Towards the Industrial Application of NMR Systems

Nicola Leone

Department of Mathematics

University of Calabria

leone@unical.it

Making the point on NMR

- Huge amount of scientific results
- Expressive NMR languages
- A number efficient systems
- Advanced academic applications
- Strong influence on the evolution of AI
- Application in industry is quite weak
  - Showing industrial usefulness is needed for NMR future!

The industrialization effort of the DLV team at University of Calabria: tools, applications, spin-off, lessons learned

Roadmap

- The ASP System DLV
- ASP Development Tools
- Some Industry Level Applications
  - Workforce Management in the Gioia Tauro Sea-Port
  - The IDUM e-Tourism System
  - Ontology-based Data Cleaning in a Medical Environment
  - Intelligent Data Extraction
  - Autonomous Agents in Business Simulations Games
- Spin-off Companies
- Lessons Learned and Conclusion

The ASP System DLV

Main Features

Advanced Knowledge Modeling Capabilities (1)

- Language:
  - Disjunctive Logic Programs under Stable Model Semantics
  - Extension with aggregates, weak constraints, functions, lists, sets...
  - Existential Quantifiers (Attend the KR talk on "**Efficiently Computable Datalog^E Programs" on Wednesday !)**

- High Expressiveness:
  - Captures Σ^P_2 (NP^{NP})
  - Able to represent complex problems not (polynomially) translatable to SAT or CSP

Advanced Knowledge Modeling Capabilities (2)

- Full Declarativeness:
  - Rules ordering and goals ordering is immaterial
  - Computation is sound and complete
  - Termination is always guaranteed

- Front ends for AI applications
  - Planning
  - Diagnosis
  - Ontology representation and reasoning
  - ...

Solid Implementation

- Database Optimization Techniques
  - Join Ordering Methods
  - Magic Sets
  - Indexing
- Search Optimization Methods:
  - Heuristics
  - Backjumping
  - Pruning Operators

Interoperability

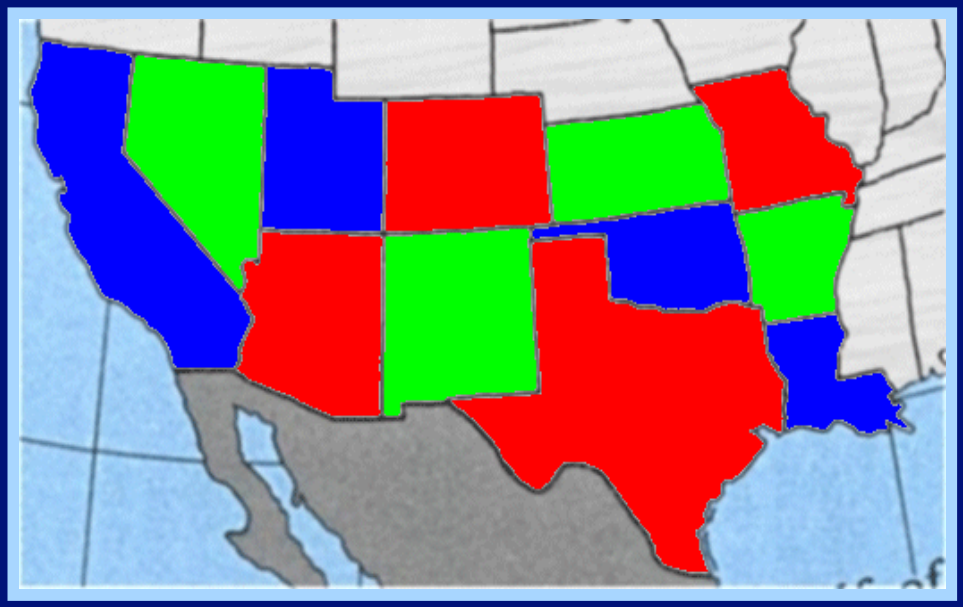
- DBMSs
  - Powerful reasoning on top of data stored in relational databases
- Semantic Web Reasoners
  - Integrate ontologies and rules
- C++ programs
  - Call C++ (application specific) functions from DLP programs
- JAVA
  - Integrate DLV with JAVA

A Flavour of DLV Language

3-colorability

Input: a Map represented by state(_) and border(_,_).

