Zaporozhye National University, Ukraine
Dept. of IT, Intelligent Systems Research Group

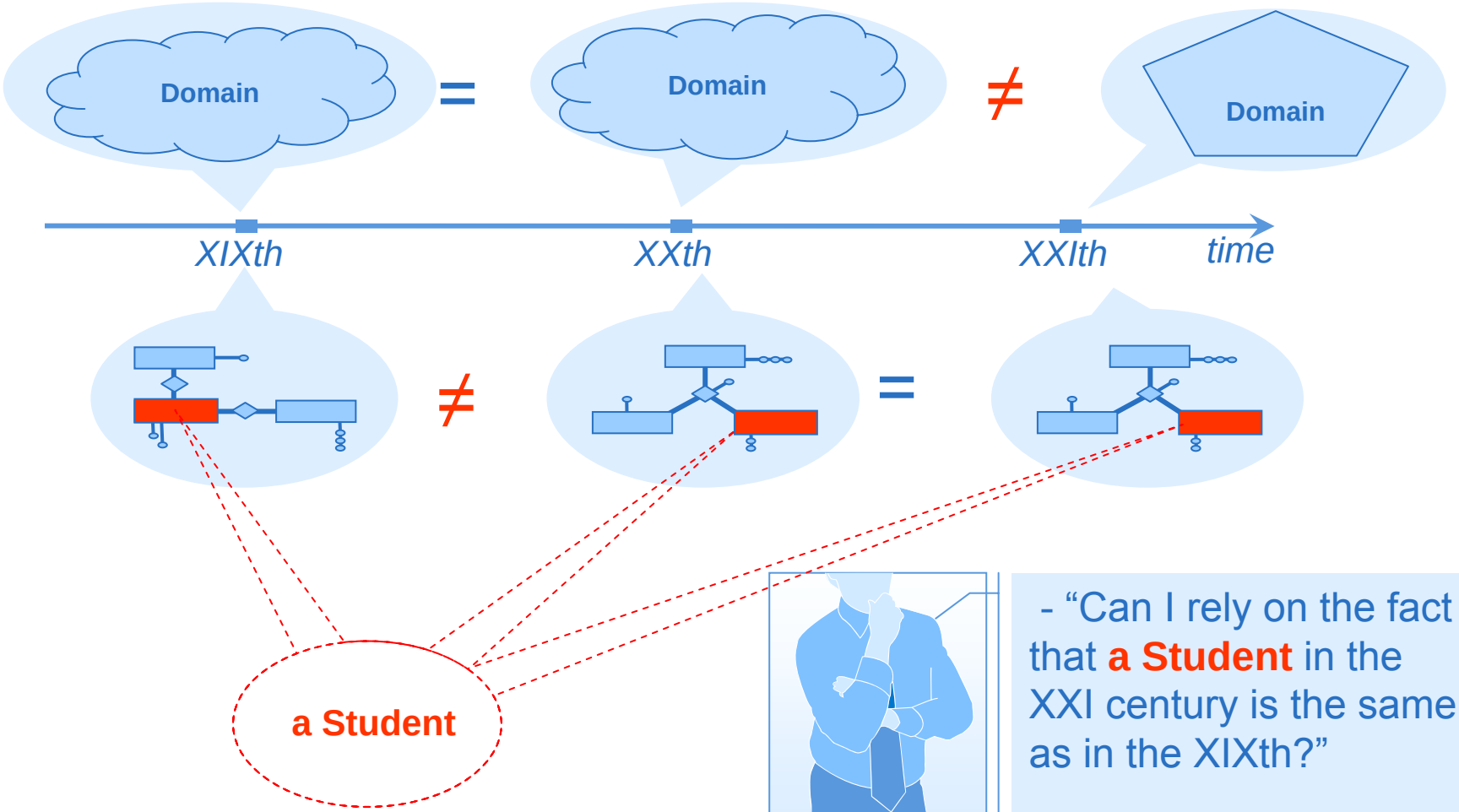
Ontology Evolution Analysis with OWL-MeT

☑ Natalya Keberle
Yuriy Litvinenko
Yuriy Gordeyev
Vadim Ermolayev

Intl. Workshop on Ontology Dynamics (IWOD) @ ESWC-2007
Innsbruck, Austria, 7th June 2007

Ontology Evolution Analysis(1)

