

Analisi e Sviluppo di Tecniche di Inferenza per Schemi ed Istanze nei Database ad Oggetti

Relatore

Chiar.ma Prof.ssa Sonia Bergamaschi

Tesi di Laurea di

Enrico Fais

Correlatore

Chiar.mo Prof. Domenico Beneventano

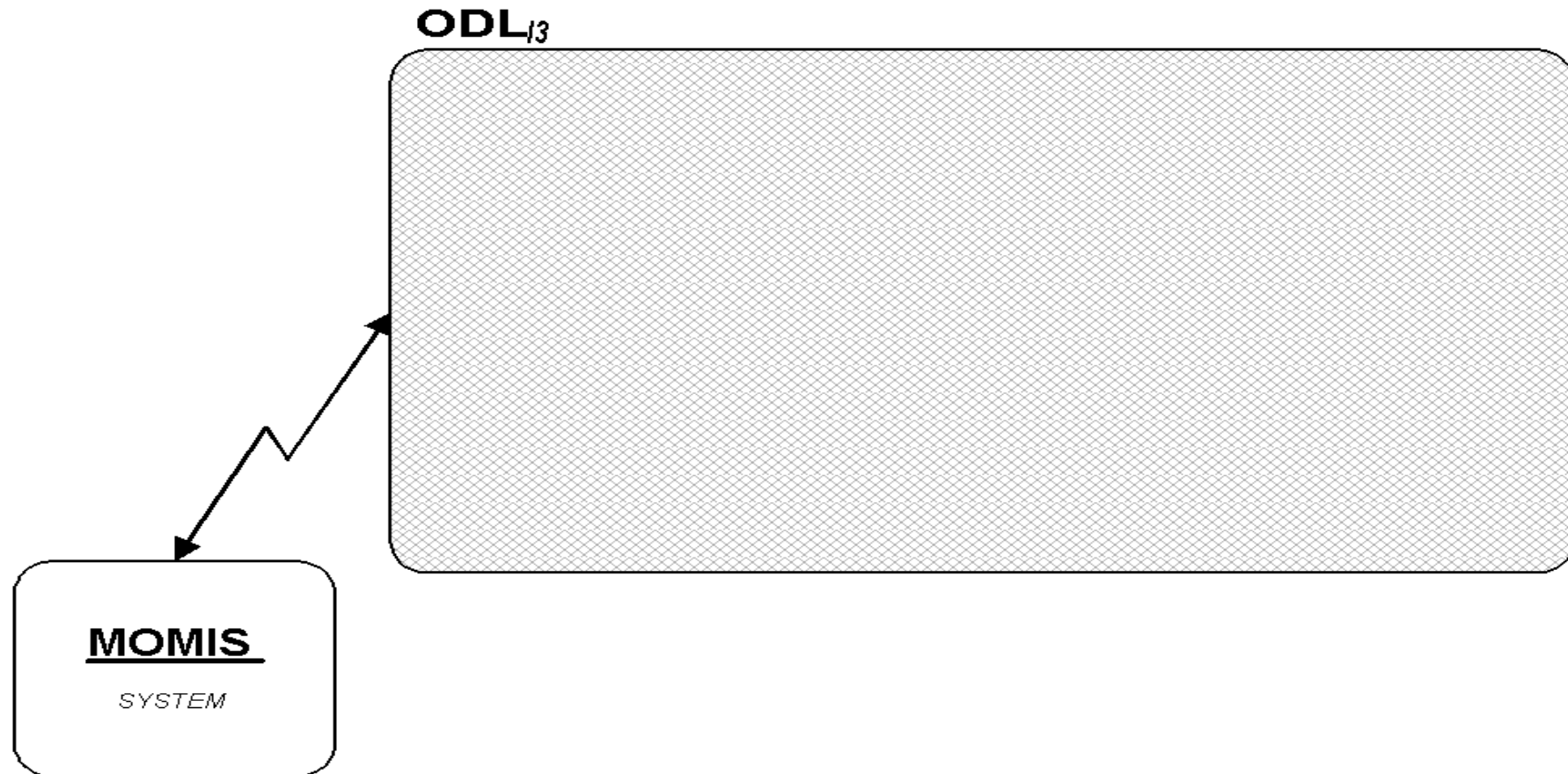
Controrelatore

Dott.ssa Federica Mandreoli

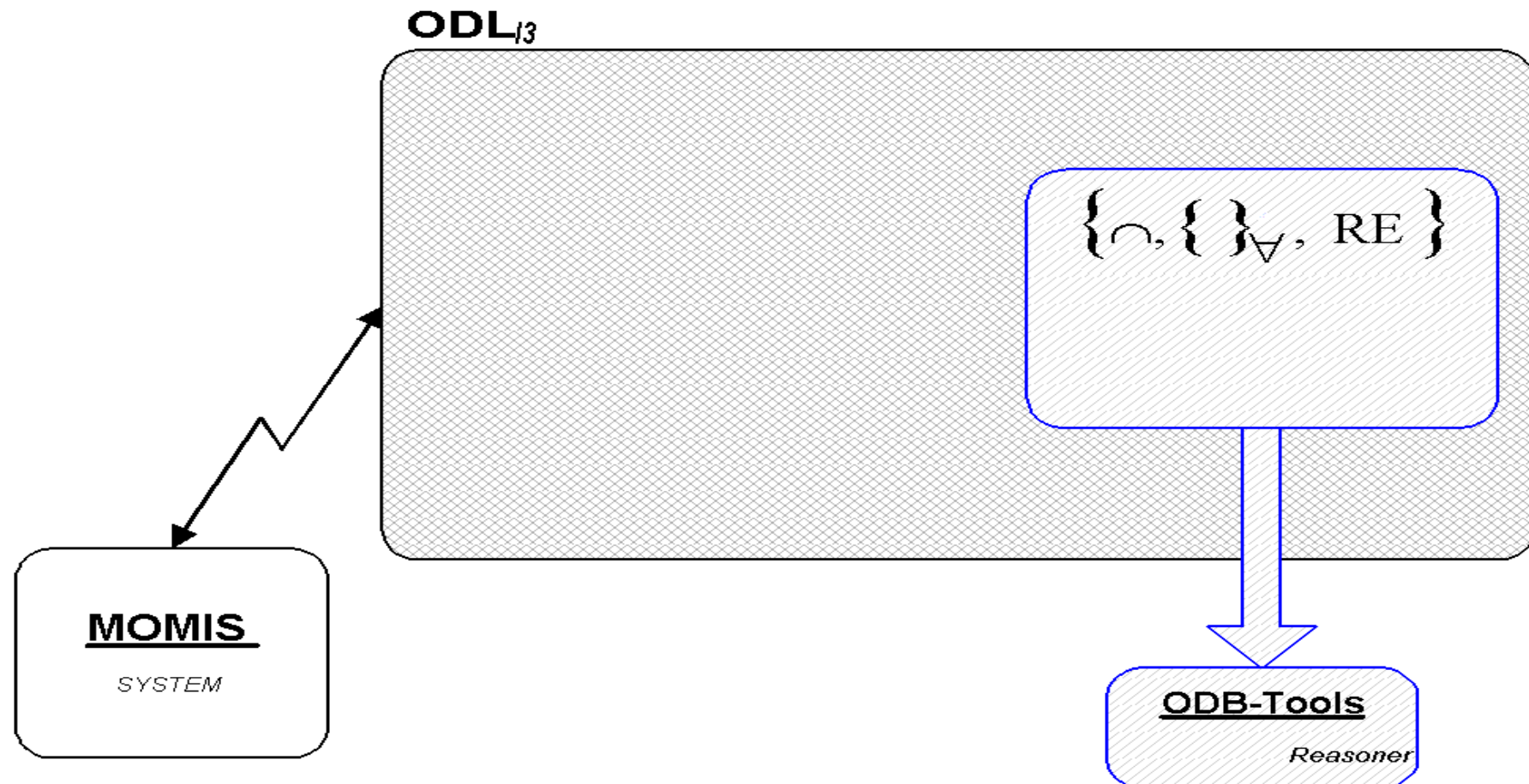
- Sistemi Intelligenti di Basi di Dati;
 - ODB-Tools:
 - semantica dei OODMs (Object Oriented Data Models);
 - controllo consistenza schemi;
 - ottimizzazione semantica delle interrogazioni;

- Sistemi Intelligenti di Basi di Dati;
 - ODB-Tools:
 - semantica dei OODMs (Object Oriented Data Models);
 - controllo consistenza schemi;
 - ottimizzazione semantica delle interrogazioni;
- Integrazione Intelligente delle Informazioni;
 - MOMIS:
 - integrazione di informazioni provenienti da sorgenti eterogenee distribuite di dati strutturati e semi-strutturati.

