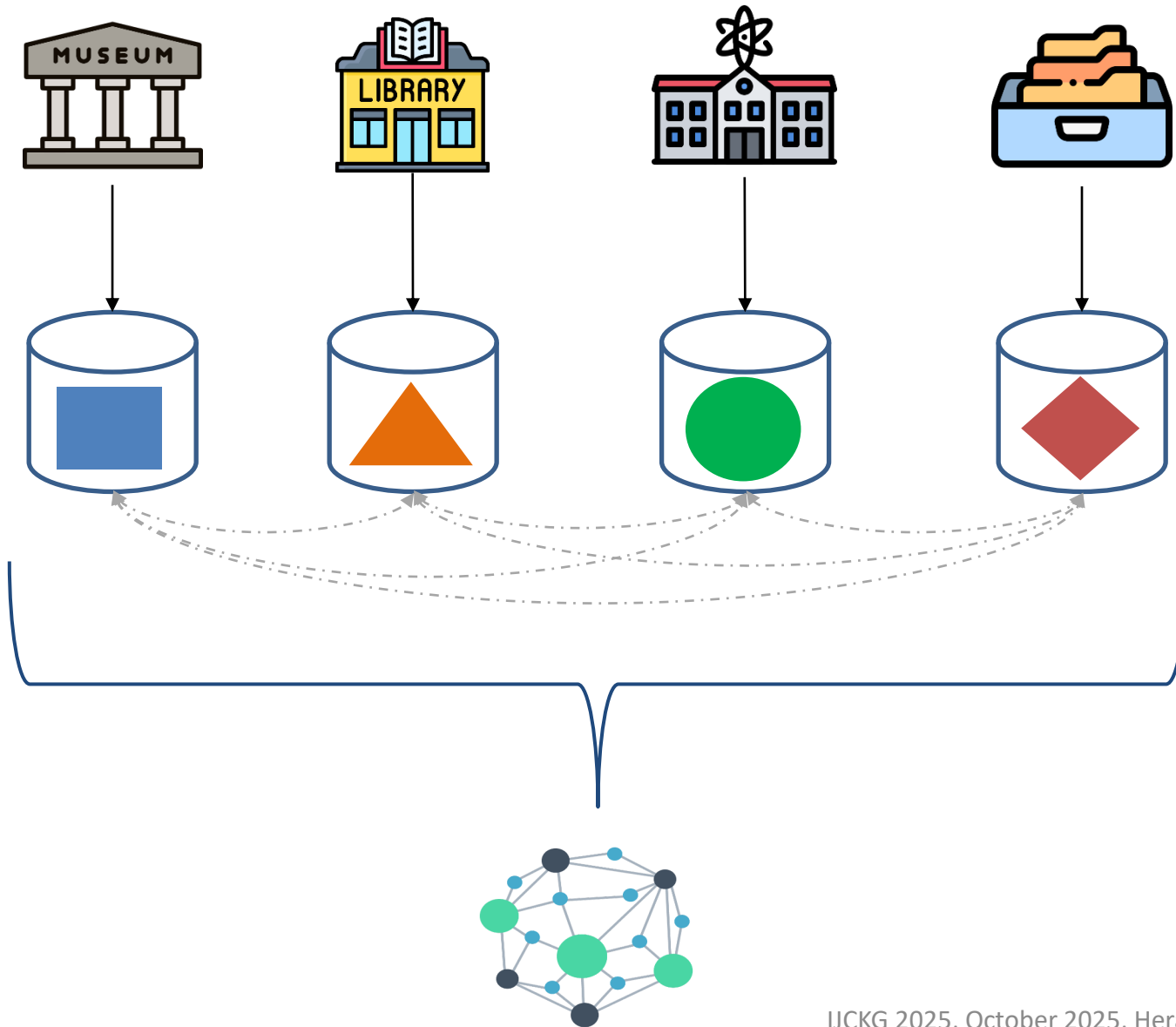
<https://ymark.github.io/X3ML-Tutorial/>



<https://github.com/ymark/X3ML-Tutorial>



# Introduction / Motivation



- **Fragmented Reality**
  - Institutions curate diverse collections using different metadata standards, languages and practices
- **The Challenge**
  - This heterogeneity limits data sharing, interoperability, and cross-domain discovery
- **The Goal**
  - Integrate and aggregate metadata into unified, semantically rich knowledge graphs
- **The Benefit**
  - Enable richer research, complex query answering, reuse and share resources

# Introduction / Motivation

## The role of schema mappings

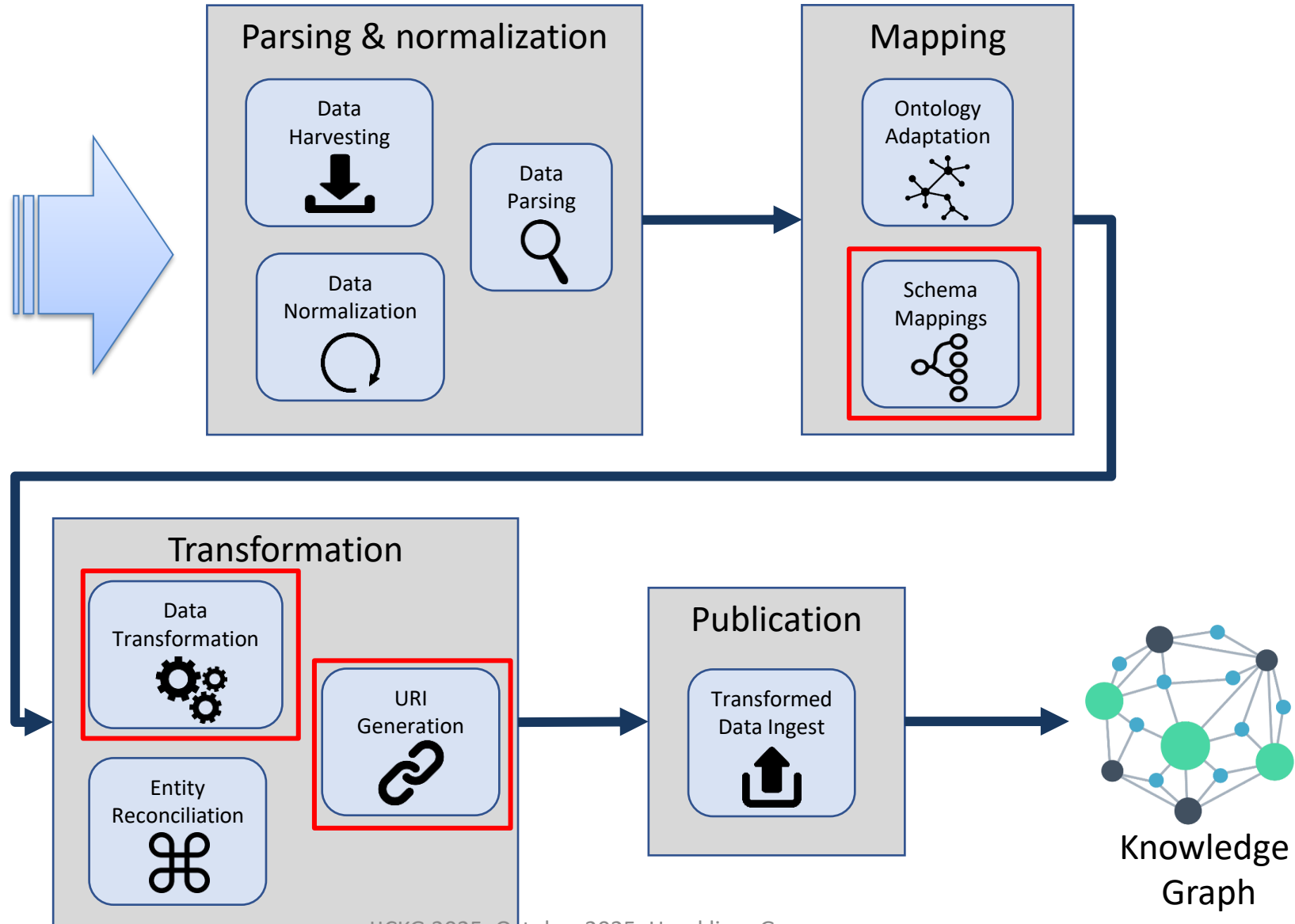
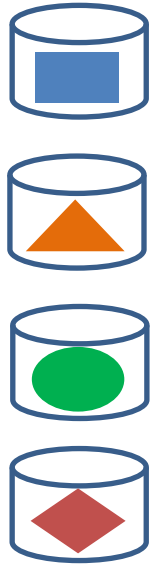
- Data sources use multiple schemas and formats to describe their contents
  - They describe similar concepts using different structures and terminologies
- To semantically integrate them we need
  - **Schema mappings**
    - Describe the correspondences between the source data elements and the target ontology concepts
  - **Data transformation tools**
    - Apply schema mappings to generate RDF representations of the original data sources aligned with a common target model (i.e. ontology)



# Introduction / Motivation

## The Data Transformation Workflow

Data Sources

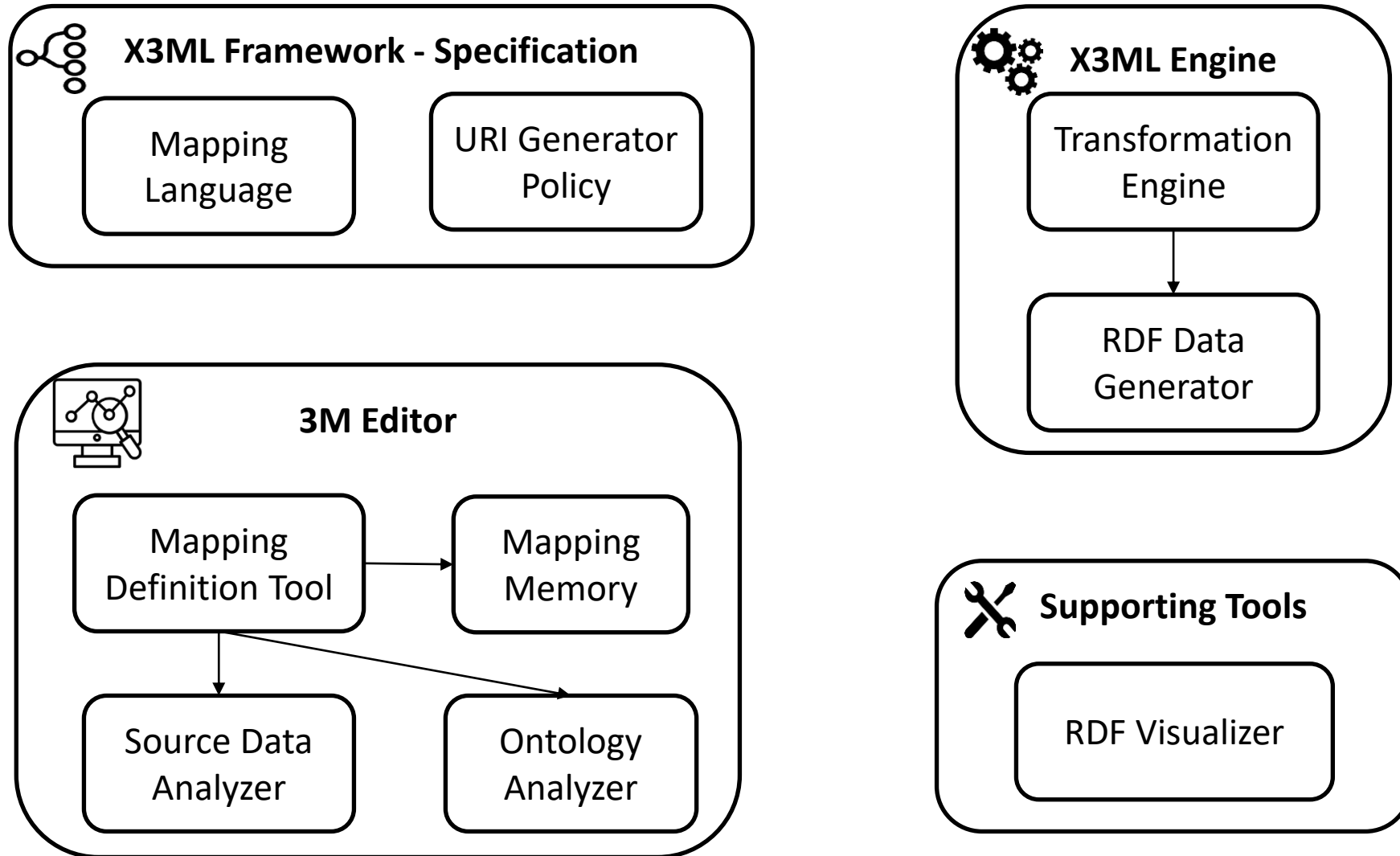


## Tutorial Objectives






- Understanding the core definitions and the role of schema mappings in data integration
- Design schema mappings using **X3ML language**
- Use tools (**3M**, **X3ML Engine**) to practice with schema mapping definition and data transformation
- Experiment with different methods for assigning URIs (Uniform Resource Identifier) and labels
- Construction of **interoperable knowledge graphs** from diverse data sources
- Assess and evaluate **data transformation results** for semantic correctness and completeness
- Explore how Large Language Models (LLMs) can **assist** or **automate** parts of the schema mapping process, reducing manual effort and accelerating semantic data integration

