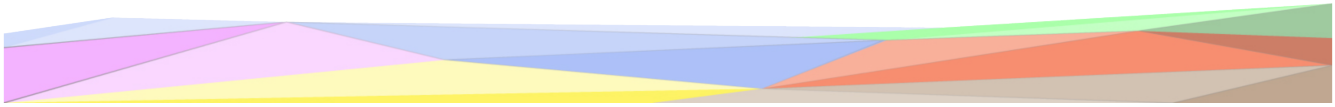
Department of Information and Computer Science

National University of Mongolia

KGE-MN 2025 - Knowledge Graph for Course Schedule Reuse at NUM

Document Data:	Reference Persons:
December 26, 2025	Baigali, Myagmarsuren, Oyu-Erdene

Ulaanbaatar, Mongolia



Index:

Revision History:

Revision	Date	Author	Description of Changes
0.1	November 20, 2025	Baigali	Document created
0.2	November 25, 2025	Oyu-Erdene	Wrote the project report introduction.
0.3	November 25, 2025	Myagmarsuren	Created the ER model and define DOI
0.4	November 25, 2025	Baigali	Define problem and developed the PFsheet
0.5	November 30, 2025	Myagmarsuren	Gathering data and combined
0.6	December 2, 2025	Baigali	Wrote the project report and helped data gathering
0.7	December 2, 2025	Oyu-Erdene	Wrote the project report
0.8	December 11, 2025	Oyu-Erdene	Refined the language resources and developed the sections on concept identification and UKC alignment.
0.9	December 11, 2025	Baigali, Myagmarsuren	Refined the filtered language resources and wrote the dataset filtering section along with the phase outcomes.
0.10	December 17, 2025	Oyu-Erdene, Baigali	Defined the Knowledge Teleontology and wrote the project report sections related to concept identification and UKC alignment
0.11	December 17, 2025	Myagmarsuren	Prepared the aligned datasets and wrote the project report section on dataset alignment
0.12	December 22, 2025	Myagmarsuren	Updated the Knowledge Definition section according to instructor feedback.
0.13	December 22, 2025	Myagmarsuren	Finalized the Knowledge Graph construction and completed the Entity Definition phase.
0.14	December 22, 2025	Myagmarsuren, Oyu-Erdene, Baigali	Completed the iTelos-based Evaluation phase of the Knowledge Graph.

1 Introduction

This document is a detailed Project Report on building an integrated Knowledge Graph (Integration KG) within the context of the National University of Mongolia, focusing specifically on academic operations—particularly course scheduling—using the iTelos Knowledge Graph Engineering (KGE) methodology. The project is conducted within the scope of ten selected academic programs, reusing existing Knowledge Graphs—such as the Courses KG, Course–Curriculum Relations KG, and Course Schedule KG—and integrating them with newly developed KGs, including the Student KG, Planned Course KG, and Curriculum–Student KG. Through this integration, the project extends existing course schedule data with curriculum-required and student-planned courses for ten programs, enabling the detection of scheduling conflicts, curriculum overlaps, and other academic inconsistencies.

2 Domain of Interest (DoI)

The Domain of Interest refers to the area of knowledge or field of study in which the project is situated. In this project, the domain focuses on the academic operations of the National University of Mongolia (NUM), specifically the processes related to course scheduling and curriculum–course alignment across a selected set of ten academic programs. The domain encompasses knowledge about courses, course schedules, curriculum structures, and student course planning within these programs. Together, these components support the detection of scheduling conflicts and overlaps among planned, selected, and curriculum-required courses during the Fall 2025 semester.

2.1 Geographical boundaries

Geographical boundaries define the spatial constraints of the project. In this project, the geographical scope is limited to the National University of Mongolia and its internal academic structures. All collected and analyzed course, curriculum, and student planning data pertain exclusively to ten selected programs within NUM.

2.2 Temporal boundaries

Temporal boundaries define the time-related constraints of the project. The temporal scope is restricted to the Fall 2025 semester. Course schedules, curriculum requirements, and student

course plans for this period serve as the basis for identifying scheduling conflicts and curriculum-related inconsistencies within the selected programs.

2.3 Domain boundaries

Domain boundaries specify the conceptual and operational limits of the project. The project focuses solely on course scheduling and curriculum alignment for the ten selected academic programs. It considers planned courses, selected (enrolled) courses, and curriculum-required but not-yet-completed courses of students within these programs. The scope explicitly excludes administrative processes, university-wide non-academic domains, and academic programs outside the chosen ten.

3 Project Development

This section describes, at top level, how the project's purpose will be satisfied. More in details the current section aims at describing how the dta production process is performed.

3.1 Data Production

The description of which (quality) data needs to be created to satisfy the project purpose. This sub-section highlights the role of the data producer. The sub-section aims at describing how the data producer creates the data required to satisfy the project's purpose.

