

A Tool for the Maintenance and Evolution of X3ML Schema Mappings

Yannis Marketakis

marketak@ics.forth.gr



In one slide (and one minute)

- **What** (*what's the problem?*)
 - The evolution of ontologies is not aligned/synchronized with the evolution of schema mappings that utilize them
- **Why** (*why it is important/difficult?*)
 - Schema mappings use deprecated resources (i.e. ontologies) that might create inconsistencies
 - Requires manual effort to spot them and update them accordingly
- **How** (*how to deal with it?*)
 - Rely on a tool that supports the transition of schema mappings from one ontology version to another
 - Describe once → apply many times
- **Next** (*what's next?*)
 - Focus on automation (identify obsolete classes/properties)
 - Integrate with existing tools

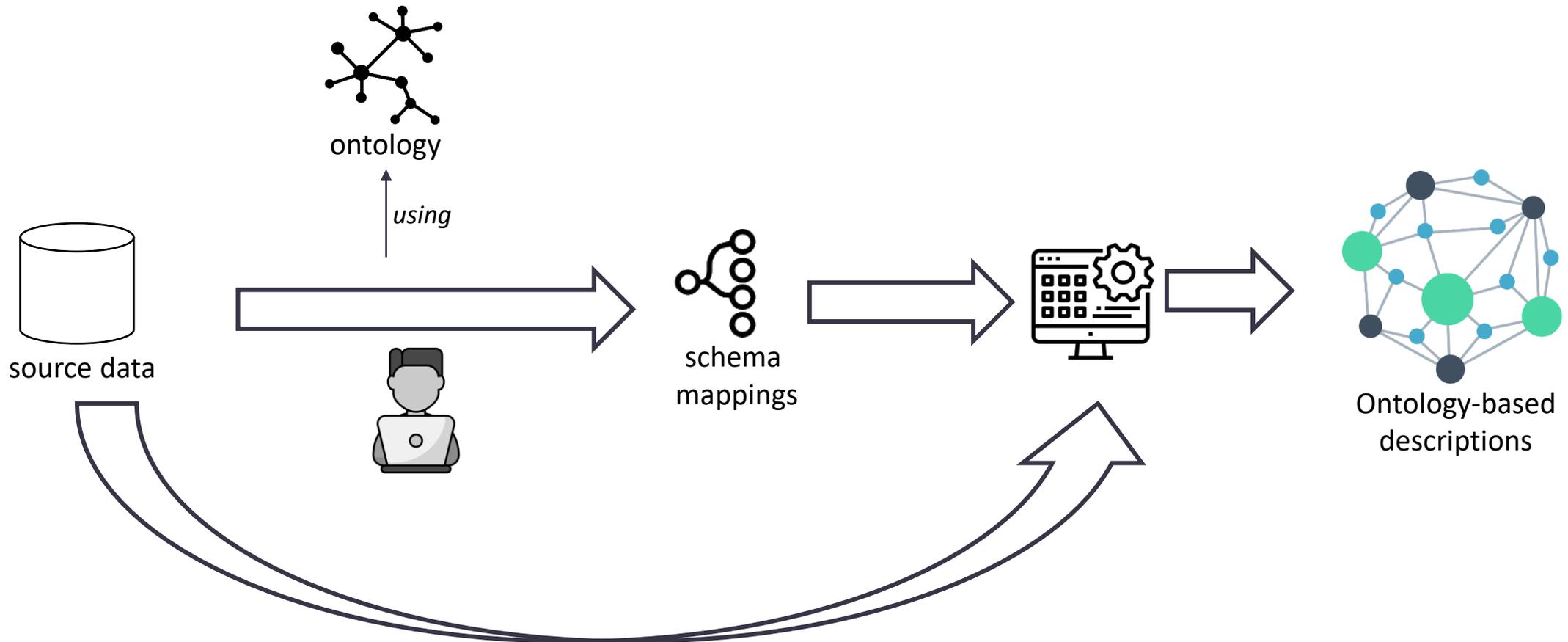
Outline

	Introduction	3'
	Background (X3ML)	5'
	XEM: X3ML Evolution Manager	5'
	Conclusion & Next Steps	2'