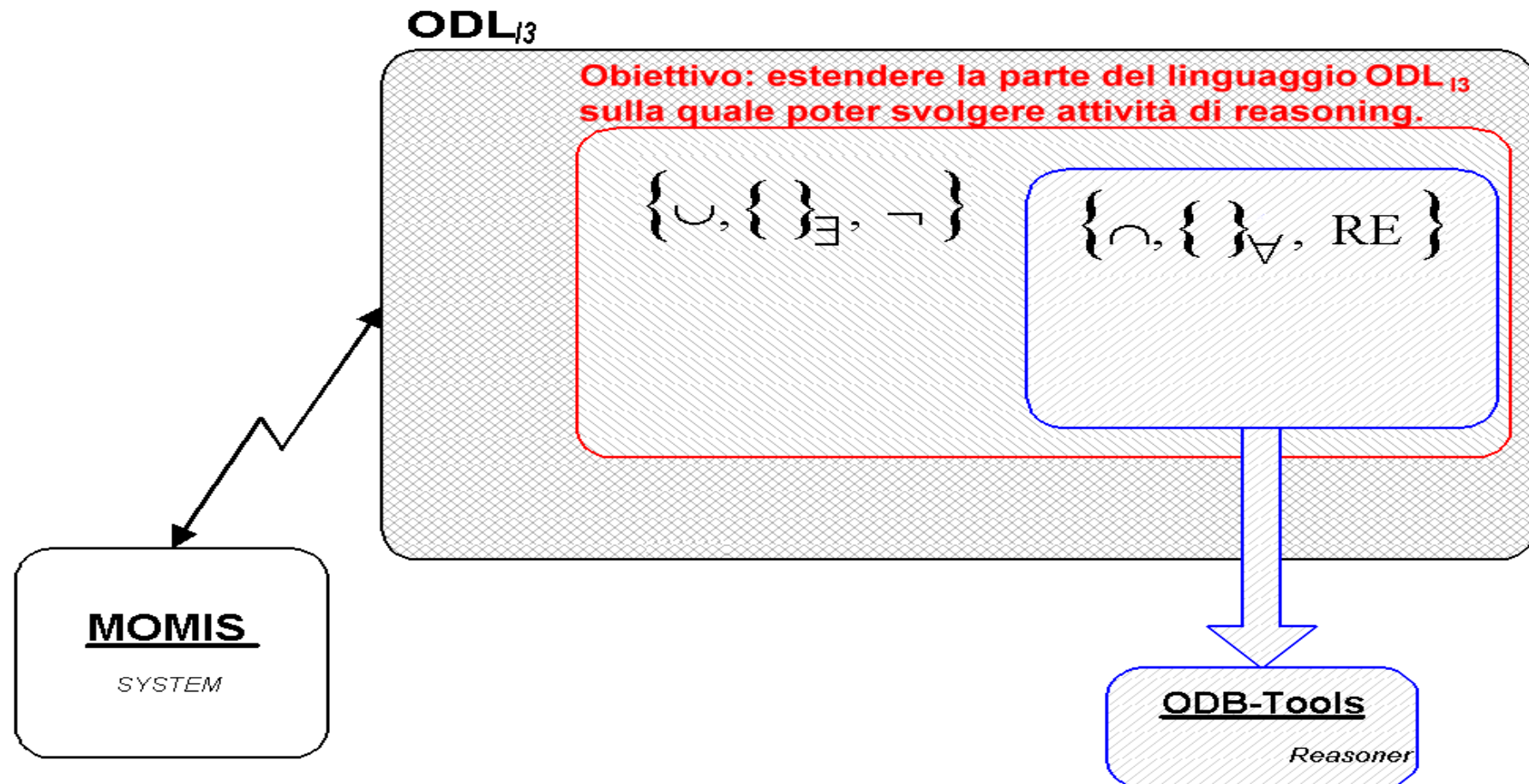
Scenario ed obiettivi



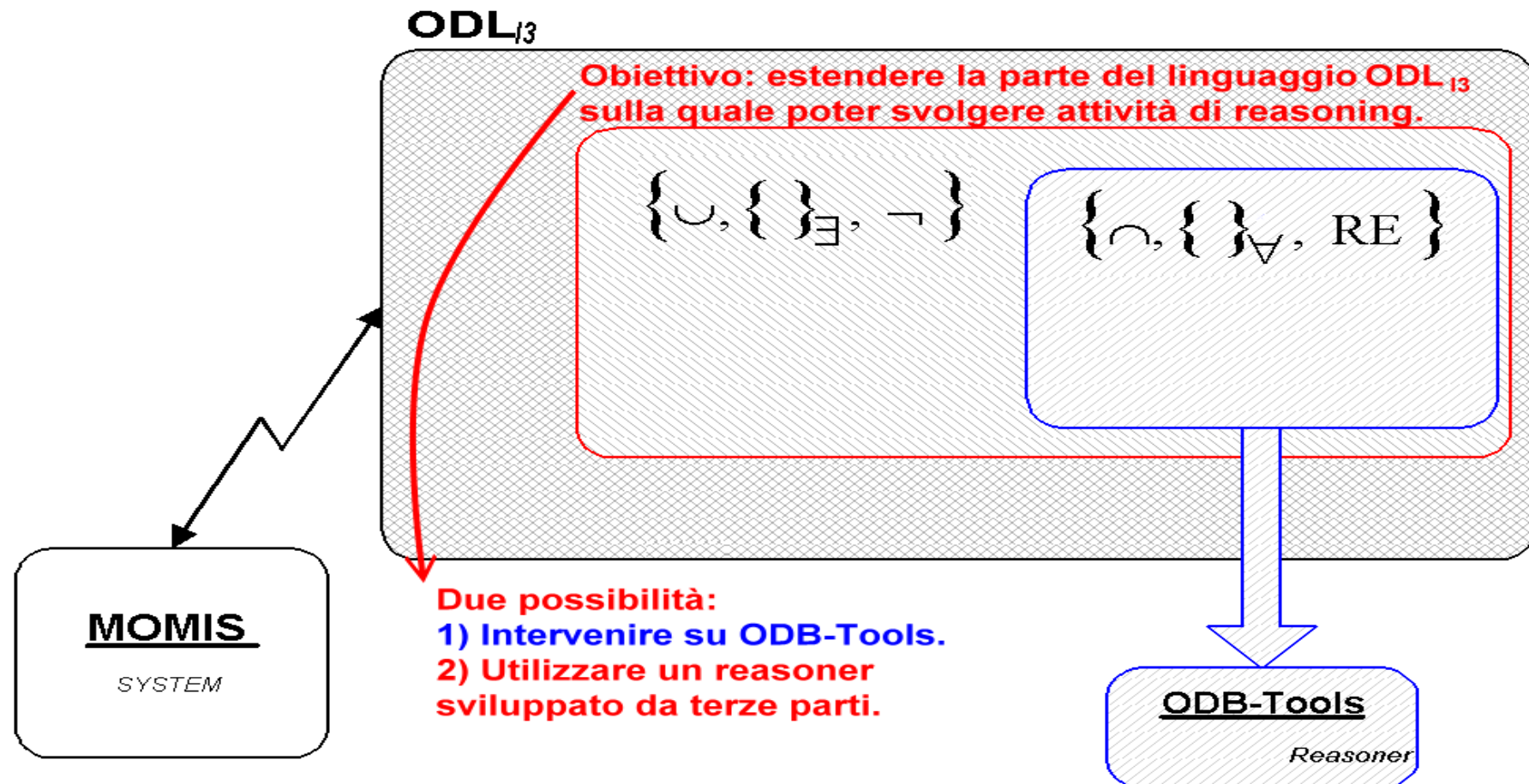
Scenario ed obiettivi



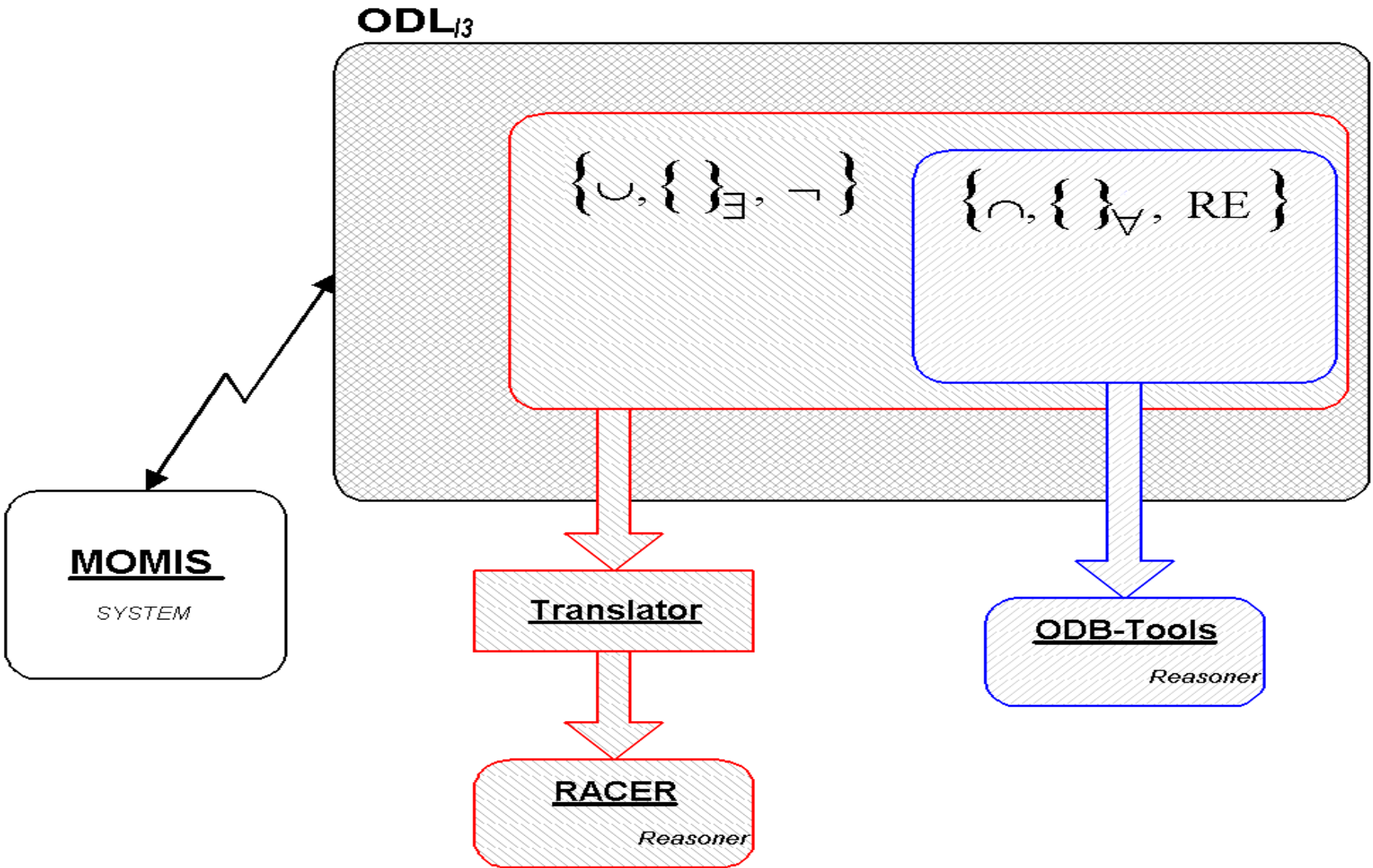
Scenario ed obiettivi



Scenario ed obiettivi



Scenario ed obiettivi



RACER (Renamed ABox and Concept Expression Reasoner)

- DL altamente espressiva $ALCQHI_{\mathcal{R}^+}(D)^-$:
 - unione (\sqcup), congiunzione (\sqcap), complemento (\neg);
 - domini concreti (ragionamento algebrico);
 - descrizioni cicliche;
- Conoscenza intensionale (TBox) ed conoscenza estensionale (ABox);
- Open World Assumption (OWA);

ODB-Tools: la logica $OD\mathcal{L}$ per OODM

- $OD\mathcal{L}$ (Object Description Logic):
 - congiunzione (\sqcap)(ereditarietà multipla);
 - domini concreti;
 - descrizioni cicliche;
 - Valori complessi :
 - distinzione tra oggetti e valori;
 - tuple e set;
- Conoscenza intensionale(TBox);
- Closed World Assumption (CWA);

ODB-Tools: la logica ODL per OODM

■ ODL (Object Description Logic):

- congiunzione (\sqcap)(ereditarietà multipla);
- domini concreti;
- descrizioni cicliche;

● Valori complessi :

- distinzione tra oggetti e valori;
- tuple e set;

■ Conoscenza intensionale(TBox);

■ Closed World Assumption (CWA);

Principali forme di ragionamento

■ Subsumption:

- Verifica che una classe **C1** sia più generale di un'altra classe **C2**
- Classificazione.