4 Initial Resources

This section describes the two kind of resources considered by a projects, by filling the two sub-sections here below.

Knowledge resources:

- **Course (course-kg.ttl)** Contains canonical course definitions at NUM, including course codes, labels, names (in multiple languages), credits, academic terms, and department affiliations.
- **Curriculum–Course (course-curriculum-kg.ttl)** Describes curriculum structures, including which courses belong to which curriculum, their categories (core/elective), and recommended semesters.

- **Course Schedule (course-schedule-kg.ttl)** Provides real course scheduling instances, including day of week, timeslots, instructors, classroom locations, and academic term/year.

Data source:

- **Student Dataset** raw student identifiers, names, enrolment years, and program information extracted from NUM's student information system. This dataset is essential for creating the Student entity and linking students to their curriculum.
- **Planned Course Dataset** collected from a prototype advising interface where students record the courses they plan to take for the Fall 2025 semester. Contains student IDs, selected course IDs, timestamps. These serve as the foundation for the PlannedCourse contextual entity used for schedule conflict detection.
- **Selected Course Dataset** represents historically chosen courses by students during enrollment periods, specifically including selections made for the Fall 2025 semester. The dataset includes student IDs and course schedule IDs. These entries are crucial for modeling student behavior patterns and validating progression toward curriculum requirements.

5 Purpose Formalization

Our project aims to integrate university course, curriculum, and scheduling data into a unified knowledge graph. This graph will allow students and academic staff to easily identify overlaps between planned courses, selected (enrolled) courses, and curriculum-required courses. By modeling relationships between course schedules, time slots, instructors, and curriculum structures, the graph will support conflict-free planning and improve the accuracy of academic decision-making.
Scenarios definition

- **Scenario 1**: A first-year student is preparing a study plan for the next semester. They want to know whether their planned courses conflict with each other in terms of time or prerequisites.
- **Scenario 2**: A student has already registered for several courses. Before the semester begins, the student wants to check if their selected courses overlap in timetable or violate curriculum rules.
- **Scenario 3**: An academic advisor needs to verify whether a student's curriculum-required courses have any timing conflicts with the student's other registered courses.
- **Scenario 4**: A student interests course that entered their free time.

- **Scenario 5**: A staff needs to know whether a student's curriculum-required courses overlap

Personas

- **Persona 1 Freshman Naraa**: Naraa is a first-year student who is creating her course plan for the next semester. Since she has not taken many courses before, she wants to ensure that the courses she selects do not overlap in time and that she meets all prerequisite requirements. Naraa mainly deals with planned (not yet registered) courses.
- **Persona 2 Senior Student Bilguun**: Bilguun is a senior student who has already enrolled in several major and elective courses. He wants a fast way to verify whether his registered courses overlap in the timetable. Unlike Naraa, Bilguun already has a fixed schedule and is checking for conflicts in an existing course list.
- **Persona 3 Academic Advisor Ariunaa**: Ariunaa works at the registrar's office and helps students choose courses that match their curriculum. She needs a fast tool to check conflicts between curriculum-required courses and a student's existing schedule.
- **Persona 4 Saraa**: Saraa needs to know course schedule that start 11:00 am in Tuesday.
- **Persona 5 Staff Naraa**: Naraa needs to know that what program's major courses's schedule overlap.