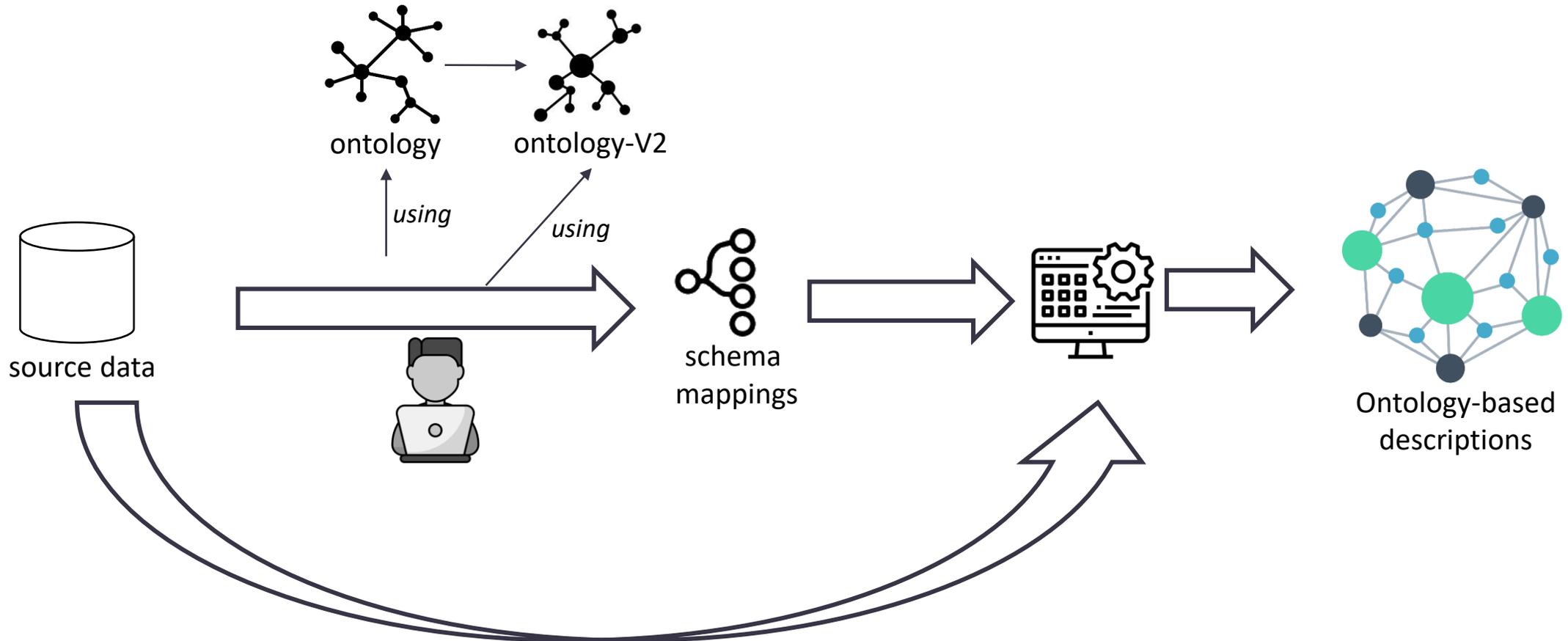
Introduction

- Ontologies provide the required semantic richness, flexibility and interoperability and are often used for semantically integrating data



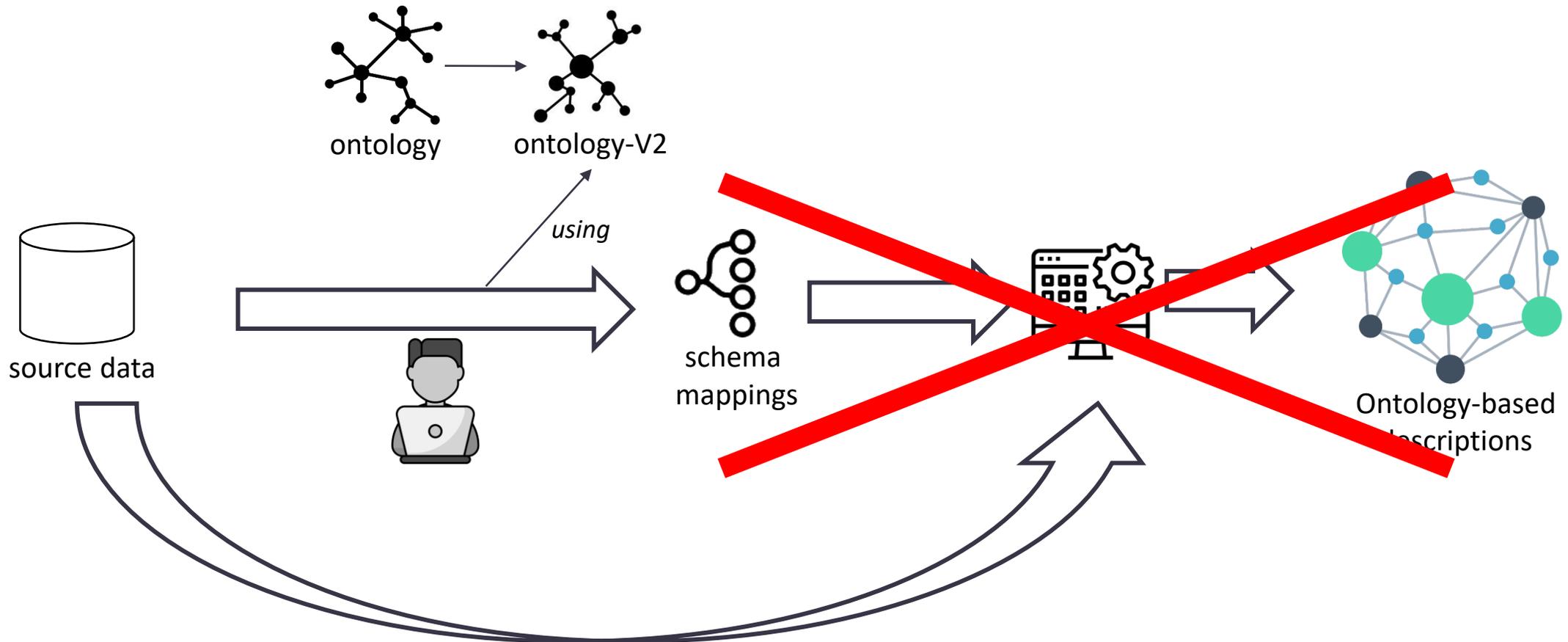
Introduction

- Ontologies evolve over time
 - If schema mappings do not follow their updates, they (and their results) become obsolete