# X3ML Framework



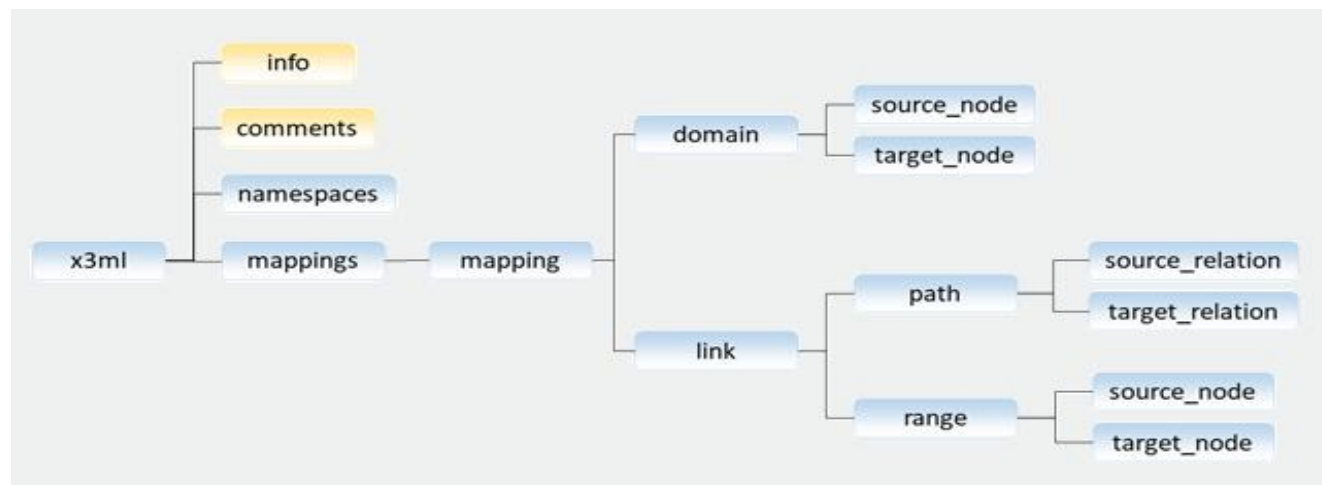




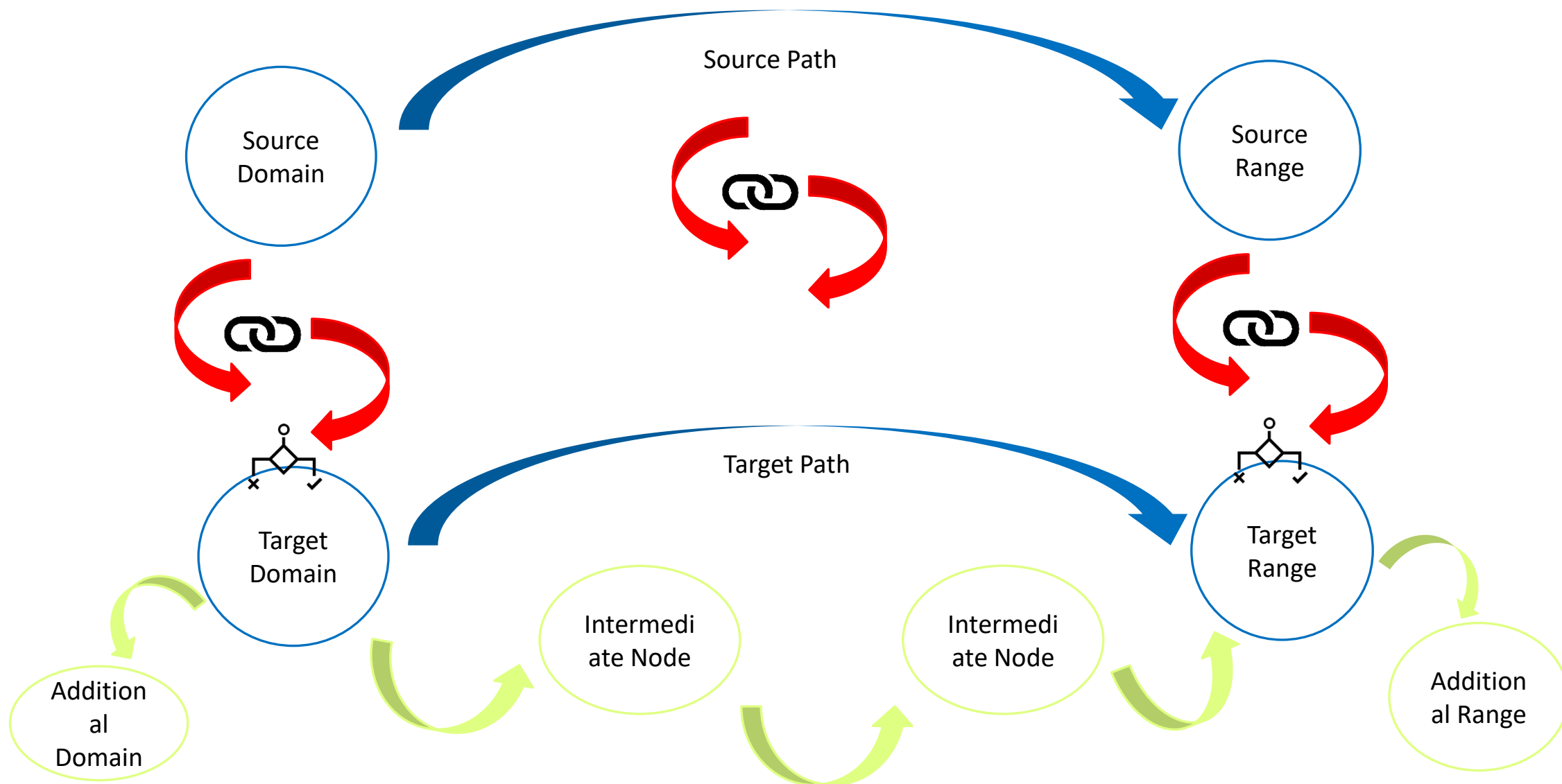
	<b>X3ML Framework - Specification</b>		
	X3ML Mapping Definition Language	9-16	7'
	X3ML Generator Policy Definition	17-18	3'

# X3ML Mapping Definition Language

- X3ML is a declarative, XML based language which describes schema mappings in such a way that they can be collaboratively created and discussed by experts.
- Key Features
  - It provides a declarative way for describing schema mappings
  - Focuses on properly mapping schema resources
  - Decoupled from the URI and values generation process
  - Mappings are described using XML serialization



# X3ML Mapping Structure



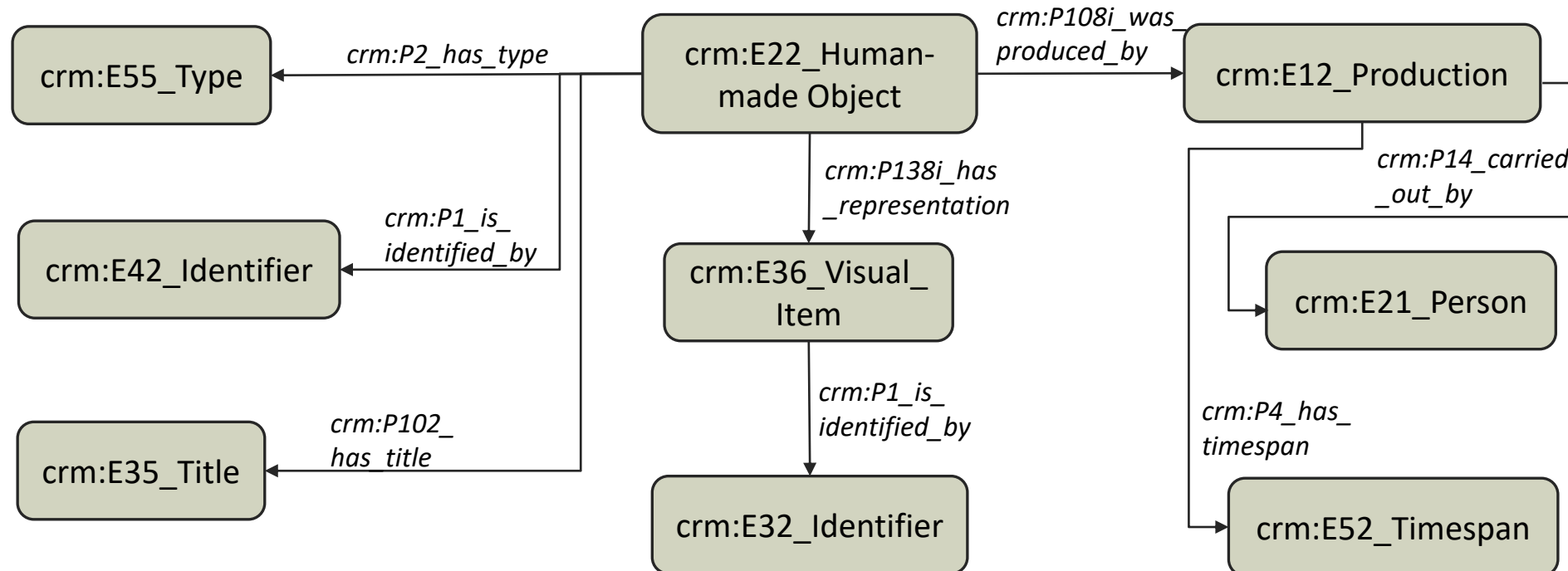
## X3ML Mapping Definition Constructs

X3ML supports **1:N mappings** and uses the following special constructs:

- **Intermediate nodes** used to represent the mapping of a simple source path to a complex target path.
- **Constant expression nodes** used to assign constant attributes to an entity.
- **Conditional statements** within the target node and target relation support checks for existence and equality of values and can be combined into Boolean expressions.
- **“Same as” variable** used to identify a specific node instance for a given input record that is generated once but is used in a number of locations in the mapping.
- **Join operator (==)** used in the source path to denote relational database joins.
- **Info and comment blocks** throughout the mapping specification bridge the gap between human author and machine executor.

# X3ML Mapping Definition Language Example

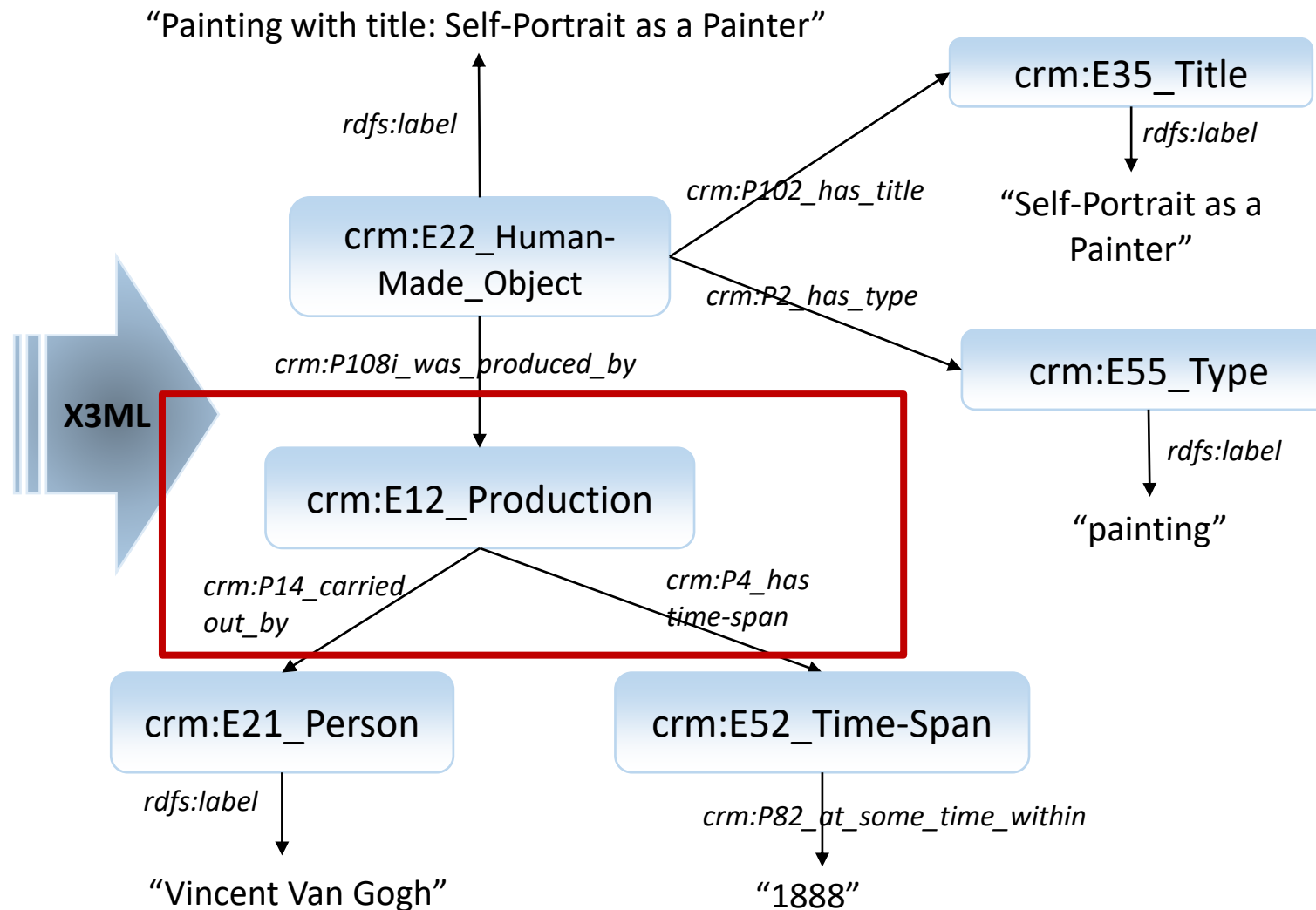
id	Painting title	painter	Creation date	Filename
p-1	Self-Portrait as a Painter	Vincent van Gogh	1888	p1.jpg
p-2	The Starry Night	Vincent van Gogh	1889	p2.jpg
p-3	The Ballet Class	Edgar Degas	1874	p3.jpg



