Objects as results from graph queries using an ORM and generated semantic-relational binding

```python
class Content(SQLAlchemyBase):
    __tablename__ = "content"
    __mapper_args__ = {
        'polymorphic_identity': 'sioc:Item',
        'polymorphic_on': 'sqla_type'
    }

    id = Column(Integer, primary_key=True)

    sqla_type = Column(String(60), nullable=False)

    iri_class = PatternIriClass(EG.Content_iri,
        'http://%{WSHostName}U/data/Content#%d', None,
        ('id', Integer, False))

    title = Column(UnicodeText,
        info={'rdf': QuadMapPattern(None, DCTERMS.title)})
    definition = Column(UnicodeText,
        info={'rdf': QuadMapPattern(None, DCTERMS.description)})

    @classmethod
    def special_quad_patterns(cls, alias_manager):
        return [QuadMapPatternS(None, RDF.type,
            IriClass(VirtRDF.QNAME_ID).apply(cls.sqla_type),
            name=EG.class_Content_class)]
```

A

F

```python
class Idea(Content):
    __tablename__ = "idea"
    __mapper_args__ = {
     'polymorphic_identity': 'idea:GenericIdeaNode’}

    id = Column(Integer, ForeignKey('content.id'), primary_key=True)

    parent_id = Column(Integer, ForeignKey('idea.id'),
        info={'rdf': QuadMapPattern(
            None, IDEA.includes,
       Content.iri_class.apply('Idea.parent_id'))})

class Excerpt(SQLAlchemyBase):
    __tablename__ = 'excerpt'
    iri_class = PatternIriClass(EG.Excerpt_iri,
        'http://%{WSHostName}U/data/Excerpt#%d', None,
        ('id', Integer, False))
    rdf_class = CATALYST.Excerpt

    id = Column(Integer, primary_key=True)

    target = Column(IRI_ID,
        info={'rdf': QuadMapPattern(
            None, OA.hasTarget,
       VirtuosoAbstractFunction(VirtRDF.id_to_iri
         ).apply('Excerpt.target'))})

    idea_id = Column(Integer, ForeignKey(Idea.id), nullable=True,
        info={'rdf': QuadMapPattern(
            None, CATALYST.expressesIdea,
       Idea.iri_class.apply('Excerpt.idea_id'),
            condition=Idea.title != None)})
```