Introduction

- Ontologies evolve over time
 - If schema mappings do not follow their updates, they (and their results) become obsolete



Introduction

- This happens quite often (and its normal)

- CIDOC 6.2.7

- E22 Man-made Object → E22 Human-made Object
- E25 Man-made Feature → E25 Human-made Feature

Class changes

- CIDOC 7.1.3

- P58 has section definition → P1 is identified by

Property changes

- E45 Address → E41 Appellation
 - P2 has type
 - E55 Type ["Address"]

More complex changes

Introduction

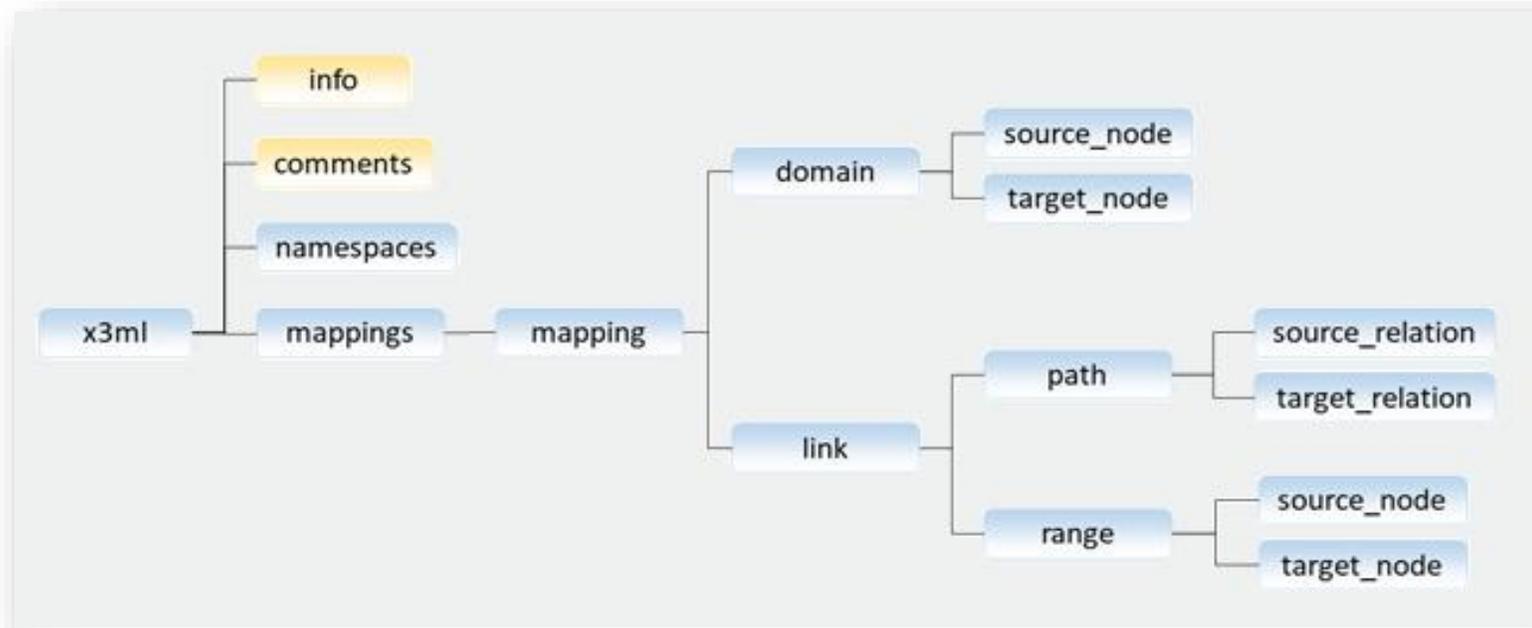
- Updating the schema mappings is laborious
 - It requires manual intervention
 - In general schema mappings definition and maintenance is a manual process
 - Confidence
 - Make sure that updates in schema mappings are aligned with ontology updates
 - Error-prone
 - Manual updates are always error-prone
 - Do the same many times
 - There are many different schema mappings using the same ontology