# X3ML Mapping Definition Language

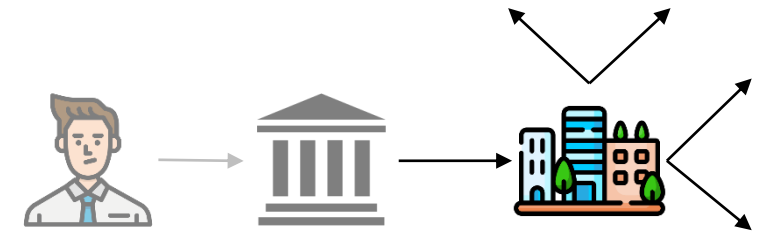
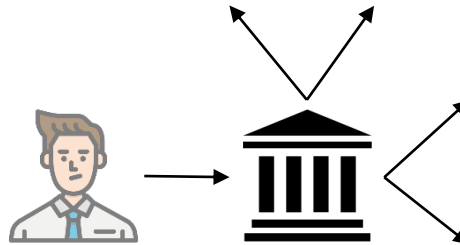
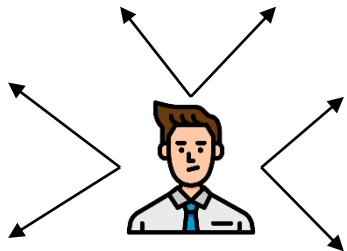
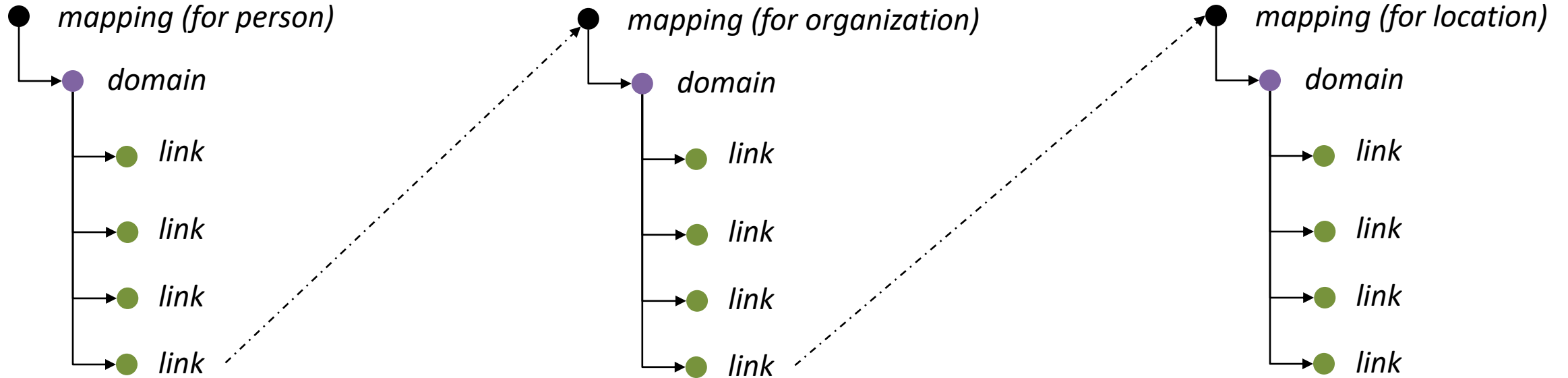
## Example

```
<root>
  <painting>
    <id>p-1</id>
    <painter_id>a-1</painter_id>
    <title>Self-Portrait as a Painter</title>
    <creation_date>1888</creation_date>
    <filename>file1.jpg</filename>
  </painting>
  <painting>
    <id>p-2</id>
    <painter_id>a-1</painter_id>
    <title>The Starry Night</title>
    <creation_date>1889</creation_date>
    <filename>file2.jpg</filename>
  </painting>
  <painting>
    <id>p-3</id>
    <painter_id>a-2</painter_id>
    <title>The Ballet Class</title>
    <creation_date>1889</creation_date>
    <filename>file3.jpg</filename>
  </painting>
</root>
```



# X3ML Mapping Definition Language

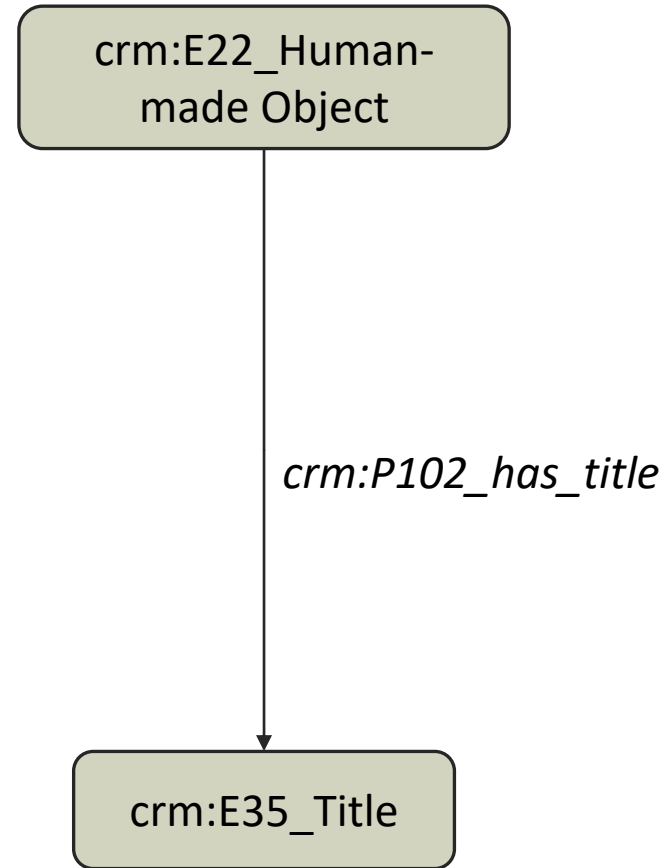
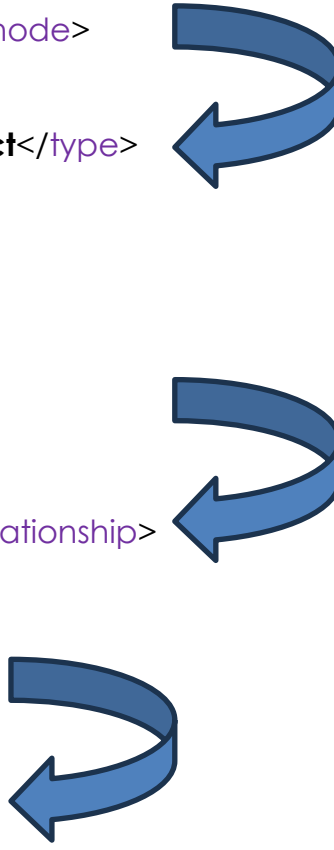
## Structure of mappings



# X3ML Mapping Definition Language

## Structure of mappings

```
<mapping>
  <domain>
    <source_node>/root/painting</source_node>
    <target_node>
      <entity>
        <type>crm:E22_Human-Made_Object</type>
      </entity>
    </target_node>
  </domain>
  <link>
    <path>
      <source_relation>
        <relation>title</relation>
      </source_relation>
      <target_relation>
        <relationship>crm:P102_has_title</relationship>
      </target_relation>
    </path>
    <range>
      <source_node>title</source_node>
      <target_node>
        <entity>
          <type>crm:E35_Title</type>
        </entity>
      </target_node>
    </range>
  </link>
</mapping>
```





# X3ML Generator Policy Definition

- Definition of rules for the generation of URIs and labels (i.e. `rdfs:label`)

```
<generator name="LocalTerm" prefix="pref">  
  <pattern> {hierarchy}/resource/{term} </pattern>  
</generator>
```

Use common  
URI prefix

```
<generator name="SimpleLabel">  
  <pattern> {label} </pattern>  
</generator>
```

Use  
parameters  
(within { }) and  
constants

Generation  
using  
templates

```
<generator name="LocalTerm-hashed" prefix="pref" shorten="yes">  
  <pattern> {hierarchy}/resource/{term} </pattern>  
</generator>
```

Create hashed  
values based  
on contents

```
<generator name="LocalTerm-uuid" prefix="pref" uuid="yes">  
  <pattern> {hierarchy}/resource/ </pattern>  
</generator>
```

Declare once,  
use multiple  
times

Create unique  
and random  
URIs

## X3ML Generator Policy Definition – cont'd

```
<painter>
  <id>a-1</id>
  <name>Vincent Van Gogh</name>
</painter>
<painter>
  <id>a-2</id>
  <name>Edgar Degas</name>
</painter>
```

```
<generator name="LocalTerm" prefix="pref">
  <pattern>{hierarchy}/resource/{term}</pattern>
</generator>

<generator name="SimpleLabel">
  <pattern>{label}</pattern>
</generator>
```

```
<instance_generator name="LocalTerm">
  <arg name="hierarchy" type="constant">persons</arg>
  <arg name="term" type="xpath">painter/id/text()</arg>
</instance_generator>
```

```
<label_generator name="SimpleLabel">
  <arg name="label" type="xpath">painter/name/text()</arg>
  <arg name="language" type="constant">en</arg>
</label_generator>
```

http://.../persons/resource/a-1

http://.../persons/resource/a-2

"Vincent Van Gogh"@en

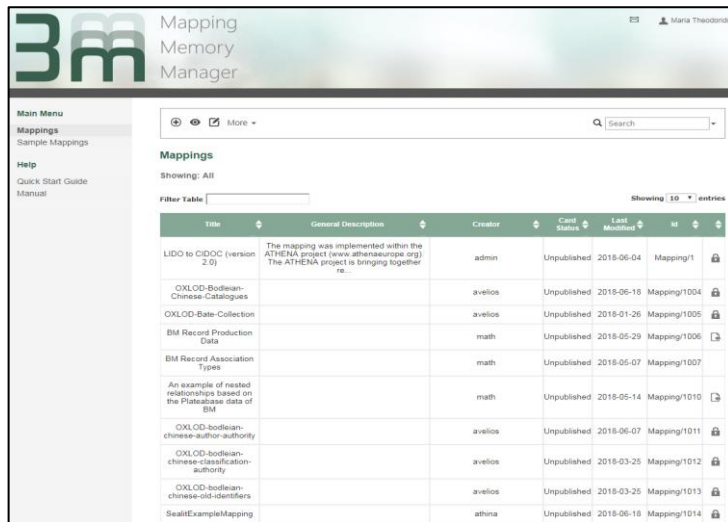
"Edgar Degas"@en



X3ML Software		
	3M Editor	20-4024'
	X3ML Engine	41-424'
	RDF Visualizer	43-442'

# 3M Editor

- Enables the creation of mapping definitions (X3ML) between source and target schemata
- Supports guided mappings by analyzing source resources and target schemata
- Provides user space and mapping storage
- Transforms data (in RDF format) using X3ML Engine



3m Mapping : X3ML Toolkit Tutorial Demo Map			
<a href="#">Info</a> <a href="#">Matching Table</a> <a href="#">Generators</a> <a href="#">Analysis</a> <a href="#">Transformation</a> <a href="#">Configuration</a> <a href="#">About</a>			
▲ TOP ▼ BOTTOM <a href="#">VIEW MODE</a> <a href="#">XML</a>			
(ALL) SOURCES ⇄		(ALL) TARGET PATHS ⇄	
#	SOURCE		TARGET
1	D	./row	E31_Document E33_Linguistic_Object
1.1	P	Type	P2_has_type
	R	Type	E55_Type
1.2	P	Creator	P94i_was_created_by E65_Creation [create1] P14_carried_out_by
	R	Creator	E39_Actor
1.3	P	Creator	P94i_was_created_by E65_Creation [create1] P14_carried_out_by
	R	Creator	E39_Actor
1.4	P	Date	P94i_was_created_by E65_Creation [create1] P4_has_time-span E52_Time-Span P82_at_some_time_within
	R	Date	rdf-schema#Literal
1.5	P	Title	P1_is_identified_by
	R	Title	E41_Appellation
1.6	P	Subtitle	P1_is_identified_by
	R	Subtitle	E41_Appellation
1.7	P	Subject	P129_is_about
	R	Subject	E55_Type
1.8	P	language	P72_has_language
	R	language	E56_Language

# 3M Editor – new version

- Implemented using modern and responsive technologies
- Faster and light-weight (at client side)
- Allows concurrent edits of mappings from different users (a la Google docs)
- In action since 2021

Project - Mappings							
Title	HRID	Author	Status	Date Created	Date Modified	Coverage	User
FIRMS Stock Mappings	278840	Yannis Marketakis	In Progress	Fri Nov 26 2021 12:02 PM	Wed May 22 2024 03:59 PM		
RTI-NEH	756044	Yannis Marketakis	In Progress	Mon Nov 29 2021 01:55 PM	Wed Oct 02 2024 12:58 PM	62%	
ARIADNEplus THANADOS mapping OFFICIAL_(cloned)	670295	Maria Theodoridou	In Progress	Thu Dec 02 2021 03:24 PM	Wed Jun 22 2022 04:35 PM		
A KTP to CIDOC -Person	317807	Athina Kitsotaki	In Progress	Mon Dec 13 2021 11:13 AM	Thu May 23 2024 03:57 PM		
A KTP to CIDOC -Organisation	495892	Athina Kitsotaki	In Progress	Wed Dec 15 2021 10:56 AM	Thu May 23 2024 04:19 PM		
Vincent Van Gogh painting	574665	Yannis Marketakis	In Progress	Wed Feb 09 2022 12:16 PM	Wed Sep 25 2024 03:37 PM		
GRSF-FishingGearsVocabulary	456359	Yannis Marketakis	In Progress	Thu Mar 03 2022 04:02 PM	Mon Mar 14 2022 03:47 PM	94%	
GRSF-SpeciesVocabulary	452869	Yannis Marketakis	In Progress	Fri Mar 04 2022 03:15 PM	Wed Jul 02 2025 12:21 PM	48%	
GRSF-WaterAreasVocabulary	478623	Yannis Marketakis	In Progress	Tue May 17 2022 04:01 PM	Mon Jun 20 2022 02:29 PM	28%	
A KTP to CIDOC -Place	649619	Athina Kitsotaki	In Progress	Thu Jun 09 2022 11:15 AM	Thu May 23 2024 04:43 PM		

Rows per page: 10 1-10 of 81

OPEN CLONE + ADD NEW IMPORT FROM ZIP

Organise Mode Delete Mode

☒ Mapping #1 - Has 3 links (3 links selected) - Named Graph: (Optionally add one)

Domain

☐ Source Node: /root/painting
 

☐ Target Entity: E22\_Human-Made\_Object
 

LocalTermURI

{painting}/{id/text0}

CompositeLabel

{Painting with title:} {title/text0}

Relation: P2\_has\_type

☐ Entity: E55\_Type
 

LocalTermURI

{object\_type}/{painting}

☒ Link #1 (of mapping #1)
 

Source Relation: title

Source Node: title

Target Relation: P102\_has\_title

Target Entity: E35\_Title

LocalTermURI

{title}/{text0}

Literal

text: {text0}

☒ Link #2 (of mapping #1)
 

Source Relation: painter\_id

Source Node: painter\_id

Target Relation: P108i\_was\_produced\_by

Target Entity: E12\_Production

production\_event

UUID

CompositeLabel

{Creation of painting with title:} {title/text0}

Target Relation: P14\_carried\_out\_by

Target Entity: E21\_Person

LocalTermURI

{painter}/{text0}

☒ Link #3 (of mapping #1)
 

Source Relation: creation\_date

Source Node: creation\_date

Target Relation: P108i\_was\_produced\_by

Target Entity: E12\_Production

production\_event

Target Relation: P4\_has\_time-span

## 3M Editor

- Key features
  - Guided mappings by analyzing source records and target schemata (i.e. ontologies)
  - Collaborative definition of X3ML mappings
  - Integration with tools (i.e. X3ML Engine for data transformation, RDFVisualization for visualizing transformed data)
  - Smart features to facilitate mapping definition (e.g. copy/clone mappings, disable mappings, etc.)
  - Hybrid definition of mappings (i.e. editing mappings through the UI as well as through their XML serialization)
  - Import and export facilities
  - Enhanced dissemination and collaboration with link-to-share mappings

# 3M Editor

- Supports the collaborative creation and exchange of (X3ML) mapping definitions
  - Mappings definition
  - Mappings storage
  - Organized in projects
  - Collaborative (many users working on the same project)
  - Supports concurrent schema editing (à la google docs)

The screenshot shows the 3M Editor web interface. At the top is a blue header with a hamburger menu icon, the '3m' logo, and a user profile icon. Below the header, there's a navigation bar with 'Project - Mappings', two toggle switches for 'Only display those that I own' (off) and 'Only display those I can edit' (on), and a search bar. The main content is a table with columns: Title, HRID, Author, Status, Date Created, and Date Modified. The table lists several mapping projects, mostly in 'In Progress' status, created between October and December 2022.

