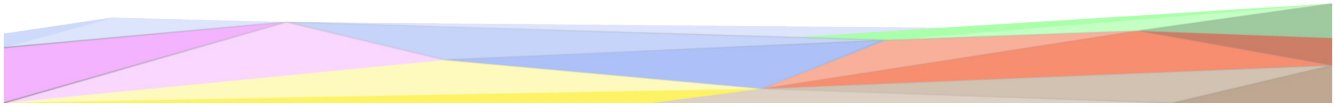
Dipartimento di Ingegneria e Scienza dell'Informazione

– KnowDive Group –

The iTelos methodology v1.0

Document Data:	Reference Persons:
August 2020	Fausto Giunchigla, Subhashis Das, Mattia Fumagalli, Alessio Zamboni, Simone Bocca, Mayukh Bagchi, Rui Zhang



Contents

Revision History:

Revision	Date	Author	Description of Changes

Glossary

CQ Abbreviation for Competency Question, real-life based examples from Personas or Users interviewed for the system, they describe a particular use-case question that the client is interested in getting answers for. 21–27, 29, 31, 33, 36, 55

DCAT Stands for Data CATalog vocabulary, it's a standard RDF vocabulary designed to facilitate the congregation of various data sets into specific catalogs, with this in mind it helps provide systems the operability of using metadata from various catalogs of merged data repositories, more information can be found here: W3C DCAT Documentation. 32

DKG Abbreviation for Data Knowledge Graph, object which represent the output of the data integration process. Based on the knowledge heterogeneity levels, it includes L1, L2, L4, and L5 data.. 56

EER Abbreviation for Extended ER Diagram, an EER diagram provides you with all the elements of an ER diagram while adding inheritances, categories or union types, specializations and generalizations plus sub and super classes for the entities defined. 5, 31, 32, 34, 36, 37, 45

EML A JSON-like schema mapping data sets into particular Ontology entities. 52, 56

ETL Abbreviation for Extract Transform Load, its the general procedure used in data science for the extraction of specific data, transformed via specific methods and then loaded into a new data set (an extraction and transformation of source data into differently context output data). 13, 23, 24, 32, 37

EType Provides a schema and set of rules for the creation of a conceptual representation of a real-world entity. "the definition and description of a set into which similar entity instances are classified (e.g. building)". It is a template that defines the constraints (set of rules) for creating attributes and relations of a real-world entity and classify them. 23, 30, 31, 33–37, 44

iteration A iteration in iTelos is the specific refinement of a phase for the purpose of advancing and refining the outputs previously made. 13–15, 22, 25, 29, 33, 37, 38, 42, 48, 55

KarmaLinker is a knowledge editor tool used for linking the data sets imported in relation to the L4 schemas constructed, various extensions provided with the tool help generate and modify the data sets in relation to the L4 schema, linking the various attributes to their corresponding L1-2 concepts. 56

KB Abbreviation for Knowledge Base, it's a technique for storing complex data structures describing facts regarding a world, useful in support of an inference engine to reason about the facts that are stored.. 11

KG Abbreviation for Knowledge Graph, it's a Knowledge based technology used for storing complex data structures. 11, 13, 55

L1 It's part of the levels of heterogeneity of Knowledge, Level 1 is the concept level knowledge called a Concept Space. 38–40, 43

L2 It's part of the levels of heterogeneity of Knowledge, Level 2 is the language level knowledge called a Lexicon. 5, 38–40, 43

L4 It's part of the levels of heterogeneity of Knowledge, Level 4 is the knowledge level that describes the SKG (L4 Knowledge Graph). 5, 30, 31, 35, 38–40, 45, 48, 56

metadata It describes the data contained from a certain data repository, the metadata stored for such data can describe its origin, author, date of construction, size, data types, keywords, abstracts, relations etc. 24

phase A phase in iTelos is a distinct group of activities grouped in regards to their specific output. 11, 13–16, 19–21, 24, 25, 27, 29, 32, 33, 36–38, 41, 42, 45, 48, 55, 56

Protégé Open source ontology editor and knowledge management system useful for the construction of SKGs, more information can be found here: Protégé Homepage. 36, 46

RapidMiner It's a platform containing various types of tools used in Data Science, it helps simplifying the activities relative to Machine Learning, Deep Learning, Data cleansing and transformation (and much more), more information can be found here: RapidMiner Website. 36, 37

RDF It's a standard model for data interchange and representation in the Web, more information can be found here: RDF Wikipage. 40

repositories Online databases containing data of certain domain(s), i.e an hospital data repository. 11, 39

SKG Abbreviation for Schema Knowledge Graph, it focuses on the concepts and relations level knowledge. 39, 44, 55, 56

UKC Abbreviation for Universal Knowldge Core, it's an expansion software based on the previous work of WordNet, it's main target is the definition of concepts in multi-lingual contexts. 40

yED It's a graph editor tool useful for creating EER, flow, activity, BPMN and other relevant diagrams, more information can be found here: yED Website. 34–36

1 Introduction & Representation Diversity

1.1 Context

The iTelos Methodology document has been written with the main objective of formally specifying and describing the activities taken place in the methodology.
As a contextual overview, Knowledge Graphs have been used to enable the usage and sharing of data since the early works of Knowledge Representation.

Knowledge Graph have enjoyed a resurgence as a research topic after the publishing of the *Google KG Project* back in 2012.

A Domain-specific Knowledge Graph is the main interesting topic output of iTelos, providing a structured data set for usage is a specific domain in a specific application or product.
Current Knowledge Graphs have a general cross-domain purpose, such as the Google one, and cannot be used for browsing the increasing huge amount of data produced in a domain-specific system.

These domain-specific data sets are sparse and diverse, collecting various huge amounts of fine-grained information and are continuously generated by various businesses and non-profit subjects.

The Knowledge Graphs being produced are cross-domain and do not have a specific scope, being produced over a broad scope.
As such most used Knowledge Graphs are very reliable in a general, cross-domain, point of view, but cannot be used for modeling the knowledge of different and multiple domain specific areas.

Another recurring problem is that the current methodologies used to refine the Knowledge Graphs do not focus on extending how the Knowledge Graphs are modified in terms of extension, integration and browsing the Domain-specific Knowledge.

1.2 Knowledge and Data Integration (KDI) Academic Course

The current documentation is the reference text for the Knowledge and Data Integration course held in University of Trento, (Dipartimento di Ingegneria e Scienze dell'informazione) by the Knowdive group. In this section is provided the structure of the course as well as the modality and objectives to reach. Following this course the students learn how to solve the data integration problem, described in section 1.3.1, through the solution proposed by the iTelos methodology (section 1.3.2.).

1.2.1 Objectives

1.2.2 Prerequisites

1.2.3 Course Modality

1.2.4 Exam Modality

1.2.5 Templates

1.3 Representation Diversity

1.3.1 The problem

1.3.2 Knowledge Graphs as solution

Knowledge Graphs [24] are a fairly new research topic that has been published in the last decade starting from 2012 after Google published the *Google KG Project* [36] .

It is an especially interesting concept usually closely associated with Semantic Web technologies and applied into cloud computing and large-scale data analytics applications.

1.3.2.1 What is a Knowledge Graph

Knowledge Graphs are often denoted with the blog entry by Google which describes them as an enhancement to their search engine with semantic properties.

On the same note Wikipedia does not provide information in general about the Knowledge Graphs and describes the Google implementation, such that no mentioning is given to the existence of other knowledge graphs.

Other definitions may lead to the assumption that knowledge graphs are a synonym for any graph-based knowledge representation but we argue that such a definition is not enough adequate for the possible applications of knowledge graphs.

By providing such a simple definition to knowledge graphs the entrance barrier for people who are unfamiliar with knowledge graphs will increase into confusion and misinterpretation of what a knowledge graph can be described and referred as.
Particularly even a graph-based vocabulary could be published as a knowledge graph.

Many definitions have been given for Knowledge Graphs and for this reason as follows some selected definitions have been compiled into a table.

Throughout various publishes the definition has been varying and for this matter a lack of the common core definition has been yet been given [13] .

According to the above definitions some possible extractions can be taken for describing what a KG is:

- Supposedly must cover at least a major portion of knowledge and not only restricted to a domain;

- A KG is generated from the extraction of information from one or more sources;
- Created for improving the accuracy of Q and A systems;
- It's serialized by a triple (s, p, o);
- Composed by a Schema Layer (see TBox) and a Data Layer (see ABox);
- Refined to improve coverage.

Definition	**Source**
"An intelligent model - in geek-speak, a graph - that understands real-world-entities and their relationships to one another: things, not strings"	Google Article
"A knowledge graph (i) mainly describes real world entities and their interrelations, organized in a graph, (ii) defines possible classes and relations of entities in a schema, (iii) allows for potentially interrelating arbitrary entities with each other and (iv) covers various topical domains."	Paulheim
"Knowledge graphs are large networks of entities, their semantic types, properties, and relationships between entities."	Journal of Web Semantics
"Knowledge graphs could be envisaged as a network of all kind things which are relevant to a specific domain or to an organization. They are not limited to abstract concepts and relations but can also contain instances of things like documents and datasets."	Semantic Web Company
"We define a Knowledge Graph as an RDF graph. An RDF graph consists of a set of RDF triples where each RDF triple (s, p, o) is an ordered set of the following RDF terms: a subject $s \in U \cup B$, a predicate $p \in U$, and an object $U \cap B \cap L$. An RDF term is either a URI $u \in U$, a blank node $b \in B$, or a literal $l \in L$."	Färber et al.
"[...] systems exist, [...], which use a variety of techniques to extract new knowledge, in the form of facts, from the web. These facts are interrelated, and hence, recently this extracted knowledge has been referred to as a knowledge graph."	Pujara et al.
"A knowledge graph (KG) is a directed, labeled multi-relational graph where nodes typically represent either entities or the at- tributes of entities, and (labeled) edges represent either relation- ships between entity-entity pairs or properties of entities. The simplest way to serialize a KG is as a set of triples, where each triple is of the form (h, r, t) and represents an edge in the graph"	Kejriwal

For what concerns their generation, KGs can be created by following different "from scratch" KG construction approaches:

i They can be the result of a well-founded conceptual modelling methodology;

ii They can be generated by the crowd

iii They can be extracted from semi-structured knowledge taken from web sources, by following different information extraction methodologies.

1.3.2.2 Applications of Knowledge Graphs

In the 1980s researchers from the University of Groningen and the University of Twente in the Netherlands initially introduced the term of KG to formally describe their knowledge-based system that integrates knowledge from different sources for representing natural language.

The authors that introduced the term of KG was relative to a limited set of relations and focusing on the qualitative modeling which included human interactions.
This was in complete contract to the idea which has been widely discussed in the recent years.
The authors proposed KGs with a limited set of relations and focusing on qualitative modeling including human interactions, which contrasts the idea of KGs that has been widely been discussed in the recent years

In 2012 Google introduced the Knowledge Graph as a semantic enhancement to the Google search function that does not match strings but enables searching for "Things" as real-world objects.
Although the implementation has not been provided in their blog posts, it has been cited over a hundred times.

Since 2012 the term KG has also been used to describe a family of applications such as DBpedia, YAGO, Freebase, Wikidata and Yahoo's competition to the Google search assistant tool.
These applications differ in characteristics and the lowest common denominator that tracts them together is their usage of Linked Data.
In the next table we provide a list of some example state of the art KGs, withe information about their construction method, publisher and status (i.e., in production or retired).

KG Name	**Publisher**	**Status**	**Construction Methodology**
Freebase	Google	Retired	Crowd sourcing
Wikidata	Wikimedia Foundation	Production	Crowd sourcing
Cyc and OpenCyc	Cycorp	Retired	Curated
DBpedia	Leipzig University	Production	KB Extraction
YAGO	Max Plank Institut	Production	KB Extraction
NELL	Carnegie Mellon University	Production	Semi-structured info extraction
Google Knowledge Graph	Google	Production	Semi-structured info extraction
Google Knowledge Vault	Google	Experimental	Semi-structured info extraction
Yahoo! Knowledge Graph	Yahoo! Holidays Inc.	Production	Semi-structured info extraction
Microsoft Satori	Microsoft	Production	Semi-structured info extraction
Facebook Entities Graph	Facebook	Production	Semi-structured info extraction

1.4 Representation Diversity State Of the Art

1.4.1 Researchers Work

The iTelos Methodology came to fruition from the development and prior works refined in the various fields specific to the tools and methods used in the distinct activities of the methodology.

As follows, a short description list of the various works produced by the group:

- Data Heterogeneity
  It's the problem found when categorizing the objects found in the data sets, in this case we present objects as concepts [22], through the categorization of concepts we are able of seeing how a Knowledge Representation that contains various types of diversities in the concepts founds in them presenting as such a Knowledge Diversity [16] . A next step in the categorization of the concepts is the work towards defining actions and recognition abilities [21].

- Dealing with data heterogeneity
  While dealing with Data Heterogeneity the group documented the possibility of moving, through the web data sets, into entity-centric knowledge modeling techniques [34] and defining a modelling system through an Extended ER model [27]. During the usage of the EER models, new tools were needed for categorizing the EER entities into the Knowledge Graph [29]. New work has been completed for systems able of producing expressive queries over the knowledge graphs constructed [17].

- Data and Schema Integration
  An important step into the integration of the Data and Schema level of the Knowledge Graph is the research completed in regards to the separation of the integration of the L4 schema and of the L2 annotation [11].
  Important work was completed with the realization of an example problem over food recipes which helped reinforce the usage of Competency Questions [37] as important tools for integrating the next steps of the Data and Schema integration.

- Schema integration examples
  Various works have been completed in regards to the usage and development of Schema-Knowledge Graphs, SKG [15] [2] [18]

- Data integration examples
  Various works have been completed in regards to the usage and development of Data-Knowledge Graphs, DKG [28] [12] [35] [5]
  An interesting example is the production of the DKG relative to the Digital University [30] application used by UNITN.

- Knowledge Graph visualization
  Various works have been completed in regards to the visualization of Open Data Entity-centric knowledge graphs [32] [23] [33]

- Linguistic Resources
  In the Linguistic area multiple works were completed through the advancement of the WordNet system, publishing works that produced an universal concept tooling called UKC [25] [20] [19] and by caterogizing such concepts into a large-scale lexical database called CogNet [1]. Various publications were made in regards to the linguistic resources, such as Scottish Gaelic [6], the resources for Mongol and Indian [31]. Tools were developed for helping the management [9] and the possibility of developing new language concepts [3] in UKC.

- Natural Language processing for data integration
  Various works have been completed in regards to the implementation and usage of Natural Language processing for the integration of data sets [10] [14] [4]

1.4.2 Students Work

The iTelos Methodology has been based on the previous case studies produced by students in particular problem solutions.

Thanks to the previous executions of the course 'Knowledge Data Integration' various projects were produced and their reports were used for categorizing and discussing the examples relative to each phase of the methodology.

In this case, three particular projects were taken as reference for particular examples:

- Space Domain [7]
  In this project the group was tasked in integrating data sources regarding places, attractions, buildings and itineraries found in the Trentino Alto Adige region to obtain a complete data collection making it easier for people to construct travel itineraries with small efforts up to the small details.
- Transportation in Trentino [26]
  In this project the group was tasked in solving the problem of finding the best itinerary and transportation choice that will impact less ecologically the province of Trentino as it might be hard for people to search the specific data on the web.
- Facilities and Events [8]
  In this project the group was tasked in constructing a Knowledge Graph relative to the possible activities and events found in Trentino facilitating the users in finding and searching activities, events and informations regarding organizations more easily.

Each project has been completed in the 2019-20 academic year and has been, in part, related to the prior works made by the group by identifying and testing the Data Integration system over the tools provided.

Each project will be used to collect and showcase examples of the activities discussed in the document.

The examples taken from the students works will be relative to the outputs produced in the various phases and activites, such as

1. Competency Questions;
2. Query Patterns;
3. Data sets extraction;
4. RapidMiner tool usage;
5. EER Models;
6. Ontologies;
7. Top-level ontologies;
8. KarmaLnker tool usage;
9. Knowledge Graph.

Given the fact that the iTelos Methodology is a new work instanced by the group, the previous students project might in part not be aligned with the document and in this case examples won't be presented.

This situation helps provide future works in executing the newly structured methodology and producing new examples for the document as presented.

The student project to which this document is taking examples from is predominately the Space Domain [7] as it's the most complete and detailed group project completed.

2 The iTelos Methodology Project

2.1 iTelos introduction

iTelos is a methodology based on the active refinement for the construction of a Domain-specific Knowledge Graph.

The main objectives of iTelos are as follow

1. Create an explicit and clear purpose for the construction of the Domain-specific Knowledge Graph;
2. To enhance the quality of the Knowledge Graph produced by defining an iterative validation of intermediate results;
3. To structuralize the activities in regards to the specific roles found in the project, providing as such a clear organizational split, exploiting as such the best identified roles in their respective activities;
4. To provide a set of tools and techniques (and guidelines) to improve the final result;
5. To make the methodology as semi-automatic and iterative as possible, resulting in a more cost efficient and time efficient methodology.