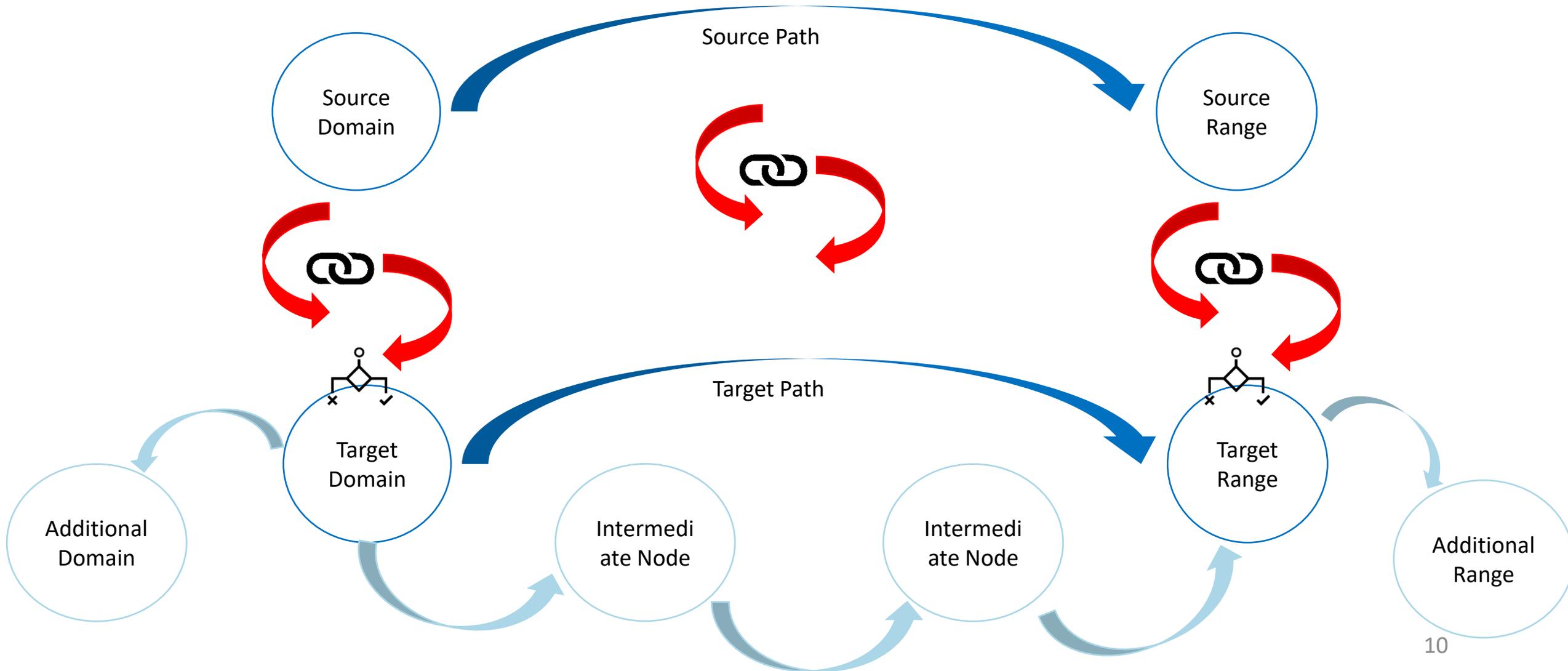
We need tools to alleviate with these problems

Background – X3ML Schema Mapping Language

- X3ML Schema mappings
 - Describe schema mappings in a declarative manner
 - Focus on collaborative creation and exchange of mappings
 - Decoupled from URI and values generation
 - XML Serialization for efficient storage and exchange