5.1 ER Model Overview

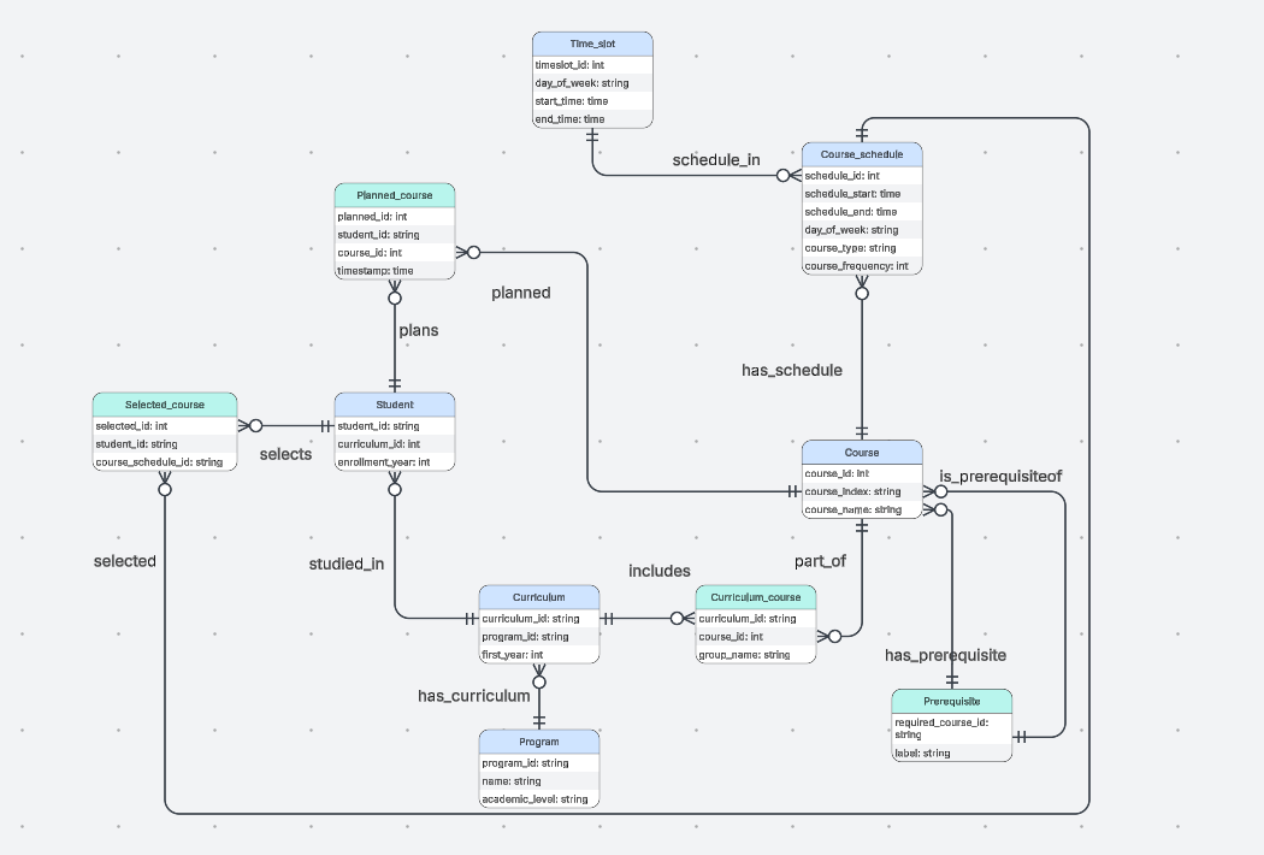


Figure 1: ER model aligning Personas and Competency Questions

Competency Questions (CQs)

- **CQ 1**: What time conflicts exist between the courses Naraa plans to take for the next semester?
- **CQ 2**: Which timetable overlaps occur among the courses that Bilguun has already enrolled in?
- **CQ 3**: Do any curriculum-required courses clash with the student's currently registered schedule?

6 Information Gathering

This section reports the activities performed during Phase 2 of the iTelos methodology. The goal of this phase was to identify, collect, clean, and standardize the datasets that will serve as the initial resources for building the Knowledge Graph (KG) on course–curriculum–schedule

relationships at NUM. All activities were conducted following the iTelos data preparation guidelines.

6.1 Data Sources

The datasets used in this project originate from two complementary sources.

(a) LiveDataNUM – Reuse Dataset (JSON)

We downloaded publicly available, FAIR-compliant educational datasets from the LiveDataNUM repository. The following conceptual datasets were reused:

- Courses
- Curriculum–Course Relations
- Course Schedules

These datasets served as an external conceptual baseline for aligning NUM's internal data with standardized educational data structures.

(b) NUM Internal Information Systems – Extracted Data (CSV)

We extracted real academic data from NUM's internal information systems in CSV format. The extracted datasets include:

- `student.csv` – student enrollment information for the current cohort;
- `curriculum.csv` – curriculum structure and program requirements;
- `program.csv` – program information;
- `planned_course.csv` – courses planned by students for upcoming semesters;
- `selected_course.csv` – courses actually selected and confirmed by students.

These datasets form the internal operational data foundation that must be aligned with the reused LiveDataNUM datasets for Knowledge Graph modeling.

6.2 Data Preparation Process

The collected datasets required extensive transformation before they could be used in KG modeling. The preparation process involved the following steps.

6.2.1 Encoding and Format Correction

All CSV files extracted from NUM systems initially contained corrupted Cyrillic characters. To resolve this, we applied:

- Removal of invisible control characters,
- Standardization of separators and line endings.

These corrections ensured that all files were machine-readable and consistent across platforms.

6.2.2 Field Naming Standardization

To ensure uniformity across sources, we defined a naming scheme compatible with iTelos and FAIR principles:

- All field names use `snake_case`;
- Primary keys follow consistent patterns (e.g., `course_id`, `student_id`);

Applying these conventions improved interoperability between datasets originating from different systems.

6.2.3 Structural Alignment Between Reuse and Internal Data

A major challenge was the mismatch between identifier systems in the reuse datasets and the internal NUM datasets. The `Courses.json` file from LiveDataNUM uses hashed/UUID-style course identifiers, whereas the internal NUM systems rely on integer-based `CourseID` values. This structural difference made direct integration impossible.
Although the reuse data contains a `courseIndex` field intended for alignment, it could not be used because multiple different courses shared the same index value. Therefore, `courseIndex` was not a reliable linking key.
To address this, we implemented a custom multi-attribute matching process using:

- instructor name,
- course title,
- course component,
- credit structure,
- day and time information,
- school/department identifiers.