In iTelos the main roles interested in the execution of the project are three:

- Domain Expert
  Coordinates the project and is interested primarily by the output of the project as he's the main client interested in the KG produced.
- Knowledge Engineer
  Tasked with the development and execution of the Schema level activities and is mainly interested into particular activities in which a knowledge of Logic and Philosophy are required.
- Data Scientist
  Tasked with the development and execution of the Data level activities and is mainly interested into particular activities in which a knowledge of ETL and Data Science techniques are required.

The iTelos process can be divided into five main blocks of development activities.
These five blocks are connected each to particular evaluation processes that are used to evaluate intermediate and final outputs produced from each block.

The five main blocks are called as such

1. Scope Definition
2. Inception
3. Informal Modeling
4. Formal Modeling
5. Data Integration

The first block is the Scope Definition which is mostly concerned with the contextualization and grounding of the purpose for the construction of the DKG through the production of the problem documentation.

Here the Domain Expert is interested into listing and selecting the scenarios, personas, geographical and temporal contexts and the various data sources from which the data sets will be extracted from.

The second block is the Inception which is concerned with the selection and extraction of the main competency questions from the scenarios and personas previously written, in regards to the Data level the Data Scientist is tasked with selecting the data sets and extracting them from the various data sources.

In this block the main roles interested are the Knowledge Engineer and the Data Scientist.

The third block is the Informal Modeling which is concerned with the definition of the entities and relations that can be found in the first informal models being constructed through an EER diagram.

In the Data level the data sets are cleaned, transformed and enriched with new metadata in alignment with the entities and relations described in the EER diagram.

The fourth block is the Formal Modeling which is concerned with the construction of an Ontology where the lexico-semantic concepts found in the informal model are described and annotated through the usage of UKC, a multilingual expansion of the WordNet.

The fifth and final block is the Data Integration where the main objective is to take the data sets extracted and the ontology created to integrate them to generate the domain knowledge graph.
The iTelos Methodology is the documentation that has been specified and transcribed as a solution for the problem of structuring the iTelos methods for the output of a knowledge graph.

Guidelines have been defined, proving as such a standard manual document which can be attested and used as a starting point and as a look-out point for the activities that must be completed.

In the iTelos Methodology two main levels have been identified as the Schema and Data level, each corresponding to a different group role.
For each main phase an indexing structure has been followed:

- Top-level view
- Schema Level
- Data Level
- Evaluation
- Deliverables
- Tools

- Examples

The Top-level view conceives the general description of the phase in regards to its inputs and outputs and the main activities being taken place while showcasing its architectural view as part of a diagram.

The Schema Level describes the logical reasoning side of the project, compiled by the Knowledge Engineer, while the Data Level is executed by the Data Scientist taking into consideration the integration and transformation of the data sets in correspondence to the project purpose.

An Evaluation section describes the methods used for evaluating the efficiency and quality of the outputs produced in both the Schema and Data level, in this case the evaluation section contains a subsection which describes one of the main features of iTelos, the iterative approach of refinement of the outputs being produced.

In the Deliverables section an explanation is given in regards to the outputs being produced from the specific phase.

The Tools section describes the possible tools that the project can use for constructing the various outputs in a formalized manner, such that they can be later on refined and checked easily.
The Examples section will take some particular examples from the referenced projects made previously by students from previous academic years and discuss their implementations and first-hand examples that can be visualized and discussed.

2.2 The Methodology

2.2.1 Top level view

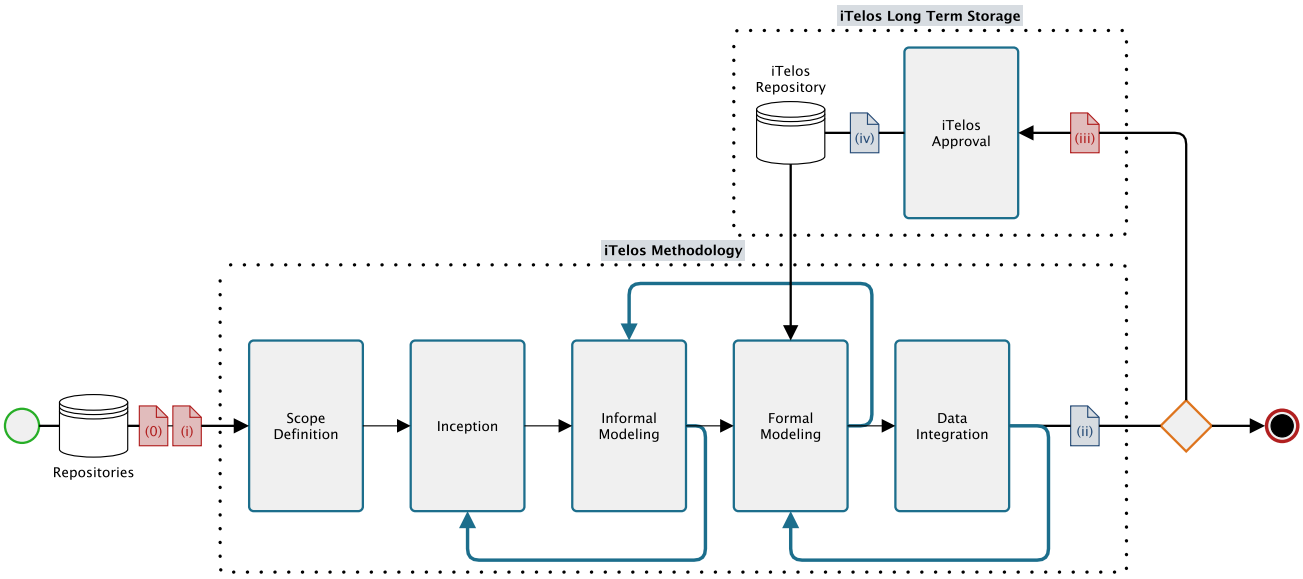


Figure 1: Top level view of iTelos

The iTelos Methodology is the structured and organized set of methods grouped into phases used for the realization of a specific problem into a Knowledge Graph, abbreviated as KG.
The document presents the methodology as a structure for understanding the categorization of the methods and best practices used for each particular phase of the iTelos Methodology.

Label	Description
0	List of Repositories and Data sets
i	Problem Purpose
ii	Metadata and Knowledge Graph plus Codebook and Project Documentation
iii	Metadata
iv	iTelos file

In the top level view we can see iTelos as a congregation of two methodologies of which this document will document the first one.

- iTelos Methodology;
- iTelos Long Term Storage.

The iTelos Long Term Storage is a second methodology used for evaluating and approving the outputs produced by the iTelos Methodology and to save its schema inside an iTelos Repository.
In the iTelos Methodology the iTelos Repository is used to find and extract a schema that will be used inside one of the main phases.
The main deliverables which can be seen in Figure 1 are represented in the approved set of documents with Label ii.

In specific the main objective of the iTelos Methodology is to, from an input given by the Domain Expert, represented as a set of repositories or Data sets and a Problem purpose to activate a methodology for the construction, via certain specific activities, of a KG which will be used for developing a KB.

2.2.2 Top level processes

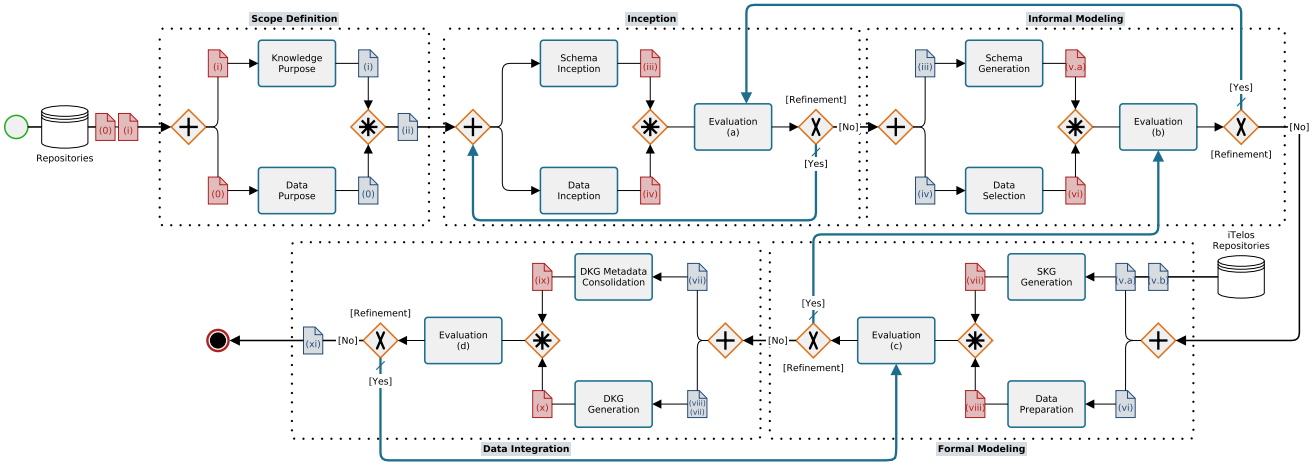


Figure 2: Top level view of the iTelos Methodology

After the view in Figure 1 we deepen the detail into the iTelos Methodology, expanding the black boxes presented as the 5 main phases into their main macro-activities.
A phase is defined in iTelos as a particular set of activities recognized as part of a macro-group containing a specific determined output.

In the iTelos Methodology we can see 5 phases as such categorized:

- Scope Definition
- Inception
- Informal Modeling
- Formal Modeling
- Data Integration

For each phase we can see a clear distinction and separation between a top and bottom level, in this case the levels identify the typology of activity being executed in such macro-activity.

Label	Description
0	List of Repositories and Data sets
i	Problem Purpose
ii	Purpose Documentation
iii	Competency Queries, eType and Properties terms
iv	Preliminary data sets and informal metadata
v.a	L4 informal schema
v.b	Teleology
vi	Data sets metadata and informal metadata
vii	L4 data set
viii	EML data sets
ix	L1,2 L4 and L5 metadata
x	L5 data set
xi	Metadata + Knowledge Graph, Codebook and Project Report

The 2 main levels of the iTelos Methodology are the Schema and Data level.

The Schema level is defined by the top macro-activities found in each phase (*Knowledge Purpose, Schema Inception, Schema Generation, SKG Generation and DSKG Metadata Consolidation*).
These activities are executed by the Knowledge Engineer.

The Data level is defined by the top macro-activities found in each phase (*Data Purpose, Data Inception, Data Selection, Data Preparation and DKG Generation*).
These activities are executed by the Data Scientist.

A third level is viewed as the combination of the Schema and Data level in which the Evaluation tests are made, identifying the possibility of proceeding or not in the iTelos Methodology.

2.2.3 Roles and Timeline

In the iTelos Methodology three main roles are identified for the implementation and execution of the activities.

1. Domain Expert
    - Coordinator of the iTelos project;
    - Main protagonist of the iTelos methodology as he will initialize the purpose documentation for the project;

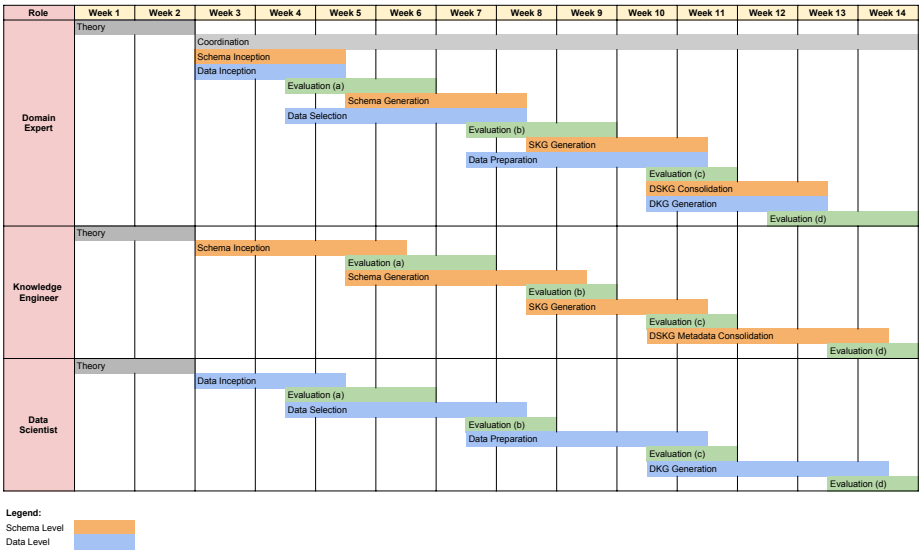


Figure 3: iTelos Methodology project GANTT chart

- Controls the evaluation activities of each phase, determining if the outputs produced are correct;
- Main client interested in the KG produced.

2. Knowledge Engineer
   - Tasked with the development and execution of the Schema level activities;
   - Mainly interested in activities in which a particular knowledge of Logic and Philosophy are required;
3. Data Scientist
   - Tasked with the development and execution of the Data level activities;
   - Mainly interested in activities in which a particular knowledge of ETL and Data Science techniques are required;

The project takes place on a schedule of 14 weeks in which the team is required to execute and communicate between each-other the various activities and outputs relative to the phases of the iTelos Methodology.

2.2.4 Process Iterations

As previously described, the iTelos methodology is constituted of phases.
These phases must be executed a number of times for constructing the correct outputs.

For this reason we introduce the need of Iteration.

Definition 2.1. Iteration A repetition of a mathematical or computational procedure applied to the result of a previous application, typically as a means of obtaining successively closer approximations to the solution of a problem.

An Iteration in the iTelos Methodology is based on the loop closing effect for each main phase (excluded the Scope Definition), where each main phase will be evaluated and re-iterated for conducting refinement activities on the outputs precedently made.

This Iteration perspective must not be mixed with the **wrongful evaluation** of the phases.
As such, if for example the Informal Modeling is not correctly produced in the Evaluation (b) activities it will be detected that the outputs are not correct, it might be reasonable to return to the Evaluation (a) in the Inception to check if, for what was produced in the Informal Modeling is aligned to what was previously designed.

If the alignment is not correct then the Inception phase is restarted, knowing in fact that the problems identified might have been produced before-hand.

Meanwhile, for an Iteration the idea is that for the example made, when the outputs are evaluated in the Evaluation(b) they result correct and a following repetition of the Informal Modeling will produce an even better and refined output following what was produced before-hand.
Each main phase will be justified by a certain amount of iterations that will be discussed in each Evaluation section.

What comes as part of the diagram shown in Figure 2 is the separation of the processes being executed during a particular phase.
The main scope of the division is the construction of non-homogeneous outputs between the Schema and Data level, elevating as such the possible construction of new ideas and possibilities to enhance the outputs.

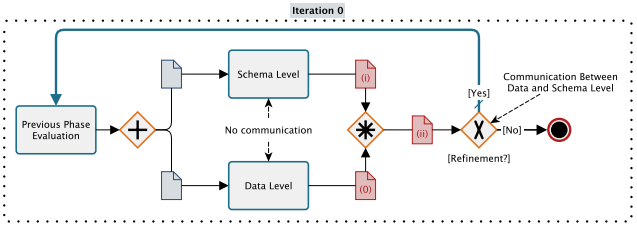


Figure 4: Open loop Iteration 0 of a generic phase

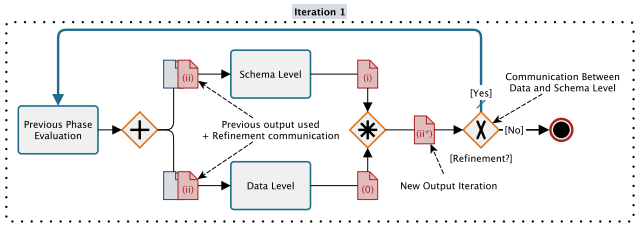


Figure 5: Closed loop Iteration 1 of a generic phase

The idea of the iteration is to, from a first execution time, not have any kind of iteration between the two levels, in this case the loop is 'open', as in there's no output previously produced that can help the two roles, Knowledge Engineer and the Data Scientist, and that they're not prone on communicating in-between the activities, producing as such a very varying output for each level.

After a first iteration is passed, and communications are made between the roles, the outputs produced before-hand will be used as standards for which new refinements and activity executions are based upon, constructing as such a 'closed' loop between the documents being used for a future refinement.

As it can be seen in Figure 4 the loop is open, as there is no previous communication between the two roles and, after a refinement process is applied, both the Data and Schema level communicate their outputs (as presented by Label ii) and determine the refinement processes or new executions needed in the next iteration.

In this case, as by Figure 5 the loop is closed, as there is a previous communication between the two roles and the outputs produced during the previous iteration are passed as input to the macro-activities in the new iteration, in this case the previous outputs are visioned and used for proceeding with the next refinements in their specific levels and activities.

With the closed loop effect, the outputs produced beforehand, in the same phase, are used as a guideline for refining or producing new outputs that will be later on evaluated and/or refined in another set of iterations.

Each phase has been set to a certain specific number of minimum iterations to follow, this to standardize and help guide the roles into identifying the types of activities that must be done in each iteration.