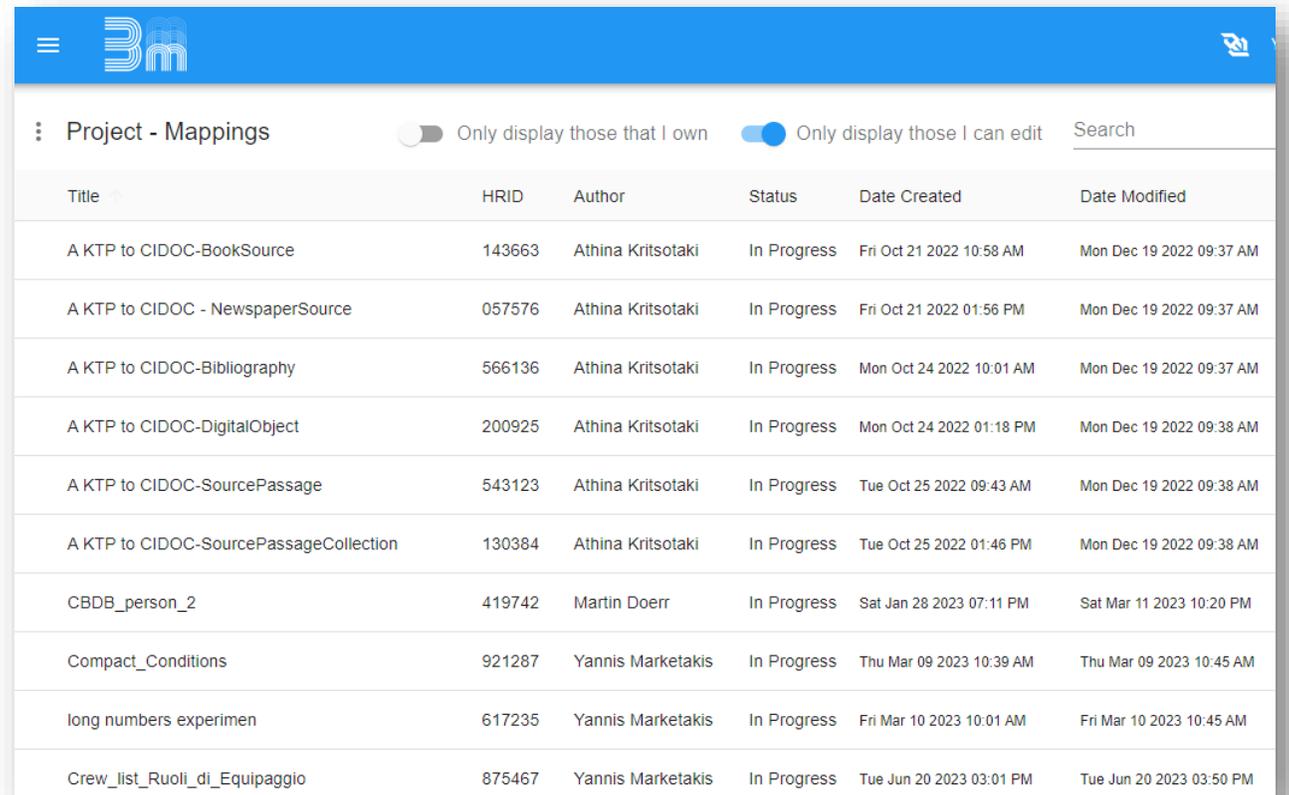


Background – X3ML Schema Mapping Language



Background - 3M Editor

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



The screenshot shows the 3M Editor web interface. At the top is a blue header with the 3M logo and a user profile icon. Below the header is a navigation bar with a hamburger menu, the text 'Project - Mappings', and two toggle switches: 'Only display those that I own' (disabled) and 'Only display those I can edit' (enabled). A search bar is also present. The main content is a table with the following columns: Title, HRID, Author, Status, Date Created, and Date Modified. The table contains 12 rows of mapping projects.

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

Background - 3M Editor – cont'd

- 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, RDFVisualizer 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

Background - 3M Editor – cont'd

```
<painting>  
  <id>p-1</id>  
  <painter id>Van Gogh</painter id>  
  <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 id>Degas</painter id>  
  <title>The Ballet Class</title>  
  <creation_date>1874</creation_date>  
  <filename>file_p-3.jpg</filename>  
</painting>
```

Domain

Source Node: /root/painting Target Entity: E22_Human-Made_Object

= 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

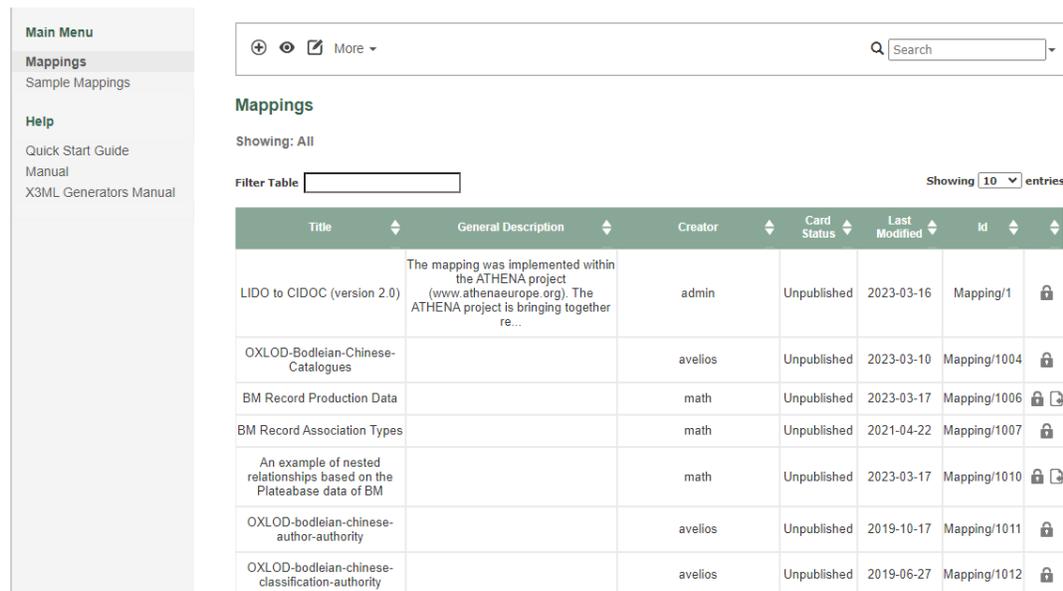
production_event

↓ Target Relation: P14_carried_out_by

Source Node: painter_id Target Entity: E21_Person

Background - 3M Editor – cont'd

- Older version of 3M (and 3M Editor)
 - Still in use



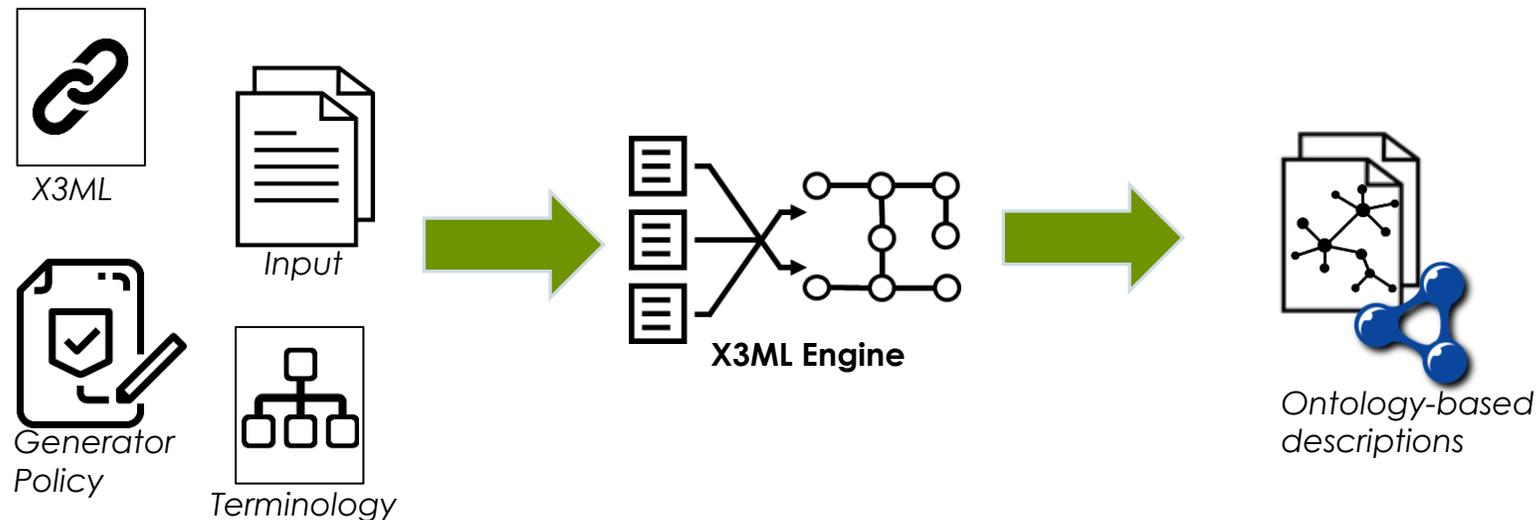
The screenshot displays the 3M Editor interface. On the left is a 'Main Menu' sidebar with options: Mappings (Sample Mappings), Help (Quick Start Guide, Manual, X3ML Generators Manual). The main area shows a search bar, a 'More' dropdown, and a 'Mappings' section. Below this is a 'Showing: All' indicator and a 'Filter Table' input field. A table lists mappings with columns: Title, General Description, Creator, Card Status, Last Modified, Id, and an edit icon. The table contains 7 entries.