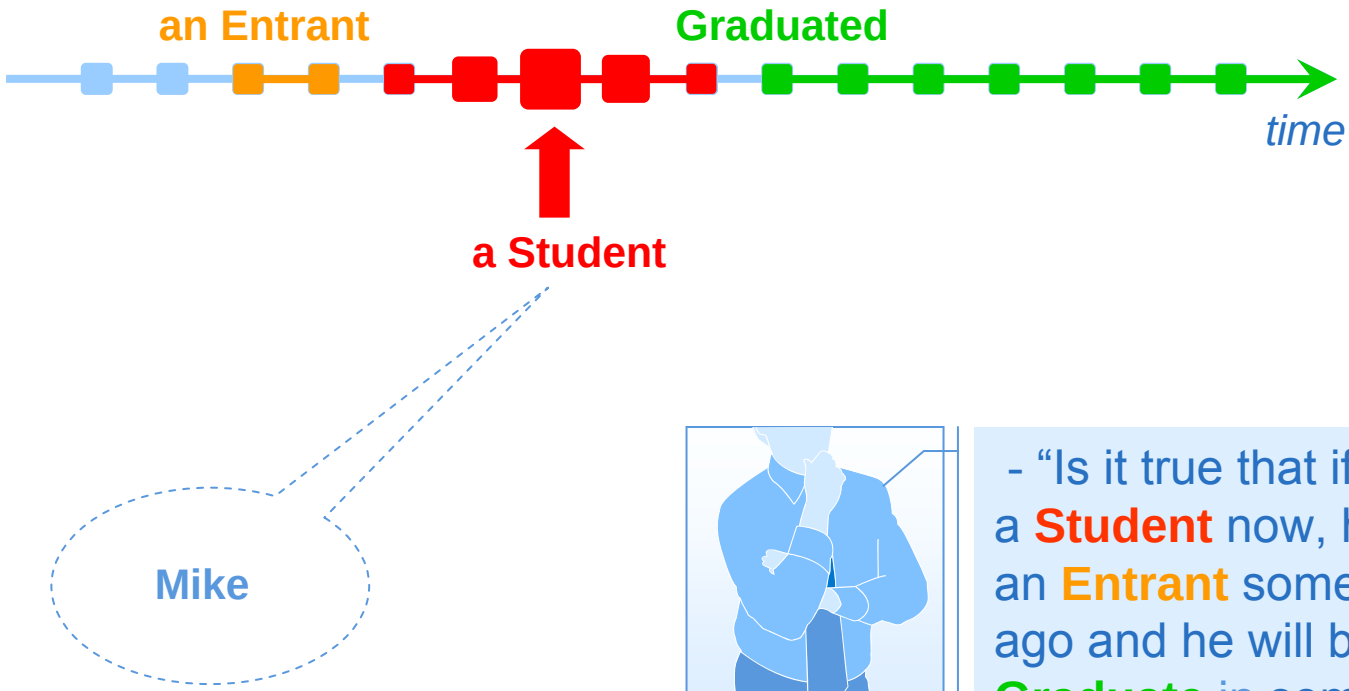
Versions compatibility



- “Can I rely on the fact that **a Student** in the XXI century is the same as in the XIXth?”

Ontology Evolution Analysis(2)

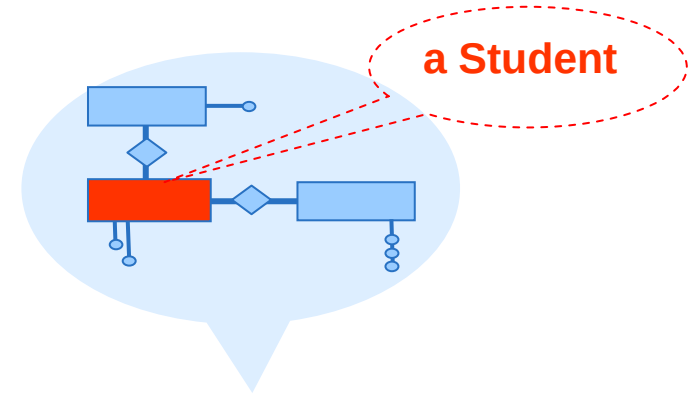
Checking derivability of a fact in different versions

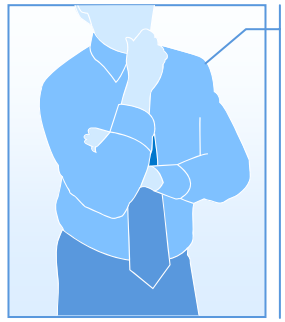


- “Is it true that if Mike is a **Student** now, he was an **Entrant** some time ago and he will be a **Graduate** in some time in the future?”