■ Satisfiability:

- Una classe **C1** è soddisfacibile se ammette almeno un oggetto.
- Riconducibile alla “non subsumption”.

Concetti fondamentali:

- *AbstractClass*: concetto padre di tutte le classi;
- *RecType*: concetto padre di tutti i tipi tupla;
- *SetType*: concetto padre di tutti i tipi insieme;

Ruoli fondamentali:

- *value*: modella l'associazione tra l'object identifier (oid) e la sua descrizione;
- *member*: permette di specificare gli elementi di un tipo set.

Esempio di traduzione

■ Linguaggio **ODL**_{I3}:

```
interface Office      {attribute String address;  
                      attribute String telephone;  
                      attribute set<Employee> employs;};
```

Esempio di traduzione

■ Linguaggio ODL_{I3} :

interface Office

```
{attribute String address;  
attribute String telephone;  
attribute set<Employee> employs;};
```

■ Formalismo \mathcal{AL} :

Office

\Rightarrow **AbstractClass** \sqcap
(\forall *value*.(**RecType** \sqcap
(= 1 address) \sqcap (\forall address.*STRING*) \sqcap
(= 1 telephone) \sqcap (\forall telephone.*STRING*) \sqcap
(= 1 employs) \sqcap (\forall employs.(**SetType** \sqcap
(\forall *member*.Employee))))))

Esempio di traduzione

■ Linguaggio ODL_{I3} :

interface Office

```
{attribute String address;  
attribute String telephone;  
attribute set<Employee> employs;};
```

■ Formalismo \mathcal{AL} :