Using this refined matching strategy, we reduced 13,117 reuse course entries to a final set of 230 courses relevant to the ten NUM programs.
During matching, several reuse courses appeared under multiple schools. We resolved this by aligning each course with the specific school structure used in internal NUM data to remove duplicates.
From the `Course Schedules.json` dataset, 6,742 raw entries were filtered down to 6,616 relevant schedule instances. Curriculum identifiers from the `Curriculum--Course Relations.json` dataset were also used to validate and align curriculum structures.
This multi-attribute matching approach provided a more robust alignment layer than the original `courseIndex`-based mapping.

6.3 Issues Encountered

The following issues arose during the data preparation process:

- Inconsistent course identifiers between reuse datasets and internal NUM data,
- Cyrillic character corruption in the initial CSV exports,
- Absence of a direct foreign key linking curriculum lines to offered courses,
- Duplicated curriculum rows caused by multi-program inheritance,
- Courses appearing in multiple programs, requiring additional disambiguation.

6.4 Solutions Applied

To produce a reliable and standardized dataset collection, we applied the following measures:

- Conversion of all raw CSV files to UTF-8 encoding and removal of corrupted characters,
- Harmonization of field names and normalization of table structures,
- Disambiguation and deduplication of program and curriculum entries,
- Standardization of numeric, temporal, and identifier formats across datasets,
- Exporting all cleaned and standardized datasets into `.xlsx` format for uniformity and improved usability.

These transformations produced a coherent and reusable dataset package fully compliant with iTelos principles.

6.5 Final Standardized Datasets

6.5.0.1 Combined Dataset Package The final dataset package submitted for Phase 2 includes the standardized reuse datasets from LiveDataNUM and the cleaned internal NUM datasets. All datasets were exported into `.xlsx` format after preprocessing:

- `course.xlsx`
- `curriculum.xlsx`
- `curriculum_course.xlsx`
- `student.xlsx`
- `planned_course.xlsx`
- `selected_course.xlsx`
- `schedule.xlsx`

All files adhere to uniform encoding, naming conventions, and structural standards, and are ready for use in the subsequent phases (Purpose Definition, Competency Questions, KG schema development).

6.6 Summary

Phase 2 successfully completed the identification, extraction, and standardization of all required datasets. The resulting dataset collection is now fully prepared for integration into the Knowledge Graph development workflow and provides a reliable foundation for Phase 3.

7 Language Definition

This section is dedicated to the description of the Language Definition phase. Like in the previous section, it aims to describe the different sub activities performed by all the team members, as well as the phase outcomes produced.

7.1 Concept Identification

This activity aimed to identify all concepts necessary to represent the academic scheduling and curriculum domain. In this project, the identification process relied on three concrete sources produced in the previous phases. Based on these sources, the following concept groups were identified:

- **Core EType concepts:** Student, Curriculum, Program, Course, Course_schedule, Time_slot, Academic_year, Academic_term, Moment. These represent the main academic objects required for describing course offerings, academic structure, and temporal organization.
- **Linking and activity entities:** Curriculum_course, Prerequisite, Planned_course, Selected_course. These concepts encode many-to-many mappings, prerequisite requirements, and student activity records, all of which are essential for reasoning tasks such as curriculum validation and schedule conflict analysis.
- **Data and auxiliary properties:** course_id, schedule_start, schedule_end, start_time, end_time, day_of_week, building_name, room_number, group_name, and similar attributes extracted directly from the source tables.

These concepts together capture the structural, relational, and operational elements required for schedule reasoning in the Knowledge Graph. During this activity, the concepts were classified into *Core*, *Contextual*, and *Custom* categories based on their contribution to CQ coverage. Concepts that did not support any reasoning tasks or did not appear in the datasets were removed from the language, as documented in Section 7.3.

7.1.1 Difficulties encountered.

The most challenging part of concept identification was aligning the ER model and the heterogeneous datasets, especially where the source data did not explicitly distinguish between Course, Course_schedule, Time_slot, and Moment. Several fields (e.g., day_of_week, course_type, frequency) required interpretation to determine whether they should be modeled as ETypes or merely as data properties. Additionally, identifying whether Academic_year and Academic_term should be modeled as independent concepts or contextual attributes required several iterations.

7.1.2 Aspects that were straightforward.

Core domain concepts such as Student, Course, Curriculum, and Program were easy to identify because they appeared consistently and explicitly in all datasets and were directly referenced in the Competency Questions. Similarly, linking entities such as Curriculum_course and Prerequisite were straightforward since they already existed as relational tables.