Title	HRID	Author	Status	Date Created	Date Modified
A KTP to CIDOC-BookSource	143663	Athina Kritsotaki	In Progress	Fri Oct 21 2022 10:58 AM	Mon Dec 19 2022 09:37 AM
A KTP to CIDOC - NewspaperSource	057576	Athina Kritsotaki	In Progress	Fri Oct 21 2022 01:56 PM	Mon Dec 19 2022 09:37 AM
A KTP to CIDOC-Bibliography	566136	Athina Kritsotaki	In Progress	Mon Oct 24 2022 10:01 AM	Mon Dec 19 2022 09:37 AM
A KTP to CIDOC-DigitalObject	200925	Athina Kritsotaki	In Progress	Mon Oct 24 2022 01:18 PM	Mon Dec 19 2022 09:38 AM
A KTP to CIDOC-SourcePassage	543123	Athina Kritsotaki	In Progress	Tue Oct 25 2022 09:43 AM	Mon Dec 19 2022 09:38 AM
A KTP to CIDOC-SourcePassageCollection	130384	Athina Kritsotaki	In Progress	Tue Oct 25 2022 01:46 PM	Mon Dec 19 2022 09:38 AM
CBDB_person_2	419742	Martin Doerr	In Progress	Sat Jan 28 2023 07:11 PM	Sat Mar 11 2023 10:20 PM
Compact_Conditions	921287	Yannis Marketakis	In Progress	Thu Mar 09 2023 10:39 AM	Thu Mar 09 2023 10:45 AM
long numbers experimen	617235	Yannis Marketakis	In Progress	Fri Mar 10 2023 10:01 AM	Fri Mar 10 2023 10:45 AM
Crew_list_Ruoli_di_Equipaggio	875467	Yannis Marketakis	In Progress	Tue Jun 20 2023 03:01 PM	Tue Jun 20 2023 03:50 PM

A new project is created in the form of a workflow (**5 + 1 steps**)

## 3M Editor – Create a new project

- Step 1: Define a title and a short description for the project

×

1 Mapping Information

2 Source Input

3 Target Schema

4 URI Generator Policy

5 Confirmation

Please fill in the following form to create a new "Mapping Project".

Title \*

Demo schema mappings

Description

A project with schema mappings created for demonstration purposes

BACK

NEXT



## 3M Editor – Create a new project

### ❏ Step 2: Upload (a sample) of the data in their original format

×

✓ Mapping Information

2 Source Input

3 Target Schema

4 URI Generator Policy

5 Confirmation

Please add your own file(s) bellow.

You can add your own "Source Input" files from here, by clicking on the "plus" blue button and then filling in the respective fields.

**File #1**

Title \*

sample file

Description


some sample data

Version

1.0

raw\_data.xml

21 KB

 DOWNLOAD

BACK

NEXT

# 3M Editor – Create a new project

## Step 3: Select or upload the target schema/ontology

×

✓ Mapping Information

✓ Source Input

3 Target Schema

4 URI Generator Policy

5 Confirmation

Please select one or more Target Schema file(s) from the available list. Additionally, you can further add your own file(s) below.

Select one or more predefined "Target Schema" files from the following list.

CRMsci (1.2.2)

☐

The Scientific Observation Model(CRMsci) is a formal ontology intended to be used as a global schema for integrating metadata about scientific observation, measurements and processed data in descriptive and empirical sciences such as biodiversity, geology, geography, archaeology, cultural heritage conservation and others in research IT environments and research data libraries.

FRBR (2.4)

☐

The FRBRoo is a formal ontology intended to capture and represent the underlying semantics of bibliographic information and to facilitate the integration, mediation, and interchange of bibliographic and museum information.

CIDOC CRM (6.2.1)

☐

The CIDOC Conceptual Reference Model (CRM) is a theoretical and practical tool for information integration in the field of cultural heritage

CIDOC CRM (7.1.1)

☒

The CIDOC Conceptual Reference Model (CRM) is a theoretical and practical tool for information integration in the field of cultural heritage

You can add your own "Target Schema" files from here.

▼

BACK

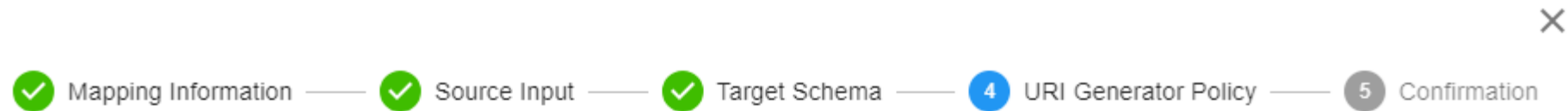
NEXT

IJCKG 2025, October 2025, Heraklion, Greece

26

## 3M Editor – Create a new project

### ❏ Step 4: Select or upload the URI and values generation policy



Please select one or more Generator Policy file(s) from the available list. Additionally, you can further add your own file(s) below.

Select one or more predefined "Generator Policy" files from the following list.

☒

Basic Generator Policy Resources (1.0)  
Some basic generator policy definitions


You can add your own "Generator Policy" files from here.


BACK

NEXT


## 3M Editor – Create a new project

### □ Step 5: Inspect the project configuration and create the project




 Mapping Information


——

 Source Input


——


 Target Schema

——


 URI Generator Policy

——


 Confirmation

 Title


Demo schema mappings

 Description


A project with schema mappings created for demonstration purposes

 Source Input

sample file v1.0 (raw\_data.xml)

 Target Schema

CIDOC CRM v7.1.1 (CIDOC\_CRM\_v7.1.1.rdfs)

 Generator Policy

Basic Generator Policy Resources v1.0 (generator-policy.xml)

BACK

FINISH








# 3M Editor – Create a new project

## ❑ Extra Step

Participating to this project users ❑ ×

Select any user to allow editing this Mapping Project

Search by name ×

<input type="checkbox"/>		Nurdan Atalan Çayırmez Natalan
<input type="checkbox"/>		Achille Felicetti akillus
<input type="checkbox"/>		Kai Salas Rossenbach kaisalasrossenbach
<input type="checkbox"/>		David Novák novak
<input type="checkbox"/>		Michael English DEA_livearchive
<input type="checkbox"/>		Pavlos Fafalios fafalios
<input type="checkbox"/>		Marie LAGASSE Marie

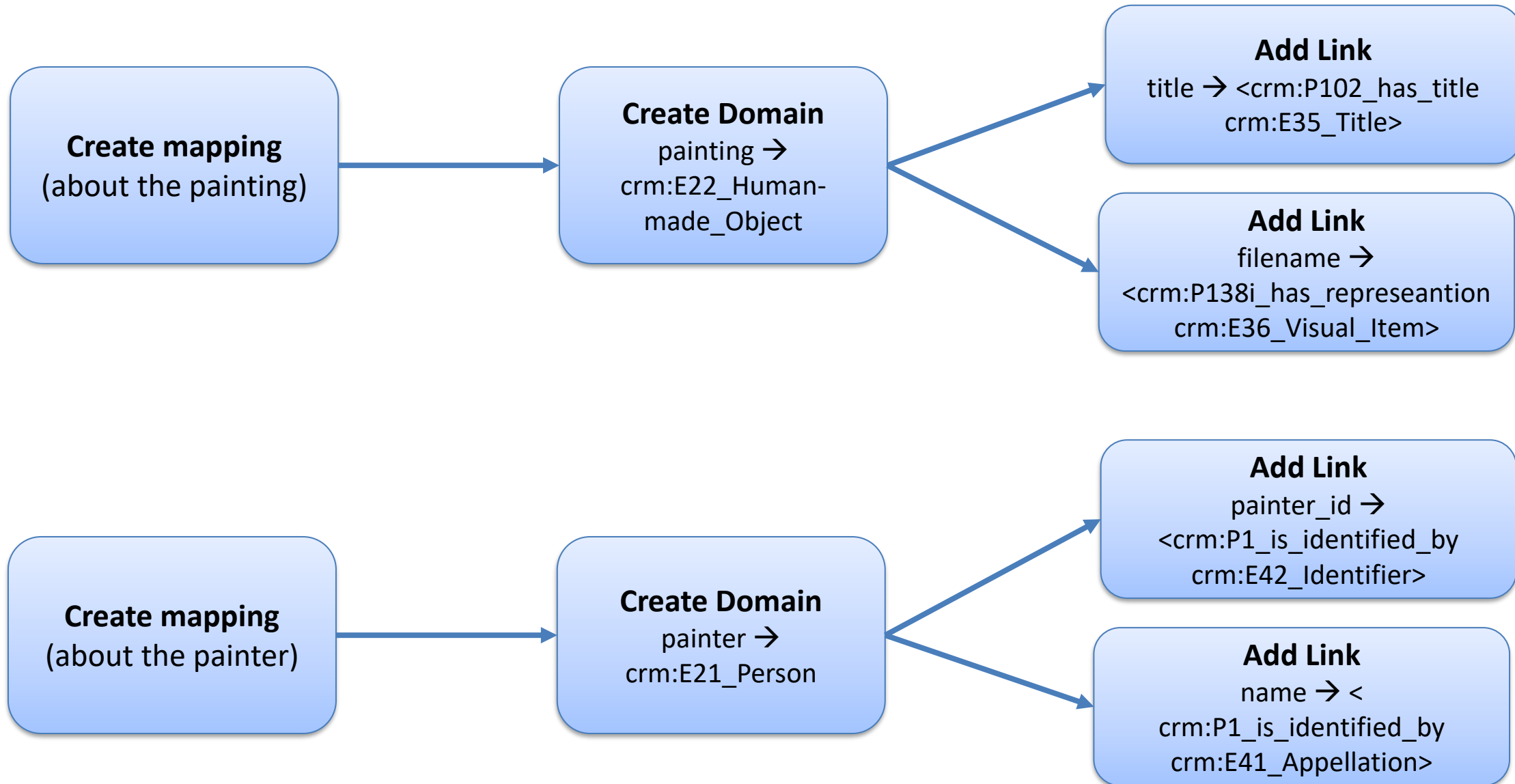
< 1 2 3 4 5 6 7 >

Invite collaborators




Start your mappings project

## 3M Editor – Defining Schema Mappings



## 3M Editor – Defining Schema Mappings

- Guided mappings by analyzing source resources and target schemata

< > **Domain** - Named Graph: *(Optionally add one)* 

 Source Node

/root/painting

/root

/root/painting

/root/painting/creator

/root/painting/title

/root/painting/creation\_date

/root/painting/filename

 Target Entity

E2

CIDOC\_CRM\_v7.1.1

E20\_Biological\_Object

E21\_Person

E22\_Human-Made\_Object

E24\_Physical\_Human-Made\_Thing




E25\_Human-Made\_Feature


E26\_Physical\_Feature


E27\_Site

## 3M Editor – Defining Schema Mappings



- Create a **mapping** and a **domain** about painting objects

✓ Mapping #1- Has no links (0 links selected ✓) - Named Graph: (Optionally add one)   

< > Domain - Named Graph: (Optionally add one) 

 Source Node  
/root/painting

CONDITION

 Target Entity  
E22\_Human-Made\_Object 

crm:E22\_Human-  
made Object



## 3M Editor – Defining Schema Mappings

- Add an **additional** node to specify the object type

✓ Mapping #1- Has no links (0 links selected ✓) - Named Graph: (Optionally add one) 🔗 📄 📁

Domain

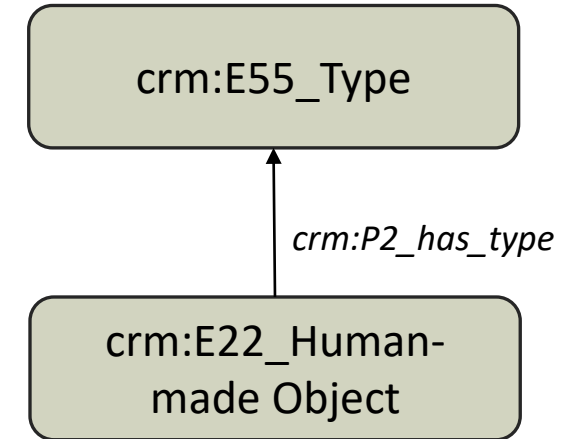
☐ Source Node: /root/painting

☐ Target Entity: E22\_Human-Made\_Object

➔ Relation: P2\_has\_type




☐ Entity: E55\_Type

An additional node



## 3M Editor – Defining Schema Mappings

- Add a **link** for the painting title

☒ Mapping #1 - Has 1 link (1 links selected ☒ ) - Named Graph: [\(Optionally add one\)](#)   