Problem: assign one color out of 3 colors to each state such that two neighbouring states have always different colors.



Solution:

col(X,red) ∨ col(X,green) ∨ col(X,blue) :-state(X). } Guess

:- border(X,Y), col(X,C), col(Y,C). } Check

Weak Constraints: *Exams Scheduling*

1. Assign course exams to 3 time slots avoiding overlapping of exams of courses with common students

r_1: assign(X,s1) v assign(X,s2) v assign(X,s3) :- course(X).

s_1: :- assign(X,S), assign(Y,S), commonStudents(X,Y,N), N>0.

2. If overlapping is unavoidable, then reduce it "As Much As Possible" – Find an approximate solution

r_2: assign(X,s1) v assign(X,s2) v assign(X,s3) :- course(X).

w_2: :~ assign(X,S), assign(Y,S), commonStudents(X,Y,N), N>0. [N:]

Scenarios (models) minimizing the total number of "lost" exams are preferred.

Aggregate Functions: *Team Building*

p_1 The team consists of a certain number of employees

p_2 At least a given number of different skills must be present in the team

p_3 The sum of the salaries of the employees working in the team must not exceed the given budget

p_4 The salary of each individual employee is within a specified limit

```
in(I) v out(I) :- emp(I,Sx,Sk,Sa).

:- nEmp(N), not #count{ I : in(I) } = N.

:- nSkill(M), not #count{ Sk : emp(I,Sx,Sk,Sa), in(I) }  ≥ M.

:- budget(B), not #sum{ Sa, I : emp(I,Sx,Sk,Sa), in(I) } ≤ B.

:- maxSal(M), not #max{ Sa : emp(I,Sx,Sk,Sa), in(I) } ≤ M.
```

Functions and Lists: *Simple Paths*

A *simple path* of a graph is a path without any node repetition.

A DLV program computing simple paths:

```
simplePath([X,Y])  :-  edge(X,Y),  X≠Y.
simplePath([X|W]) :-  edge(X,Y), simplePath(W),
                      #head(W,Y), not  #member(X,W).
```

alternative encoding:

```
simplePath([X|[Y|T]]) :- edge(X,Y), simplePath([Y|T]),
                         not  #member(X,[Y|T]).
```

Infinitely large Herbrand models, but stable models are finite

DLV computations are sound and complete here

ASP Development Tools

Motivation

- Lessons learned from Application Development:
  - Viability of the industrial exploitation of ASP
    - complex business-logic at a lower (implementation) price than usual languages
    - flexibility, readability, extensibility, ease of maintenance, etc.
  - Practical obstacles to ASP-based development:
    - ASP is not a full general-purpose language
      - Some components are better built with O.-O. Programming
      - ASP solutions must be embedded at some point
    - ASP systems are not supported by effective development tools

Need for Development Tools

- Application developer needs:
  - Support for development tools/workbenches
  - Seamless embedding of ASP
  - Integration in standard software processes/technologies
- Popular programming languages come with SDKs and IDEs
  - ① Tools for simplifying development and maintenance
    - programmers accustomed to Workbenches (e.g. eclipse,...)
    - graphic tools simplify the approach of novice users
  - ② Integrate different programming tools
- No Tool → usage of ASP may be discouraged

Development Tools for ASP

① Integrated Development Environment for ASP:

- ASPIDE

② A framework integrating ASP with Java

- The hybrid language JASP
- The JDLV plugin for Eclipse platform

ASPIDE

Integrated Development
Environment for ASP

Integrated Development Environment for ASP: ASPIDE

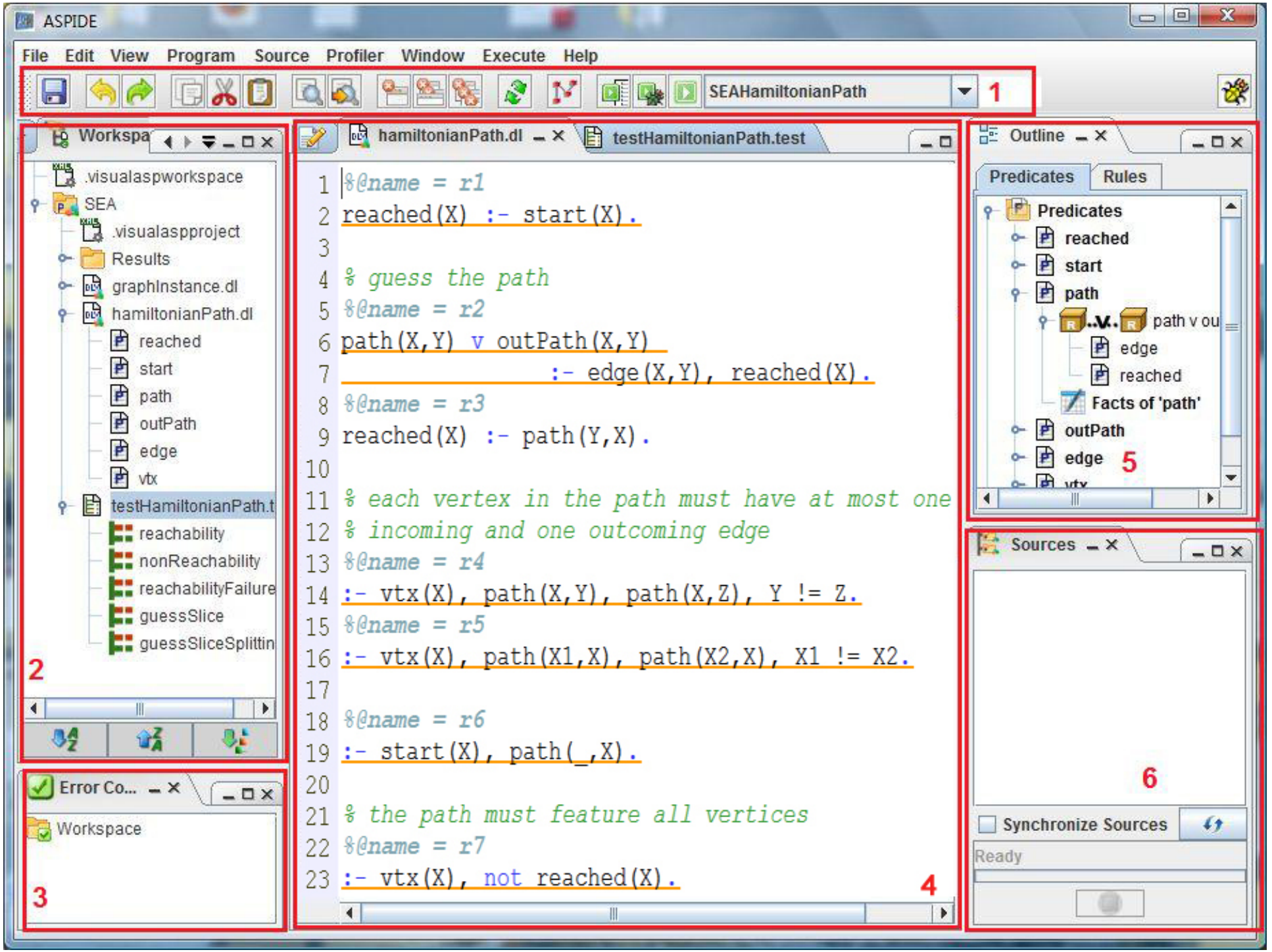
- Supports the entire life-cycle of ASP development
- Assisted composition of programs
- debugging
- profiling
- testing
- output-handling
- DBMS access
- execution configuration