7.2 UKC Alignment

The second activity involved aligning the selected concepts with the Universal Knowledge Core (UKC) to ensure semantic consistency and reuse of established ontology definitions. Each

concept was reviewed to determine whether a corresponding UKC entry existed and whether the meaning matched the requirements of the academic scheduling domain.

- **Concepts found in the UKC:** Student, Program, Curriculum, Course, Time_slot, Academic_year, Academic_term. These concepts had clear UKC equivalents, enabling the reuse of established semantic structures and reducing ambiguity in the language.
- **Concepts not available in the UKC:** Course_schedule, Curriculum_course, Planned_course, Selected_course, Prerequisite, Moment. These concepts represent domain-specific structures or operational activities not covered in UKC; therefore, they were formally defined as new concepts and assigned project-specific identifiers (GID-500XX). All glosses were written following UKC conventions to maintain terminological consistency.

Through this alignment, standard concepts benefited from existing ontology definitions, while domain-specific concepts were precisely formalized to fit the project's reasoning needs. This produced a hybrid conceptual model that balances reuse (via UKC) and customization (via project-specific definitions), ensuring coherence, reusability, and clarity in the purpose-specific language used to construct the Knowledge Graph.

7.2.1 Difficulties encountered

The main challenges were determining whether certain temporal concepts (*Moment*, *Time_slot*, and *Course_schedule*) matched existing UKC entries, and deciding how deeply to reuse UKC's event-related hierarchy. Several concepts had partial similarities but did not fully align, which required creating new definitions instead of forced mapping.

7.2.2 Aspects that were straightforward

Aligning Student, Course, Program, and Curriculum was simple, as all have well-established equivalents in UKC and map consistently to schema.org and VIVO. This allowed quick confirmation of their semantic grounding and reduced the need for additional clarification.

7.3 Dataset filtering

In this stage, all collected resources were examined to determine whether they were necessary, connected, and semantically useful for the knowledge graph. The items highlighted in red in the original spreadsheet were identified as unused, redundant, and were therefore removed. The goal of filtering was to keep the graph minimal, coherent, and focused on entities and relations that directly support reasoning tasks such as course schedule validation and duplicate detection.

7.3.1 Removed elements and justification

Several classes and properties were excluded because they did not contribute to any graph operations:

- **Academic_year and Academic_term**: These entities were unnecessary because academic periods (e.g.,“Fall 2025”) are already handled through duplicate schedule checking. Explicitly modeling them would introduce additional nodes without providing any additional reasoning capability.
- **course_name as a separate Name entity**: The course name is sufficiently represented as a literal attribute of the `Course` class. A standalone Name entity is not required, since there is no multi-language or identity-linking requirement in this project scope.
- **credits**: Credit hours are not used in the knowledge graph queries and do not affect the course scheduling or relationship reasoning. Therefore, this attribute was removed to avoid storing unused data.
- **planned_id, selected_id, timestamp**: These fields represent system metadata and were not connected to any other entities. Including them would create isolated nodes that have no semantic role in the academic domain model.

7.3.2 Filtering rationale

The filtering decisions were guided by the following principles:

1. **Eliminate unused information**: Only data that directly contributes to the knowledge graph structure was kept.
2. **Avoid redundant representations**: Concepts already validated by the scheduling process were not modeled separately.
3. **Prevent isolated nodes**: Attributes or identifiers that were not linked to any other entities were removed to maintain graph integrity.
4. **Reduce complexity**: We kept the knowledge graph small so it would be easier to understand and faster to use.

7.4 Phase Outcomes

The Language Definition phase produced the following key outputs:

1. **Purpose-specific conceptual vocabulary** A complete set of ETypes, linking entities, activities, object properties, and data properties required to model the academic course scheduling domain.
2. **UKC-aligned concept set** Reusable concepts were matched with UKC entries, while domain-specific concepts were formally defined and assigned identifiers.
3. **Filtered and aligned dataset schema** All unnecessary elements were removed, resulting in a simplified yet semantically powerful resource set for reasoning tasks.
4. **Final Language Resource Sheet** A consolidated spreadsheet containing identifiers, labels, glosses, superclasses, and mappings, ready for Phase 4 (Resource-to-KG Mapping).

7.5 Summary

The Language Definition phase established a well-structured, purpose-specific semantic language for constructing the knowledge graph. Through Concept Identification, UKC Alignment, and Dataset Filtering, only essential concepts were retained, terminology was standardized, and the final dataset remained compact and reasoning-efficient. This provides a strong foundation for the next phases of the iTelos methodology.

8 Knowledge Definition

In this section, we describe how the prepared datasets and the purpose-specific language are transformed into a formal Knowledge Graph representation. Using the standardized datasets from Phase 2 and the conceptual vocabulary defined in Phase 3, a knowledge teleontology is defined and all datasets are aligned to it.