Title	General Description	Creator	Card Status	Last Modified	Id	
LIDO to CIDOC (version 2.0)	The mapping was implemented within the ATHENA project (www.athenaeurope.org). The ATHENA project is bringing together re...	admin	Unpublished	2023-03-16	Mapping/1	🔒
OXLOD-Bodleian-Chinese-Catalogues		avelios	Unpublished	2023-03-10	Mapping/1004	🔒
BM Record Production Data		math	Unpublished	2023-03-17	Mapping/1006	🔒
BM Record Association Types		math	Unpublished	2021-04-22	Mapping/1007	🔒
An example of nested relationships based on the Plateabase data of BM		math	Unpublished	2023-03-17	Mapping/1010	🔒
OXLOD-bodleian-chinese-author-authority		avelios	Unpublished	2019-10-17	Mapping/1011	🔒
OXLOD-bodleian-chinese-classification-authority		avelios	Unpublished	2019-06-27	Mapping/1012	🔒

- Three distinct deployments
 - New 3M
 - **497** mapping projects
 - Old 3M
 - **1480** mapping projects
 - D4Science-powered 3M
 - **650** mapping projects

Background - X3ML Engine

- A tool that realizes the transformation of data resources to a target format with respect to X3ML Mappings
 - Efficient construction of RDF data from structured (XML) source data
 - Transparency in terms of expected output
 - Re-use of standards and technologies as much as possible
 - Facilitates the instance matching process



Background - X3ML Engine

- Handles the transformation of data expressed in XML to RDF
- Support various RDF formats: RDF/XML, NTRIPLES, TRIG, TRIX, TURTLE, JSON-LD
- Execution modes

Application with UI



CLI tool

```
C:\Repositories\Github\X3ML\target>java -jar x3ml-engine-1.9.4-SNAPSHOT-exe.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 @

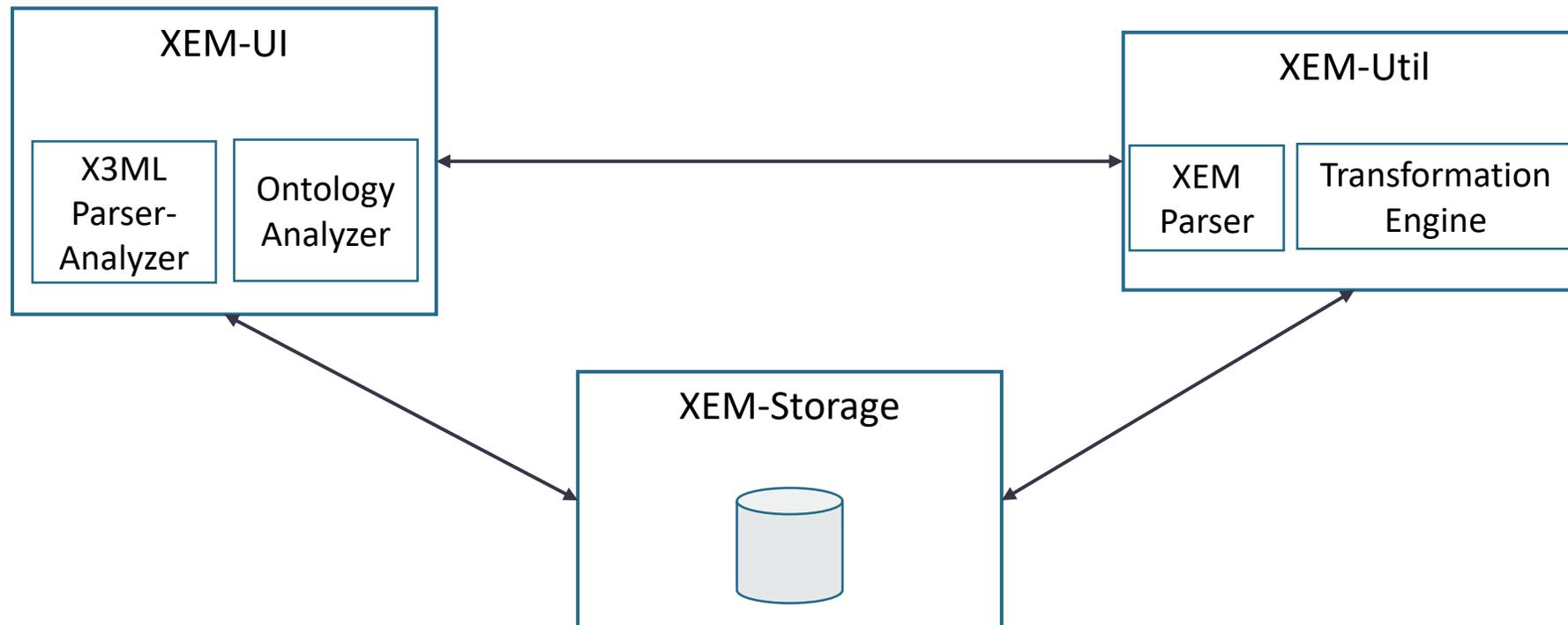
Missing required options: i, x
```