I

C

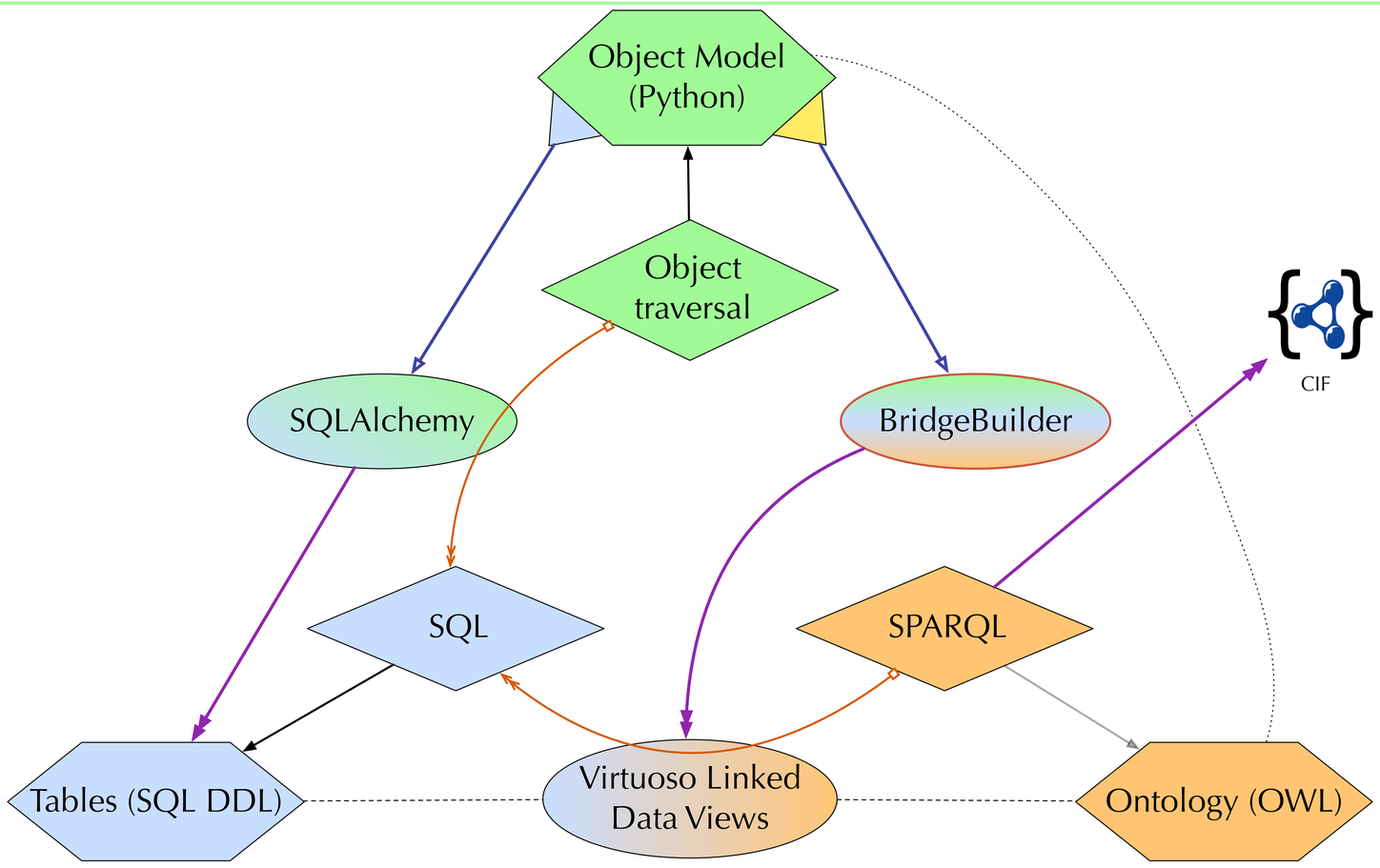
specify
generate
query
convert
link

Attributes

Inheritance

Function calls

Conditions




```
CREATE TABLE content (
   id INTEGER NOT NULL IDENTITY,
   sqla_type VARCHAR(60) NOT NULL,
   title LONG NVARCHAR NULL,
   definition LONG NVARCHAR NULL,
   PRIMARY KEY (id)
)

CREATE TABLE idea (
   id INTEGER NOT NULL,
   parent_id INTEGER NULL,
   PRIMARY KEY (id),
   FOREIGN KEY(id) REFERENCES content (id),
   FOREIGN KEY(parent_id) REFERENCES idea (id)
)

CREATE TABLE "excerpt" (
   id INTEGER NOT NULL IDENTITY,
   target IRI_ID NULL,
   idea_id INTEGER NULL,
   PRIMARY KEY (id),
   FOREIGN KEY(idea_id) REFERENCES idea (id)
)
```

I

```
create iri class eg:Content_iri "http://%{WSHostName}U/data/Content#%d" (
  in id INTEGER NOT NULL) .
create iri class eg:Excerpt_iri "http://%{WSHostName}U/data/Excerpt#%d" (
  in id INTEGER NOT NULL) .
create quad storage eg:quad_storage
FROM "eg.DBA".content AS alias_content_0
FROM "eg.DBA".excerpt AS alias_excerpt_0
FROM "eg.DBA".idea AS alias_idea_0
FROM "eg.DBA".content AS alias_content_1
FROM "eg.DBA".excerpt AS alias_excerpt_1
WHERE (^{alias_content_1.}^.id = ^{alias_idea_1.}^.id
     AND ^{alias_content_1.}^.title IS NOT NULL) {

    graph eg:graph_qmp {

        eg:Content_iri (alias_content_0.id)
         rdf:type virtrdf:QNAME_ID (alias_content_0.sqla_type)
              as eg:class_Content_class;
         dcterms:description alias_content_0.definition
              as eg:col_pattern_Content_definition;
         dcterms:title alias_content_0.title
              as eg:col_pattern_Content_title.

        eg:Content_iri (alias_idea_0.id)  idea:includ
         eg:Content_iri (alias_idea_0.parent_id)
              as eg:col_pattern_Idea_parent_id.

        eg:Excerpt_iri (alias_excerpt_0.id)  oa:target
         virtrdf:id_to_iri (alias_excerpt_0.target)
              as eg:col_pattern_excerpt_target.

        eg:Excerpt_iri (alias_excerpt_1.id) catalyst:expressesIdea
         rdf:type catalyst:Excerpt as eg:class_Excerpt_class;
         eg:Content_iri (alias_excerpt_1.idea_id)
         option(using alias_content_1, using alias_idea_1)
              as eg:col_pattern_Excerpt_idea_id.
    }.
    create virtrdf:DefaultQuadMap using storage eg:quad_storage.
}
```