The goal of this section is to bring all information into a unified and consistent structure and to prepare both the data and the schema for Knowledge Graph construction and querying.

8.1 Knowledge Teleontology and kTelos

In this phase, the concepts defined in the Language Definition phase were used as the primary input for constructing an OWL knowledge teleontology. The objective of this activity was to transform the previously defined conceptual vocabulary into a formal and machine-readable schema suitable for Knowledge Graph instantiation.

Following the iTelos methodology, the kTelos process was applied by reviewing the finalized language resource and encoding the identified entity types, object properties, and data prop-

erties as OWL classes and properties. This ensured direct traceability between the language definition and the resulting ontology.

Core domain concepts such as *Student*, *Program*, *Curriculum*, *Course*, and *Time_slot* were modeled as general classes, reflecting their central role in the academic scheduling domain. Project-specific concepts including *Course_schedule*, *Curriculum_course*, *Planned_course*, and *Selected_course* were introduced to explicitly represent relationships and course selection states that are not directly captured by generic educational ontologies.

The ontology distinguishes between *Action* and *Event* to separate student-driven planning and selection operations from time-bound academic occurrences. *Action* represents conceptual operations such as planning or selecting courses, which describe a student's academic intent or state rather than a temporal occurrence. In contrast, *Event* represents scheduled academic activities that occur at specific times, such as course schedules and time slots. This separation avoids conflating decision-related concepts with temporal events and reflects the structure of the datasets.

In the current project, *Prerequisite* represents a simple dependency indicating that one course must be completed before another course can be taken within a specific curriculum. The prerequisite information is defined by three identifiers: *curriculum_id*, *pre_course_id*, and *course_id*. Although this dependency could be modeled as a direct object property between Course entities, it was modeled as a separate class to ensure a direct and lossless mapping from the source dataset, where prerequisite information is represented as an explicit relational table. In the current implementation, the *Prerequisite* class does not introduce additional semantics beyond representing this relationship.

The *Program* concept was modeled as a subclass of a general educational organization concept to reflect that academic programs are defined and managed within an institutional context. This choice was made to remain consistent with the structure of the dataset and to support conceptual coherence, without introducing additional organizational semantics beyond those required by the project.

The ontology was kept lightweight, aligned with the language resource, and created without reusing external ontologies. It serves as the schema for the Knowledge Graph.

8.2 Dataset Alignment

In this activity, the datasets prepared in Phase 2 were aligned with the conceptual vocabulary defined in the Language Definition phase and formalized in the ontology. The main objective was to ensure consistency between dataset structures and the ontology schema.

Column names in all datasets were adjusted to match the corresponding classes and properties defined in the language resource, and data types were verified for compatibility with the

OWL model. No semantic reinterpretation or restructuring of the data was performed beyond this alignment step.

All aligned datasets were consolidated into a single Excel file (`aligned_datasets.xlsx`), where each dataset is represented as a separate sheet:

- `Student` → Student entities
- `Curriculum` → Curriculum entities
- `Program` → Program entities
- `Course` → Course entities
- `Course schedule` → Course_schedule entities
- `Time_slot` → Course schedule time slots
- `Prerequisite` → Prerequisite relations
- `Curriculum_course` → Curriculum–Course relations
- `Planned_course` → Planned course selections
- `Selected_course` → Selected course records

This step ensures that all datasets are structurally consistent with the ontology and ready for instantiation in the Knowledge Graph.

8.3 Phase Outcomes

The Knowledge Definition phase resulted in the following outcomes:

1. An OWL knowledge teleontology formally encoding the conceptual vocabulary defined in the Language Definition phase.
2. Knowledge Graph schema describing students, programs, curriculum, courses, schedules, and their relationships.
3. A set of datasets aligned with the ontology, whose structure and data types are consistent and ready for Knowledge Graph instantiation in the subsequent phase.

8.4 Summary

In this phase, the conceptual language defined earlier was transformed into a formal OWL-based knowledge representation. The ontology was created by faithfully encoding the language resource, and the datasets were aligned accordingly. This provides a consistent and well-grounded foundation for Knowledge Graph instantiation and querying in the next phase.

9 Entity Definition

This section describes the Entity Definition phase of the iTelos methodology. Building on the outcomes of the Data Preparation and Language Definition phases, this phase focuses on translating the abstract conceptual model into concrete, instantiable entities. The main objective is to ensure that all real-world instances from the prepared datasets are consistently identified, correctly matched, and accurately mapped to the knowledge teleontology before constructing the Knowledge Graph.

9.1 Entity Identification

Entity identification was performed by analyzing the aligned datasets and determining which records correspond to concrete instances of the conceptual entities defined in the Language Definition phase. Core entities such as *Course*, *Course_schedule*, *Time_slot*, *Curriculum*, *Program*, and *Student* were directly instantiated from their respective datasets, as they represent fundamental academic objects within the domain.