Domain





☐ Source Node: /root/painting

☐ Target Entity: E22\_Human-Made\_Object

→ Relation: P2\_has\_type

☐ Entity: E55\_Type

---

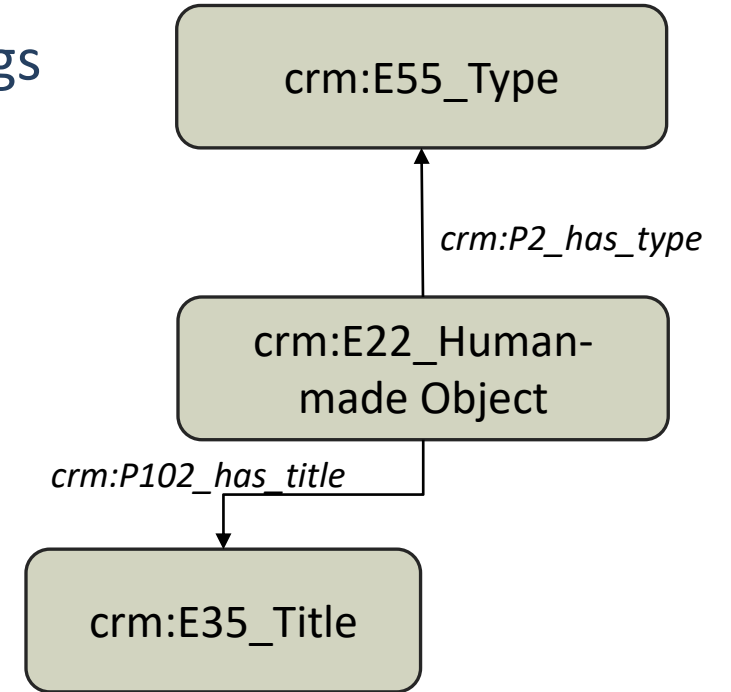
 ☒ Link #1 (of mapping #1)   

↓ Source Relation: title

↓ Target Relation: P102\_has\_title

☐ Source Node: title

☐ Target Entity: E35\_Title



# 3M Editor – Defining Schema Mappings

- Add a **link** for the painter

✓ Mapping #1 - Has 2 links (2 links selected ✓) - Named Graph: (Optionally add one)

Domain

☐ Source Node: /root/painting ☐ Target Entity: E22\_Human-Made\_Object

→ Relation: P2\_has\_type  
☐ Entity: E55\_Type

---

= ✓ Link #1 (of mapping #1)

↓ Source Relation: title ↓ Target Relation: P102\_has\_title

☐ Source Node: title ☐ Target Entity: E35\_Title

---

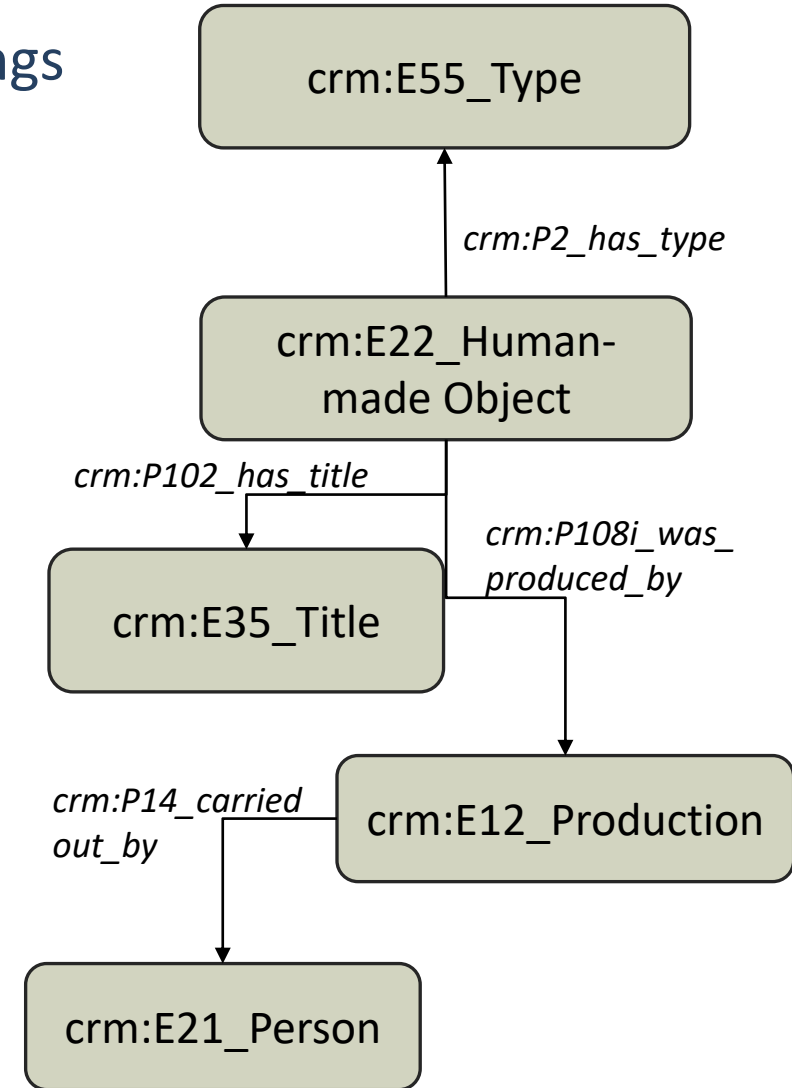
= ✓ Link #2 (of mapping #1)

↓ Source Relation: painter\_id ↓ Target Relation: P108i\_was\_produced\_by

☐ Target Entity: E12\_Production  
↓ Target Relation: P14\_carried\_out\_by




☐ Source Node: painter\_id ☐ Target Entity: E21\_Person

An intermediate node



## 3M Editor – Defining Schema Mappings

- Add a **link** for the date of creating the painting




Mapping #1 - Has 3 links (3 links selected ☒) - Named Graph: (Optionally add one)   

Domain

☐ Source Node: /root/painting ☐ Target Entity: E22\_Human-Made\_Object

→ Relation: P2\_has\_type  
☐ Entity: E55\_Type




---

☒ Link #1 (of mapping #1)   

↓ Source Relation: title ↓ Target Relation: P102\_has\_title

☐ Source Node: title ☐ Target Entity: E35\_Title

---




☒ Link #2 (of mapping #1)   

↓ Source Relation: painter\_id ↓ Target Relation: P108i\_was\_produced\_by

☐ Target Entity: E12\_Production  
↓ Target Relation: P14\_carried\_out\_by

☐ Source Node: painter\_id ☐ Target Entity: E21\_Person

---

☒ Link #3 (of mapping #1)   

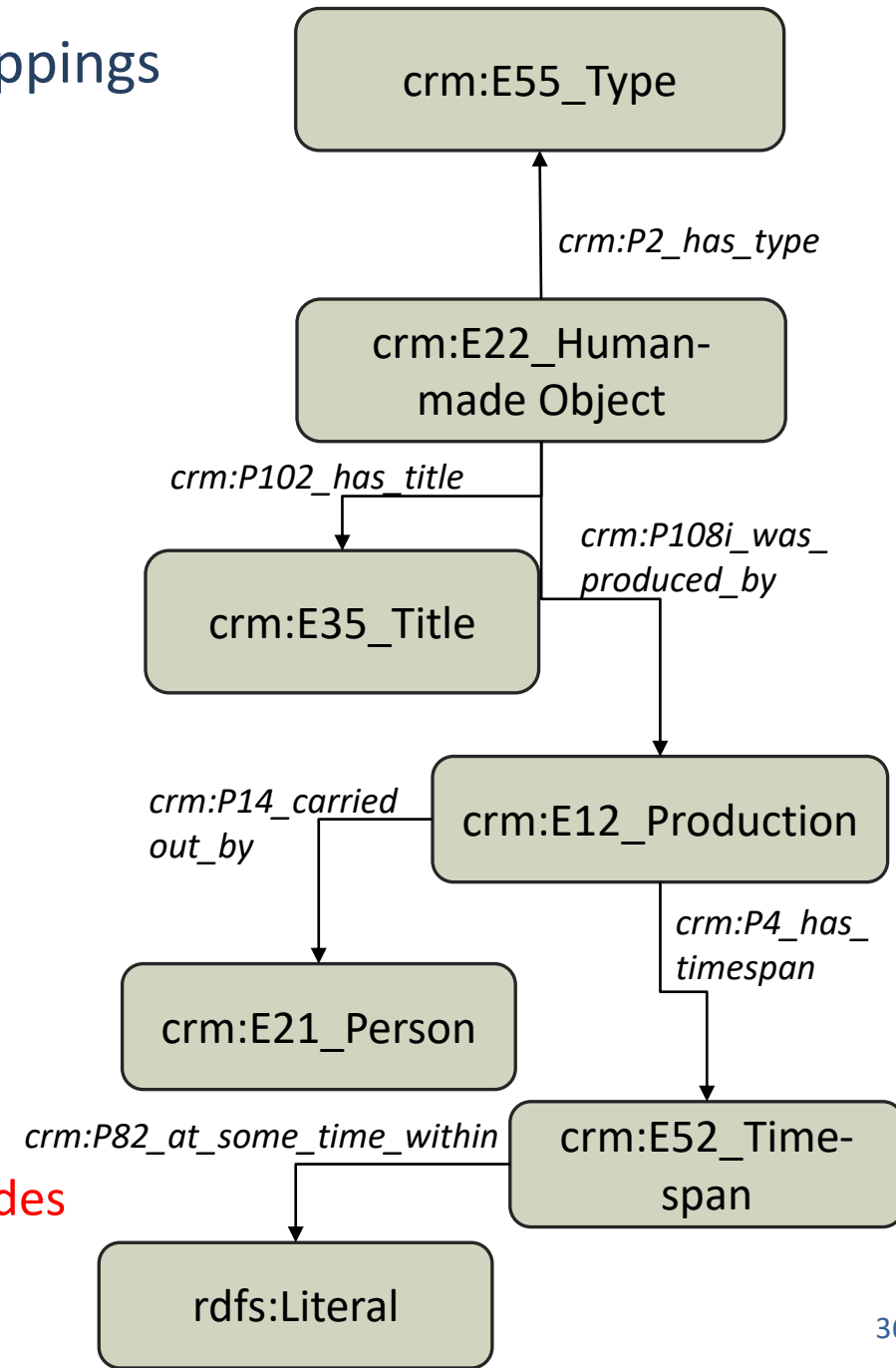
↓ Source Relation: creation\_date ↓ Target Relation: P108i\_was\_produced\_by

☐ Target Entity: E12\_Production  
↓ Target Relation: P4\_has\_time-span

☐ Target Entity: E52\_Time-Span  
↓ Target Relation: P82\_at\_some\_time\_within

☐ Source Node: creation\_date ☐ Target Entity: Literal

Two intermediate nodes



## 3M Editor – Defining URI & Label Generators

- Define a **URI Generator** for the painting

< > Domain - Named Graph: (Optionally add one)

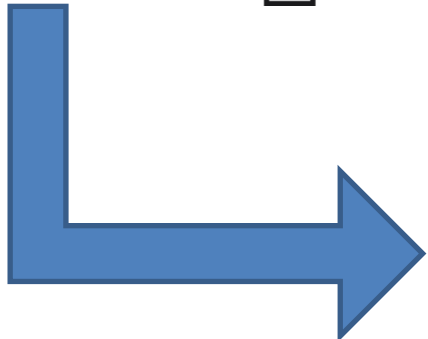
Source Node  
/root/painting

CONDITION

Target Entity  
E22\_Human-Made\_Object Please select target

ADDITIONAL GENERATOR VARIABLE

INSTANCE INFO



Will create URIs of the form  
**<http://www.example.com/resource/painting/p1>**

### Generator

INSTANCE LABEL

Please select some definition and then fill in the respective form. Don't forget to save before closing this dialog.

Definitions

☐ UUID

☐ Literal

☐ SimpleLabel  
Pattern: {label}

☐ CompositeLabel  
Pattern: {term1} {term2}

☒ LocalTermURI  
Prefix: pref  
Pattern: {hierarchy}/{term}

☐ TypedLiteralGen

☐ URIorUUID

☐ UlanUri  
Prefix: ulan  
Pattern: {ulan\_id}

Instance Declaration

LocalTermURI

Argument #1: hierarchy  
Select Type

Constant

Value \*

painting

Argument #2: term  
Select Type

Xpath

Select or add your own XPath \*

id/text()