ASPIDE Features (1)

- Advanced Program Editing
  - Text Editor
    - Text coloring, automatic completion, refactoring, code templates, etc.
  - Visual Editor with Reverse-engineering
    - Drawing logic programs in a QBE-like style
  - Outline navigation
- Run Configuration and Presentation of Results
  - Setup execution & friendly browsing of results
- Debugger and Profiler
  - Spock, DLV Profiler

ASPIDE Features (2)

- Unit Testing for ASP
  - a framework in the style of JUnit
- Annotations for ASP programs
  - rule names, predicate schemas, etc.
- User-defined Plugins
  - SDK free for download
- Database Access
  - Connect to DBMSs via JDBC, DLV^{DB}
  - TYP files, ODBC import/export
  - Support for Data Integration


ASPIDE
File Edit View Program Source Profiler Window Execute Help
SEAHamiltonianPath
1
Workspa
.visualaspworkspace
SEA
.visualaspproject
Results
graphInstance.dl
hamiltonianPath.dl
reached
start
path
outPath
edge
vtx
testHamiltonianPath.t
reachability
nonReachability
reachabilityFailure
guessSlice
guessSliceSplittin
2
Error Co...
Workspace
3
hamiltonianPath.dl
testHamiltonianPath.test
1 %@name = r1
2 reached(X) :- start(X).
3
4 % guess the path
5 %@name = r2
6 path(X,Y) v outPath(X,Y)
7 :- edge(X,Y), reached(X).
8 %@name = r3
9 reached(X) :- path(Y,X).
10
11 % each vertex in the path must have at most one
12 % incoming and one outcoming edge
13 %@name = r4
14 :- vtx(X), path(X,Y), path(X,Z), Y != Z.
15 %@name = r5
16 :- vtx(X), path(X1,X), path(X2,X), X1 != X2.
17
18 %@name = r6
19 :- start(X), path(_,X).
20
21 % the path must feature all vertices
22 %@name = r7
23 :- vtx(X), not reached(X).
4
Outline
Predicates
Rules
Predicates
reached
start
path
path v ou
edge
reached
Facts of 'path'
outPath
edge
5
Sources
6
Synchronize Sources
Ready