In addition, contextual and activity-based entities, including *Curriculum_course*, *Planned_course*, *Selected_course*, and *Prerequisite*, were identified and modeled as separate entities. These entities capture curriculum structure and student behavior rather than standalone academic objects, enabling reasoning tasks such as schedule conflict detection and curriculum validation.

9.2 Entity Matching

Entity matching ensured that the same real-world object was represented by a single entity across all datasets. Stable identifiers such as *course_id*, *curriculum_id*, and *student_id* were used to generate consistent IRIs for core entities. For composite or context-dependent entities, such as time slots, a hash-based identifier strategy was adopted to guarantee uniqueness. In particular, each *Time_slot* entity was generated using a hash of its schedule identifier, day of week, start time, and end time, preventing duplicate time slot instances across schedules.

This matching strategy ensures consistency while remaining robust against heterogeneous identifier schemes in the source data. Its main strength lies in reproducibility and collision avoidance, while its limitation is that identifier semantics are derived from data values rather than explicit global identifiers.

9.3 Data Mapping and Model Refinements

During data mapping, dataset fields were systematically linked to the corresponding classes and properties defined in the ontology. The entity definitions were applied by distinguishing between core entities (*Course*, *Course_schedule*, *Time_slot*, *Curriculum*, *Program*, and *Student*) and contextual entities (*Curriculum_course*, *Planned_course*, *Selected_course*, and *Prerequisite*), ensuring a clear separation between standalone academic objects and relationship- or state-dependent entities.

Data properties were mapped according to the role of each entity. *Course* entities were associated with identifiers and descriptive attributes such as *course_id*, *course_index*, and *course_name*. Scheduling-related attributes including *schedule_start*, *schedule_end*, *course_type*, *course_frequency*, and *cycle* were mapped to *Course_schedule*, while concrete temporal information (*day_of_week*, *start_time*, *end_time*) was captured by *Time_slot*. Location-related properties (*building_name*, *room_name*, and *room_number*) were associated with scheduled course instances. Program- and curriculum-level attributes such as *program_id*, *program_name*, *academic_level*, *first_year*, and *curriculum_id* were mapped to the corresponding entities. Student-specific attributes including *student_id* and *enrollment_year* were assigned to *Student* entities, while curriculum groupings were represented through *Curriculum_course* using the *group_name* and *required_course_id* attributes.

Several model refinements were applied to improve semantic clarity and to support the project's competency questions. The *degree_program* attribute was removed from the *Course* entity, as it introduced redundancy with program and curriculum relationships already captured through object properties. The *room_number* attribute in *Course_schedule* was converted from a numeric type to a string to accommodate alphanumeric room identifiers (e.g., "1000A"). Furthermore, an *academic_level* attribute was added to the *Program* entity to explicitly distinguish between different program levels, such as Bachelor and Master programs.

Object properties were instantiated to represent structural, temporal, prerequisite, and behavioral relationships between entities. *has_schedule* links courses to their schedules, while *includes* and *has_course* capture curriculum–course relationships. The *has_curriculum* and *part_of* properties represent institutional and structural hierarchies. Course dependencies were modeled using *has_prerequisite* and its inverse *is_prerequisite_of*. Student behavior was captured through the *plans* and *selects* properties, linking students to planned and selected courses. Together, these mappings ensure a consistent and semantically grounded Knowledge Graph representation aligned with the ontology and the prepared datasets.

9.4 Entity Instantiation Using Python

After completing entity identification, matching, and mapping, the Knowledge Graph was instantiated using Python. The implementation relied on the `rdflib` library to load the existing OWL teleontology, define namespaces, and generate RDF triples programmatically. Each dataset was processed independently, and entities were created using stable IRIs derived from dataset identifiers.

Object and data properties were assigned according to the mappings defined in this phase, with explicit XML Schema datatypes applied to all literals. The Python-based construction approach ensures repeatability, scalability, and consistency between the datasets and the ontology. The final Knowledge Graph was serialized in both OWL/XML and Turtle formats and prepared for querying and validation using SPARQL.

9.5 Summary

The Entity Definition phase transformed the abstract conceptual model into concrete, interoperable entities ready for Knowledge Graph construction. Through careful entity identification, robust matching strategies, targeted model refinements, and programmatic instantiation using Python, this phase ensured semantic consistency across datasets and provided a reliable foundation for subsequent querying and reasoning activities.

10 Evaluation

This section presents the evaluation of the final Knowledge Graph produced by applying the iTelos Knowledge Graph Engineering methodology. The evaluation assesses the quality, consistency, and suitability of the resulting Knowledge Graph with respect to the project purpose and the defined Competency Questions (CQs).