Definition namespace

Prefix: pref

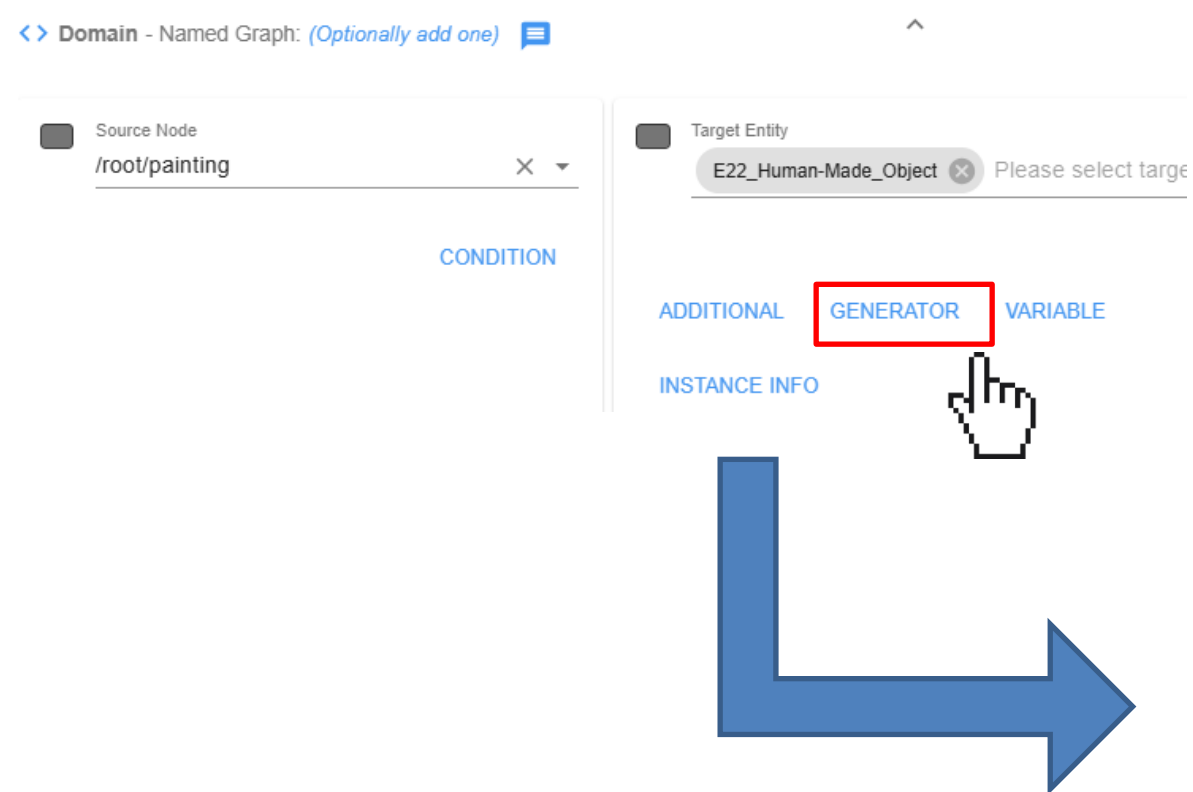
Value \*

<http://www.example.com/resource/>

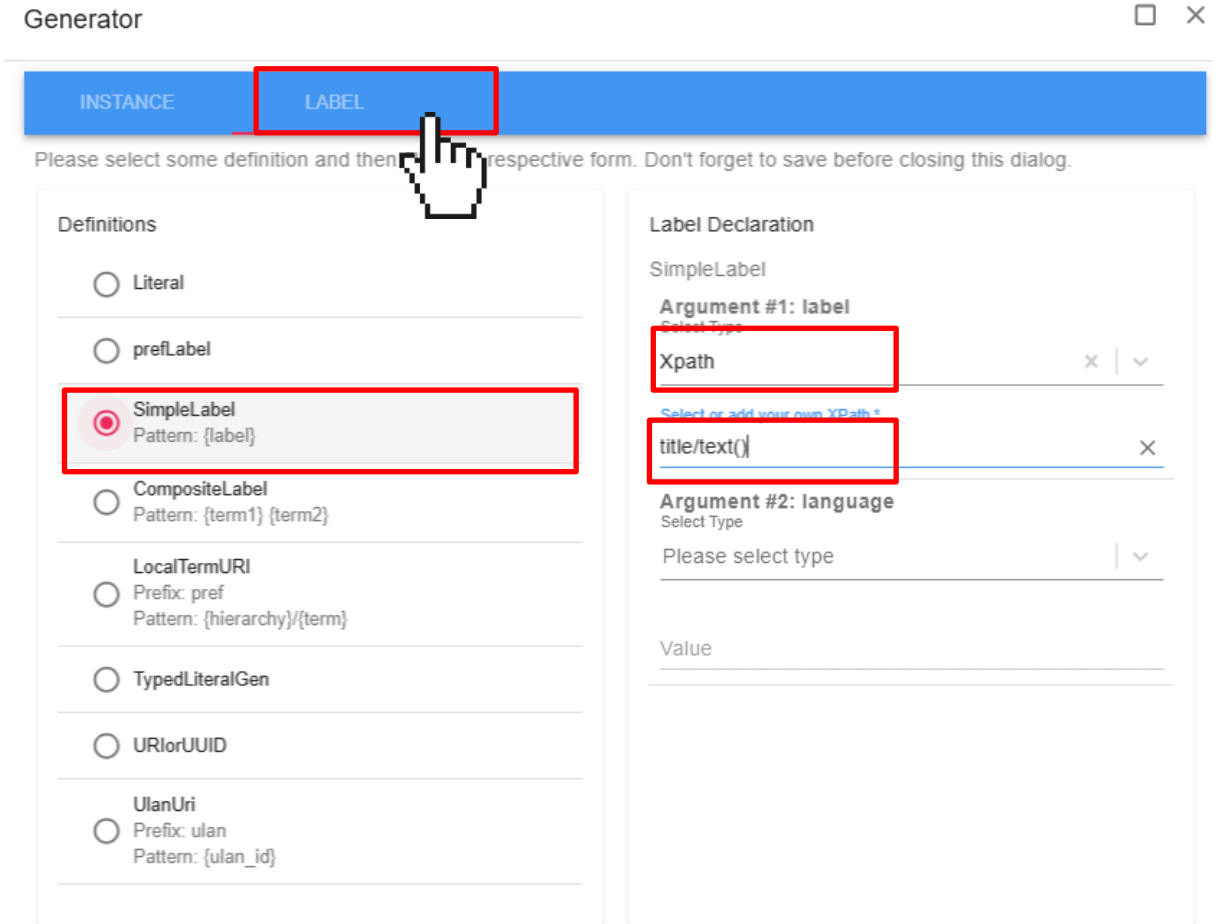
SAVE INSTANCE GENERATOR &amp; NAMESPACE CLOSE

## 3M Editor – Defining URI & Label Generators

- Define a **label Generator** for the painting



Will create labels with the title of paintings  
e.g. **“Self-Portrait as a Painter”**



SAVE LABEL GENERATOR CLOSE

## 3M Editor – Defining URI & Label Generators

- Define a **URI Generator** for the painting

The screenshot shows the 3M Editor interface. At the top, a status bar indicates 'Mapping #1 - Has 3 links (3 links selected) - Named Graph: (Optionally add one)'. Below this, the 'Domain' section is set to 'Source Node:/root/painting'. The 'Target Entity' is 'E22\_Human-Made\_Object'. Under the 'Target Entity', two options are listed: 'LocalTermURI' and 'SimpleLabel'. The 'LocalTermURI' option is selected, and its value is '{painting}/{id/text0}'. The 'SimpleLabel' option is also listed, and its value is '{title/text0}'. Two callout boxes with arrows point to the configuration: 'The URI Generator' points to the 'LocalTermURI' option and its value, and 'The label Generator' points to the 'SimpleLabel' option and its value.

Mapping #1 - Has 3 links (3 links selected) - Named Graph: (Optionally add one)

Domain

Source Node:/root/painting

Target Entity: E22\_Human-Made\_Object

LocalTermURI  
{painting}/{id/text0}

SimpleLabel  
{title/text0}

The URI Generator

The label Generator

**Green** arguments: constant values

**Yellow** arguments: XPATH expressions (for collecting data from XML input)

## 3M Editor – Defining URI & Label Generators

```
<painting>
  <id>p-1</id>
  <painter>Van Gogh</painter>
  <title>Self-Portrait as a Painter</title>
  <creation_date>1889</creation_date>
  <filename>file_p-1.jpg</filename>
</painting>
```

```
<painting>
  <id>p-3</id>
  <painter>Degas</painter>
  <title>The Ballet Class</title>
  <creation_date>1874</creation_date>
  <filename>file_p-3.jpg</filename>
</painting>
```

☐ **Target Entity:** E22\_Human-Made\_Object



LocalTermURI  
*{painting}/{id/text()}*

http://.../painting/p-1



CompositeLabel  
*{Painting with title:} {title/text()}*

Painting with the title: Self-Portrait as a Painter

☐ **Target Entity:** E12\_Production



production\_event

urn:uuid:5178d556-d670-4bf8-b7dc-48e78433edd3



UUID



CompositeLabel  
*{Creation of painting with title:} {../title/text()}*

Creation of the painting with the title: Self-Portrait as a Painter



**Target Relation:** P14\_carried\_out\_by

☐ **Target Entity:** E21\_Person



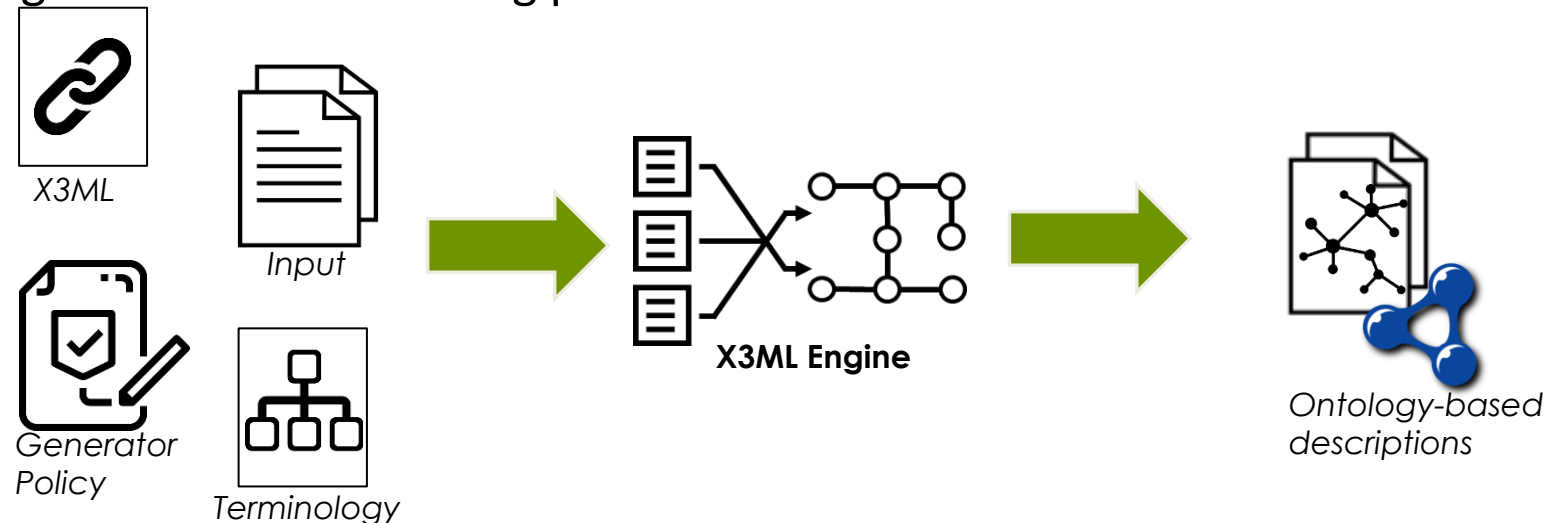
LocalTermURI  
*{painter}/{text()}*

http://.../painter/VanGogh



# X3ML Engine

- **X3ML Engine** is a tool that realizes the transformation of data resources to a target format with respect to an X3ML Mapping definition language.
- Main principles:
  - Simplicity by design
  - Transparency in terms of expected output
  - Re-use of standards and technologies as much as possible
  - Facilitating the instance matching process



# X3ML Engine

- X3ML Engine has been designed by FORTH.
  - The initial development has been carried out by DELVING B.V. under the support and contribution of FORTH (until version 1.3 – March 2015)
  - FORTH took over the full development of X3ML Engine since March 2015.
- 30 Releases in total (Latest one: version 2.2.2 May 2025)
- Available as: API, executable (console-based & GUI), service

## Application with UI



## CLI tool

```
C:\Repositories\Github\X3ML\target>java -jar x3ml-engine-1.9.4-SNAPSHOT-exej.jar
usage: X3ml -xml <input records> -x3ml <mapping file> hello
Options
  -a,--assocTable <arg>      export the contents of the association table in XML format
  -f,--format <arg>          Output format. Options:
                              --format application/rdf+xml (default)
                              --format application/n-triples
                              --format application/trig
                              --format text/turtle
  -i,--input <arg>           XML input records.
                              Option A-single file: --input input.xml
                              Option B-multiple files (comma-sep): --input input1.xml,input2.xml,input3.xml
                              Option C-folder: --input # folder_path
                              Option D-URL: --input @input_url
                              Option E-multiple URLs: --input @input_url1,input_url2,input_url3
                              Option F-stdin: --input @
  -m,--mergeAssocWithRDF     merge the contents of the association table with the RDF output
  -o,--output <arg>          The output file name: --output output.rdf
  -p,--policy <arg>          The value policy file: --policy policy.xml
  -r,--reportProgress         reports the progress of the transformations
  -t,--terms <arg>           the SKOS taxonomy
                              Option A-single file: --terms skosTerms.nt
                              Option B-URL: --terms @skos_terms_url
  -u,--uuidTestSize <arg>    Create a test UUID generator of the given size.
                              Default is UUID from operating system
  -x,--x3ml <arg>           X3ML mapping definition.
                              Option A-single file: --x3ml mapping.x3ml
                              Option B-multiple files (comma-sep): --x3ml mappings1.x3ml,mappings2.x3ml
                              Option C-URL: --x3ml @mappings_url
                              Option D-stdin: --x3ml @

Missing required options: i, x
```

## JAVA API

```
X3MLEngineFactory.create()
    .withMappings(new File("path/to/mappings.x3ml"))
    .withInputFiles(new File("path/to/input.xml"))
    .withGeneratorPolicy(new File("path/to/generator-policy.xml"))
    .withTerminology(new File("path/to/terms.nt"), Lang.NT)
    .withOutput("output.ntriples", X3MLEngineFactory.OutputFormat.NTRIPLES)
    .withProgressReporting()
    .execute();
```

# RDF Visualizer

RDF Visualizer is a generic browsing mechanism that gives the user a flexible, highly configurable, detailed overview of an RDF dataset/database

**Symbol & color Legend**

- Other Entities
- Temporal Entities
- Actors
- Physical Things
- Conceptual things
- Places

**Entity Hierarchy:**


- ship Andrea [ sealit#Ship ]
  - sealit#was\_constructed\_by
    - Construction of Andrea [ sealit#Ship\_Construction ]
      - P7\_took\_place\_at
        - Sestri Ponente [ E53\_Place ]
      - sealit#under\_name
        - Andrea
      - P4\_has\_time-span
        - Construction date 1859 [ E52\_Time-Span ]
      - P2\_has\_type
        - brigantino [ E55\_Type ]
      - P1\_is\_identified\_by
        - Andrea [ sealit#Ship\_Name ]
      - sealit#is\_registered\_by
        - Registration of Andrea [ sealit#Ship\_Registration ]
      - sealit#has\_tonnage
        - tonnage of Andrea [ sealit#Tonnage ]
      - sealit#has\_phase
        - Ownership of Andrea [ sealit#Ship\_Ownership\_Phase ]
          - sealit#has\_owner
            - Gaetano Ogno [ E21\_Person ]
      - sealit#voyages
        - Voyage of ship Andrea [ sealit#Voyage ]

**Callouts:**

- Instance label and type (i.e. Class)
- View URI on hover  
urn:uuid:344e313e-766c-4f29-b9ff-fb02ecd52615
- Expand/collapse information
- Display images and galleries

**Image Gallery**

## RDF Visualizer – cont'd

> ☐ Expand all mappings 

☒ Mapping #1 - Has 3 links (3 links selected ☒ ) 

☒ Mapping #2 - Has 2 links (2 links selected ☒ ) 

RDF Visualizer can be  
loaded directly from 3M  
Editor

VALIDATE PATHS

PRODUCE X3ML

PRODUCE RDF



VIEW IN THE RDF VISUALISER

Enter subject

<http://www.example.com/resource/painting/p-1>

Choose a Template:

Template 1

Mark same instances



Painting with title: Self-Portrait as a Painter [ E22\_Human-Made\_Object ]

P2\_has\_type

[http://www.example.com/resource/object\\_type/painting](http://www.example.com/resource/object_type/painting) [ E55\_Type ]

P108i\_was\_produced\_by

Creation of painting with title: Self-Portrait as a Painter [ E12\_Production ]

P14\_carried\_out\_by

Vincent Van Gogh [ E21\_Person ]

P1\_is\_identified\_by

a-1 [ E42\_Identifier ]

<http://vocab.getty.edu/page/ulan/500115588> [ E42\_Identifier ]

P4\_has\_time-span

urn:uuid:c4cddcd6-115e-456c-9c22-92b220c3abb6 [ E52\_Time-Span ]

P82\_at\_some\_time\_within

1889

P102\_has\_title

Self-Portrait as a Painter [ E35\_Title ]



## Looking to the Future



Introduction

46-47

2'



How LLMs can support?