2.3 Projects Illustration

3 Scope Definition

3.1 Top level view

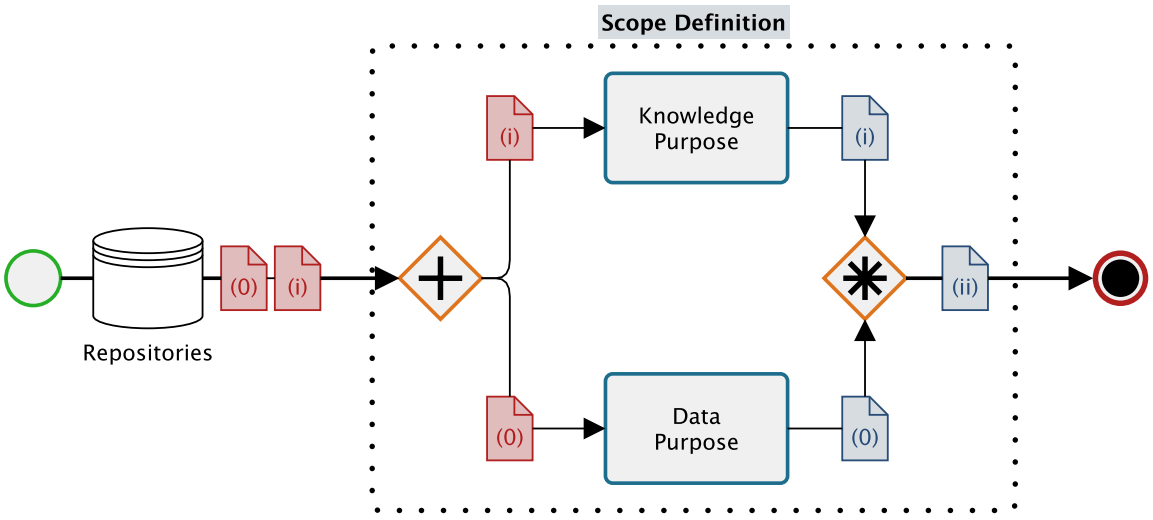


Figure 6: Scope Definition Diagram

The Scope Definition is the first phase of the iTelos Methodology. It aims to define the project's purpose, and more in detail it has to answer to the following questions:

- Why the iTelos Methodology has to be adopted ?
- Which is the problem's context and how is defined ?
- Which problem the Methodology will solve in the context ?

Label	Description
0	List of Repositories and Data sets already available.
i	Problem Purpose
ii	Purpose Documentation

Keeping the focus on the questions above, the first phase aims also to identify and localize the data needed to solve the problem. The main role interested in both macro-activities is the Domain Expert which describes the problem both at Schema and Data level.

3.2 Knowledge Purpose

In the Knowledge Purpose the sub-activities being executed are the following:

- Problem Context definition
- Problem Why definition

In the current macro-activity the Domain Expert has to identify and define the context in which the project will be focused on, and so why to solve the problem in that context, as well as why the iTelos Methodology is needed to solve it. Moreover the documentation produced as output of this phase has to specify who will get benefits from the solutions achieved.

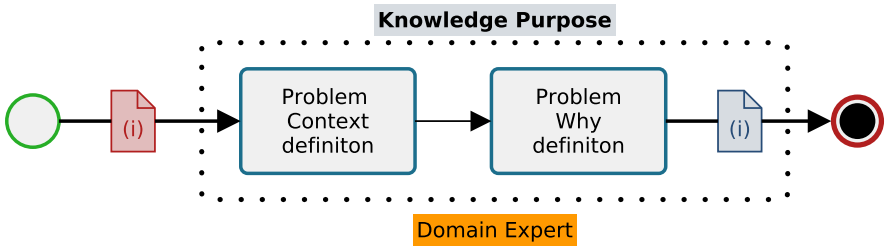


Figure 7: Knowledge Purpose Diagram

3.2.1 Problem Context definition

In this sub-activity the Domain Expert identifies and defines the context in which the application will work in. The context definition is crucial in order to identify the correct datasets and how to use them. The Domain Expert has to define the fundamental aspect of the problem's context:

- Geographical aspects;
  These aspects regard the geographical scope in which the problem is trying to solve a certain set of tasks, if specific to a certain location. (i.e. a Transportation solution in Trentino)
- Temporal aspects;
  These aspects regard the temporal scope in which the problem wants to solve a certain set of tasks such as the future, past or present data being given to study and execute a certain task. (i.e a Museum requiring an easy-to-use paintings or monuments information visualization product)
- Domain general aspects;
  These aspects regard the domain in which the project is carried on. Defining these aspects the Domain Expert gives the fundamental elements to define the problem as well as the objects used to solve it.

3.2.2 Problem Why definition

In the definition of the *Why* the Domain Expert has to integrate the documentation, started in the previous step, with the information regards why the problem has to be solved. More in detail the main aspects to define are:

- Purpose for starting the Project;
- Problem description;
- Personas and Scenarios relative to the problem;

The three points listed above follow an order starting from the most general one to the most specific, with the objective to define precisely Why the project was started. Note that this definition is done following the context elements defined in the previous step.

In a first moment the Domain Expert starts to identify and provide more details on the reasons and motivations required for the construction and beginning of the iTelos methodology, for the construction of a Knowledge Graph to achieve a solution for the problem. Then, in a second moment, the problem is better specified defining

the real objective to achieve, in terms of context elements. In the end of the sub-activity, to improve the description of the problem, the Domain Expert has to provide descriptions of the scenarios and actors who plays relevant role in the problem definition as well as in the solutions achievement.

These sub-activities are mostly documented with a textual document, generated in output, which can be composed of tables and of sections relative to the various arguments being described.

3.3 Data Purpose

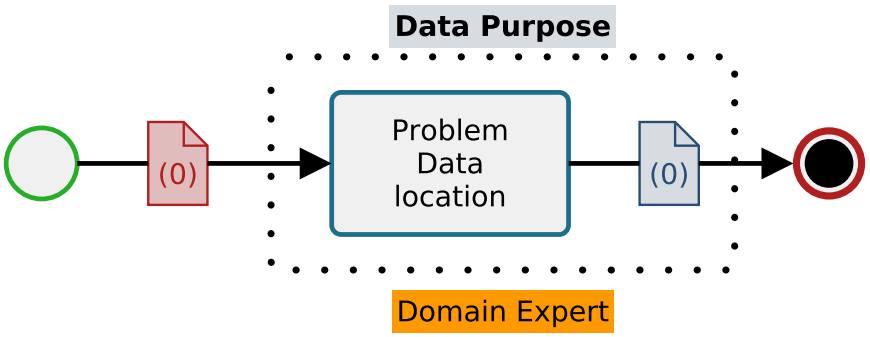


Figure 8: Data Purpose Diagram

In the Data Purpose the main activity being executed is the following:

- Problem Data location

This macro-activity describes how the project's purpose definition influence the identification of the datasets used for instancing the project.

3.3.1 Problem Data location

In this current sub-activity the Domain Expert is tasked with producing a list of the main data sources used to collect data for the project. The list can contains different types of data sources, such as:

- Open data repositories
- Data sets, such as private databases
- Web pages to gather data

The Domain Expert catalogs the various data sources into a sheet which can later on be used as a foundation from which data sets will be picked and extracted from. The data sets used by the project will be categorized in regards of their provenance, helping in future activities the description of the methods of extracting the data sets. The output of this sub-activity will integrate the documentation produced in the previous step, with the catalog list of the data sources.

3.4 Languages & Standards

During the Scope definition phase the usage of Overleaf tool is required to document the phase's output. For this reason the language requested to complete the work, is LaTeX. Regarding the data level output, the usage of Excel of Google Sheet is requested, so the Domain Expert has to be able to produce document following those standards.

3.5 Tools

In the Scope Definition the tools that are mostly used are divided between the two macro-activities. For the Knowledge purpose the main output is the documentation, due to that the tool adopted is Overleaf through which is possible to produce PDF documents. For the Data purpose the main output is a spreadsheet with the relative list of data sources, due to that the tool used are Excel or Google Sheets.

3.6 Deliverables

In the Scope Definition phase the output consists of a documentation that include:

- **iTelos project report** : This document will be, in the end of methodology, the main report for the project. The creation of the project report starts in the current phase, with the definition of the initial most important aspects which define the project itself:
  - **The context** : Definition of the domain and environment where the project lives.
  - **The actors** : Definitions of the actors (persons or things) which play in the context defined.
  - **The scenarios** : Definitions of all the possible scenarios played by the actors.
  - **Project purpose** : Definitions of the problem to solve through the adoption of the data integration methodology.
- **Data sources sheet** : This sheet (i.e. Excel sheet) created by the Data Scientist, collect all the possible data sources that will be used in the project.

3.7 Examples

The particular example that we will be looking at is referring to the student project named Space Domain [7].

In this first phase we will discuss examples regarding the definition of the Context, What and Why of the problem purpose and data sources collected.

3.7.1 Examples of Knowledge Purpose

> Trips organization is a complex problem, which people don't want to spend time on. The solution we propose is an integration of data regarding places, famous attractions, itineraries and any kind of points of interest. In order to obtain a complete data collection, we intend to integrate data about facilities like hospitals, shops, and public structures, and transports. It is important to adapt the trip to your needs in order to let you better enjoy the experience. Crossing data, it is possible to plan the whole vacation with no effort up to the smallest detail.

Figure 9: Space Domain Problem purpose

NAME	AGE	INTEREST	USAGE	DESCRIPTION
Maria (1)	25	Visit cities, popular attractions and museums.	Travel more and spent less. Trips focus on culture.	Maria is a student. She wants to use the system to organize her trips. She wants to spend as little as possible but at the same time, she wants to visit many places. Using the system could be an optimal solution, crossing the data and prices you can get a list of accommodation and facilities with the most advantageous price. Organizing a trip by public transport requires a huge amount of data to find the best connections, the proposed system will also provide such data.
Giovanni(2)	45	Enjoy holidays with his family, Skiing on winter and biking/hiking/trekking on summer.	Organize the trip, suitable sports with different levels of challenge. Parking lots (accommodation).	Giovanni works as a professional and goes on holidays with his family. He wants to use the system to organize their vacations in Trentino. Their holidays differ in length, type of sport and the level of challenge of the sports they practice. So, for each of their holidays, they have different needs about the type of staying structure. They travel with their own vehicle.

Figure 10: Space Domain Personas

Given the purpose of the project as seen in Figure 9, the main activities being identified is the selection of specific Personas and Scenarios that will be used to explain and evolve the project upon.
In the description of the Personas as seen in Figure 10 the important properties to define are the interest and usage of which they are interested in the problem.

In this case the Interest is related to the types of activities they would be satisfied in doing, in the real world.
The usage is related to the types of benefits the persona is interested in obtaining from the application.

As it can be seen in Figure 10 two personas are identified, they both have different types of interests and possible application usages.

In their respective Description columns importance is given in respect to the types of activities and backgrounds the specific persona has, enriching the possibility of developing various types of possible Competency Questions in the Inception phase.

Nowadays people spent their free time travelling around. In our frantic world, time is running short and people want to improve the quality of their trips. Let us explore some possible personas.

Maria is a young woman; she is still studying at the university and next year she will graduate. She has free weekends because lessons end on Thursday morning. She travels with her friends who are students at the university like Maria. Maria plan one trip per month, she uses public transport, and she prefers to spend as little money as possible. As lessons start on Monday morning, she has only the weekend available, and she looks for accommodation for a maximum of 3 nights. She is really interested in visiting cities, the most popular attractions and museums.

Giovanni is a husband and a father of two young children. He usually travels with his family for two different weekly trips a year, and occasionally they take weekend trips during the year. Travelling with the family requires planning the whole trip to avoid problems. As Giovanni is a precise person, he takes care of the whole organization. He and his wife love to practice sports in the visiting areas, but at the same time, they look for attractions, like adventures parks, suitable for their 7 and 11 years old children. The budget is high because both Giovanni and his wife have a substantial salary: they are lawyers. Giovanni has a beautiful car, so he wants to travel in comfort using it.

Figure 11: Space Domain Scenarios

In the definition of the Scenarios in Figure 11 the importance of the development of the scenarios of the personas is relative to the amount of detail and interest a specific persona has in the application.

In this case the Scenarios are important for the future development of the various types (Core, Common and

Contextual) of Competency Questions found in the Inception phase.

3.7.2 Examples of Data Purpose

```
https://www.booking.com
https://scrapy.org/
https://dati.trentino.it/dataset/esercizi-alberghieri
https://dati.trentino.it/dataset/trasporti-pubblici-del-trentino-formato-gtfs
http://sasabus.org/it/opendata
https://developers.google.com/transit/gtfs
https://www.istat.it/it/archivio/6789
https://www.openstreetmap.org/
```

Figure 12: Space Domain Data Sources

In the project, importance is given to the data sources from which data sets will be taken from.
In this case, as by Figure 12 the group has selected 8 data sources from where data sets will be extracted.

A justification to this example is that the data sources must be linked via a spreadsheet or table, not as a footnote to the project report.

4 Inception

4.1 Top level view

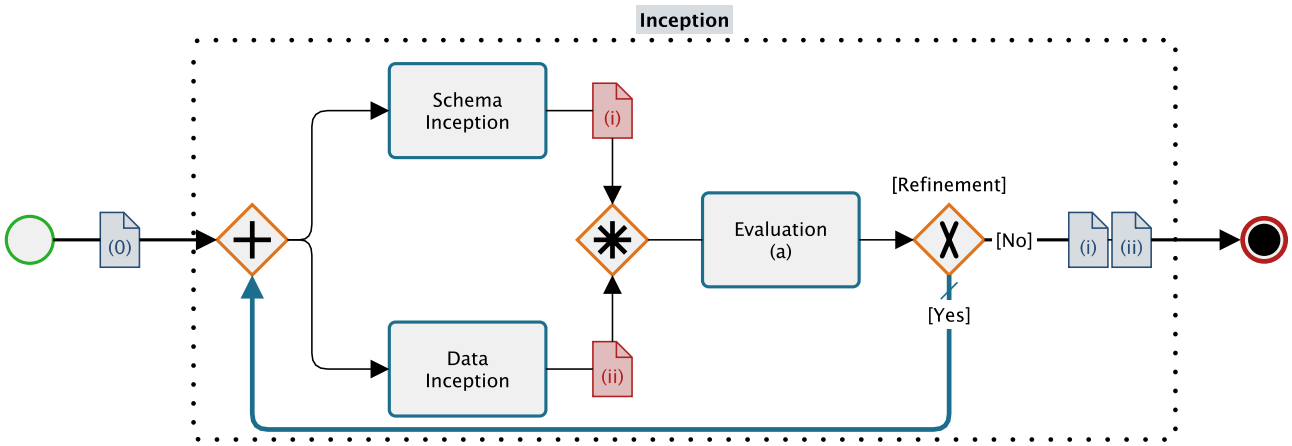


Figure 13: Inception Diagram

The second phase of the iTelos Methodology, called *Inception* aims, starting from the Purpose documentation coming from the previous phase, to define what are called CQ, *Competency Questions*, that in the end of this phase will became *Competency Queries* defining more precisely all kinds of queries can be generated to solve the problem as defined in the previous phase. As the diagram above shows, in parallel, a first data and metadata selection is performed.

Label	Description
0	Purpose Documentation
i	Competency Queries with Data Objects definition
ii	Preliminary datasets and informal metadata

The Knowledge Engineer and the Data Scientist are respectively in charge of the Schema Inception and Data Inception activities. Starting from this phase the work of the actors in the Methodology is scheduled following the iterative process, as described in the diagrams. As described in the Phase Iterations sections presents for each future phase (starting from the current one), the activities are executed following a certain number of iterations, in which the input (for both schema and data level) is the output of the previous iteration, performing in this way a refinement procedure. In the end of each iteration the output is evaluated and will be either iterated for refinement or denied for not following correctly the purpose documentation.

4.2 Schema Inception

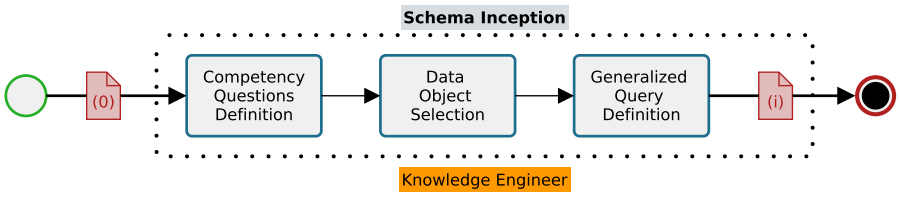


Figure 14: Schema Inception Diagram

In the Schema Inception the main sub-activities being executed are the following:

- Competency Questions Definition
- Data Object Selection
- Generalized Query Definition

The objective of this macro-activity is to describes the first steps needed for defining the structure of the data schema for the problem, that the Knowledge Engineer has to produce. The first step for the data schema definition, that this macro-activity has to perform, is the definition and documentation of all the possible questions (later specified more precisely as queries) that can help to solve the problem.

4.2.1 Competency Questions Definition

In this first sub-activity the Knowledge Engineer using the Purpose documentation, has to define the CQ. In order to do that She/He starts from the actors and roles as well as the scenarios definitions coming from the previous phase, and proceed to list (i.e. interviewing specific actors in specific scenarios) all the possible questions, and relative answers, relevant to solve the problem.

The Knowledge Engineer categorizes the CQ collected during the phase iterations, following the different data typologies:

- Core data
  Describes the most important entities and properties relevant to the problem space. These entities are the subject of the project, the most important in the solution achievement.