Ontology Evolution Analysis(3)

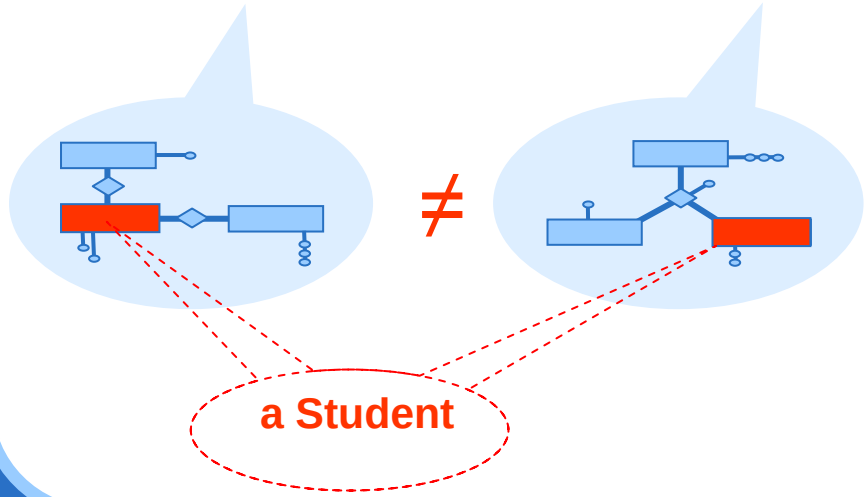
Structural analysis of versions and of version changes

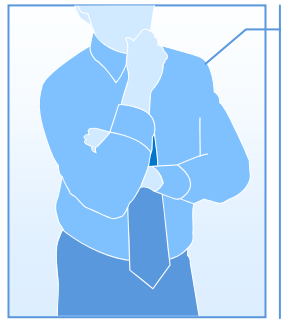




- “Get the properties of a **Student** of the XIX century”

XIXth *XXIth* *time*





- “What was changed in the definition of a **Student** of the XXI century as compared to the one of the XIX century?”

Ontology Evolution Analysis "wishlist”

- Ontology Version Management System
  - All ontology versions are available
  - Or, there is a version log
  - Or, both versions and a version are available
- Explicit referencing of ontology versions
- Different Query Types
  - Reasoning queries
  - Meta - level queries on versions compatibility
  - Retrieval queries

Existing approaches to ontology evolution analysis

- **Versioning and structural analysis of version logs**

  OntoView [Klein 2004]

- **Proof-theoretic approach – usage of temporal logic**

  MORE tool [Huang & Stuckenschmidt, 2005]

  - **LTLm**

Requirements for Temporal Logic

- The notion of distance
  - Metric logic
- Explicit version names addressing
  - Hybrid logic
- Semantic Web oriented
  - Description logic

Temporal Logics overview

- **Propositional:**
  - **LTL, CTL**
  - **MT [Hustadt et al. 2005]**
  - **PTC(MT) [Keberle 2005]**

  Reasoning support : LoTREC (refl.& trans. frames), MetTel, …
- **DL-oriented:**
  - **Schild's logic [Schild 1993]**
  - **Family of CIQ $_{US}$ [Wolter & Zakharyashev 1999]**
  - **TL-ALCF [Artale & Franconi 2000]**

  Reasoning support : open question

ALCIO(MT) proposal

$$E, F \rightarrow A \mid top \mid bottom \mid E \sqcap F \mid E \sqcup F \mid \neg E \mid \exists R.E \mid \forall R.E \mid \{o\}$$

$$P \rightarrow R \mid P^{-1}$$

$$C, D \rightarrow E \mid \{a\} \mid C \text{ intersection } D \mid C \text{ union } D \mid \text{not } C \mid C@\{a\} \mid \text{future } n\ C \mid$$
$$\mid \text{past } n\ C \mid \text{somefuture } C \mid \text{somepast } C \mid \text{allfuture } C \mid \text{allpast } C$$

ALCIO(MT)

Specific semantics of **ALCIO(MT**) is defined on reflexive and transitive frames

$$\mathrm{M} = < \Delta, dist, \{R_F, R_P\}, I, V >$$

$$(future\ n\ C)^{I(k)} = \{o \in \Delta^k : \exists j = k + n, o \in C^{I(j)}\}$$

$$(past\ n\ C)^{I(k)} = \{o \in \Delta^k : \exists j : k = j + n, o \in C^{I(j)}\}$$

$$(somefuture\ C)^{I(k)} = \{o \in \Delta^k : \exists j \geq k, o \in C^{I(j)}\}$$

$$(somepast\ C)^{I(k)} = \{o \in \Delta^k : \exists j \leq k, o \in C^{I(j)}\}$$

$$(allfuture\ C)^{I(k)} = \{o \in \Delta^k : \forall j \geq k, o \in C^{I(j)}\}$$

$$(allpast\ C)^{I(k)} = \{o \in \Delta^k : \forall j \leq k, o \in C^{I(j)}\}$$

$$(C@\{a\})^{I(k)} = \{o \in C^{I(den(a))}\}$$

ALCIO(MT)