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();
```

XEM: X3ML Evolution Manager

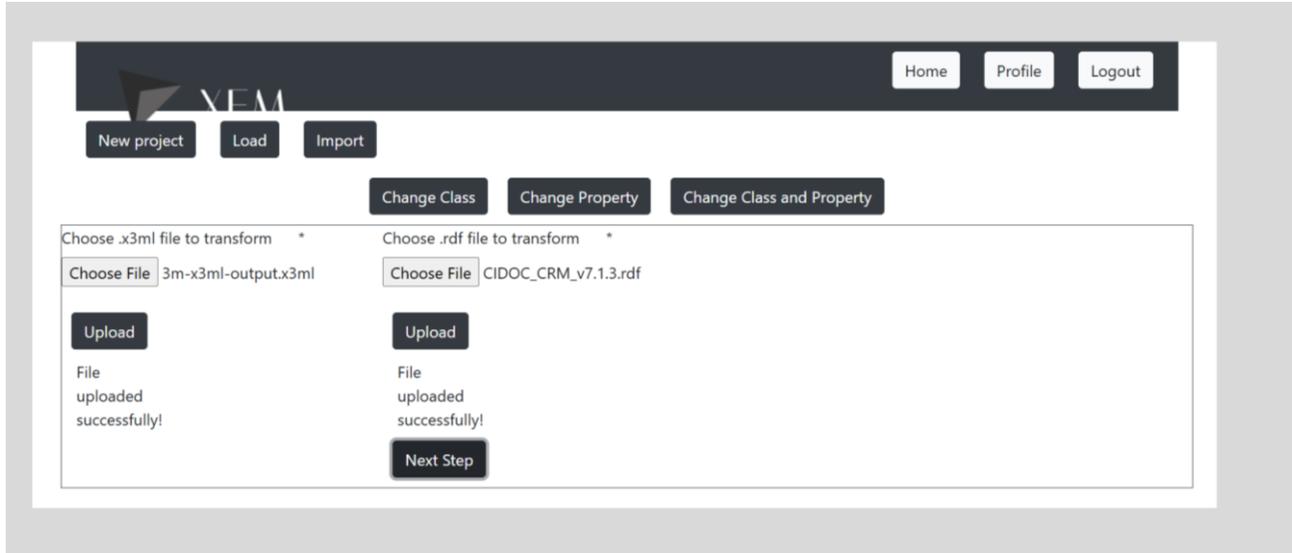
- A tool to facilitate the evolution of X3ML Schema mappings based on target ontology updates
- Describe once → apply many times



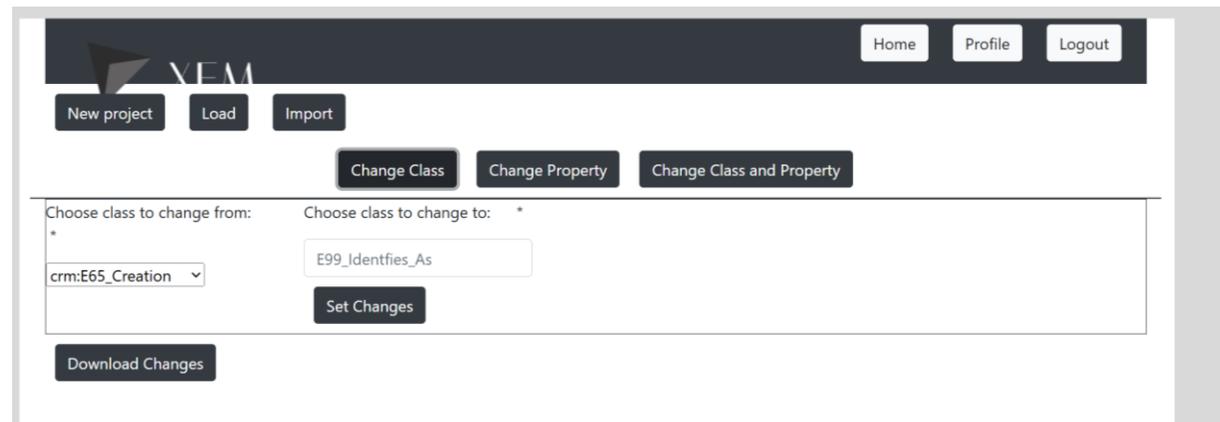
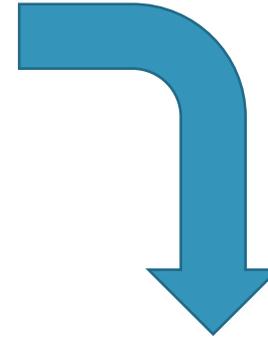
XEM: Requirements

- Load X3ML mappings and ontology resources
- Support class conversions (target path)
- Support property conversions (target relation)
- Provide recommendations based on X3ML mappings and ontology resources
- Definition of multiple conversions
- Persistence storage of projects directly in XEM
- Download projects in structured format
- Creation of updated X3ML mappings
- Offline creation of X3ML mappings

XEM: Screenshots



This screenshot shows the XEM web application interface. At the top, there is a navigation bar with the XEM logo and links for Home, Profile, and Logout. Below the navigation bar, there are three buttons: New project, Load, and Import. Further down, there are three buttons: Change Class, Change Property, and Change Class and Property. The main content area is divided into two columns. The left column is titled "Choose .x3ml file to transform" and contains a "Choose File" button, the filename "3m-x3ml-output.x3ml", an "Upload" button, and a message "File uploaded successfully!". The right column is titled "Choose .rdf file to transform" and contains a "Choose File" button, the filename "CIDOC_CRM_v7.1.3.rdf", an "Upload" button, a message "File uploaded successfully!", and a "Next Step" button.



This screenshot shows the XEM web application interface. At the top, there is a navigation bar with the XEM logo and links for Home, Profile, and Logout. Below the navigation bar, there are three buttons: New project, Load, and Import. Further down, there are three buttons: Change Class, Change Property, and Change Class and Property. The main content area is divided into two columns. The left column is titled "Choose class to change from:" and contains a dropdown menu with the value "crm:E65_Creation". The right column is titled "Choose class to change to:" and contains a text input field with the value "E99_Identifies_As". Below the input fields, there is a "Set Changes" button. At the bottom of the main content area, there is a "Download Changes" button.

XEM: XEM descriptions