- Common data
  Describes the common entities and properties relevant to the problem space (such as time and location). These entities are the most used in order to define common aspects of the world in which the data live.

- Contextual data
  Describes the extensions of the entities and properties relevant to providing specific details to the problem space. These entities are used to describe specific aspect of the data.

It is important to note that the three data typologies listed above, define a *dependency hierarchy* among the data. The Common data have the strongest impact in terms of dependencies because they describe common aspect used as base from the other kinds of data. In the second position there are the Core data that are the most important entities regard the project's solution. They create dependencies on the Contextual data, but are often defined using the Common data for this reason they depend on the latter. In the end, the Contextual data are a specification of the previous typologies of data, so they depend on both the previous kinds of.

4.2.2 Data Object Selection

After the previous sub-activity, the Data Object Selection starts using the CQ definitions. The scope of this internal step is to identify and list, in a preliminary and general way, the main data object involved in the questions defined before. This general object are the first version of what will be called EType, and in the current phase are used in the next sub-activity to improve the CQ definition.

4.2.3 Generalized Query Definition

The last sub-activity of the schema level in this phase, aims to define more precisely all kinds of queries which can be useful in the solution achievement. To obtain this result, the Knowledge Engineer uses the preliminary defined CQ together with the data object defined in the previous step, and proceed to write a list of queries in a more precise format (i.e. SQL-like language). The output of this sub-activity will be a document defining a semi-structured version of the queries needed to support, merged with a first definition of the kinds of object that have to be handled within the project.

4.3 Data Inception

In the Data Inception the sub-activities being executed are the following:

- Data sets Selection

- Data sets Metadata Enrichment

This data level macro-activity of the iTelos Methodology describes the extraction of the datasets starting from the data sources list in the Purpose documentation coming from the previous phase. The Data Scientist is in charge of this task which will use his skills in ETL techniques to extract the data and to enrich it with its metadata.

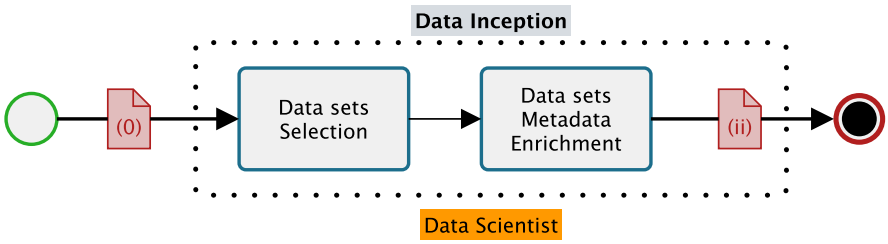


Figure 15: Data Inception Diagram

4.3.1 Data sets Selection

In this first data level sub-activity, the Data Scientist, has to analyze all the data sources listed in the Purpose documentation. The objective of this step is to identify, within those sources, the single datasets needed to solve the project's problem. Having the documentation from the previous phase, the Data Scientist (working in parallel with the Knowledge Engineer) has to identify the correct datasets respect to the users and scenarios previously defined, as well as understand the data types and formats within the datasets which have to be collected. It is important to note that the data sources could be different, local or remote, free or under specific permissions. Moreover, some data sources could be not a repository but instead specific web location from where a scraping procedure is needed in order to extract the data. Due to these reasons, in the current sub-activity the Data Scientist has to use some ETL techniques through the usage of specif tools, better described in the *Tools* section of this phase.

4.3.2 Data sets Metadata Enrichment

In the Data sets Metadata Enrichment the Data Scientist is tasked on enriching the datasets selected and extracted with record-level metadata, such as its provenance, historical value or any other interesting metadata relative to the problem. Part of the metadata might have to be discovered by reading the importance of the problem context, as such the Data Scientist will have to document himself by reading the Purpose documentation once again, finding if certain specific context might be documented via metadata relative to the data sets selected.

This first metadata documentation together with the datasets selected in the previous sub-activity form the data level output of this phase.

4.4 Evaluation (a)

Main aspects for Inception evaluation:
- alignment between project informal knowledge collected and datasets collected.

In the Evaluation (a) the objective is to evaluate the output of the Inception phase.
The output of this phase is made out of CQs plus the preliminary list of data sets with informal metadata selected out of the problem documentation.

The class and properties are collected into two types of sets, S_c and S_p from the CQs and the reference alignment data set.

There are two quotas between the sets from different sources, the coverage and the flexibility.

Definition 4.1. Coverage (Cov) is the coverage between two sets α onto β, the percentage of the difference from α from β.

$$Cov(\alpha, \beta) = 1 - \frac{|\alpha - \beta|}{|\alpha|}$$

Definition 4.2. Flexibility (Flx) is the flexibility between two sets α onto β, the percentage of the difference from α from β.

$$Flx(\alpha, \beta) = \frac{|\beta - \alpha|}{|\beta|}$$

The Coverage calculates the ratio $\alpha \cap \beta$ to α which is the percentage of the join set to the source set α.

The Flexibility returns the ration $\beta - \alpha$ to β which is the percentage of the leftover of the target set β to itself.

The evaluation is focused on the quality of the CQs and the preliminary selection of the data sets.

$Cov(C_{c/p}, D_{c/p})$ evaluates the percentage of the overlapped part of CQs, where C and D stand for CQ and the referenced alignment data set.

c/p stands for the type of set, classes as c and properties as p.
It is rational to have a relatively higher value of this quota, such as 0.8 for instance, because by increasing the overlap it increases the overlap by respect to the CQs selected.

$Flx(C_{c/p}, D_{c/p})$ evaluates the leftover percentage of the reference alignment data sets.
It is rational to have relatively low value of this quota, such as 0.2 for instance, because by decreasing the value it increases the overlap by respect to the data sets selected.

4.5 Phase Iterations

In the Inception phase the minimum number of iterations required for the production of an high quality output is equal, or more than, four iterations. The iterative process provide at each iterations a more defined and precise input to the activities. In this way the schema and data level work at certain iteration is guided by both the schema and data level output of the previous one. More in detail each iteration is focused on a specific data typology, among the three already mentioned, Common, Core and Contextual.

4.5.1 Iteration Zero

In the first iteration the main output of the schema level is a document with the definitions of general queries with the general object definition for the Common data typology. Regarding the data level, instead, the first iteration aims to identify eventually missing data sources.

4.5.2 Iteration One

In the second iteration, at schema level, the Knowledge Engineer has to define the general queries and the general objects definition for the Core data typology, using the previous defined Common elements if necessary. At data level the Data Scientist has to extract the dataset and collect metadata for the Common typology, using the schema level output of the previous iteration. The evaluation activity at the and of the iteration, verifies the datasets extracted using the schema elements defined in the parallel activity

4.5.3 Iteration Two

In the third iteration, at schema level, the Knowledge Engineer has to define the general queries and the general objects definition for the Contextual data typology, using the previous defined Core elements if necessary. At data level the Data Scientist has to extract the dataset and collect metadata for the Core typology, using the schema level output of the previous iteration. The evaluation activity at the and of the iteration, verifies the datasets extracted using the schema elements defined in the parallel activity.

4.5.4 Iteration Three

In the fourth iteration, at schema level, the Knowledge Engineer has to perform a general check on the documentation produced in the previous iteration, in order to identify eventually missing elements. At data level the Data Scientist has to extract the dataset and collect metadata for the Contextual typology, using the schema level output of the previous iteration. The evaluation activity at the and of the iteration, verifies the datasets extracted using the schema elements defined in the parallel activity.

As a fourth output given by the phase the documents created are of higher quality and closer to the problem, future iterations might be done for completing the selection of the datasets being taken in correspondence to the advanced CQs but the main scope of the phase should have been achieved after the iteration four.

4.6 Languages & Standards

In the Inception phase the Knowledge Engineer has to produce as final schema level output the Generalized Query Definitions, which have to be expressed in a semi-formal way using. For this reason the usage of a SQL-based Language is requested. The Data Scientist, instead starts to collect the dataset needed from the data sources identified in the previous phase. Depending by the type of data sources identified the datasets collected can be expressed using one or more of the following format:

- XML
- HTML
- CSV
- JSON

The datasets collected in this phase can be obtained after scraping procedures. To implement those procedure the usage of programming languages (usually Python) is requested.

4.7 Tools

As in the previous phase also for the Inception phase the tool needed are different in the schema and data level. For the schema level the Knowledge Engineer can use a spreadsheet tool such as Excel or Google Sheet to produce the documentation containing initially the CQs, and, more detailed in the final activities, the Generalized Queries. For the data level the Scientist has to collect and manage data from the data sources defined in the previous phase. In order to do that a Python Jupyter environment is used. Thanks to this tool, that can be used both locally (standalone instance of the environment on the Data Scientist machine) and online, (using an external server where the environment is already set up) the Data Scientist can use several libraries, easily integrable in the Jupyter environment. Here a brief list of those libraries, grouped by purpose:

- **Data management**: Pandas, NumPy, Scikit Learn
- **Data scraping**: Beautiful Soup, Scrapy, Selenium, LXML
- **Data formatting**: Arrow, PrettyPandas, datacleaner

The List above reports only some of the libraries that can be used to handle the datasets.

4.8 Deliverables

In the Inception phase the main deliverables produced are:

- **iTelos project report** : The main project report document is updated with the list of CQs defined, as well as the Generalized Query defined formally in the document using an SQL-based language. Moreover the Data Scientist will update a specific section of the document describing the programming libraries used to extract and manage the data coming from the data sources defined previously.
- **Preliminary datasets sheet** : This sheet (Excel file) is an improvement of the data source sheet defined in the previous phase. It includes the list of the preliminary datasets extracted form the sources.
- **Metadata sheet** : This sheet (Excel file) is created in the current phase by the Data Scientist. It start to collect the metadata associated to the data, keeping the focus, in the current phase, on the provenance information of the data.
- **Metadata description** : Another document related to the metadata is generated in this phase, containing the description for each metadata collected in the above mentioned metadata sheet.

4.9 Examples

The particular example that we will be looking at is referring to the student project named Space Domain [7].

In this second phase we will discuss examples regarding the construction of the Competency Questions, extraction of patterns and revealing the type of entities and properties interesting to the problem.
Another important part of the phase is the explanation and selection of the data sources from which the data sets are taken from.

PERS	NUM	QUESTION	ACTION
Maria	1.1	Give the list of hotels near the station of Bolzano	The system search all the hotels within 5 km near the station of the city and returns all the fields.
Maria	1.2	Give all trains from the station of Bolzano to the station of Trento	Select the station of departure in Bolzano and all the trains available to reach the station of Trento will be provided with timetables.
Maria	1.3	Give the list of museums of Bolzano open on Sunday	A list with distances and other info will be provided (description).
Maria	1.4	Give the religious attraction timetable of Merano	In order not to lose the best monuments and churches of the place, the structure with the relative timetable will be provided.

Figure 16: Space Domain Competency Questions

Giovanni	2.1	Give the list of family accommodations near Garda Lake Trentino having parking	Extracts and returns all the accommodations within 15 km from municipality Riva del Garda that have four or more NumOfBeds and have parking.
Giovanni	2.2	Give the list of all cultural options in Trento, Rovereto, Arco, Riva d/Garda for the week 23/12/2019 - 29/12/2019	Extract and returns all attractions of type "culture", "NightlifeEntertainment" ocurring within 5 km from municipality of Trento, Rovereto, Arco and Riva del Garda; scheduled for the required period 23/12/2019 - 29/12/2019.
Giovanni	2.3	Give the list of smooth biking paths activity options close to Dro and upcoming for the next three days from 15/12/2019	From attractions of type "SportLeisure", select those with activityPath-s within 5km of distance from municipality of Dro, having the SuggestedType "bike" from activityPath, having difficulty Low (L), and scheduled in the next three days after 15/12/2019.
Giovanni	2.4	Give the list of natural climbing areas of Trentino	Extract and returns all points of interest in province of Trento being attractions of type SportLeisure with ActivityPath-s of Medium and High difficulty, PositiveGradient of more than 25%, and suggestedType: walk or other.

Figure 17: Space Domain Competency Questions

4.9.1 Examples of Schema Inception

All the Competency Questions must be placed inside a table or spreadsheet such as in Figure 16 and Figure 17, enumerating them for easier selection in the patterns evaluation and table construction.

In this particular activity the group was tasked with extracting from the Personas and Scenarios previously defined the types of Competency Questions and their relative actions, purposely answers to the questions being selected.

Here the main columns of the table are the Person to which the Question and Action are linked with.

The question is the particular type of activity a Persona is interested in doing or finding information about and for such question an Action is described, where a simple explanation of the application execution must be given.

NUM	TYPES	PROPERTIES
2:1-13-19, **4:**1-23-24	Generic (Accommodation)	type
1:1-24, **2:**1, **4:**3-10-23-24	Hotel (Accommodation)	price, stars, parking, number of beds, wellness
2:8, **3:**21-22-23	Lodge (Accommodation)	-
2:6, **3:**5-6-10-11-12	Camping (Accommodation)	price, parking, number of spots

Figure 18: Space Domain Query Patterns

After the Competency Questions are selected the patterns for such queries must be extracted, categorizing the specific types of properties and activity interested in the query.

All the Query patterns are placed inside a table or spreadsheet such as in Figure 18, categorizing the enumeration previously given to the CQs in respect to the type and properties found in such CQs.
In the case given by the first row in Figure 18 the first CQ selected from Giovanni is part of a generic type of action of which only the type accommodations is interested in solving the question.

4.9.2 Examples of Data Inception

Empty

5 Informal Modeling

5.1 Top level view

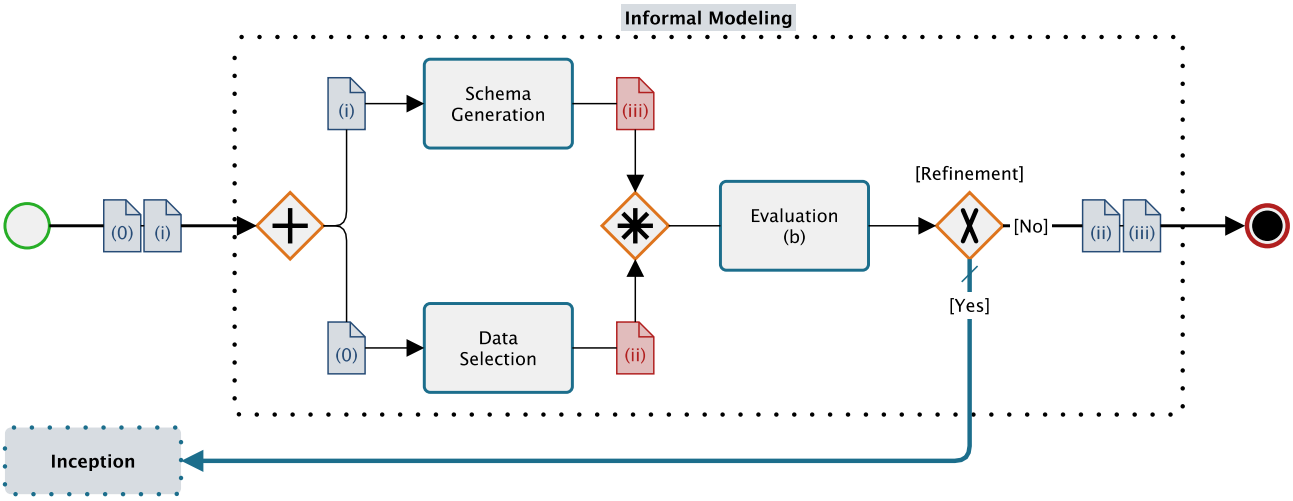


Figure 19: Informal Modeling Diagram

The Informal Modeling is the third phase of the iTelos Methodology containing the definition of the informal model which describes the main entity types related to the problem and the filtering of the preliminary data sets produced in the Inception phase. The Knowledge Engineer, during this phase, is interested in the schema level macro-activity of the Schema Generation while the Data Scientist is interested in the data level macro-activity of the Data Selection. Later on, following the phase iterations, they will be combining their outputs for refining the previously produced outputs. The outputs of each level will be evaluated and will be either iterated for refinement (or denied for not following correctly the various inputs given by the Inception).

Label	Description
0	Preliminary data sets and informal metadata
i	Competency Queries with Data Objects definition
ii	Project Data sets and metadata
iii	Informal L4 Schema (EER)

5.2 Schema Generation

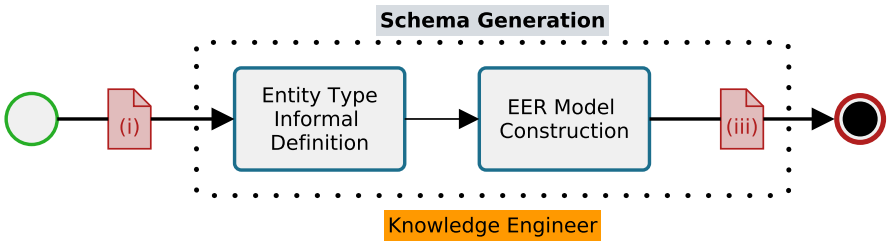


Figure 20: Schema Generation Diagram

In the Schema Generation the main activities being executed are the following:

- Entity Type Informal Definition
- EER Model Construction

In this first schema level macro-activity, the Knowledge Engineer generates the L4 informal Schema model of the problem, starting from the Competency Queries and data objects definitions coming from the previous phase. This model include the definitions of all the entity type, and relative attributes, needed to achieve the project purpose, as well as the relations among those kinds of entity types.