Following the iTelos framework, the evaluation is conducted at both the Knowledge Layer and the Data Layer, addressing the primary objective of purpose satisfaction and the secondary objective of reusability. In addition, the evaluation is complemented by the execution of representative SPARQL competency queries to validate the operational correctness of the Knowledge Graph.

10.1 Knowledge Graph Statistics

The final Knowledge Graph consists of a structured set of entity types, object properties, and data properties derived from the aligned datasets and instantiated according to the knowledge

teleontology.

The schema includes core academic entities such as *Student*, *Course*, *Curriculum*, *Program*, *Course_schedule*, and *Time_slot*, as well as contextual and activity-based entities including *Curriculum_course*, *Planned_course*, *Selected_course*, and *Prerequisite*.

At the data layer, the Knowledge Graph contains instances for all students, courses, curricula, programs, schedules, and student activity records included in the prepared datasets. Each entity type is represented by a consistent and non-empty set of instances, and all object and data properties defined in the ontology are populated according to the mapping rules described in the Entity Definition phase. No isolated entities were observed, indicating a well-connected graph structure suitable for querying and reasoning.

10.2 Knowledge Layer Evaluation

10.2.1 Purpose-based Evaluation (Primary Objective)

The primary objective of the knowledge layer evaluation is to verify whether the Knowledge Graph satisfies the project purpose by supporting the defined Competency Questions.

Following the iTelos methodology, a coverage-based evaluation was applied by comparing the entity types and properties extracted from the Competency Questions with those defined in the knowledge teleontology. All entity types required by the Competency Questions, including *Student*, *Course*, *Course_schedule*, *Time_slot*, *Curriculum*, and *Program*, are explicitly represented in the teleontology. Similarly, all properties required to express temporal relations, course planning and selection, curriculum membership, and prerequisite constraints are present and correctly defined.

As a result, the entity-type coverage (CovE) and property coverage (CovP) of the teleontology with respect to the Competency Questions are effectively complete, corresponding to coverage values close to 1.0. This confirms that the teleontology fully supports the reasoning tasks implied by the project purpose.

10.2.2 Reusability Evaluation (Secondary Objective)

The secondary objective of the knowledge layer evaluation concerns the reusability of the Knowledge Graph. Core concepts such as *Student*, *Course*, *Program*, *Curriculum*, and *Time_slot* were aligned with corresponding entries in the Universal Knowledge Core (UKC) during the Language Definition phase. This alignment ensures semantic interoperability with external educational knowledge graphs and supports reuse in future iTelos-driven projects.

Domain-specific concepts such as *Course_schedule*, *Planned_course*, and *Selected_course* were intentionally defined as project-specific extensions, reflecting operational aspects not cov-

ered by reference ontologies. This design balances reuse and customization while preserving conceptual clarity.

10.3 Data Layer Evaluation

The data layer evaluation focuses on assessing the connectivity, consistency, and completeness of the instantiated Knowledge Graph.

10.3.1 Identifier and Datatype Consistency

All entities were instantiated using stable and reproducible IRIs derived from dataset identifiers. Composite entities such as *Time_slot* were generated using hash-based identifiers to guarantee uniqueness and avoid duplication across schedules. All literal values were assigned explicit XML Schema datatypes, and datatype inconsistencies were resolved during data preprocessing, including the conversion of room identifiers to string values to support alphanumeric formats.

10.3.2 Connectivity Evaluation

In accordance with the iTelos methodology, the evaluation considered both entity connectivity and property connectivity. Although a full numerical connectivity matrix was not computed, qualitative inspection confirmed that all core and contextual entities participate in at least one object property relation and are associated with non-null data property values where applicable. No disconnected subgraphs were identified in the final Knowledge Graph.

This indicates a high level of entity and property connectivity, consistent with the iTelos requirements for a well-integrated Knowledge Graph.

10.4 Competency Query Execution and Validation

To validate purpose satisfaction at the operational level, the Competency Questions were translated into SPARQL queries and executed against the final Knowledge Graph. The executed queries include detecting time conflicts among planned courses, identifying overlaps among selected courses, verifying conflicts between curriculum-required courses and existing student schedules, retrieving courses scheduled at specific times and days, and detecting overlapping major course schedules within a curriculum.

All competency queries were successfully executed and returned correct and meaningful results. This confirms that the Knowledge Graph structure and data representation adequately support the intended reasoning and validation tasks.

10.5 Summary

The evaluation demonstrates that the final Knowledge Graph satisfies both the primary and secondary objectives defined by the iTelos methodology. At the knowledge layer, the teleontology provides complete coverage of the concepts and properties required by the Competency Questions and supports reusability through UKC alignment. At the data layer, the instantiated entities are consistent, well-connected, and free of isolation or datatype inconsistencies.

The successful execution of all competency queries confirms that the Knowledge Graph is suitable for detecting scheduling conflicts, validating curriculum constraints, and supporting academic decision-making at the National University of Mongolia.