Ontology-Driven Sub-Query Extraction in UnIT-Net IEDI

☑ **Vadim Ermolayev**
Natalya Keberle
Vladimir Shapar
Vladimir Vladimirov

http://eva.zsu.zp.ua/

Zaporozhye State University, Ukraine

http://www.zsu.edu.ua/

UnIT-Net: **IT in University Management Network**
TEMPUS/TACIS MP-JEP-2010-2003

http://www.unit-net.org.ua/

ISTA'04, Salt Lake City, 15-17.07.2004

UnIT-Net – TEMPUS/TACIS MP-JEP-2010-2003

- **Title: IT in University Management Network**
- **Objective(s):**
  - Creation of the Ukrainian National "**Network of Excellence**"
  - Dissemination of the **best practices** – IT in University Management
  - Elaboration of the **Specifications** recommending reasonable ways of using IT in University Management
  - Design and implementation of the **Research Prototype** of the National **Infrastructure for Electronic Data Interchange** (mediator-wrapper, hybrid knowledge representation)
- **Participants:**
  - Kherson State University (project coordinator)
  - Ministry of Education and Science of Ukraine
  - Kharkiv national University
  - Zaporozhye State University
  - University of Nice – Sofia Antipolis, France
  - Glasgow Caledonian University, UK
- **URL:** http://www.unit-net.org.ua/

Problems to Solve:

- **Q1**: **How to add a new IR to a Retrieval System**?
  - **A1.1**: Wait until its owner volunteers to provide the annotation (**IR registration**) – **UnIT-Net**
  - **A1.2**: **Crawl** the Web for publicly available IR-s

- **Q2**: **How to align our beliefs on IR to its actual state**?
  - **A2.1**: **Wait until the owner informs** about the changes, then align – **UnIT-Net**
  - **A2.2**: **Mine** the changes from IR-s regularly. Align accordingly

Problems to Solve:

- **Q3**: **How to formulate & perform queries to such an IR Grid**?
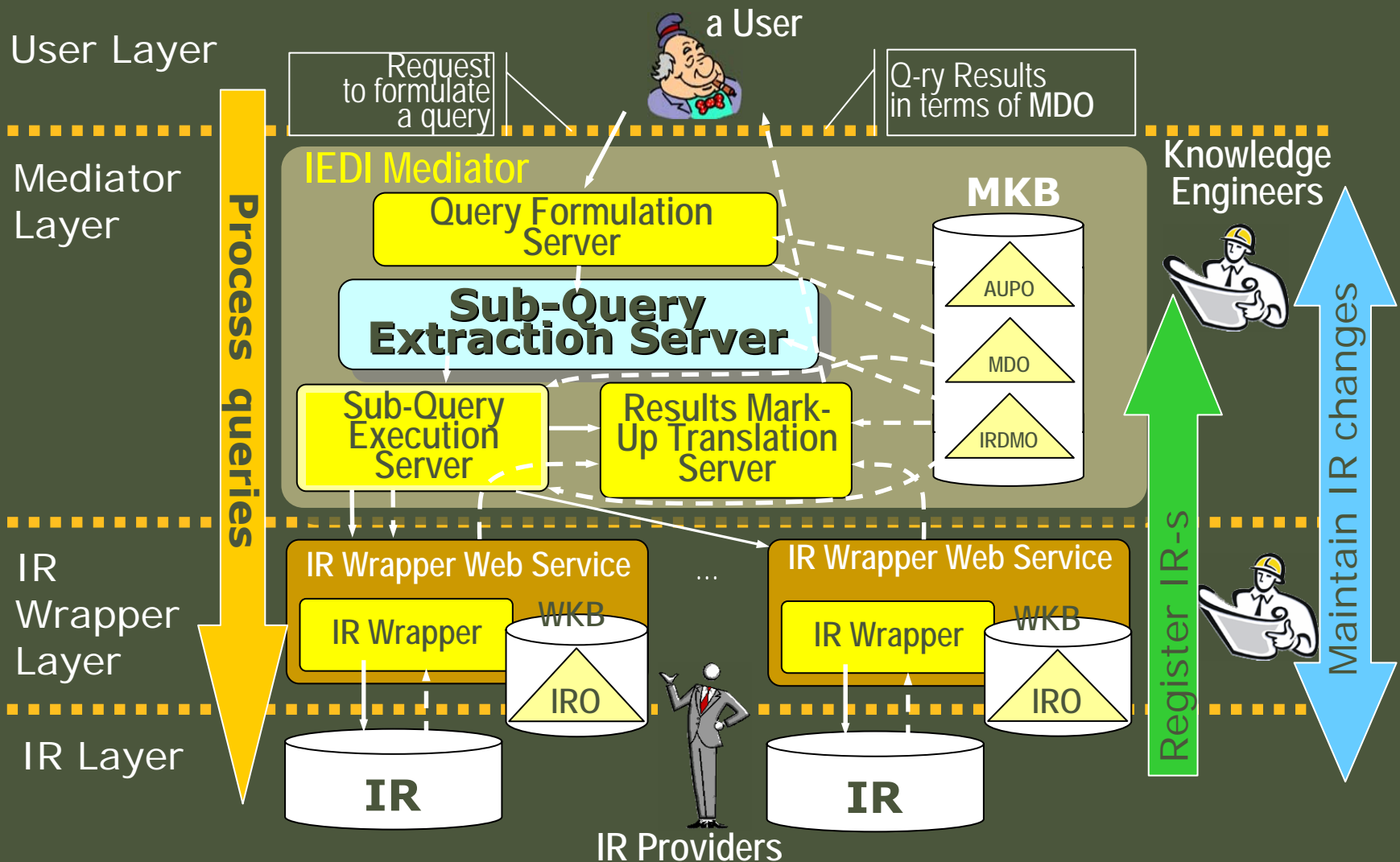
- **A3**: **UnIT-Net**
  - Assist a User to **formulate** his query in terms familiar to him
  - **Transform** the query to the terms commonly accepted for the Domain
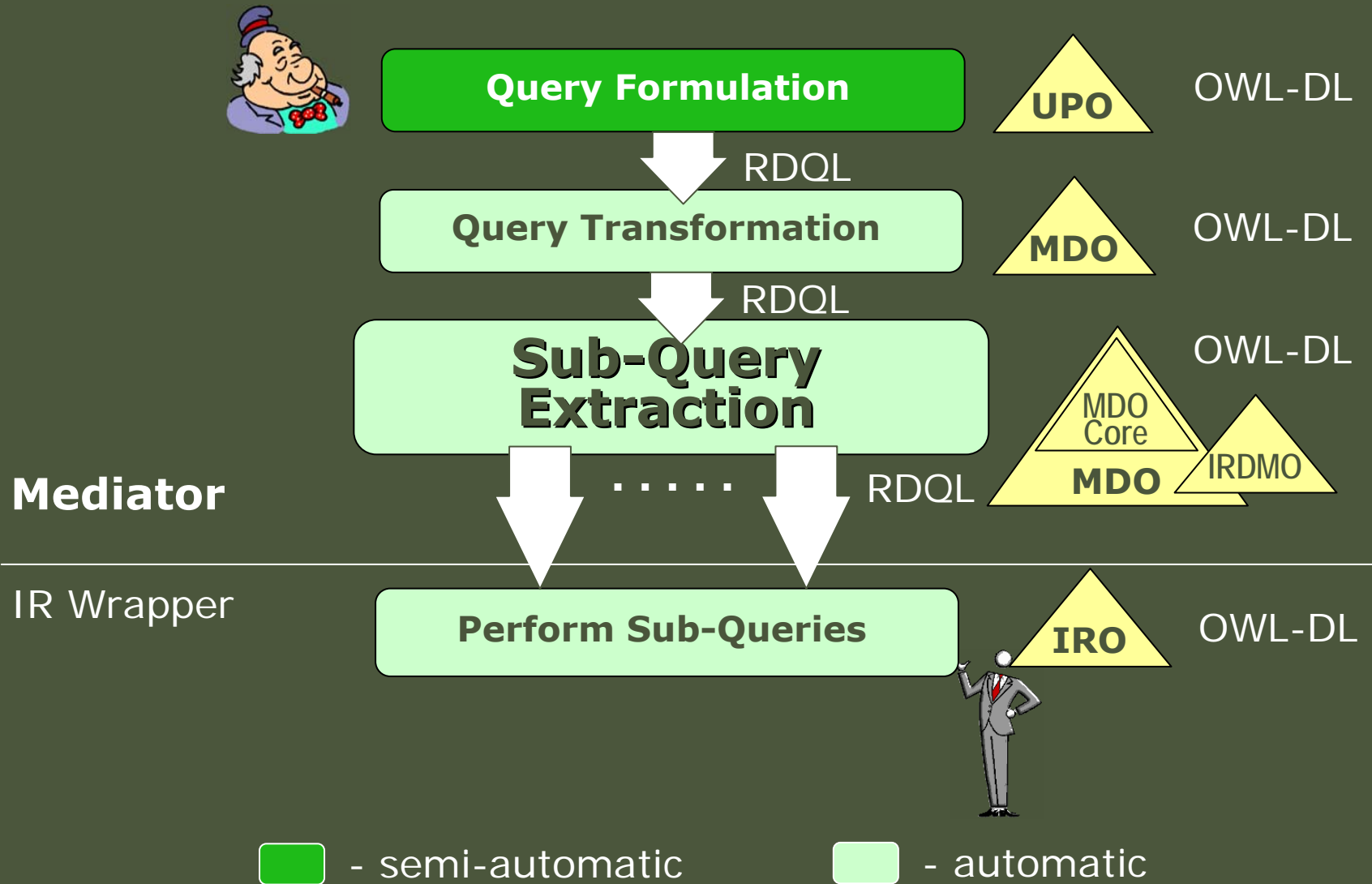 - **Decompose** the query to the set of sub-queries
  - **Forward** sub-queries to appropriate IR wrappers
  - …
  - (**Fuse** and) **Return** the **results** to the User

IEDI Architecture in a Nutshell

Ontology and Language Aspect:

Query Formulation → UPO — OWL-DL

↓ RDQL

Query Transformation → MDO — OWL-DL

↓ RDQL

Sub-Query Extraction → MDO Core, IRDMO, MDO — OWL-DL

Mediator

· · · · · RDQL

IR Wrapper

Perform Sub-Queries → IRO — OWL-DL

■ - semi-automatic   ■ - automatic

A Walkthrough Example

Why?