```
[  
  {  
    "classBefore": "crm:E22_Man-Made_Object",  
    "classAfter": "crm:E22_Human-Made_Object"  
  },  
  {  
    "propertyBefore": "crm:P58_has_section_definition",  
    "propertyAfter": "crm:P1_is_identified_by"  
  }, {  
    "classBefore": "mtlo:BC10_Event",  
    "classAfter": "crm:E5_Event"  
  }  
]
```

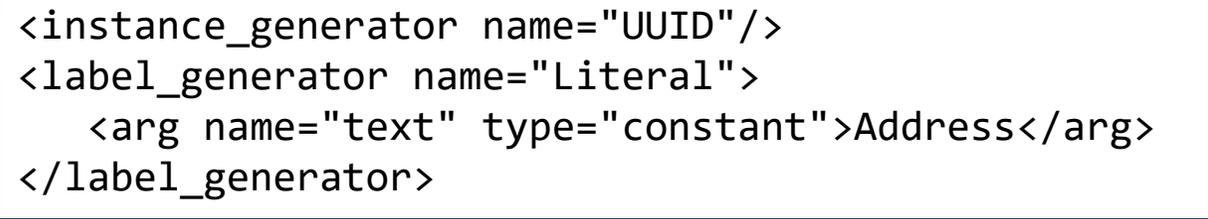
Updated class name (in the same namespace)

Updated property name (in the same namespace)

Updated class URI (under a new namespace)

XEM: XEM descriptions

```
[  
  {  
    "classBefore": "crm:E45_Address",  
    "classAfter": "crm:E41_Appellation",  
    "propertyAfter": "crm:P2_has_type",  
    "additionalClass": "crm:E55_Type"  
    "generation_info": "</>"  
  }  
]
```



```
<instance_generator name="UUID"/>  
<label_generator name="Literal">  
  <arg name="text" type="constant">Address</arg>  
</label_generator>
```

XEM: Execution (updated X3ML Schema mappings)

- Directly from XEM-UI
- Through XEM-Util
 - Useful for offline conversion of schema mappings
 - Provides a JAVA-API
 - Programmatically we can updated multiple schema mappings
 - Can be integrated with other tools

XEM: Details

- XEM was implemented as a student's thesis project (Jun 2024 – October 2024)
 - Implemented by Gerasimos Logiotatopoulos
 - Graduate of Computer Science Department – University of Crete
- Source code
 - [https://github.com/isl/X3ML Evolution Manager](https://github.com/isl/X3ML_Evolution_Manager)

Conclusions & Next Steps

- In a nutshell
 - XEM is a tool that facilitates the maintenance of schema mappings against target schemata (i.e. ontology) updates
 - Supports the definition of schema mappings updates in an intuitive manner (through a user interface)
 - Updates schema mappings based on XEM descriptions
- Next steps / much room for improvements
 - Not heavily tested on “production” environment
 - In future we could also
 - Integrate it with existing tools/systems
 - Focus on automation
 - Identification of required changes in X3ML schema mappings
 - Identification of changes in ontology versions

Thank you

Feel free to provide your feedback and ideas

(https://github.com/isl/X3ML_Evolution_Manager/issues)