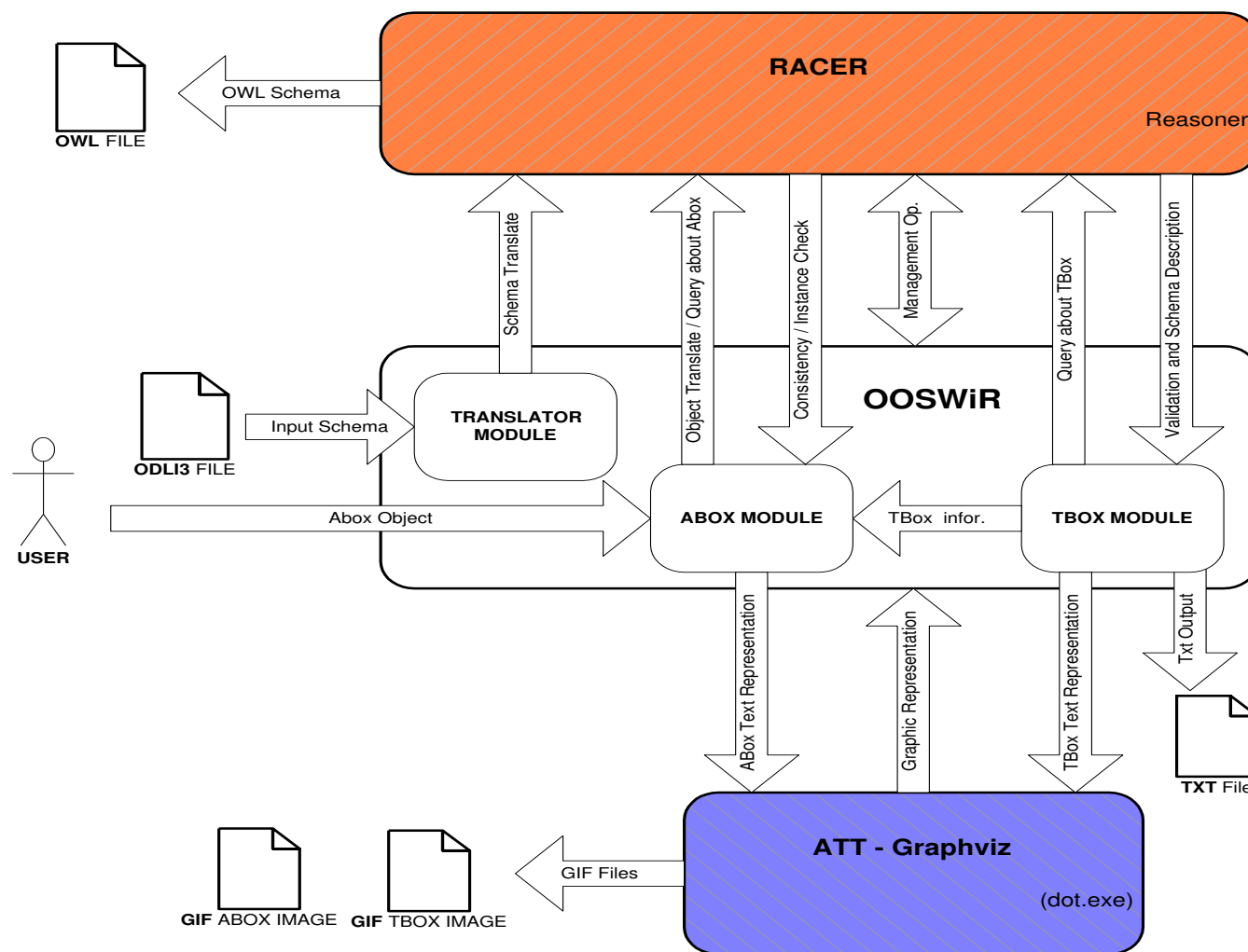
Office

\Rightarrow **AbstractClass** \sqcap
($\forall value.$ (**RecType** \sqcap
(= 1 address) \sqcap ($\forall address.$ *STRING*) \sqcap
(= 1 telephone) \sqcap ($\forall telephone.$ *STRING*) \sqcap
(= 1 employs) \sqcap ($\forall employs.$ (**SetType** \sqcap
($\forall member.$ Employee))))))

+ assiomi di definizione della “struttura di base”.

Architettura di OOSWiR

OOSWiR (Object Oriented Schema With Racer)



Schema ODL_{I^3} : “FACTORY”

(*Tipi Tupla*)

```
typedef LEVEL {  
  attribute RANGE{1,10} parameter;  
  attribute String qualification; };
```

```
typedef TOPLEVEL: LEVEL {  
  attribute RANGE{8,10} parameter;};
```

```
typedef DENOMINATION {  
  attribute String name;  
  attribute String address; };
```

Schema ODL_{I^3} : “FACTORY”

(*Tipi Classe*)

```
interface EMPLOYEE {  
    attribute String name;  
    attribute Integer salary *; };
```

```
interface MANAGER: EMPLOYEE {  
    attribute LEVEL level; };
```

```
interface MATERIAL {  
    attribute String name;  
    attribute RANGE{1,20} risk; };
```

```
interface FLUID: MATERIAL {  
    attribute Real viscosity; };
```

```
interface SOLID: MATERIAL, NOT FLUID {};
```

Schema ODL_{I^3} : “FACTORY”

(Tipi Classe con Inconsistenza)

```
interface REPOSITORY{  
  attribute UNION<String, DENOMINATION> denomination;  
  attribute SET<MATERIAL> stock; };
```

```
interface DEPARTEMENT{  
  attribute String denomination;  
  attribute MANAGER managedby; };
```

```
interface WAREHOUSE:DEPARTEMENT,REPOSITORY{};
```

```
interface AWAREHOUSE:WAREHOUSE{  
  attribute SET<FLUID> stock; };
```

```
interface BWAREHOUSE:AWAREHOUSE{  
  attribute EXISTS<SOLID> stock; };
```

Schema ODL_{I^3} : “FACTORY”

(Tipi Classe con Inconsistenza)

```
interface REPOSITORY{  
  attribute UNION<String, DENOMINATION> denomination;  
  attribute SET<MATERIAL> stock; };
```

```
interface DEPARTEMENT{  
  attribute String denomination;  
  attribute MANAGER managedby; };
```

```
interface WAREHOUSE:DEPARTEMENT,REPOSITORY{};
```

```
interface AWAREHOUSE:WAREHOUSE{  
  attribute SET<FLUID> stock; };
```

```
interface BWAREHOUSE:AWAREHOUSE{  
  attribute EXISTS<SOLID> stock; };
```

Schema ODL_{I^3} : “FACTORY” (Viste)

```
view DANGERMATERIAL: MATERIAL{  
attribute RANGE{15,20} risk; };
```

```
view OIL: FLUID{  
attribute RANGE{4.0,8.0} viscosity; };
```

```
view TOPMANAGER: MANAGER{  
attribute TOPLEVEL level; };
```

Conoscenza estensionale: oggetti e valori complessi in RACER

- Simulare la CWA in RACER:
 - specificare quali attributi sono definiti per i tipi tupla.
 - specificare il numero di riempitivi per i tipi set.
- Verificare la consistenza degli oggetti inseriti.
- Fornire “un’interpretazione grafica” alla conoscenza estensionale.
- Sfruttare le capacità di classificazione di RACER per catalogare gli oggetti inseriti.