- *Return the list of the 1-st year CS students who:*
  - *had received **maximal grade** in **Mathematics** at the entrance examinations*
  - *and have **failed** to pass the 1-st Term examination in **any basic course in Mathematics***

A Walkthrough Example

- Return the list of the 1-st year CS students who:

  - had received **maximal grade** in **Mathematics** at the entrance examinations

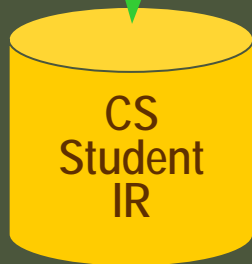 - and have **failed** to pass the 1-st Term examination in **any basic course in Mathematics**

Univ. Entrant IR

CS Student IR

Ontology

Mathematics:
├─Math Analysis
├─Linear Algebra
├─Analytical Geometry
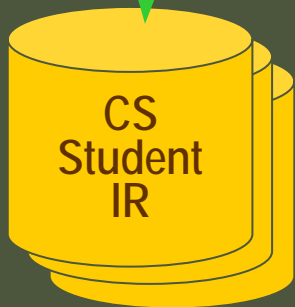├─...

A Walkthrough Example

of different Universities

☐ *Return the list of the 1-st year CS **students** who:*

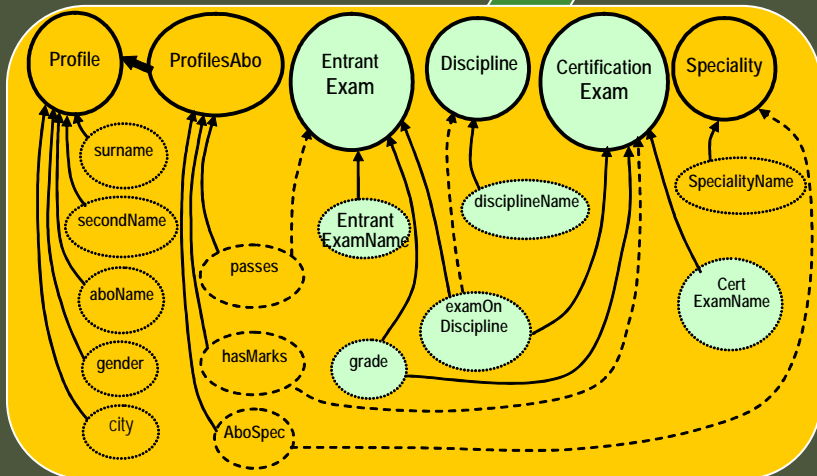▪ *had received **maximal grade** in **Mathematics** at the entrance examinations*
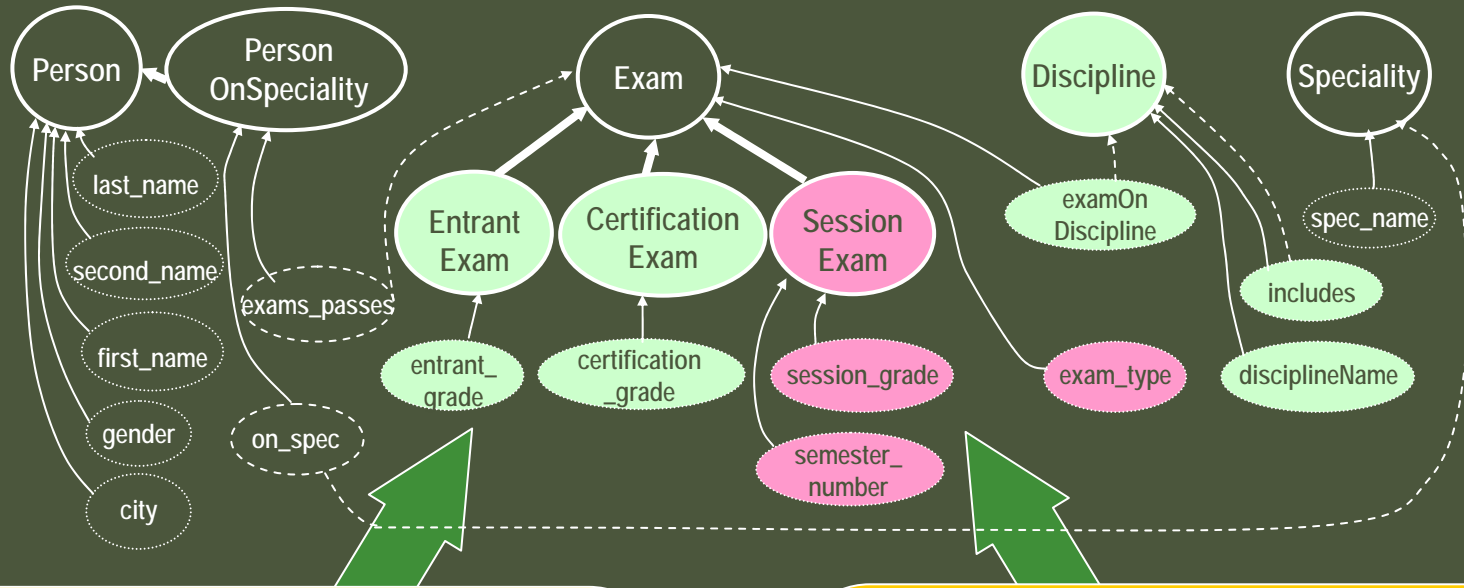
Univ. Entrant IR