ALCIO (**MT**) is decidable as the syntactic variant of **CIQ** $_{US}$

SAT problem for **ALCIO(MT**) **is EXPTIME-hard** [Areces, Blackburn & Marx 1999]

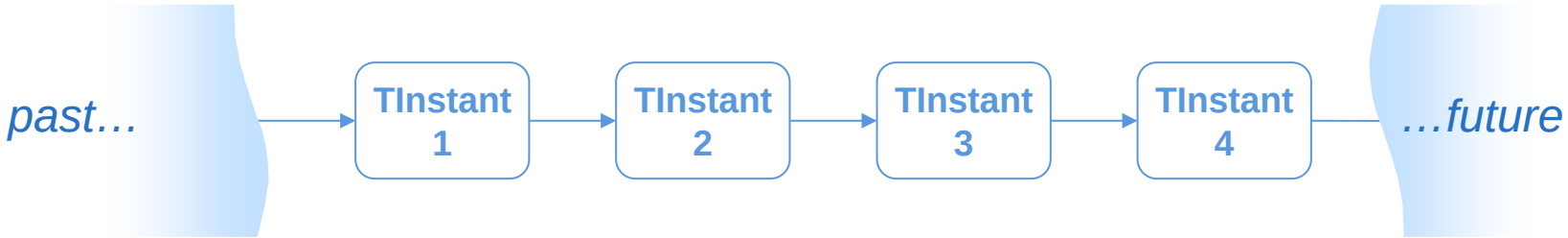
Tableau-based procedure of SAT checking is developed

OWL-MeT proposal

- **OWL-MeT**: **O**ntology **W**eb **L**anguage for **Me**tric **T**ime
- Metric and Temporal extension of OWL
- Based on ALCIO(MT)

PLUS

- Definition of TimeStructure for versions identification and ordering. TimeStructure is a finite set of version IDs .



OWL-MeT examples

Student is…

```
<TClass rdf:ID="Entrant"/>
<TClass rdf:ID="Graduated"/>
<TClass rdf:ID="Student">
  <equivalentClass>
    <intersectionOf>
      <TRestriction>
       <somepast rdf:resource="#Entrant"/>
      </TRestriction>
      <TRestriction>
        <allfuture>
          <TClass>
            <unionOf>
              <TClass about="#Student"/>
              <TClass about="#Graduated"/>
            </unionOf>
          </TClass>
        </allfuture>
      </TRestriction>
    </intersectionOf>
  </equivalentClass>
</TClass>
```