5.2.1 Entity Type Informal Definition

In the Entity Type Informal Definition the Knowledge Engineer is tasked with defining all the possible kinds of entities (called in the following Entity Type or EType) being used in the problem and to find the attributes and properties that describe each entity.

An Entity Type can be of three types, following the three typologies of data already mentioned in the previous phase:

- Core Entity Type
- Contextual Entity Type
- Common Entity Type

Each EType, is formed by several EType Attributes and Properties. The attributes are the elements that define the structure of the EType (i.e. for the EType *Person* we can have the attributes *Name*, *Surname* and *Date of birth*). A property of an EType is composed by an Entity Attribute and the possible relation with another EType through that attribute. It might happen that, while defining a Core EType some attributes related to it might be of a Common or Contextual level (i.e. Given a Car, as Core entity, there might be some common level properties, such as its production date or as a contextual level property, its mileage), in this case there is a relation between

ETypes. In other words, one attribute in the EType takes as value an instance of another EType, creating so a relation among them. These two elements of the EType defines respectively the EType schema and the "position" of the EType, respect to the others, in the overall knowledge schema, which will be defined informally at the end of this phase.

Below a brief description of the three kind of EType is provided, regards the attributes they might include.

- **Core Entity**: A Core EType is defined from the Knowledge Engineer as a specific EType that includes particular attributes important to the CQs and queries given as input from the previous phase. The Core EType will be the core part of the L4 schema, they represent the classes for the most important data instances (Entity) defined in the Inception phase. To provide an example of this kind of EType:
  Car is a Core EType for a transportation problem, so the most important attributes relative to it are engine type, size, seats available and so on. Regard the relations we can have the Driver attribute which is defined as an EType value of type Person, that could be a second Core Entity.

- **Common Entity**: The Common ETypes define the classes for those entities that are used to represent common aspect of the world to represent, such as space and time aspects. These ETypes include attributes which help to represent those common aspect. To provide an example of this kind of EType:
  Address is a Common EType that can have attributes like *City*, *Street* and *Number*. This EType expresses a location concept used in several domains, it can be related to a the *Person* Core EType in the previous example.

- **Contextual Entity**: The Contextual ETypes define the classes for those entities that are used to represent specific aspects of the problem to solve. This ETypes together with their attributes, are defined to add specific peaces of information about the domain considered in the project. For Example:
  Dosage con be a Contextual EType for medications within medical domain. It can include attributes like *times per day* and/or *quantity per day*.

5.2.2 EER Model Construction

In the EER Model Construction the Knowledge Engineer is tasked with taking the ETypes defined beforehand and to create the first informal L4 model using an EER Model. Here the Knowledge Engineer is tasked with producing a detailed Entity-Relation model based on the ETypes and following the relations defined within their attributes. This model is constructed using an EER Model, this makes it easier for the Knowledge Engineer to edit and transform the model as needed and by making it easier to visualize the entire problem through a relationship model. The EER model created will be used, in the next phase as a base to build the formal L4 schema (SKG).

5.3 Data Selection

In the Data Selection the main activities being executed are the following:

- Data sets Filtering
- Metadata (DCAT) Specification

This macro-activity, relative to the data level of the iTelos Methodology, and performed by the Data Scientist, aims to provide a final selection of the datasets needed in the project, later on they will be manipulated to be compliant

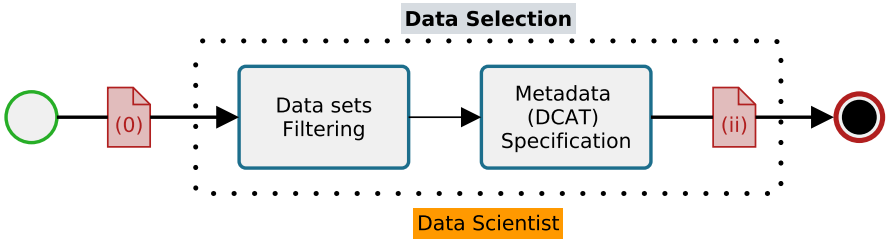


Figure 21: Data Selection Diagram

with the knowledge specification, but in the current activity the dataset selection process is finalized. Together with the data, also the metadata are specified better, eventually collecting new information, and standardized using DCAT. As in the previous phase, both the schema and data level activities are structured and work within an iterative process, where the output of an iteration is the input of the next one.

5.3.1 Data sets Filtering

In this data level sub-activity the preliminary datasets extracted in the Inception phase are filtered through various ETL procedures performed by the Data Scientist. The objective is to finalize the dataset selection in order to have all the datasets needed for the project, to reach a data status where no other information has to be collected anymore (at most manipulated). This task, which operates considering the parallel informal knowledge definition, includes the last dataset extraction procedure and/or the deletion of those data recognized as useless for the project purposes.

5.3.2 Metadata (DCAT) Specification

In the Metadata (DCAT) Specification the Data Scientist is tasked with improve the metadata collection, adding new information regarding the operations performed on the data in the previous sub-activity, and also adopting a standard why to categorize all the metadata collected, which will be one of the main outputs of the project, The standard adopted is DCAT, that allows to handle metadata in structural way easy to consult. Within the DCAT object, the Data Scientist has to add some mandatory metadata like *Provenance*, *Creation Date*, *Author*, *Owner* and *Source* and any other kind of important metadata that might be interested in the problem, for each dataset finally selected int he previous sub-activity.

5.4 Evaluation (b)

Main aspects for Inception evaluation:
- alignment between project informal knowledge collected and datasets collected.

In the Evaluation (b) the objective is to evaluate the output of the Informal Modeling phase.
The output of this phase is made of an informal model, in form of an EER diagram.
Such a model are schema level abstractions from the data sets.

Suppose the output of the phase is a model M, with a set of classes and properties.

$Cov(C_{c/p}, M_{c/p})$ verifies the extend of overlapping of the CQ and the model M selected with respect to the CQs.

$Flx(C_{c/p}, M_{c/p})$ verifies the extend for which the model covers the CQs selected with respect to the model M.

$Cov(D_{c/p}, M_{c/p})$ verifies the extend to which the data sets and the model M selected are with respect to the model M.

$Flx(D_{c/p}, M_{c/p})$ verifies the extend to which the model M covers the data sets with respect to the model M.

Although the model M selected is mainly abstracted from D and is constructed for the purpose of answering C.

5.5 Phase Iterations

In the Informal Modeling phase the minimum number of iterations required for the production of high quality output is equal to, or more than, four iterations. Also in this phase the iterative process is scheduled on the three different data typologies previously mentioned, Common, Core and contextual. The output of each iteration is used to improve the work done in the following. Thanks to the informal schema definition generated in the first iteration, the Data Selection activity have more information to work in the second iterations, and in the same way for the third one and subsequent.

5.5.1 Iteration Zero

In the first iteration the main output of the schema level activity is given by the identification and definition of the Common informal ETypes in the problem, given the CQs, and general data objects defined in the previous phase.

In the first iteration, at data level, the Data Scientist has to finalize the selection of the datasets in which she/he will extract the different types of data based on the informal schema definition.

5.5.2 Iteration One

In the second iteration the main output of the schema level is the identification and definition of the Core informal ETypes.

In the data level the previous datasets selected are checked for satisfying the Common ETypes defined in the previous schema level activity iteration, in this way the datasets with their entities and attribute values are aligned to the Common knowledge defined.

5.5.3 Iteration Two

In the third iteration the main output of the schema level is the identification and definition of the Contextual informal ETypes.

In the data level the previous datasets selected are checked for satisfying the Core ETypes defined in the previous schema level activity iteration, in this way the datasets with their entities and attribute values are aligned to the Core knowledge defined.

5.5.4 Iteration Three

In the fourth iteration the main output of the schema level is an overall check on eventually missing informal ETypes or EType attributes definitions.

In the data level the previous datasets selected are checked for satisfying the Contextual ETypes defined in the previous schema level activity iteration, in this way the datasets with their entities and attribute values are aligned to the Contextual knowledge defined.

As fourth output given by the current phase the documents created are of higher quality and closer to achieve the problem solution. Future iterations might be done for aligning even more the datasets.

5.6 Deliverables

In the Informal Modelling phase there aren't new deliverable produced, but some crucial improvements are done on the documents already existing.

- **iTelos project report** : The main project report is updated with the informal definition of the ETypes. Note that in this first definition of the main object classes, the classification on the three possible kind of entities has to be specified (Common, Core and Contextual). Moreover a specific section in the document is filled with the description of the EER model developed as output of the current activity.
- **Metadata sheet and description** : The metadata documents are updated extending both the set of metadata collected and the relative description in the description document.

5.7 Languages & Standards

In the Informal modeling phase the Knowledge Engineer has to create the EER model for the informal ETypes and their relations. Due to that the usage of the EER standard is required to produce high quality output in the end of the phase. The Data Scientist in the current phase, needs to know the DCAT standard to collect and correctly structure the metadata of the project.

5.8 Tools

In the Informal Modeling there are different tools that can be used in relation to the activities levels. In the Schema level, the documentation reporting the informal definition of the ETypes can be made with the use of overleaf integrating the documents coming from the previous phases. In a second moment, for the construction of the EER Model, yED is used as a modelling tool. In the Data level, the Data Selection can be supported through the Jupyter environment already set up in the Inception phase, for the datasets filtering activity and to extract the metadata that will be stored within the Data Catalog.

5.8.1 yEd Usage

A specific palette has been developed for the usage of yED and as such it will help formalize in a structured way the production of the EER diagram.

The palette can be found here : **Insert Link**
After downloading it, install it onto yED via the Palette manager.
(Open yED → Click View and Palette Manager → Import Selection and Select the downloaded .graphml file)

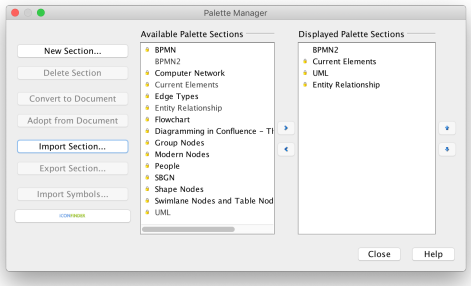


Figure 22: yED Palette Manager

After installing the palette, it is easy to follow and understand how to use the palette.
Inside it you can find there are 3 entities and a bunch of ER styled arrows.

The arrows can be customized while using the palette, to make it show the relationship between the entities in both way.
(You'll have to edit the Source and Target Arrow type in the properties view)

Here is an example of the yED Palette used for constructing the simple problem of the transportation product:

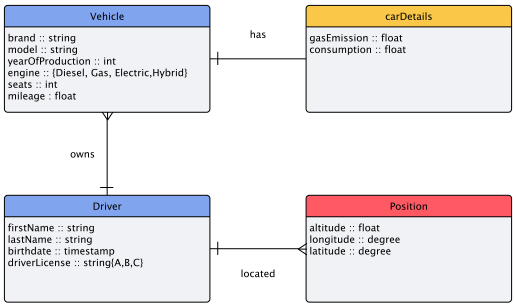


Figure 23: yED Palette usage example

5.9 Deliverables

In Informal Modeling the main output being created is divided into the Schema and Data level.
In the Schema level we have two types of documents relative to the productions made by the Knowledge Engineer.

- Document with written explanation of the EType defined;
- Informal L4 model as an EER Model.

In the Data level we have the filtered data sets with their corresponding metadata and enriched metadata made by the Data Scientist.

5.10 Examples

The particular example that we will be looking at is referring to the student project named Space Domain [7].

In this third phase we will discuss examples regarding the definition of the ETypes and the construction of the EER model using yED.
Another important part of the phase is the usage of RapidMiner for the cleaning and transformation activies on the data sets selected beforehand.

5.10.1 Examples of Schema Generation

In the Schema Generation macro-activity the group is tasked with defining the ETypes and the construction of the EER informal model with the usage of yED.
It's important to describe the types of entities being defined and to which type of EType they will be categorized as.
In this case, as shown in Figure 24 the work is related to the identification and grouping of the entities and properties relative to the correct EType.

These identifications are based in regards to the Figure 18 which categorizes the types and properties of the CQs selected in the Inception phase.
In this case the Core, Common and Auxiliary (correctly renamed as Contextual in the iTelos Methodology) are the three types of entities identified in the EER model.
After explaining the decisions made in regards to the model constructed, an explanation must be made for the three particular types, describing here which type of entities and possible properties are part of the Core, Common or Auxiliary ETypes.

In specific, as an example, in Figure 25 we can see the developed EER informal model constructed in the 'Space Domain' project with the usage of yED.

The relations between the entities in the EER model are of multiple type and can be used to relate entities into cardinality sets, this feature will be lost when exporting the informal model into a formal model with Protégé.

Starting from the generalized queries represented in the table above, this section provides the diagram that represents the EER model. We have used the tool **yEd** to represent the model. The column of the types represents the entities. The column of the properties represents the attributes.
The EER model is very similar to the ER model, the main distinction consists in the subdivision of the entity in three different types. The colour of the model can help the user to distinguish between these types. The types, ordered by decreasing importance, are:

1. core entity types (plus relations and attributes), the blue ones
2. auxiliary entity types (plus relations and attributes), the red ones
3. common entity types (plus relations and attributes), the yellow ones

In order to better represent the entities in our model, we have decided to add another type that is the violet one. It represents the union of two pre-existent types that are common entity types and structured attributes. We have used the new type to define the Address entity. This representation is an adaptation to our model of the original one.
Furthermore, there are two types of attributes, the simple ones and the structured ones. Attributes are defined as structured, if a possible value is a tuple of values, in other words, there are attributes that includes a series of attributes.

Figure 24: Space Domain Entity Definition

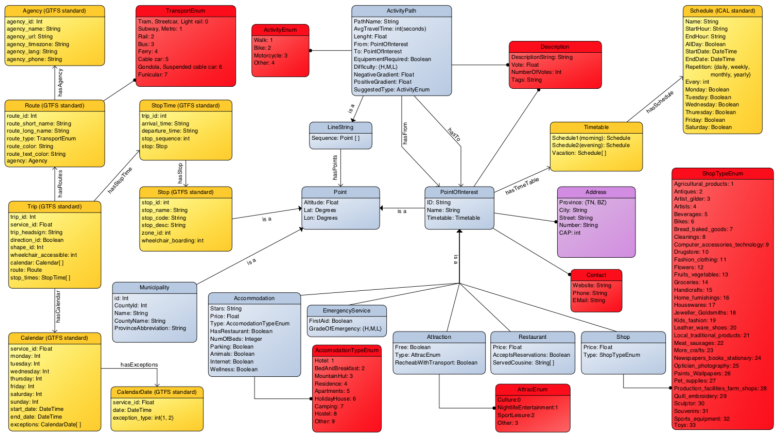


Figure 25: Space Domain EER Model

5.10.2 Examples of Data Selection

In the Data Selection macro-activity the group is tasked with extracting and executing various ETL activites over the data sets selected in the Inception phase.

In this case, as seen in Figure 26, RapidMiner is used as a tool for extracting, cleaning and transforming the data sets.

Figure 26: Space Domain RapidMiner usage

After the data sets are extracted and cleaned, they will be mapped over the EER model constructed in the Schema-level, after some particular amount of iterations have been completed.
In this case a mapping of the types found in the data sets are mapped to the EER ETypes as seen in Figure 27.

EER Entity	Type(-Subtype) list
PointOfInterest	"Service providers", "Traffic and Transport (without Bus stops, Taxis, and Railways subtypes)", "Public institutions (without Hospitals subtype)"
Attraction	"Culture and sights", "Sports and leisure", "Nightlife and entertainment"
Shop	"Shops", "Craft"
EmergencyService	"Public institutions-Hospitals", "Doctors, Pharmacies"

Figure 27: Space Domain Data Sets Mapping

6 Formal Modeling

6.1 Top level view

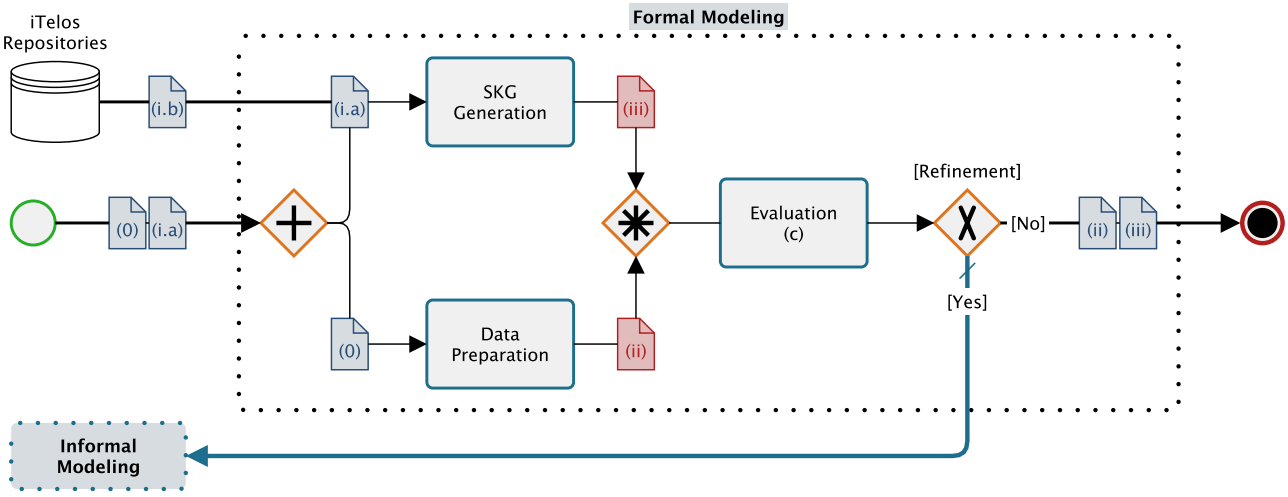


Figure 28: Formal Modeling Diagram

The Formal Modeling is the fourth phase of the iTelos Methodology containing the construction of the formal L4 Schema Knowledge Graph and a first annotation regarding the L1 and L2 lexical and semantic concepts of the SKG, another important step in the phase is the transformation of the previously generated data sets into the EML format.
The main roles interested in each macro-activity is related to their level.