▪ *and have **failed** to pass the 1-st Term examination in **any basic course in Mathematics***
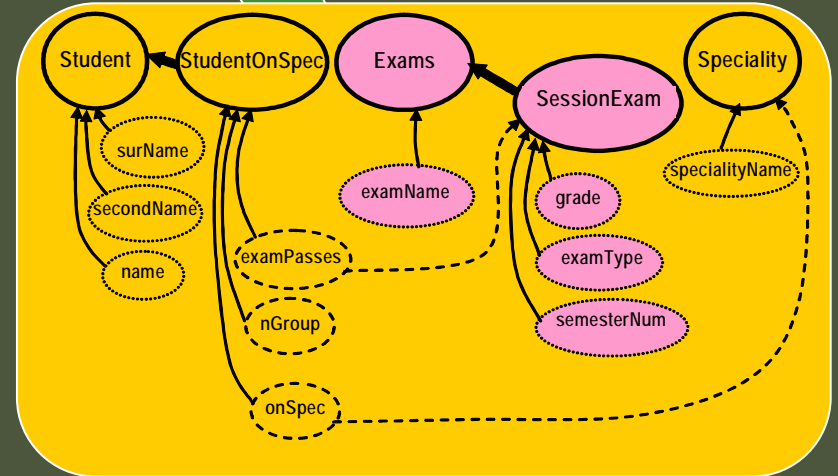
CS Student IR

..and different basic courses in the 1-st term

Mathematics:
- Math Analysis
- Linear Algebra
- Analytical Geometry
- ...

MDO and IRO-s for the Example

IRO: University Entrants
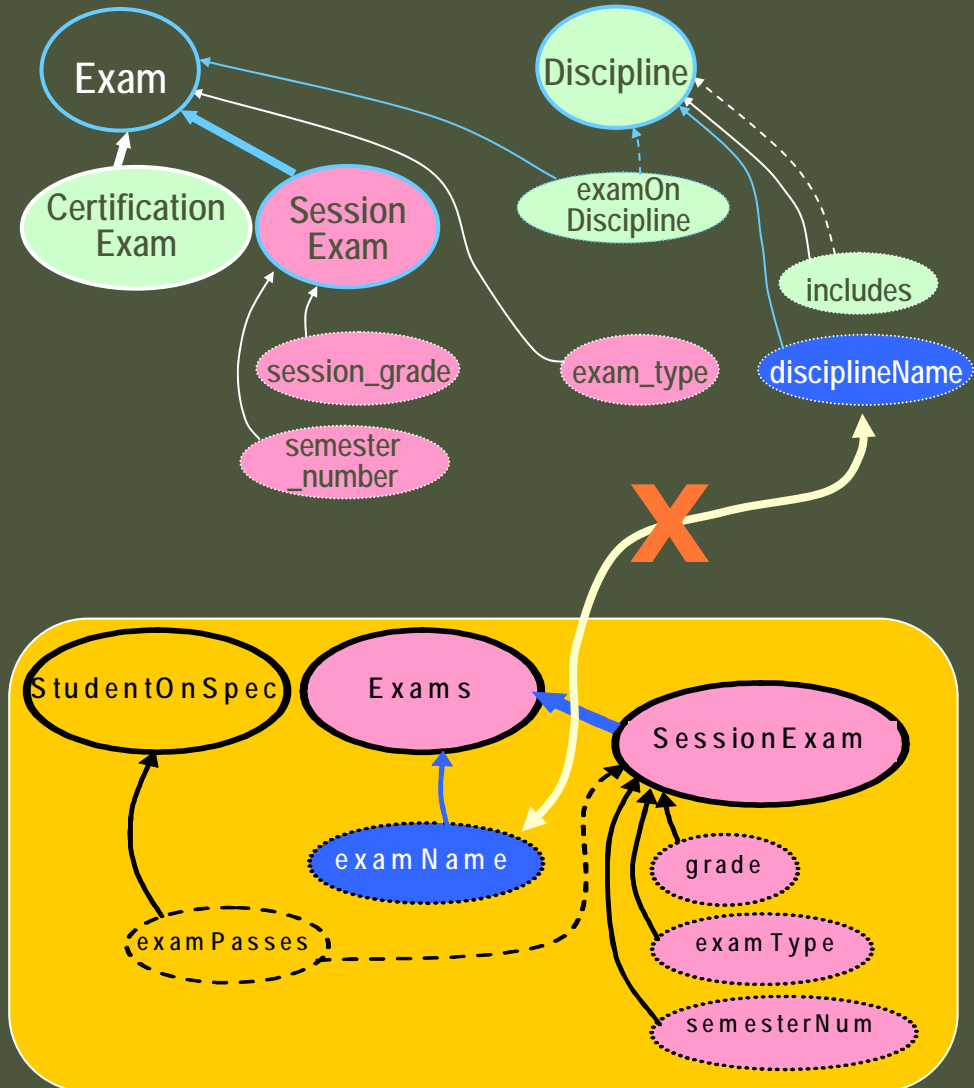
IRO: CS Students

IRO Registration is NOT a Simple Join

Note:

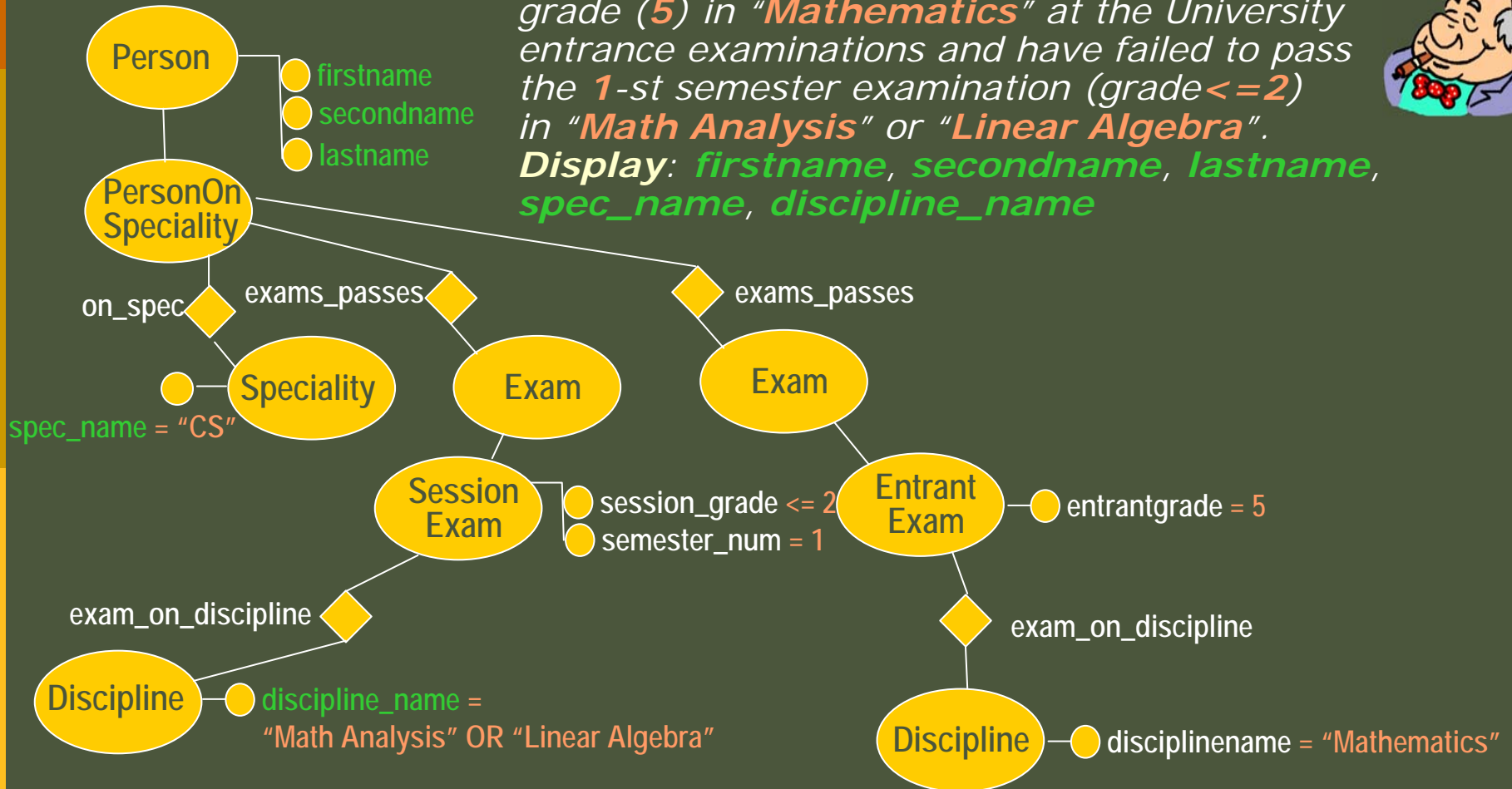- IR-Domain Mapping Ontology (**IRDMO**) provides only **minimally necessary mappings**:
  - Slots defined for the concept
  - But NOT the slots inherited from its superclass
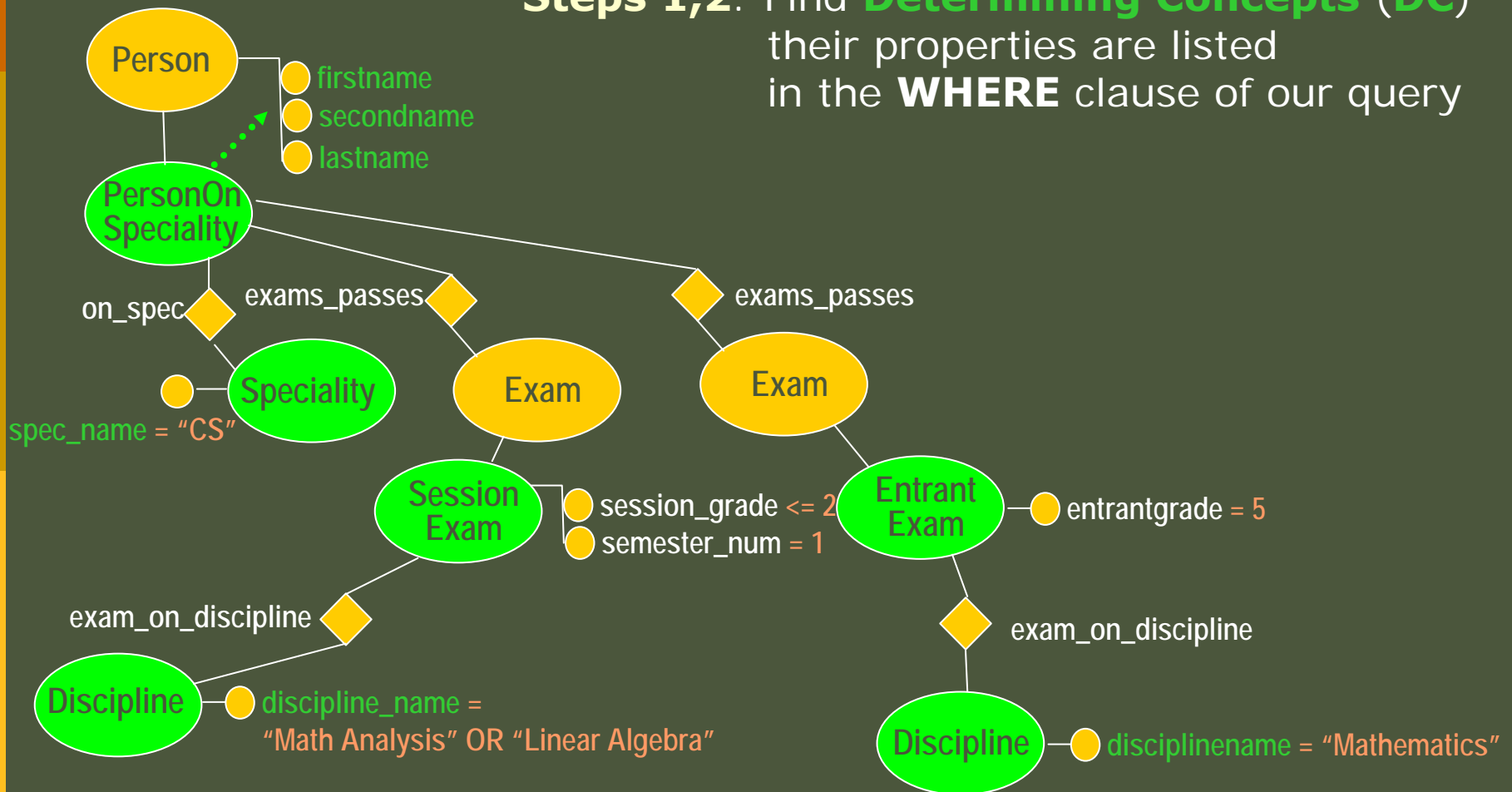
Requires **Late Binding**