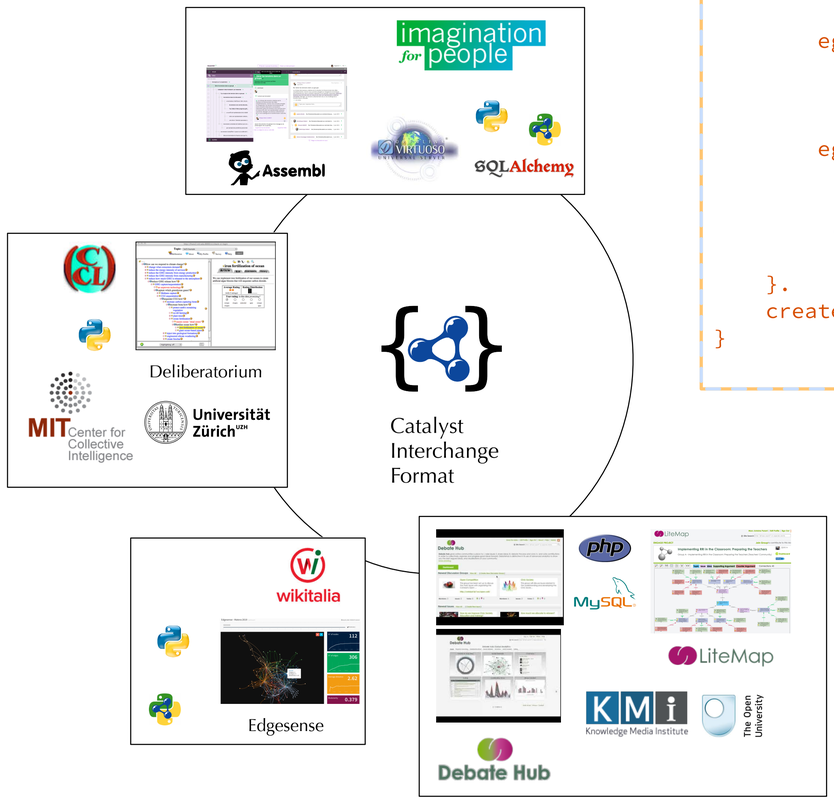
C

F

A

```
@prefix sioc: <http://rdfs.org/sioc/ns#> .
@prefix oa: <http://www.openannotation.org/ns/> .
@prefix idea: <http://purl.org/catalyst/idea#> .
@prefix catalyst: <http://purl.org/catalyst/core#> .
@prefix eg: <http://example.com/> .

sioc:Item a owl:Class.

oa:Annotation a owl:Class.

oa:hasTarget a owl:ObjectProperty;
  rdfs:domain oa:Annotation.

catalyst:Idea owl:subClassOf idea:GenericIdeaNode,
    sioc:Item.

idea:includes
  a owl:AsymmetricProperty;
  rdfs:range idea:GenericIdeaNode;
  rdfs:domain idea:GenericIdeaNode.

catalyst:Excerpt owl:subClassOf oa:Annotation.

catalyst:expressesIdea a owl:ObjectProperty ;
  rdfs:domain catalyst:Excerpt;
  rdfs:range catalyst:Idea.
```

or: 3 models, 1 source of truth…
means a lot of code generation!



CATALYST			
SIOC	OA	IBIS	
DC	FOAF	IDEA	AIF

Catalyst Ontologies

http://github.com/maparent/virtuoso-python

Marc-Antoine Parent
maparent@acm.org

Catalyst
catalyst-fp7.eu