Label	Description
0	Data sets metadata and informal metadata
i.a	L4 informal schema
i.b	Teleology
ii	Aligned/Formatted data sets
iii	L4 SKG

The Knowledge Engineer is interested in the Schema-level macro-activities of the SKG Generation while the Data Scientist is interested in the Data-level macro-activities of the Data Preparation.
Both roles describe and perform activities related to the advancement of the project.

Later on, with iterations they will be combining their outputs for refining the previously produced outputs.
The outputs of each level will be evaluated and will be either iterated for refinement (or denied for not following correctly the informal L4 schema previously made).

If the outputs have followed the phase correctly and executed a certain amount of iterations as described per the Evaluation section, they will be accepted as formalized documents to be used in the next phase.

6.2 SKG Generation

In the SKG Generation the main activities being executed are the following:

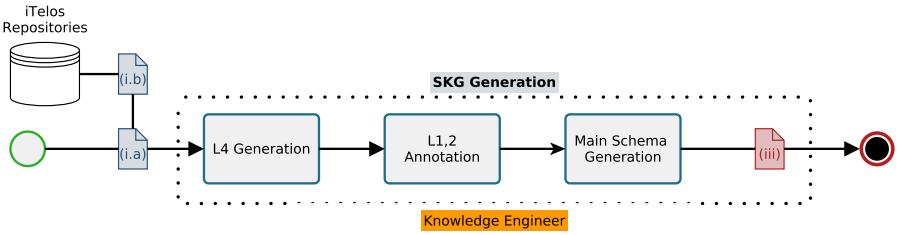


Figure 29: SKG Generation Diagram

- L4 Generation
- L1,2 Annotation
- Main Schema Generation

This macro-activity is found in the Schema level of the iTelos Methodology and the Knowledge Engineer is tasked with constructing the formal L4 schema knowledge graph which will be annotated through the L1 and L2 concept definitions and finally exported as the complete SKG which will be used as formal knowledge in the next section.

The main important objective is to relate the teleology found in the iTelos Repositories to help support and facilitate the construction of the L4 formal model, the Knowledge Engineer will later on relate the concepts as in the teleology to the entities generated for the problem and slowly categorize and export the new L1 and L2 concepts which will need to be fixed.

A Teleology is constituted of three particular levels of knowledge:

- L1 regarding the Concept Space;
  It is interested in the linguistic interpretation of the verbs and nouns interesting to the problem.
  In here words are categorized into speech categories and are described according to their relation with other words in the system.
- L2 regarding the Lexical-Semantic Space;
  It is interested in the definition of the group of synsets organized according to the linguistic principles.
  L2 is usually denoted as a detailed annotation over the L1 lexicons and are described in multiple languages.
- L4 regarding the representation of the SKG for a specific problem;
  This schema is generated based on a data-driven requirement and uses standards already implemented (thanks to the teleologies).
  A data structure graph-like is used where the nodes define the entities and their attributes and the edges define the relations between the nodes (such as entity relation attributes).
  In this level the entities and relations are constrained by properties and logics which define theorems to proof the existence of such node or edge.

6.2.1 L4 Generation

In the L4 Generation the Knowledge Engineer aims to obtain a new formal version of the knowledge schema (L4) starting from the Teleology, selected from the iTelos Repositories, and the L4 informal schema produced as output

of the previous phase. The new L4 schema will be formally defined using RDF/OWL. To achieve this objective, the informal L4 schema is compared to the Teleology in order to identify in the latter, those nodes which are already able to represent the informal ETypes defined in the informal L4 schema. In the case the Teleology doesn't contain enough knowledge to cover all the informal ETypes (in other words there are no nodes in the Teleology that can be used to represent some informal ETypes), new nodes and edges are added in the new L4 formal schema (the new knowledge schema elements will be added, in the future, within the Teleology, if necessary, to produce a more defined Teleology).

6.2.2 L1-2 Annotation

In the L1-2 Annotation the Knowledge Engineer has to identify within the Teleology those lexicon-semantic elements of L1 and L2, which are concepts in the UKC, that can be used to define ETypes, ETypes attributes and relations for the data which have to be integrated. The main purpose of the Annotation activity is to contribute into finding most, if not all, the possible concepts in the UKC that can help to describe the entities and relations found in the finalized ontology which will be used as formal knowledge definition. During this phase, the Knowledge Engineer, could discover within the dataset analyzed, some concepts that have not yet been added in the UKC (so not available in the Teleology), in this case the new concepts will be added as formal knowledge and they will go to integrate the existing Teleology. Moreover some different representations of already existing concepts can be found within the data analyzed, so in this case those will be annotated as synonyms of the respective concepts found in the UKC. This sub-activity can modify the RDF/OWL file produced in the previous one, adding the L1,2 annotation, and/or produce an *Import File* for those new concepts/senses which have to be imported in order to integrate the UKC. That import file is an Excel file that lists, using a specific structure the knowledge elements to import.

6.2.3 Main Schema Generation

Once the previous two sub-activities are completed, the outputs generated are the RDF/OWL file and if new knowledge elements have to be imported, the Excel import file. In the current sub-activity these two objects are imported together, using the appropriate tools, in the Data Integration Platform where they will be merged in a single, and more precise, SKG that will be exported and provided as the output of this sub-activity.

6.3 Data Preparation

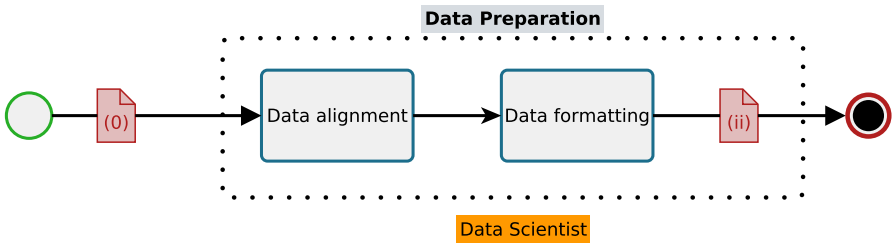


Figure 30: Data Preparation Diagram

In the Data Preparation the main activities being executed are the following:

- Data alignment

- Data formatting

The Data Preparation macro-activity aims to handle the data identified and extracted during the previous phases (Inception and Informal modelling), and perform shaping operations in order to provide them to the next phase in the bast form as possible. These operations regard two different aspects, the data alignment respect the EType used to represent the data, and the data formatting following the correct data types, the two sub-activities manage these actions respectively.

6.3.1 Data alignment

The Data alignment activity compares the data extracted from the Informal Modelling phase, with the informal definition of the ETypes, trying to understand if the data are correctly shaped to be represented by those ETypes. In case differences appear between the data form and the ETypes structure, this activity aims to reorganize the data with the objective to reduce as much as possible the gap between the data layer and knowledge layer.

6.3.2 Data formatting

The Data formatting activity has the same objective of the previous activity, but focused on the values of the data instead of their structure. It checks and models the data values to ensure that those values respect the data types for the data that they represent. The information about the correct data types to adopt comes from the knowledge defined for the respective data. This activity aims to identify and, if necessary provide solutions to those data values that are no type compliant.

6.4 Evaluation (c)

Main aspects for Inception evaluation:
- [schema level] alignment between project formal knowledge defined for the data to integrate and the reference ontology used to solve the project's problem.
- [data level] quality measures on datasets filtered.

In the Evaluation (c) the objective is to evaluate the output of the Formal Modeling phase.
The output of this phase is made of a formal model which takes the format of an ontology, it is straight forward to check the model with the criterias for ontologies.
There are three levels of criteria to which the ontology is evaluated against.

1. Schema Level
   Schema level evaluation checks the logical perspective of the conceptual modeling, which covers the following three dimensions.

   - Consistency
     It's the logical satisfactory of the knowledge system, covers various topics.
     - No cycles must be modeled in the class hierarchy (isA relations).
       Circulatory models are typically errors.
     - Polysemous terms are not welcomed, especially when the same name is used in different specific nodes, such as as a class and as a property.
       (i.e contact can be an entity and a property)

- Different classes in the domain of a property should not conflict with each other.
  It is necessary to model the domain/range of a property as complex classes (i.e union or intersection of multiple classes).
- Tangles are not welcomed in hierarchies.
  Tangles are potential causes of miss-classification of entities.
  As such a classical structure of a taxonomy is a tree-like structure where the most general class *owl:thing* is the root of all the classes and the tree grows more and more in detail up until the leaf classes.
- No different/multiple names for the same object, if such objects exists it is possible with owl to use the *owl:sameAs* tag.
  (i.e surname and family name are equivalent)

- Accuracy
  Verifies the correctness of the modeling.
  - Incorrect relationships usually lie in semantical differences.
  - Over-specialization occurs in a leaf node when the entity does not have an individual in the knowledge base.
  - Misc./Other is not welcomed as it does not provide clear intentions to the knowledge base.
  - Chains of hierarchy are not welcomed, constructing a chain/thread like structure for the taxonomy.
- Completeness
  Is to verify the model as an organic whole.
  - Isolated entities are not welcomed, no matter its construction (entity, individual, user defined data-type, or property)
  - Properties should be defined with a well-defined domain and range.

2. Linguistic Level
   Linguistic level evaluation aims at the usage perspective of the model, which covers the compliance and understand- ability dimensions.
   - Compliance
     TBD : Ask Rui
   - Understandability
     It's based on the user-friendly perspective of the model.
     - Clumping properties is not welcomed if there doesn't exist a more general property.
       (i.e street/road, city, state should be organized under the sub property of *address*.

3. Metadata Level
   In the Metadata level the evaluation is concentrated on the information of the model construction, such as the *creator*, *version*, *construction date*, *purpose*.

6.5 Phase Iterations

In the Formal Modeling phase the bare minimum iterations required for the production of an high quality output is expected after four iterations. The iterative process in this phase is scheduled in order to assign one main type

of data (Core, Common and Contextual) to each iteration, plus one final iteration for a final check. Each iterations aim to improve step by step, the ontology that will be used in the next phase as formal knowledge of the data to be integrated, for the schema level, and for the data level, a set of data as much as possible aligned to the ontology and well shaped in terms of values and data types.

6.5.1 Iteration Zero

In the first iteration the main output of the Schema level is the definition of formal ETypes for the Core entities, starting from the Informal definition provided by the previous phase. As part of the formal definition the EType are annotated with the lexicon-semantic element of L1 and L2.

During the first iteration the Data level is less considered due to the fact that the formal definition of the ETypes is needed to perform the data preparation sub-activities. However a first general iteration of Data Alignment and Data Formatting activities, can be done on the base of ETypes informal definitions coming from the previous phase.

6.5.2 Iteration One

In the second iteration, in the Schema level, the ontology that includes the formal definition of the Core ETypes, is improved adding the formal definition of the Common ETypes annotated with the lexicon-semantic element L1 and L2.

In the Data level the Core entity data are handled, following the two sub-activities of the Data Preparation macro-activity, on the base of the Core formal ETypes defined at Schema level in the previous iteration.

6.5.3 Iteration Two

In the third iteration, in the Schema level, the ontology that includes the formal definition of the Core and Common ETypes, is improved adding the formal definition of the missing Contextual ETypes annotated with the lexicon-semantic element L1 and L2.

In the Data level the Common entity data are handled, following the two sub-activities of the Data Preparation macro-activity, on the base of the Common formal ETypes defined at Schema level in the previous iteration.

6.5.4 Iteration Three

In the fourth iteration, in the Schema level, the ontology includes the formal definition of the Core, Common an Contextual ETypes for the data that have to be integrated, for this reason this iteration is used as a final check in order to identify missing knowledge definitions.

In the Data level the missing Contextual entity data are handled, following the two sub-activities of the Data Preparation macro-activity, on the base of the Contextual formal ETypes defined at Schema level in the previous iteration..

After this minimum number of iteration the result of this phase is produced and can be moved on to the next phase. More in specific the ontology containing the knowledge formal definition for the dataset and the dataset aligned to the knowledge and well shaped.

6.6 Languages & Standards

During the Formal modeling the Knowledge Engineer define formally the EType and their relations producing the L4 schema annotated with the L1-2 elements coming from the UKC. The L4 schema is defined using the RDF-OWL format. For this reason the knowledge of that standard is required in the current phase. Moreover the Knowledge Engineer has to be able to import new L1-2 knowledge elements, if they are missing, in order to correctly define the data used in the project. To do that she/he has to know the Knowdive internal standard used to define the Excel Import file (L1 Import File standard).

6.7 Tools

In the Formal Modeling phase the Knowledge Engineer have to define the Schema Knowledge Graph (SKG). In order to do that the tool used are:

- **Protégé**: This tool is used to define the SKG using the OWL format. The OWL file ,created with this tool, can be used as knowledge input for the KarmaLinker tool in the net Data Integration phase.
- **UKC**: The usage of the UKC is provided to the Knowledge Engineer, in order to annotate the L4 schema, previously generated with the already existing L1-2 elements.
- **Knowledge Importer**: Thanks to this tool, the Knowledge Engineer can import new L1-2 elements (new concepts and relations) in the UKC, and so let them available to annotate the SKG that has to be produced.
- **OWL importer**: This tool allows the import of the whole SKG within the Data Integration platform, in order to be able to improve the knowledge layer of an eventually existing SKG within the platform itself.
- **OWL exporter**: This tool allows to export, from the Data Integration platform the OWL file which defines the whole SKG contained in the platform. This OWL file can be used as knowledge input for the KarmaLinker tool in the net Data Integration phase.

At data level the Data Scientist can use the Jupyter environment, previously set up, to align and format the data, providing so well formed dataset to the next final phase.

6.8 Deliverables

In the Formal Modelling phase there aren't new deliverable produced, but some crucial improvements are done on the documents already existing.

- **iTelos project report** : The main project report is updated with the formal definition of the ETypes, and most important, the definition of the L4 Schema produced in output in the current phase, together with the description of the Schema annotation using the L1-2 elements.
- **Metadata sheet and description** : The metadata documents are updated extending both the set of metadata collected and the relative description in the description document. In the current phase, these documents are update with the L1-2 and L4 metadata information.

6.9 Examples

The particular example that we will be looking at is referring to the student project named Space Domain [7].

In this fourth phase we will discuss examples regarding the construction of Ontology and it's grounding with the Teleology selected from Schema.org and the lexical annotations defined for the entities and relations.

6.9.1 Examples of SKG Generation

In the SKG Generation macro-activity the group is tasked with constructing the Ontology and to compare and ground it with the Teleology selected.

In the 'Space Domain' project the group annotated the Ontology created and explained its main differences from the EER constructed in the Informal Modeling phase.
As seen in Figure 31 the Ontology created followed various types of entities aggregations and in regards to the rules needed for the construction of an Ontology specific to the three dimensions (Consistency, Accuracy and Completeness)

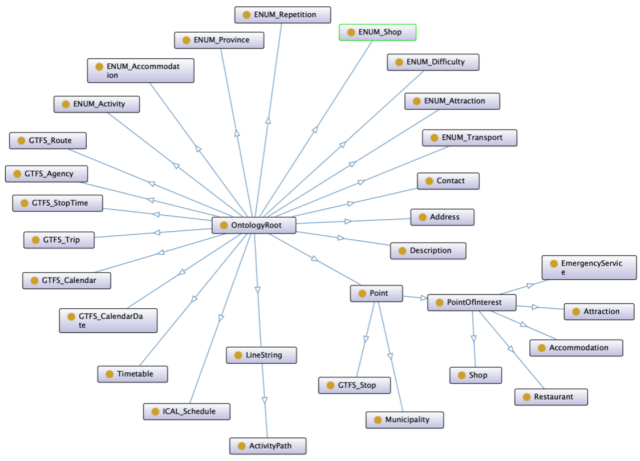


Figure 31: Space Domain Ontology Creation

After the group constructed its specific Ontology, it took some specific types of Teleologies on which it compares the Ontology created and improves it, structuring it over the previously constructed L4 schemas.

In this case, the group used the GeoVocab and GeoCoordinates schema to solidify and produce a reusable and improved schema.

The main objective of the top-level grounding is to identify the new root of the L4 schema and to adjust the Ontology created in comparison to the Teleology entities.
An important step in addressing heterogeneity in regards to the entity types is to describe and annotate the entities found in the Ontology through an annotation tool, such as WordNet.