Schema di Esempio: "FACTORY"

(Oggetti)

OID1(MANAGER) = [name:"PETER", salary:44000, level: [parameter:9, qualification:"MARKETING"]];

OID2(MANAGER) = [name:"LIZ", salary:40000, level: [parameter:5, qualification:"ADMINISTRATION"]];

OID3(REPOSITORY) = [denomination: [name:"METALLURGY", address:"IRON STREET"], stock:{**OID4**, **OID5**}];

OID4(SOLID) = [name:"IRON", risk:1];

OID5(SOLID) = [name:"STEEL", risk:2];

OID6(WAREHOUSE) = [denomination:"GOLD-WAREHOUSE", managedby:**OID2**, stock:{**OID8**}];

OID7(SOLID) = [name:"GOLD", risk:16];

OID8(FLUID) = [name:"OIL-API5", risk:1, viscosity:4.02];

OID9(EMPLOYEE) = [name:"PAUL", salary:3300.33];

Schema di Esempio: "FACTORY"

(Oggetti)

OID1(MANAGER) = [name:"PETER", salary:44000, level: [parameter:9, qualification:"MARKETING"]];

OID2(MANAGER) = [name:"LIZ", salary:40000, level: [parameter:5, qualification:"ADMINISTRATION"]];

OID3(REPOSITORY) = [denomination: [name:"METALLURGY", address:"IRON STREET"], stock:{**OID4**, **OID5**}];

OID4(SOLID) = [name:"IRON", risk:1];

OID5(SOLID) = [name:"STEEL", risk:2];

OID6(WAREHOUSE) = [denomination:"GOLD-WAREHOUSE", managedby:**OID2**, stock:{**OID8**}];

OID7(SOLID) = [name:"GOLD", risk:16];

OID8(FLUID) = [name:"OIL-API5", risk:1, viscosity:4.02];

OID9(EMPLOYEE) = [name:"PAUL", salary:3300.33];

Schema di Esempio: "FACTORY"

(Oggetti)

OID1(MANAGER) = [name:"PETER", salary:44000, level: [parameter:9, qualification:"MARKETING"]];

OID2(MANAGER) = [name:"LIZ", salary:40000, level: [parameter:5, qualification:"ADMINISTRATION"]];

OID3(REPOSITORY) = [denomination: [name:"METALLURGY", address:"IRON STREET"], stock:{**OID4**, **OID5**}];

OID4(SOLID) = [name:"IRON", risk:1];

OID5(SOLID) = [name:"STEEL", risk:2];

OID6(WAREHOUSE) = [denomination:"GOLD-WAREHOUSE", managedby:**OID2**, stock:{**OID8**}];

OID7(SOLID) = [name:"GOLD", risk:16];

OID8(FLUID) = [name:"OIL-API5", risk:1, viscosity:4.02];

OID9(EMPLOYEE) = [INCONSISTENTE];

- Problema della traduzione OODM-DL:
 - mapping **ODL**_{I³}-RACER;
 - estesa la parte del linguaggio sulla quale poter svolgere attività inferenziali;

- Problema della traduzione OODM-DL:
 - mapping **ODL**_{I³}-RACER;
 - estesa la parte del linguaggio sulla quale poter svolgere attività inferenziali;
- Sviluppo prototipo:
 - calcolo della sussunzione tra i tipi;
 - consistenza e minimalità di schemi;
 - riconoscimento delle istanze;

- Problema della traduzione OODM-DL:
 - mapping **ODL**_{I³}-RACER;
 - estesa la parte del linguaggio sulla quale poter svolgere attività inferenziali;
- Sviluppo prototipo:
 - calcolo della sussunzione tra i tipi;
 - consistenza e minimalità di schemi;
 - riconoscimento delle istanze;
- Il prototipo è potenzialmente integrabile in MOMIS:
 - evoluzione nell'ambito del progetto SEWASIE;