Conceptual Graph for Example Query

Retrieve "**CS**" *students who had got maximum grade (**5**) in* "**Mathematics**" *at the University entrance examinations and have failed to pass the* **1**-*st semester examination (grade**<=2**) in* "**Math Analysis**" *or* "**Linear Algebra**". ***Display****: **firstname**, **secondname**, **lastname**, **spec_name**, **discipline_name***

Person — firstname, secondname, lastname

PersonOn Speciality

on_spec

exams_passes

exams_passes

spec_name = "CS"

Speciality

Exam

Exam

Session Exam — session_grade <= 2, semester_num = 1

Entrant Exam — entrantgrade = 5

exam_on_discipline

exam_on_discipline

Discipline — discipline_name = "Math Analysis" OR "Linear Algebra"
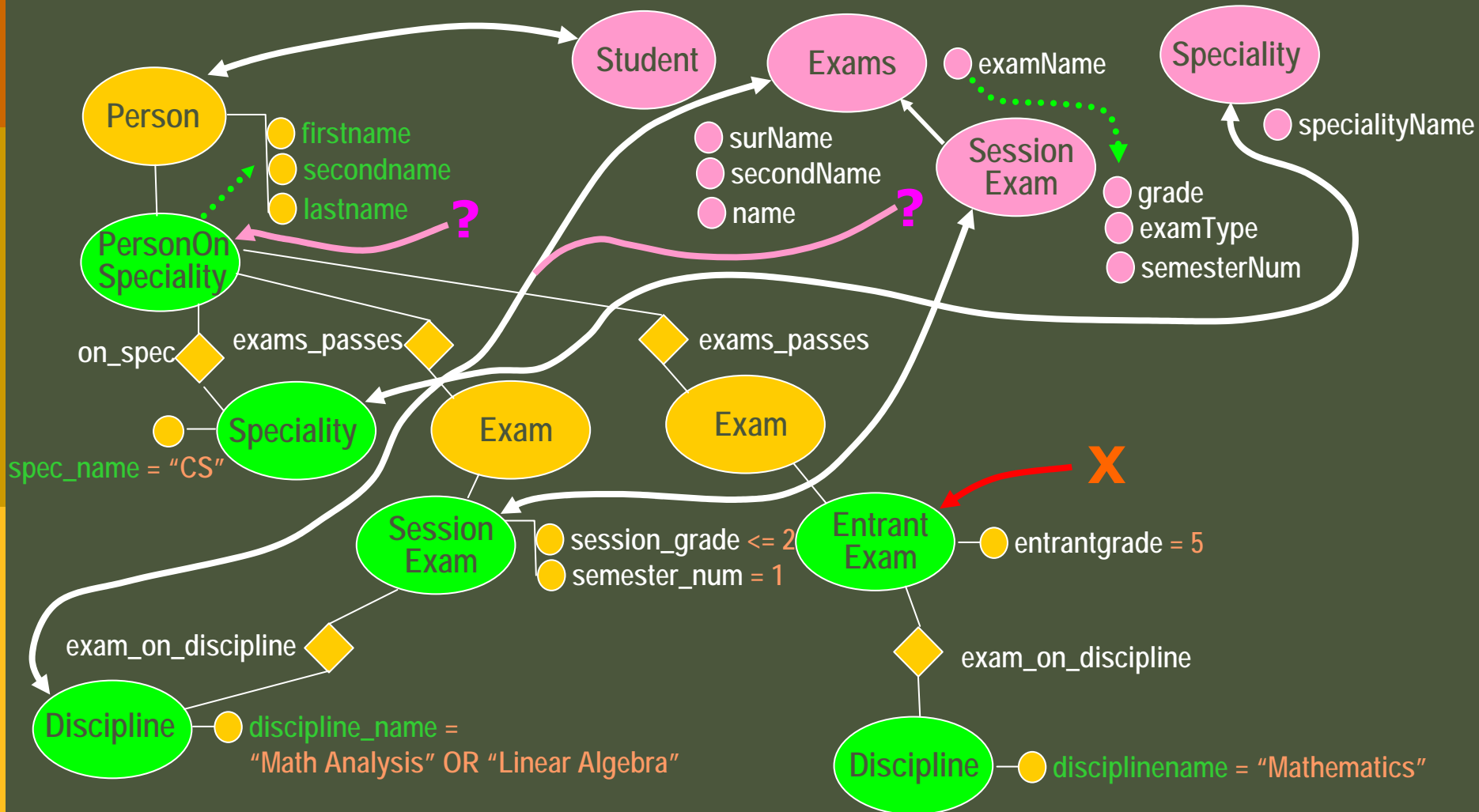
Discipline — disciplinename = "Mathematics"