48

1'



Approaches / Prompts

49-51

2'



First Results

52-53

1'



Issues / Challenges

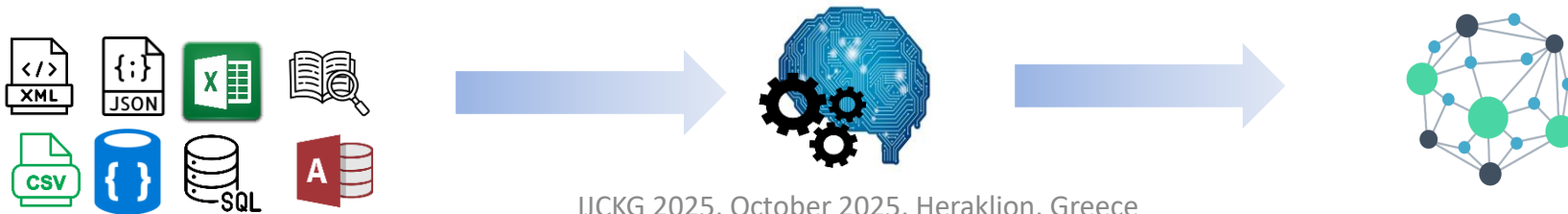
54

1'

# Looking to the Future

## Automatic the Definition of X3ML Schema Mappings

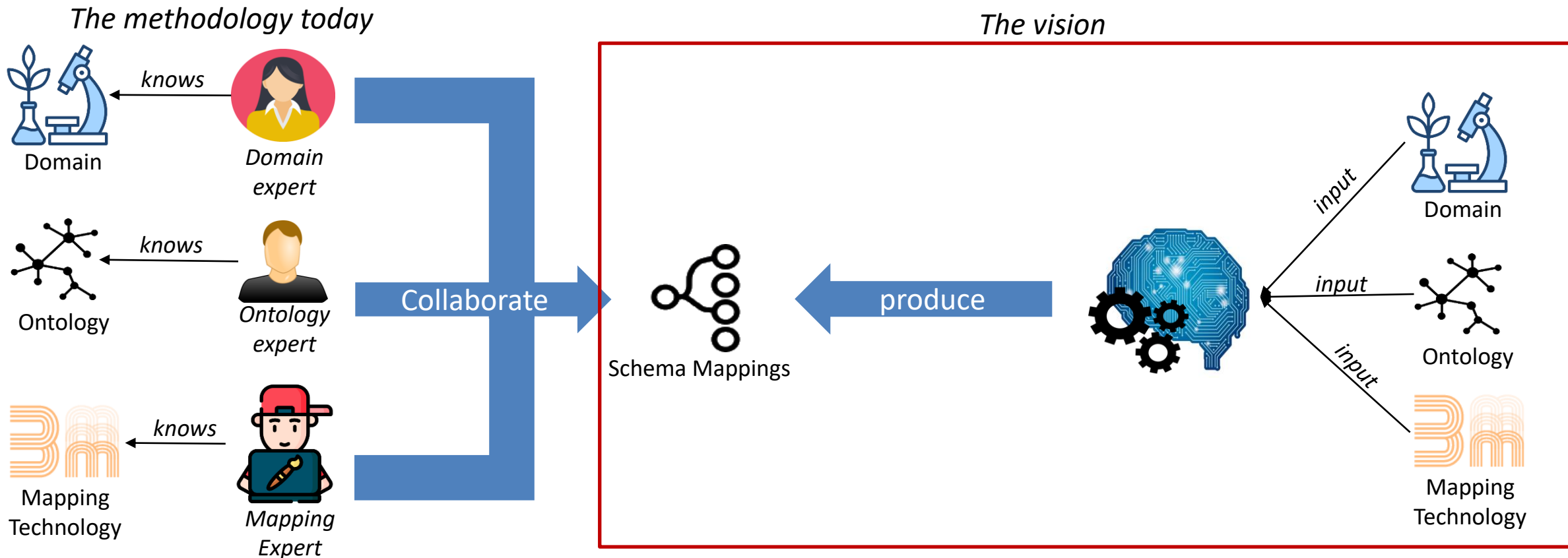
- **What?**
  - Automate the transformation of large-scale structured data to RDF Knowledge Graphs
  - Reduce the manual intervention of the schema mapping process
- **Why do we need it?**
  - Considering the tremendous volumes of data and their diversity, speeding up the semantic data integration process is the only way
- **Why it is difficult?**
  - The definition of schema mappings is a manual process
  - It requires domain expertise and mapping experience
  - Ensuring that mappings are accurate is a complex task (i.e. depends on the ontology, the input data, etc.)
- **How?**
  - Focus on automation
    - Investigate how Large Language Models (LLMs) can support the definition of schema mappings
    - Explore different methods and approaches
    - Focus on mappings validation and evaluation



# Looking to the Future

## How LLMs can support?

- Assign LLMs the roles of domain experts, ontology experts, and mapping experts to **automate** the data integration process



## Looking to the Future

### Why focus on Schema Mappings?

- While other approaches utilize LLMs for transforming data **we focus on generating schema mappings**
  - It is an intermediate step that allows for verification and adjustments before the actual transformation
  - Easier debugging as potential errors (in the data transformation) can be traced by inspecting the mappings that were generated (data transformations using LLMs is a black box)
  - Direct data transformations using LLMs might introduce variability
  - Require to use only a small subset of the actual data
  - More efficient compared to transforming entire data collections using LLMs

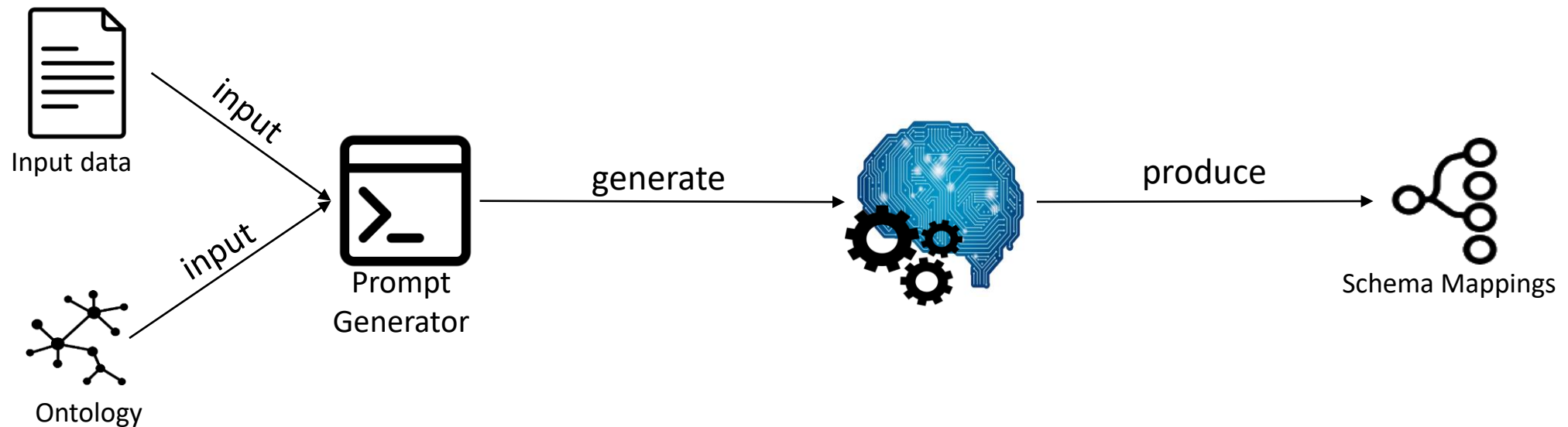


# Looking to the Future

## LLM-Supported Methods and Approaches

- **zero-shot**

- Create a prompt asking the LLM to design X3ML schema mappings using as input
  - the original data
  - the target ontology



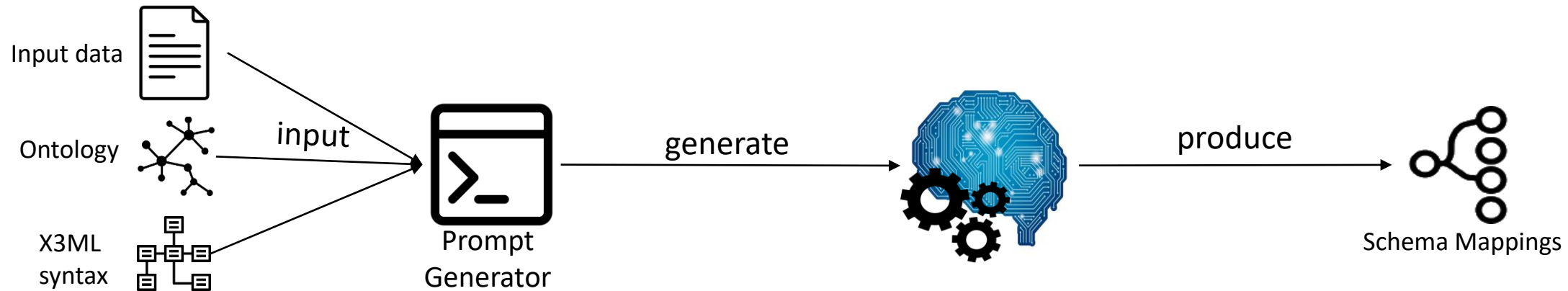
> Generate the schema mappings using X3ML mapping language for transforming the given dataset to CIDOC CRM ontology.

# Looking to the Future

## LLM-Supported Methods and Approaches

- **syntax-aware**

- Create a prompt asking the LLM to design X3ML schema mappings using as input
  - the original data
  - the target ontology
  - x3ml syntax



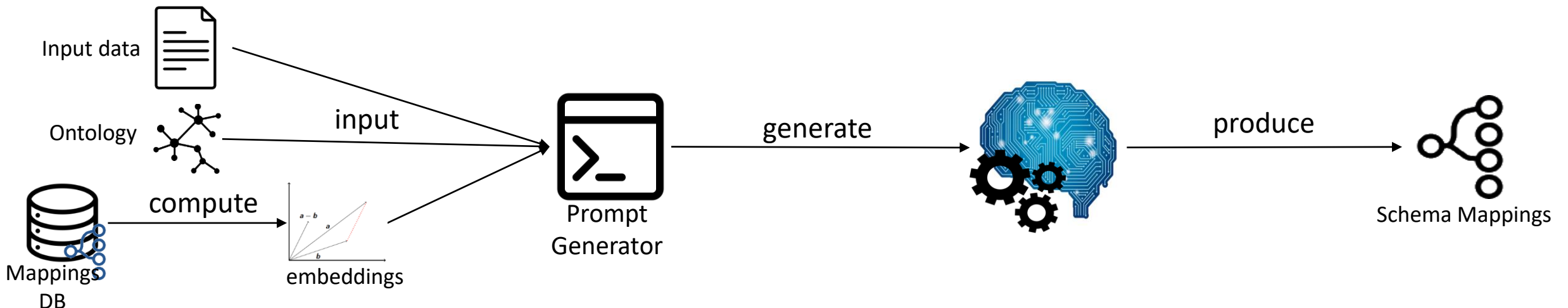
```
> Generate the schema mappings using X3ML mapping language for transforming the given dataset to CIDOC CRM ontology. You must generate a valid X3ML file, so you are given a sample X3ML file to understand how it is structured.
```

# Looking to the Future

## LLM-Supported Methods and Approaches

- **in-context mapping**

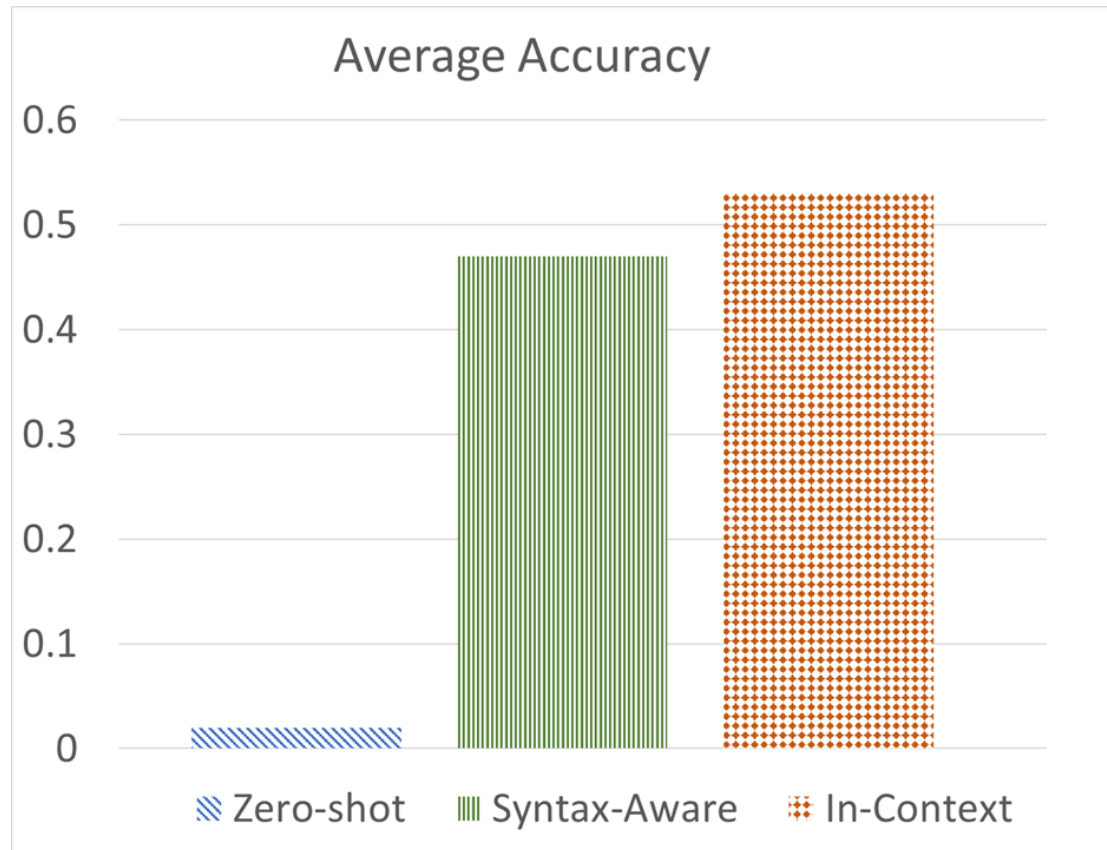
- Create a prompt asking the LLM to design X3ML schema mappings using as input
  - the original data
  - the target ontology
  - relevant x3ml mappings