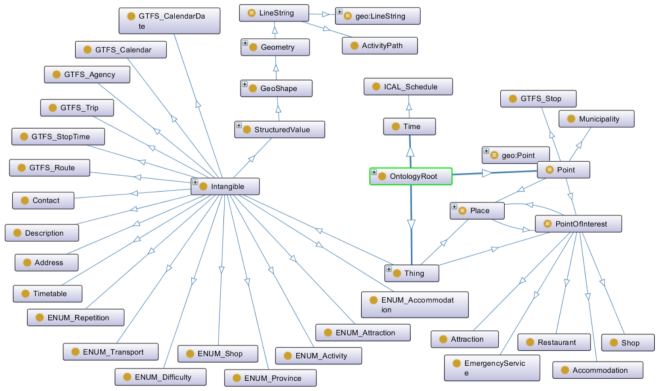


Figure 32: Space Domain Teleology Grounding

The group project in this case used WordNet and annotated the entities on Protégé.

As it can be seen in Figure 33 the annotation is based on four particular values separated by a semicolon:

1. Ontology Entity name;
2. WordNet ID mapped over Protégé;
3. WordNet precise item with respecitve category;
4. Synset of the term.

- ***OntologyRoot***; {08524579}; ⟨*noun.location*⟩ [15] S: (n); beginning, origin, root, rootage, source (the place where something begins, where it springs into being)
- ***Address***; {08508037}; ⟨*noun.location*⟩ [15] S: (n); address (the place where a person or organization can be found or communicated with)
- ***Contact***; {07293990} ;⟨*noun.communication*⟩ [10] S: (n) ;contact, touch (a communicative interaction)
- ***Description***; {06737512}; ⟨*noun.communication*⟩ [10] S: (n) ;description, verbal description (a statement that represents something in words)
- ***GTFS_Agency***; {08073958}; ⟨*noun.group*⟩ [14] S: (n); agency (a business or organization that provides a particular service, especially the mediation of transactions between two parties)

Figure 33: Space Domain Lexical Annotations

6.9.2 Examples of Data Preparation

Empty

7 Data Integration

7.1 Top level view

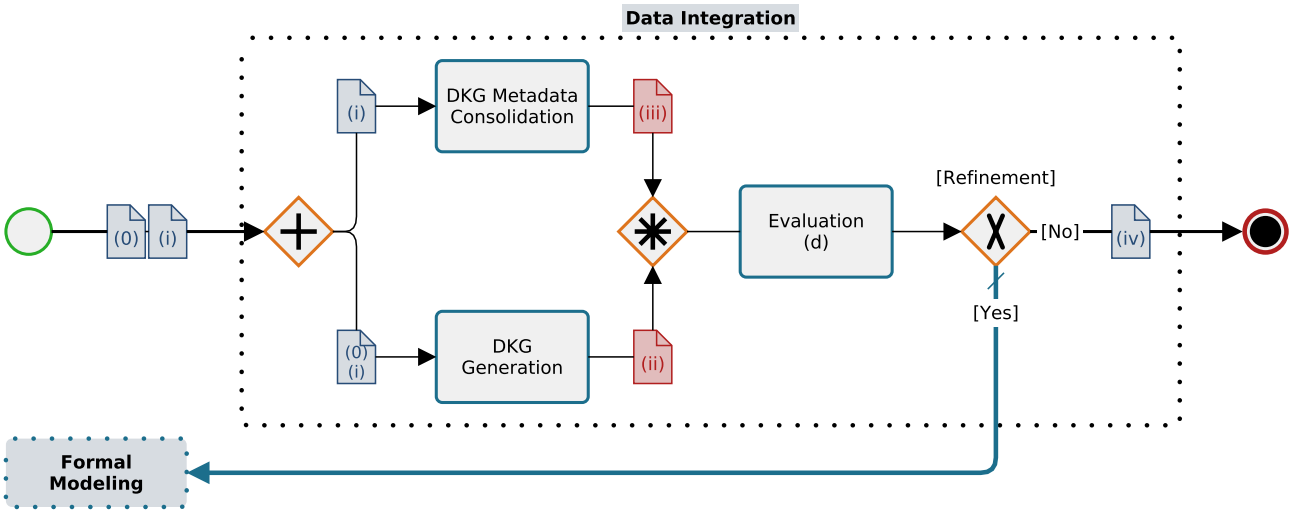


Figure 34: Data Integration Diagram

The Data Integration is the fifth and final phase of the iTelos Methodology containing the finalization of the process with the construction of the Knowledge Graph and the relative Codebook for the metadata collection as well as the relative Project Report as main deliverable. The main roles interested in each macro-activity are the two related to their level, so the Knowledge Engineer and the Data Scientist.

Label	Description
0	L4 formal schema (SKG)
i	L5 entity data
ii	L5 DKG data
iii	L1,2 L4, L5 metadata plus Project Report and Slides
iv	Metadata + DKG, Codebook, Project Report, Slides, Project purpose

The Knowledge Engineer is interested in the Schema-level macro-activities of the DKG Metadata Consolidation with the purpose to finally define the set of metadata within a Codebook. While the Data Scientist is interested in the Data-level macro-activities called DKG generation, which aims to generate the final instance of the DKG.
The outputs of each level will be evaluated and will be either iterated for refinement (or denied for not following correctly the formal L4 schema taken in input). If the outputs results correct after a certain amount of iterations as described per the Evaluation section, they will be accepted and a formal deliverable document will be created as project report.

7.2 DKG Metadata Consolidation

In the DKG Metadata Consolidation the main activities being executed are the following:

- DKG metadata collection
- Codebook Documentation

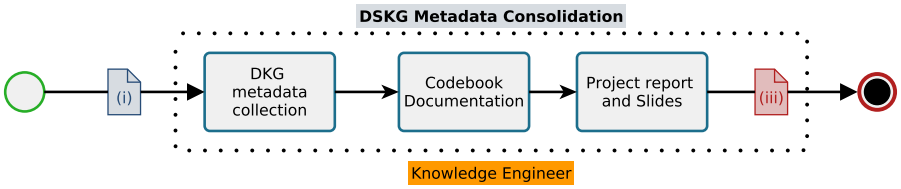


Figure 35: DKG Metadata Consolidation Diagram

- Project report and Slides

This macro-activity, performed by the Knowledge Engineer, aims to identify and collect all the metadata needed to guarantee the quality of data. The metadata collected in this macro-activity regards both the Schema and Data level. Through the iterations of the current phase the collection of the metadata is performed on the always more well formed DKG instance (SKG plus entity data), which will be the output of the whole phase. The final output of this activity is a documentation object called Codebook which contains a structured definition of all the metadata collected during the iterations. Moreover in the last sub-activity the Knowledge Engineer has to produce a final version of the Project report document as well as the Project Slides.

7.2.1 DKG metadata collection

The metadata collection sub-activity takes in input both the formal L4 schema (SKG), defined in the previous phase, and the L5 data, with the scope to identify and/or collect useful metadata regarding the Schema elements and the data extracted. Within the two input, there are several kinds of metadata, listed below a list of main metadata areas are reported:

- Regards the Schema level, there are metadata defined at level of:
  - ETypes
  - ETypes attributes
  - concepts
- Regards the Data level, there are metadata defined at level of:
  - Dataset
  - Entity
  - Entity attribute value

For each level mentioned above there are several metadata the can be found within the L5 data and/or the SKG received as input of the current phase. Moreover there are some kinds of metadata associated to the DKG instance, that is under creation in this phase, which are defined in during the Data level activity of *DKG Generation* defined below. These metadata are collected iteration after iterations during the execution of the current phase. The role of the Knowledge Engineer has to identify all this kinds of metadata through the different iterations, passing them in input to the next sub-activity, described in the following section.

7.2.2 Codebook Documentation

In the Codebook Documentation the Knowledge Engineer is tasked with collecting together and describe all the metadata identified in the previous sub-activity, into a structured document called Codebook, describe below. This document makes it possible to view and read in a humanly fashion the metadata related to the whole DKG.

The Codebook is a document which describes the layout of the data and of the data selected in respect to their quality through the metadata information. The Codebook document generated will follow an index splitted for Schema and Data level.

For the Schema level the Codebook is internally divided describing each EType formally defined in The Formal Modeling phase, and for each EType, description and metadata information are provided for all the relative EType attributes. In this way the document provide a description of the formal schema defined for the data, as well as the quality level of those through the metadata information associated to.

For the Data level the document aims to report and give a description for all the metadata associated to the general datasets handled as well as for the single values within them. Due to that, the structure of this sub-section of the document is divided dataset by dataset, and within each dataset, also in this case, following the different kinds of entities identified in the datasets, but focused on the data values instead of schema elements like in the previous section of the Codebook.

For each EType reported in the Codebook, an explanation of the EType attributes is reported providing at least some mandatory details reported below:

- Attribute data type;
- Attribute range/values;
- Attribute description;

7.2.3 Project report and Slides

In the last sub-activity the Knowledge Engineer has to finalize the documentation which has to be produced has part of the final output of the whole Methodology. More in specific she/he has to produce a final version of the Project report document that describes in details the different phases executed and the results obtained. Moreover the final version of the Project Slides is completed in the current sub-activity. The Slides aims to summarize and present the work done.

7.3 DKG Generation

In the DKG Generation the sub-activities being executed are the following:

- Data Mapping
- EML data import

This macro-activity performed by the Data Scientist produces, after the needed iterations, the final output of the data integration process, the DKG instance. During this final part of the process the Data Scientist maps the

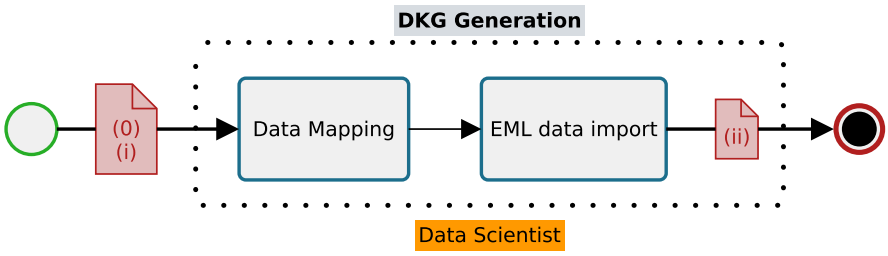


Figure 36: DKG Generation Diagram

"well formed" entity data, received as input from the previous phase with the Schema Knowledge Graph (L4 SKG) formally defined, creating so a more precise as possible, association between Data level and Schema level. After that the data correctly associated with the knowledge, can be imported and maintained within a data integration platform (provided as base element of the iTelos Methodology) [– ADD REFERENCE –] and being available for any external service which wants to exploit them.

7.3.1 Data Mapping

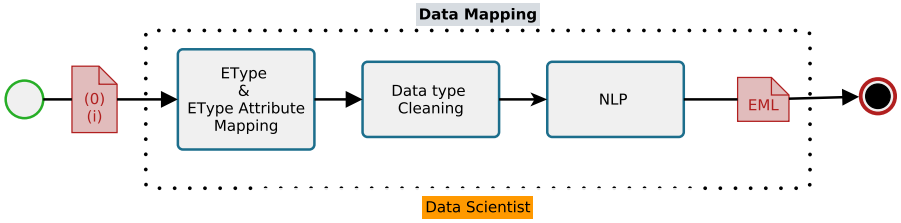


Figure 37: Data Mapping Diagram

The Data mapping sub-activity is executed through the following internal activities:

- EType & EType Attribute Mapping
- Data Type Cleaning
- NLP

The Data Mapping sub-activity plays a crucial role in the overall data integration process. In this step the Data Scientist, using the KarmaLinker tool (described in the tools section of the current phase), define the mapping between the data entities identified within the datasets and the ETypes defined formally in the SKG. In other words, during this sub-activity the Data Scientist define the data, retrieved from external sources and expressed with different format and data structure, using the knowledge formalized in the previous phases. There are three different kinds of operations performed on the data during the Data Mapping step, which are identified in the three internal activities, described in the paragraphs below.

During this mapping activity some minor changes on the data values are allowed, in order to be compliant with the knowledge defined in the SKG. The Data Scientist can be helped out with some important features of KarmLinker

tool. All the mapping operations as well as the minor modifications on the data values are stored, following the execution order, within what is called *Mapping Mode* or also *Mapping Recipe*. This model can be re-applied on dataset having the same structure of the one used to create the model itself, in order to automatically reproduce the mapping operations on different data values. This helps a lot the work of the Data Scientist introducing an automatic step in the process which can be used for potentially huge amount of data.

The output of this sub-activity is a specific file in a specific format called EML. This file can be considered as a first "data-specific" instance of DKG, in the sense that it contains both the Schema and Data layer related to data handled using KarmaLinker, but the resources within the EML file are not yet part of a complete Knowledge Graph. In the next sub-activity the final integration in the KG will be performed.

7.3.1.1 EType & EType Attribute Mapping

The first type of operation performed on the data, using the KarmaLinker tool, regard the mapping between the data entity and the EType defined in the SKG. This kind of mapping is performed for all the EType attributes define for each EType. The data scientist has to associate the structure of the data to the structure defined in the formal schema, at EType level (for each data Entity) and EType Attribute level (for each fields that compose a data Entity in the dataset).

7.3.1.2 Data Type Cleaning

The second type of operations regard some minor modifications on the data values that the KarmaLinker tool allows to perform on the data values. If some data arrive at this step without the correct form, in terms of format (i.e. the dates) or allowed values, the data scientist can clean those values in order to obtain a correct integration.

7.3.1.3 NLP

The third possible type of operation allow to run NLP procedures on the data values in order to extract concepts and other information "hidden" in the dataset as natural text. In this way the data scientist can exploit better the knowledge level provided by the UKC, linking the data attributes with concepts handled in the Knowledge base.

— TO INTEGRATE.. —

7.3.2 EML data import

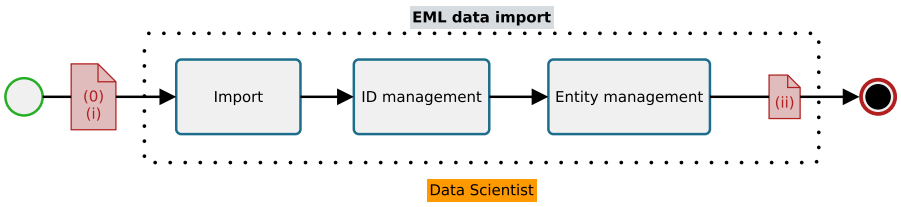


Figure 38: EML Data Import Diagram

The EML data import sub-activity is executed through the following internal activities:

- Import
- ID management
- Entity management

The last sub-activity aims to complete the Data Integration process, through the full integration of the "data-specific" EML file within a complete KG stored and maintained in the Data Integration Platform [– ADD REFERENCE –]. EML data import activity is completed running in line some specific tools, provided by the Data Integration Platform, which are able to perform automatically the internal activities described below, taking in input the EML file produced in the previous sub-activity.

7.3.2.1 Import

The first internal activity represents the starting point of the EML data import process. The *Data Importer* tool is run by the Data Scientist with the ML file as input. This first tool aims to concretely store the L5 data (both knowledge and entity data) contained in the EML file) in the Data Integration platform in charge of manage the data. Within the EML file the *Data Importer* tool recognizes the IDs of the concepts used to define the EType and the ETypes attributes, and thanks to that is able to store correctly the input information inside the Data Integration platform. It is important to note that the platform can contain some already imported data and knowledge, for this reason the Importer has to well integrate the new input with the resources already existing.

7.3.2.2 ID management

The second automatic internal activity has the purpose to manage the identification aspects of the data integration regard the entities imported in the previous internal activity. As already mentioned, this is a fully automatic procedure started by the Data Integration platform after new imports. More in specific this internal activity aims to analyze the entity identifiers attributes as well as those attribute compositions able to uniquely identify an entity, in order to recognize if those entities are the same as others already present in the entity base or they are new entities, or new representations of already existing entities, so carrying new information to store. Once discovered the "nature" of the new entities to import, this internal activity proceeds to merge the entities that are recognized the same, so there is no new information to store, in this way there will be no entities duplicated after the import procedure.

7.3.2.3 Entity management

The third internal automatic activity is in charge of manage the new entities imported in order to align the information which they carry on with the existing entities in the data integration platform. More in specific the task performed in this internal step is to manage different representation of entities (due to the import of new data) both at values level and schema level.

The final sub-activity, composed by the three just described above, can be defined as "platform-activity" due to the fact that is automatically executed by the Data Integration platform. For more details regards the platform components which perform this procedures see [ADD REFERENCES TO KHUB AND SWEB ARCH. DOCS]. If the EML data import is completed successfully the data, and relative knowledge, are well integrated and ready to be exported and exploited by external services.

7.4 Evaluation (d)

Main aspects for Inception evaluation:
- [schema level] quality of metadata collected.
- [data level] quality of data at knowledge graph level. How much the data exploitable in the KG are useful to solve the project's problem.

In the Evaluation (d) the objective is to evaluate the output of the Data Integration phase.
The output of this phase is made of the KG generated, the evaluation is taken place on checking the validity and coherency to the inputs and valid in answering the CQs with the integrated data sets.