Sub-Query Extraction

Steps 1,2: Find **Determining Concepts** (**DC**)
their properties are listed
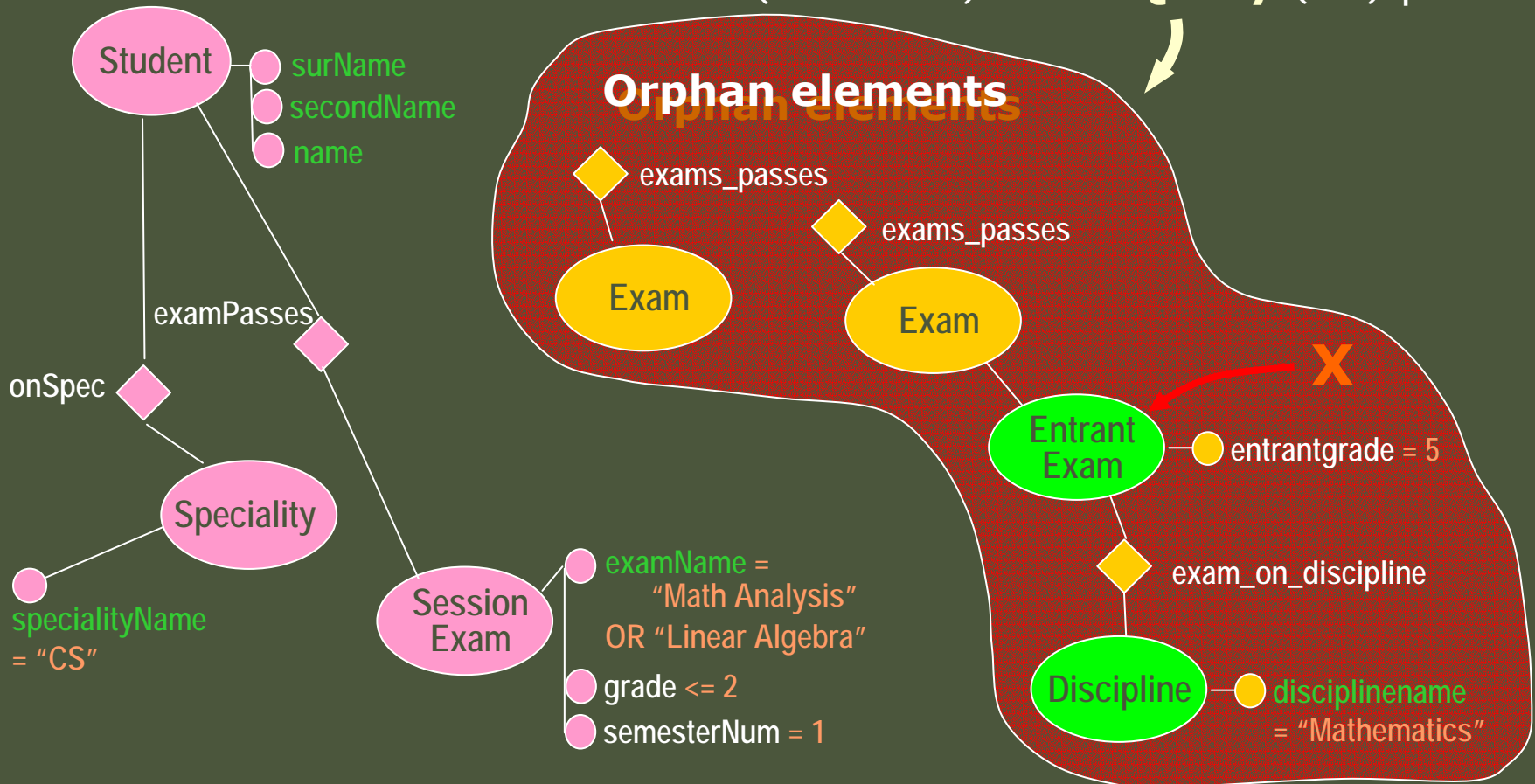in the **WHERE** clause of our query