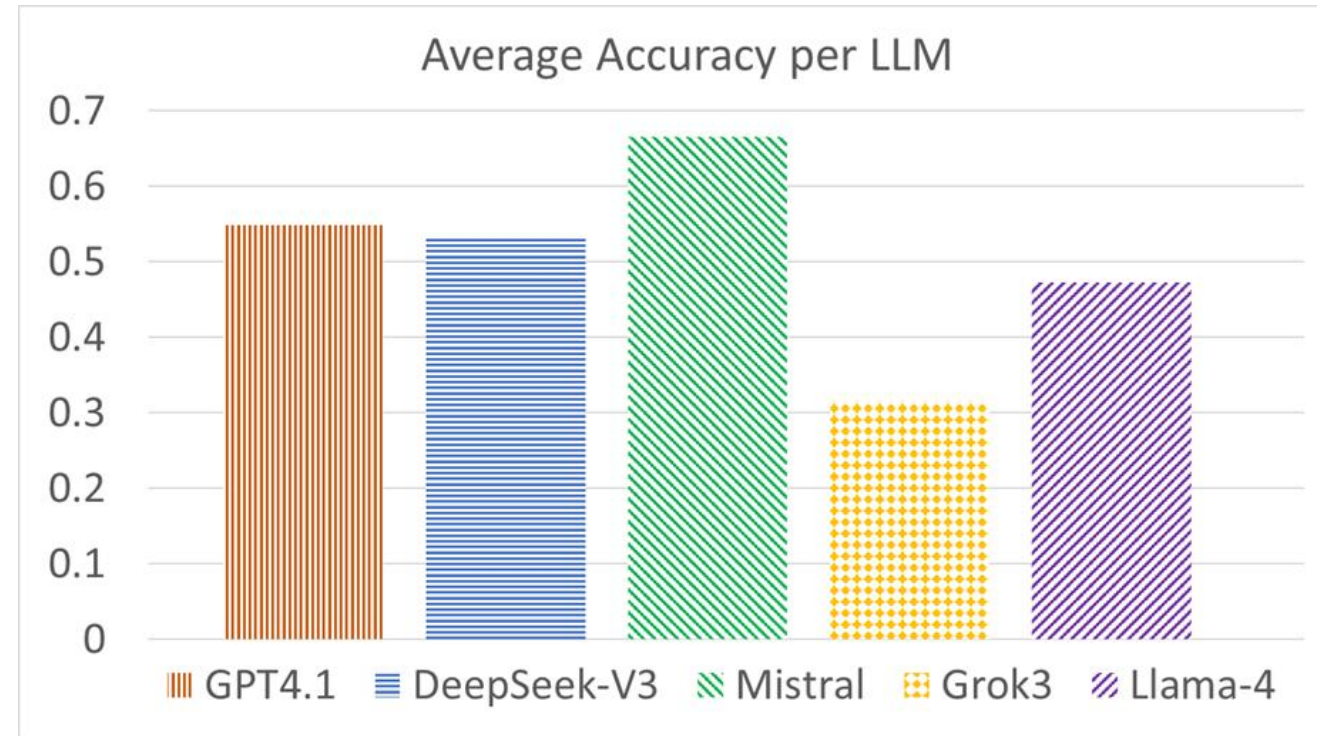
> Generate the schema mappings using X3ML mapping language for transforming the given dataset to CIDOC CRM ontology. You are given a relevant X3ML mapping file used to describe the schema mappings for similar input like the one provided.

## Looking to the Future

### Some first results



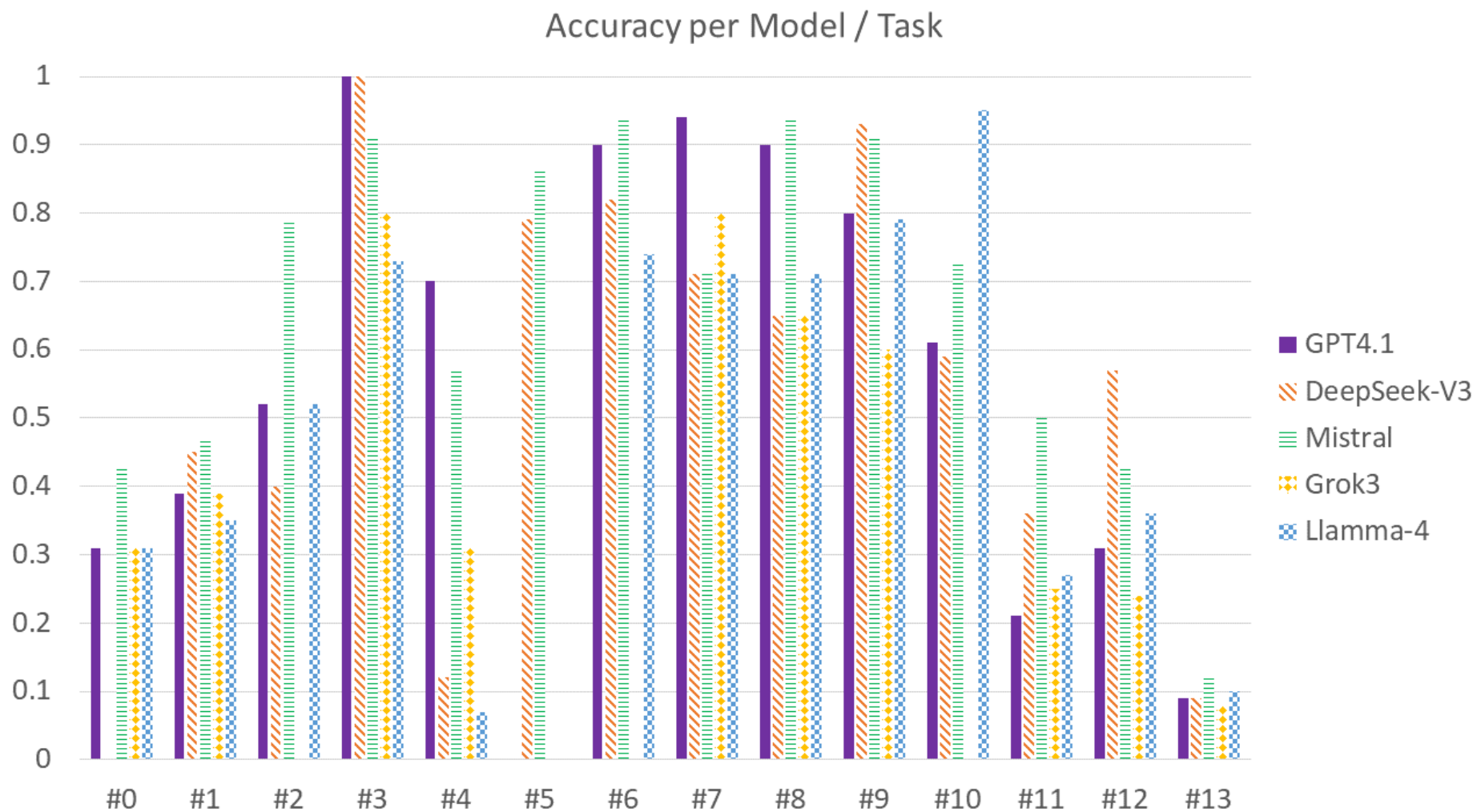
average accuracy (in-context)  $\approx$  **53%**



average accuracy (in-context) of the best performing LLM  $\approx$  **65%**

# Looking to the Future

## Some first results – cont'd



## Looking to the Future Issues & Challenges

- The generated X3ML mappings need to be validated
  - Validity of X3ML: is X3ML valid with respect to X3ML Schema?
  - Validity of resources: are resources (e.g. classes) syntactically correct?
  - Validity of resource connectivity: do classes and properties relate (in the adopted ontology)?
  - Validity of semantics: are mappings semantically correct?
- How to evaluate ?
  - Gold standard: real existing mapping projects (from various 3M installations)
  - More than 1500 X3ML mapping projects
- How to deal with the construction of URIs and labels ?



	Conclusion		
	Conclusions and Next Steps	56	1'
	Some Statistics	57	1'
	References and Links	58	1'

# Conclusions & Next Steps

## Key takeaways

- Integrating heterogeneous data requires **semantic alignment** through schema mappings
- The **X3ML framework** offers a declarative, transparent, and reusable way to transform data into RDF knowledge graphs
- A well-defined schema mapping process ensures the construction of **interoperable and semantically rich knowledge graphs**
- **Tools** that facilitate the definition of schema mappings and promote collaboration of users

## Looking Ahead

- X3ML continues to **evolve** toward greater usability
- Large Language Models (LLMs) can assist or **automate** parts of the mapping process
- Integration workflows are moving toward **semi-automated, intelligent pipelines**
- **Community collaboration** promote high-quality schema mappings and strengthen best practices



## Some Statistics

Project / Activity	# Mappings	# Links	# RDF Triples
The Global Record of Stocks and Fisheries (GRSF) <i>(so far)</i>	86	315	6.4 million
SemantyFish <i>(so far)</i>	8	210	6.5 million
SeaLit	266	2284	18.5 million
RICONTRANS	116	683	? millions
DLN Sip Archiver	42	164	On demand
SKOS FoodEx2	5	51	2.8 million

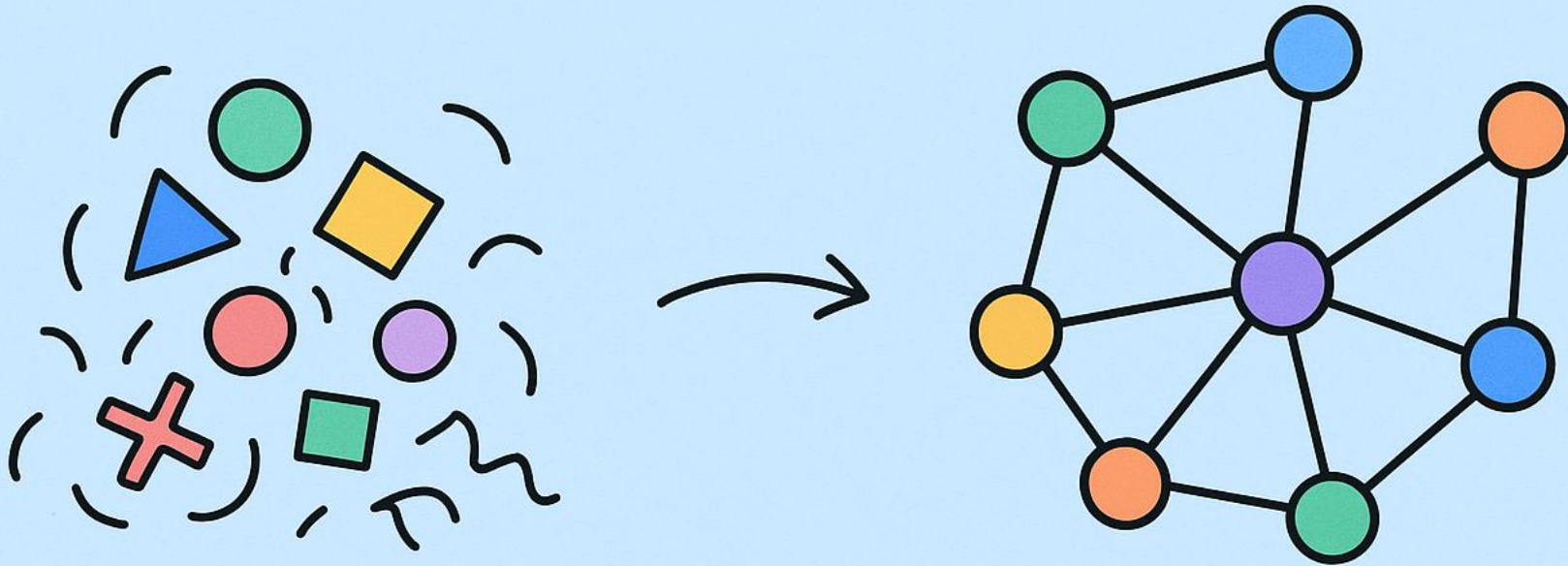


## References & Links

- Tutorial-related
  - <https://ymark.github.io/X3ML-Tutorial/>
  - <https://github.com/ymark/X3ML-Tutorial> (with exercises)
- X3ML-related
  - <https://github.com/isl/x3ml>
  - <https://demos.isl.ics.forth.gr/3m/Projects>
- Related Publications
  - Marketakis, Y., Lintanff--Castel, M., and Tzitzikas, Y., 2025. **Using LLMs to Automate the Transformation of Any Structured Data to Ontology-based Descriptions** (submitted – under review)
  - Marketakis, Y., Minadakis, N., Kondylakis, H., Konsolaki, K., Samaritakis, G., Theodoridou, M., Flouris, G. and Doerr, M., 2017. **X3ML mapping framework for information integration in cultural heritage and beyond. International Journal on Digital Libraries**, 18(4), pp.301-319.
  - Minadakis, N., Marketakis, Y., Kondylakis, H., Flouris, G., Theodoridou, M., de Jong, G. and Doerr, M., 2015, September. **X3ML Framework: An Effective Suite for Supporting Data Mappings**. In EMF-CRM@TPDL (pp. 1-12).

# From Chaos to Knowledge Graphs – Mission Accomplished! 🚀

Thanks for mapping with us!



Any Questions ?

Scan for more