7.5 Phase Iterations

In the Data Integration phase the minimum number of iterations required for the production of high quality outputs, regarding both the DKG generated and the Codebook, is expected in regards to the number of datasets being mapped, coming from the previous phase. In order to guarantee the correct data integration, due to the fact that the data (so the entities within the datasets) have dependencies among the different datasets, the iterations have to respect a specific order regarding the kind of data to import. In specific the Data Scientist together with the Knowledge Engineer have to handle first the Common data, because they are the data which create more dependencies on the others. After those there will be the import of Core data, as main entities in the domain they have a strong impact in term of dependencies (not so strong as the Common data), and in the end the Contextual data can be imported as well.

At Data Level the iterations on the DKG Generation activity aim to map the entity data, dataset by dataset. Due to the fact that the data information about the domain considered are often widespread on different datasets, the iterations are not completely separated because mapping a certain dataset the Data Scientist has to consider the dependencies or references, on other different datasets which will be mapped later or have been already mapped.

In the DKG Metadata Consolidation activity, at Schema Level, the Knowledge Engineer follows the iterations collecting the metadata datasets by dataset and so updating the Codebook document. Moreover during the DKG Generation activity, executed in parallel, some new metadata are generated regarding the import of new data on the Data Integration platform, such as the import time, who (Data Scientist) perform the import, for this reason, iteration by iteration the Knowledge Engineer has to collect those metadata coming from the previous iteration in the Data Level.

7.6 Languages & Standards

In the Data Integration phase the Data Scientist produces, through the KarmaLinker tool, the import file which will be imported in the Data Integration platform, in order to create the whole Knowledge Graph (DKG). That import file follows the EML language.

7.7 Tools

In the last phase the most important tools used by the Data Scientist is KarmaLinker. This tool is used to map the datasets, cleaned and well, with the Schema Knowledge Graph (SKG), both coming from the Formal Modelling

phase. The output of KarmaLinker can be an RDF file defining the Data Knowledge Graph (DKG) which can be represented using the GraphDB tool, and/or an EML file that is used to integrate the DKG instance produced with the project dataset, in the Data Integration platform which can contain an extended Knowledge Graph that will be improved with the new data. In order to import the new data in the Data Integration platform the Data Scientist has to use the L5 Importer tool provided by the platform itself.

7.8 Deliverables

In the Data Integration phase as the last phase of the methodology, the finalization of the deliverables is required, plus the creation of the presentation for the final demo of the project.

- **iTelos project report** : The main project report is finalized describing the last steps of the process where the data are concretely integrated in the Knowledge Graph. Moreover a specific section of the project report is filled with the description of the final demo, to present as final examination of the KDI academic course.
- **Metadata sheet and description** : The metadata documents are updated extending both the set of metadata collected in the last phase of the process, and the relative description in the description document.
- **Project slides** : A presentation is created in the current phase, in order to describe the project and the outcomes achieved.

7.9 Examples

The particular example that we will be looking at is referring to the student project named Space Domain [7].

In this fifth phase we will discuss examples regarding the mapping of the EML data sets into the SKG with the usage of KarmaLinker.

7.9.1 Examples of DSKG Metadata Consolidation

Empty

7.9.2 Examples of DKG Generation

In the DKG Generation macro-actvity the main activity described by the group was the mapping of the particular data sets into the L4 model.

In this case KarmaLinker was used for mapping efficiently and easily the data sets.
KarmaLinker helped the group into mapping the particular data sets fastly and, in the case of the yellow blocks, transform the data sets in some specific ways.

In this case, for examples in the point_uri, the mapping is formed by two different data sets, the Latitude and Longitude.

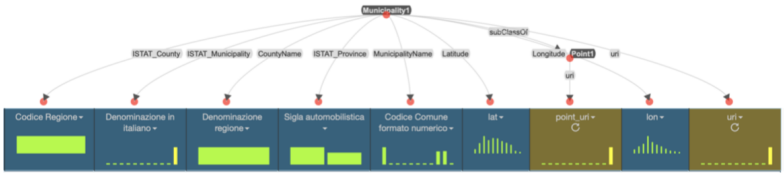


Figure 39: Space Domain KarmaLinker Mapping

8 Conclusion

The complete work of the iTelos Methodology helped rework the structure of the methods and finalizing the categorization of the phases.

During the beginning of the work the Methodology was split into four main phases, after a long discussion the phases got increased to five, introducing the Scope Definition.

The case study used in the document was helpful with managing and checking the consistency of the procedures between the students projects and the future methods and tools planned to be used in the upcoming academic years.

8.1 Future works

In terms of future works the iTelos Methodology helps in establishing the foothold for two different types of works.

1. New case studies;
2. An extension of the iTelos Methodology.

With new case studies we propose the approach of conducting new projects for the course ‘Knowledge Data Integration‘ in relation to the documentation written and to extract from such project new case studies and observations to fix and upgrade the document as is.

An extension of the iTelos Methodology is the work related to the congregation of the Methodology into a single black box called Adaptation, to this a new methodology will be inserted next to it, called the iTelos Evolution.

The future work of extending the iTelos Methodology is aimed at the upgrade of the Knowledge Graph generated from the Adaptation such as given new data to import, modify or for possibly new schema features to implement into the Knowledge Graph.

References

[1] Khuyagbaatar Batsuren, Gabor Bella, and Fausto Giunchiglia, *Cognet: A large-scale cognate database.*, Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics, 2019, https://drive.google.com/open?id=1VWuUoNsKvII2vriLgPkzHwKcFas-yufY, pp. 3136–3145.

[2] Gabor Bella, Fausto Giunchiglia, and Fiona McNeill, *Language and domain aware lightweight ontology matching.*, Journal of Web Semantics **43** (2017), 1–17, https://drive.google.com/open?id=16pIchMG4npjaFla8ajYtgho0g7yIxZyb.

[3] Gabor Bella, Fiona McNeill, David Leoni, Francisco Jose Quesada Real, and Fausto Giunchiglia, *Diversicon: Pluggable lexical domain knowledge*, Journal on Data Semantics **8** (2019), no. 4, 219–234.

[4] Gabor Bella, Alessio Zamboni, and Fausto Giunchiglia, *Domain-based sense disambiguation in multilingual structured data*, DIVERSITY Workshop at ECAI 2016, 2016, https://drive.google.com/open?id=1g_O7PMVtrMrlan0dUDH5YcInF3uUVzi7.

[5] Gábor Bella, Liz Elliot, Subhashis Das, Stephen Pavis, Ettore Turra, David Robertson, and Fausto Giunchiglia, *Cross-border medical research using multi-layered and distributed knowledge*, Proceedings of Prestigious Applications of Intelligent Systems (PAIS@ECAI) (Santiago de Compostela, Spain), 2020, https://drive.google.com/open?id=1Ia5TOHgNVxWDl-uiuwrpnbgwU8tLSlSE.

[6] Gábor Bella, Fiona McNeill, Rody Gorman, Caoimhín Ó Donnaíle, Kirsty MacDonald, Yamini Chandrashekar, Abed Alhakim Freihat, and Fausto Giunchiglia, *A major wordnet for a minority language: Scottish gaelic*, Proceedings of the 12th Language Resources and Evaluation Conference (Marseille, France), 2020, https://drive.google.com/open?id=1aZwUm9LqzeOrjY67WIDNvzAyzmhVXluI.

[7] Alessandro Cacco, Martina Battisti, Bertiana Balliu, Sara Callaioli, Daniele Isoni, Stefano Leonardi, and Michela Lorandi, *Kdi course project - space domain*, https://github.com/alecacco/KDI-Project-2019-2020-Geospace, 2019.

[8] Davide Cappellaro, Andrei Conti, Alessandro and Diaconu, Valentina Sofia Pigato, Andrea Hassanli, Seyyed Arya and Iossa, and Andrea Mattè, *Kdi course project - facilities and events*, https://github.com/vale17accidentidellastoria/KDI-Project-FacilityDomain, 2019.

[9] Yamini Chandrashekar, *A system for large-scale multilingual lexicon management*, Master's thesis, DISI, University of Trento, Trento, Italy, 2019, https://drive.google.com/open?id=16cx51ofy-77gIzo2yHSjLQobhaNz8W7r.

[10] Danish Asghar Cheema, *Nlp for data integration*, Master's thesis, DISI, University of Trento, Trento, Italy, 2019, https://drive.google.com/open?id=1cY_xOhShnlkx8xqs5Klwa54gQQ2NMRLm.

[11] Subhashis Das and Fausto Giunchiglia, *Geoetypes: Harmonizing diversity in geospatial data*, Proceedings of the 15th International Conference on Ontologies, DataBases, and Applications of Semantics (ODBASE), 2016, https://drive.google.com/open?id=1iU9ZSPVVxa_RGayMyZRW8vRsWiHdgoBp.

[12] Subhashis Das, Sajan Raj Ojha, and Fausto Giunchiglia, *Atom: Ontology aware transportation model*, Proceedings in 11th International Conference on Semantic Computing (ICSC 2017), 2017, https://drive.google.com/open?id=1_vAKuKwXis1n9mA0Iv1urfm8OGYGapIj.

[13] Lisa Ehrlinger and Wolfram Wöß, *Towards a definition of knowledge graphs.*, SEMANTiCS (Posters, Demos, SuCCESS), 2016.

[14] Abed Freihat, Gabor Bella, Mubarak H., and Fausto Giunchiglia, *A single-model approach for arabic segmentation, pos-tagging and named entity recognition*, second International Conference on Natural Language and Speech Processing ICNLSP (Algiers, Algier), April 25-26 2018, https://drive.google.com/open?id=1xg2qsQxHaN5Idb6Rd5DoOQqI248uyCaL.

[15] Mattia Fumagalli, Gabor Bella, and Fausto Giunchiglia, *Towards understanding classification and identification*, Pacific Rim International Conference on Artificial Intelligence (2019), 71–84, https://drive.google.com/open?id=1MT86A1dPBjPL7_SjoseurCg3QNTSlR01.

[16] Mattia Fumagalli and Fausto Giunchiglia, *On knowledge diversity*, Proceedings of the 2019 Joint Ontology Workshops (JOWO), WOMoCoE 2518 (CEUR-WS: 2019), 2019, https://drive.google.com/open?id=1OhQEmh7cCJg_efRA8_OuWWMXiNrHILNu.

[17] F. Giunchiglia, Dutta, and V. Maltese, *From knowledge organization to knowledge representation*, Knowledge Organization **41** (2014), no. 1, https://drive.google.com/open?id=1pmSLjDd1WvHK378srOuci5QqzI7NLFN-.

[18] F. Giunchiglia, V. Maltese, and B. Dutta, *Domains and context: first steps towards managing diversity in knowledge*, Journal of Web Semantics, special issue on Reasoning with Context in the Semantic Web (2012), 53–63, https://drive.google.com/open?id=12UmzozfFOu4re3WRuoOCxKXYjvZUku2A.

[19] Fausto Giunchiglia, Khuyagbaatar Batsuren, and Gabor Bella, *Understanding and exploiting language diversity*, Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence (IJCAI-17), 2017, https://drive.google.com/open?id=11_tM4DLRKXHiDbExrL6BgCrEu-c7F0A5, pp. 4009–4017.

[20] Fausto Giunchiglia, Khuyagbaatar Batsuren, and Abed Freihat, *One world - seven thousand languages*, 19th International Conference on Computational Linguistics and Intelligent Text Processing (Hanoi, Vietnam), 2018, https://drive.google.com/open?id=1-wEtnzLWRzHMgVKIaSpv4m7ENm9SuFdE.

[21] Fausto Giunchiglia and Mattia Fumagalli, *Concepts as (recognition) abilities*, IOS Press in electronic format with permanent open access (2016), https://drive.google.com/open?id=1Gb9fZj87GzPR2SDPK2cv7Odea85El87Y.

[22] ———, *Teleologies: Objects, actions and functions*, International Conference on Conceptual Modeling, ICCM, 2017, https://drive.google.com/open?id=1nIKyvINUrvsIsxTBu_GfkK0lDQI96kh6, pp. 520–534.

[23] Fausto Giunchiglia, Sajan Raj Ojha, and Subhashis Das, *Semui: A knowledge driven visualization of diversified data*, Proceedings in 11th International Conference on Semantic Computing (ICSC 2017), 2017, https://drive.google.com/open?id=1Eqjfsbo2rwjg8k3CNGrZVpjO8JC4rlot.

[24] Aidan Hogan, Eva Blomqvist, Michael Cochez, Claudia d'Amato, Gerard de Melo, Claudio Gutierrez, José Emilio Labra Gayo, Sabrina Kirrane, Sebastian Neumaier, Axel Polleres, Roberto Navigli, Axel-Cyrille Ngonga Ngomo, Sabbir M. Rashid, Anisa Rula, Lukas Schmelzeisen, Juan Sequeda, Steffen Staab, and Antoine Zimmermann, *Knowledge graphs*, 2020.

[25] Mercedes Huertas-Migueláñez, Natascia Leonardi, and Fausto Giunchiglia, *Building a lexico-semantic resource collaboratively*, Proceedings in 18th EURALEX International Congress, 17-21 July 2018, (Lubljana, Slovenia), 2018, https://drive.google.com/open?id=17QghC4weRNptQrwXLzaTUeGPlsbBUNYS.

[26] Bogdan Kostić, Evidence Monday, Andrea Montibeller, Federico Calabrese, Giacomo Callegari, and Fabio Molignoni, *Kdi course project - public transportation in trentino*, https://github.com/fabiomolignoni/KDI-transport, 2019.

[27] Massimiliano Luca, *Extended entity relationship model and entity model*, Bachelor's thesis, DISI, University of Trento, Trento, Italy, 2017, https://drive.google.com/open?id=1D14yyttaOmLDhpkYT9FsMzVuhfxfeBTw.

[28] Eddy Maddalena, Luis-Daniel Ibáñez, Elena Simperl, Mattia Zeni, Enrico Bignotti, Fausto Giunchiglia, Claus Stadler, Patrick Westphal, Luís PF Garcia, and Jens Lehmann, *Qrowd: Because big data integration is humanly possible*, Proceedings of the Project Showcase Track of KDD2018 (2018), https://drive.google.com/open?id=1X5UpIzi49VFZcULGsZoI_qmohMTheaqD.

[29] V. Maltese, F. Giunchiglia, A. Sarangi, and S. Margonar, *efrbr: An entity model for frbr*, ISKO UK conference, 2015, https://drive.google.com/open?id=1NjzKzJodUNyIAYchOpqWb9-1BGsDFEe6.

[30] Vincenzo Maltese and Fausto Giunchiglia, *Foundations of digital universities*, Cataloging & Classification Quarterly (2016), 1–25, https://drive.google.com/open?id=1pR0lUz9c8fVEKwjmML8MusHPVgcaUY_k.

[31] Nandu Chandran Nair, Rajendran Sankara Velayuthan, and Khuyagbaatar Batsuren, *Aligning the IndoWordNet with the Princeton WordNet*, Proceedings of the 3rd International Conference on Natural Language and Speech Processing (ICNLSP) (Trento, Italy), Association for Computational Linguistics, 12–13 September 2019, https://drive.google.com/open?id=1ZXjJd27IW0eSz3SK_SBMa0k-pvxQbCRI, pp. 9–16.

[32] Sajan Raj Ojha, *Diversity aware visualization*, Ph.D. thesis, DISI, University of Trento, Trento, Italy, 2018, https://drive.google.com/open?id=1TM8h62nSYZK3xUsOAsvAmSeX4OsjmQMT.

[33] Sajan Raj Ojha, Mladjan Jovanovic, and Fausto Giunchiglia, *Entity-centric visualization of open data*, Human-Computer Interaction – INTERACT **9298** (2015), no. Lecture Notes in computer Science, 149–166, https://drive.google.com/open?id=1iKaEKPhaAJ4VX9WqSekGUoW6UlcBNm2e.

[34] A.R.D Prasad, Fausto Giunchiglia, and Devika P. Madalli, *Dera: From document centric to entity centric knowledge modelling*, International UDC Seminar - Faceted Classification Today (London), 2017, https://drive.google.com/open?id=1FL6o9K-l6X7HrvxqAHL6KBNVxI3u7Nol.

[35] David Robertson, Fausto Giunchiglia, Stephen Pavis, Ettore Turra, Gabor Bella, Elizabeth Elliot, Andrew Morris, Malcolm Atkinson, Gordon Mcallister, Areti Manataki, Petros Papapanagiotou, and Mark Parsons, *Healthcare data safe havens: Towards a logical architecture and experiment automation*, The Journal of Engineering, Special Issue: Engineering for Health (2016), https://drive.google.com/open?id=1McUzoa1DxXZBYlhPvR6PeMrGf-dp7z2u.

[36] Amit Singhal, *Introducing the knowledge graph: things, not strings*, 2012, https://blog.google/products/search/introducing-knowledge-graph-things-not/.

[37] Chatterjee Usashi, Fausto Giunchiglia, Madalli Devika P., and Vincenzo Maltese, *Modeling recipes for online search*, The 15th International Conference on Ontologies, DataBases, and Applications of Semantics, 2016, https://drive.google.com/open?id=13XgFwS__SOXcUjclvd-epqHrYfi_vVrK.