Mappings Recorded in IRDMO
(MDO ↔ Stud IRO)

? – **Late Binding** required to resolve the mapping

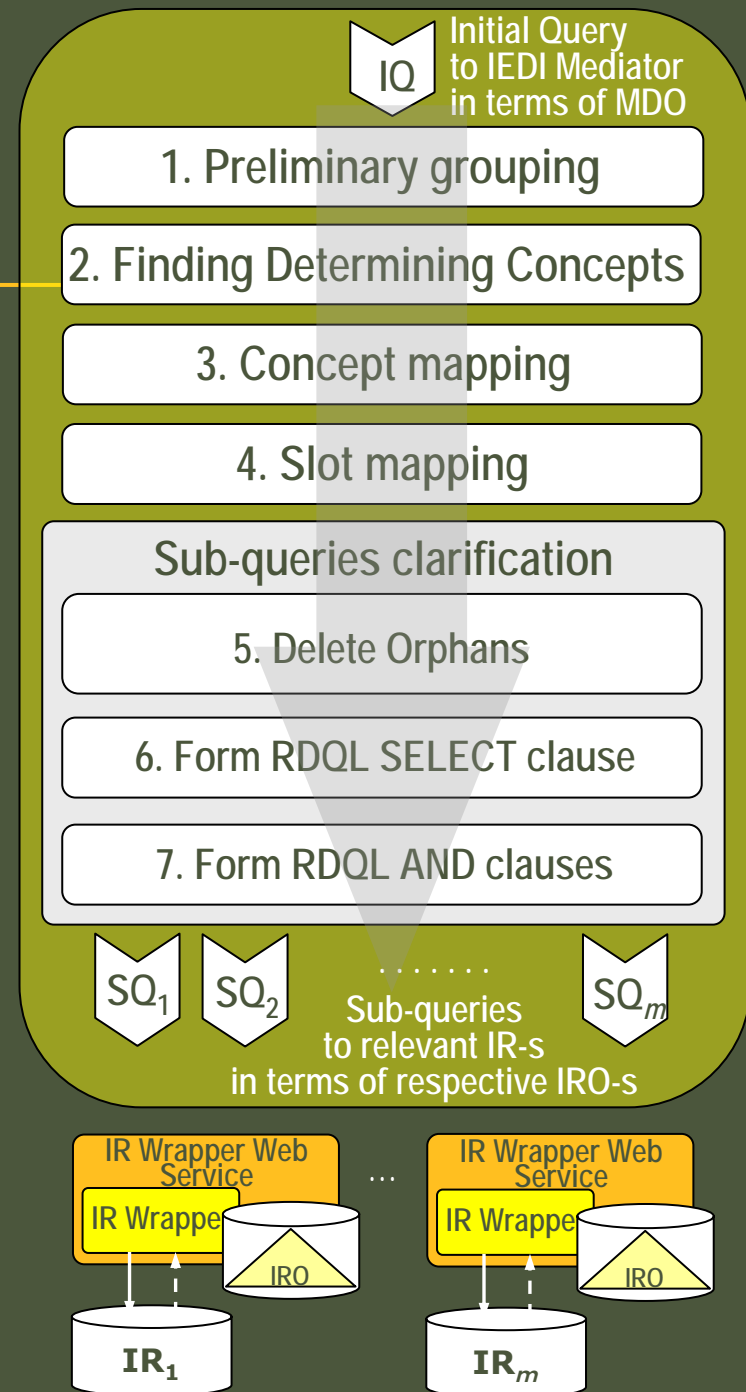# **Resolved Mappings**: MDO↔Stud IRO

Steps 3,4: Resolve **Concept and Slot Mappings**
Result: a (subset of) **Billet Query** (ies) per **IR**

Orphan elements – to be deleted at Query Clarification - Step 5

ODSQE at a Glance

Features:

- IRDMO provides only minimally necessary mappings
- Late Binding is used to resolve mappings in subclass/superclass chains (further development - meronymy)
- Multiple mappings result in series of Billet Queries per IR
- MDO Core is used to fix the minimal set of concepts common to all IR-s

IQ — Initial Query to IEDI Mediator in terms of MDO

1. Preliminary grouping

2. Finding Determining Concepts

3. Concept mapping

4. Slot mapping

Sub-queries clarification

5. Delete Orphans

6. Form RDQL SELECT clause

7. Form RDQL AND clauses

SQ_1 SQ_2 ...... SQ_m

Sub-queries to relevant IR-s in terms of respective IRO-s

IR Wrapper Web Service — IR Wrapper — IRO ... IR Wrapper Web Service — IR Wrapper — IRO

IR_1 ... IR_m

ODSQE Properties

- Builds the set of IRO queries for a specific MDO query (**existence**)

- Builds the unique set of IRO queries for a specific MDO query (**uniqueness**)

- The recall of a hypothetic MDO query is less or equal to the combined recall of the set of the produced IRO queries (**complete coverage**)

To Conclude …

- Initial **proof-of-concept implementation** of ODSQE is done
- **UnIT-Net IEDI** Sub-Query Extraction Server is **under development**

- **Further development:**
  - Richer semantic relationships
  - More intelligence in Late Binding for non-equvalent concepts

That's it …

Shall be happy
to know the answers

These slides are available from: http://eva.zsu.zp.ua/eva_personal/evapubs.htm