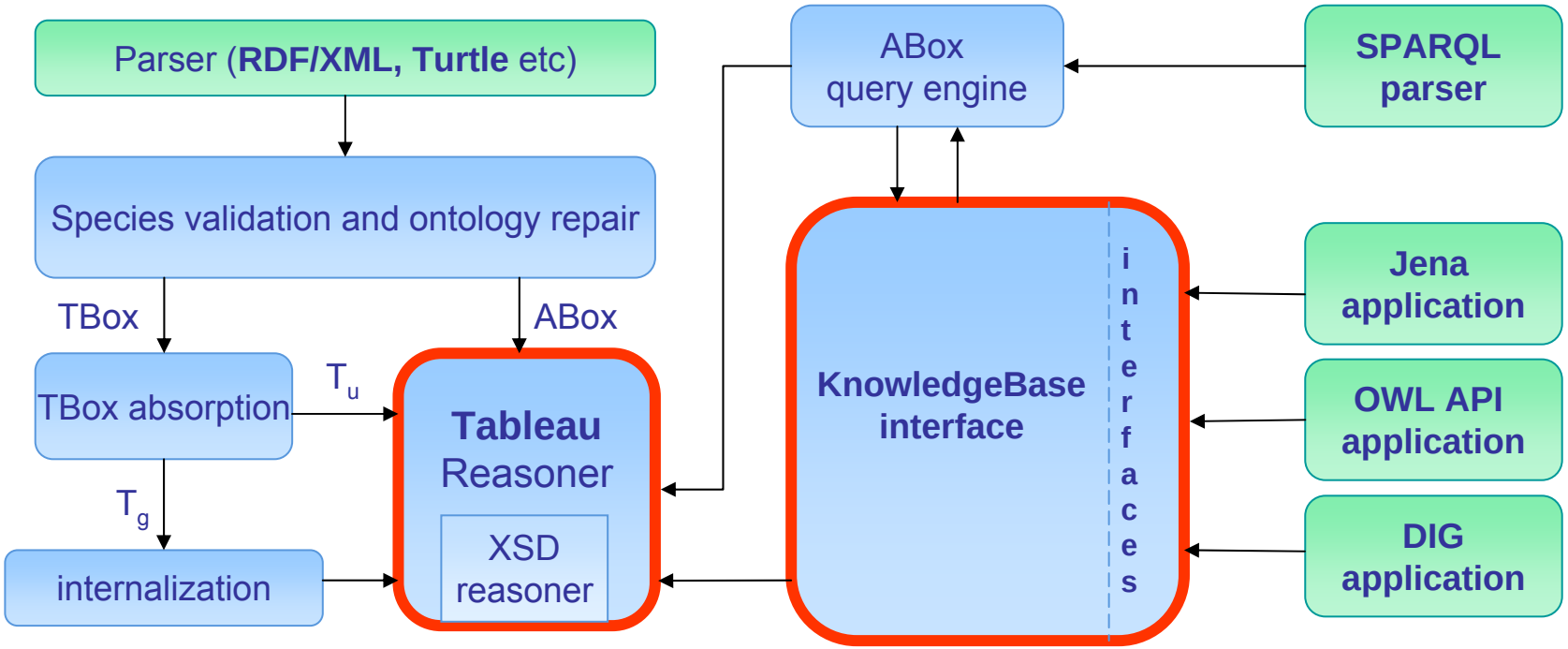
3rd year student is…

```
<TClass ID="Entrant"/>
<TClass rdf:ID="Student">
 <rdfs:subClassOf>
  <TRestriction>
    <past rdf:datatype=
          "&xsd;#NonNegativeInteger">
          1 </past>
      <equivalentClass>
       <TClass rdf:about="#Entrant"/>
      </equivalentClass>
  </TRestriction>
 </rdfs:subClassOf>
</TClass>
```

Sources of Reasoning Support for OWL-Met

Engine	OWL Support	Status
KAON2	incomplete OWL DL	Freeware
FaCT++	OWL DL	Opensource
RacerPro	OWL DL	Commercial
Pellet	OWL DL	Opensource
Jena	incomplete OWL DL	Opensource

Changes in Pellet



Changes in Jena

- added file **owlmet.owl** to Jena
- owlmet:**TClass** is subClassOf **rdf:Class**
- owlmet:**TRestriction** is subClassOf **TClass**
- owl:**Class** is subClassOf **owlmet:TClass**
- owlmet:Instant is subClassOf **owlmet:TClass**

- Redifined properties like "**equivalentClass**", "**disjointWith**" to operate on TClasses

- Added properties for "**allfuture**"/"**somefuture**"/"**future n**"
- Added properties for "**at**" (rdfs required also to add property "**happens**")

Back to Evolution Analysis

- **Reasoning queries**

e.g.

(*C intersection* ((*past 2*) *not C*)) @{*v5*}

meaning:

“What are the new individuals of concept *C* in a version *v5*, which were not present two versions before?”

Back to Evolution Analysis

- **Meta-level queries**

Given version i, ontology Oi, concept Gi –intersection of the definitions of all concepts and individuals in Oi.

Then

|— Gi @ {i} – checking SAT for Oi

|— Gi @ {j} – checking SAT for Oi in j

|— GE,i @ {i} – checking SAT for concept E (from i) in version i

|— GE,i @ {j} – checking SAT for concept E (from i) in version j

|— (Gi intersection Gj) @ {j}

GE,i - compiled [Stuckenschmidt & Klein 2003] concept from all explicit and implicit definitions of E in version i

Back to Evolution Analysis

- **Retrieval queries**

e.g.

Child (C,B)@{j} intersection (not Child(C,B)@{i})

meaning

“Get new children *B* of concept *C* appeared in the version j as compare to the version i”

Might require new roles/role restrictions to be introduced

Future Work

- Real cases (propositions are welcome)
- Optimization
- Combination of TimeStructure concept with an ontology of temporal aggregates (years, monthes, days,….) – e.g. with OWL-Time [Hobbs&Pan 2004]
- Fusion (decidability in mind) between pure DL and temporal parts – like roles/role restrictions over TClasses

Additional info

http://ermolayev.com/owl-met/

Shall be happy to answer your questions