Integrating ASP with Java

The JASP Language

The JDLV Eclipse plugin

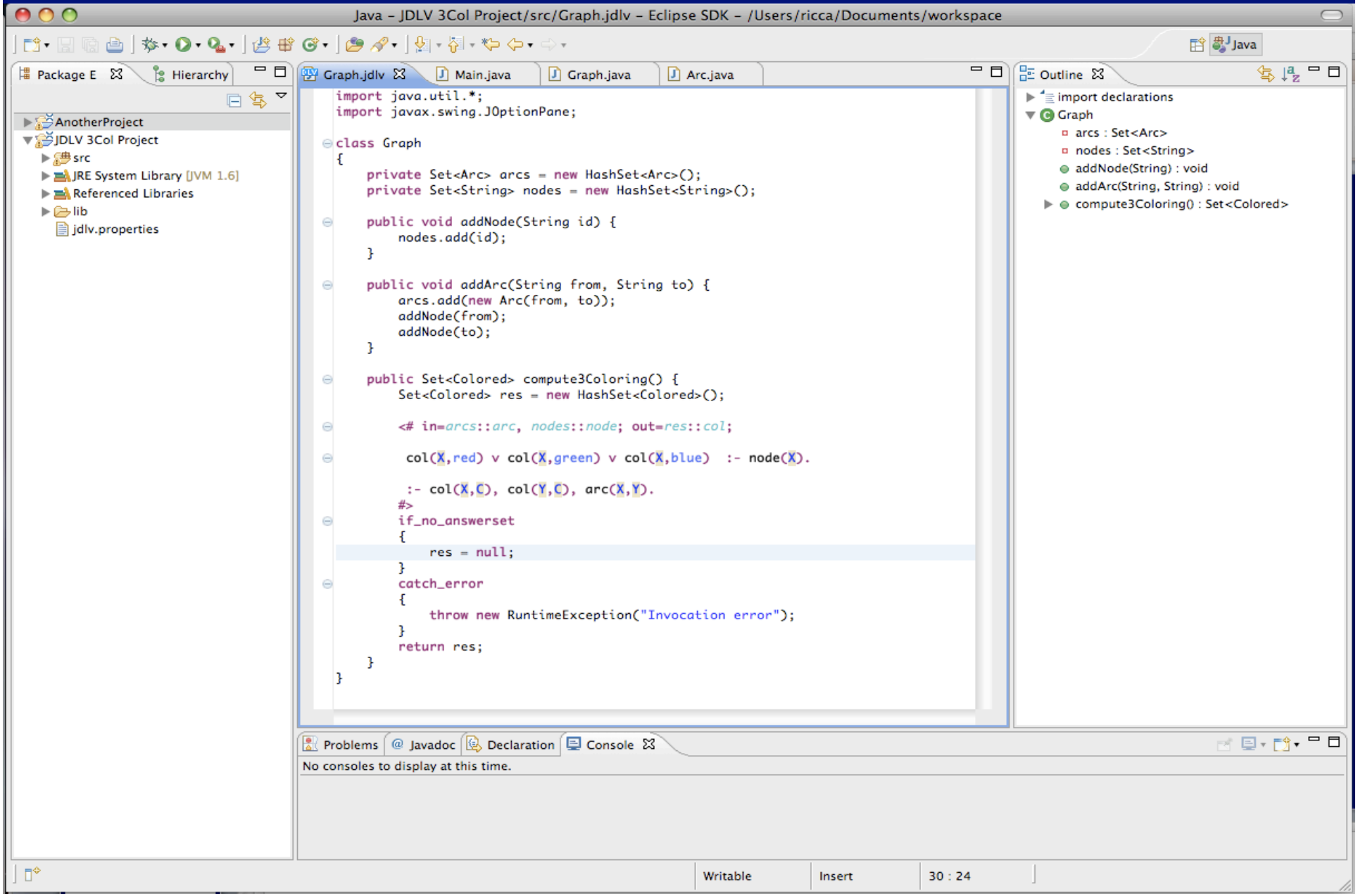
A framework integrating ASP with Java

- The hybrid language JASP
  - simply embed ASP code in a Java program
  - bilateral interaction between ASP and Java
  - exploit standard ORM technologies
    - direct access to DBMS
- The JDLV plugin for Eclipse platform
  - Includes a compiler from JASP to Java
  - Exploits the DLV system

JASP Example: Graph Coloring

```
1  class Graph {
     private Set<Arc> arcs = new HashSet<Arc>();
3    private Set<String> nodes =
                                 new HashSet<String>();
5    public void addNode(String id){
       nodes.insert(id); }
7    public void addArc(String from, String to){
       arcs.insert(new Arc(from,to)); }
9    public Set<Colored> compute3Coloring(){
       Set<Colored> res = new HashSet<Colored>();
11     <# in=arcs::arc,nodes::node out=res::col
          col(X,red) v col(X,green)
13                   v col(X,blue) :- node(X).
         :- col(X,C), col(Y,C), arc(X,Y).
15     #>
       if_no_answerset { res = null; }
17     return res; }
   }
19 public class Arc {
       public String start; public String end;   }
21 public class Colored {
       public String node;  public String color;}
```

The JDLV Plugin for Eclipse



More on JASP/JDLV at KR

MONDAY, JUNE 11

16:00-17:00 TECHNICAL SESSION

Reports from the Field - room B

"JASP: A Framework for Integrating Answer Set Programming with Java"
by Febbraro, Grasso, Leone, and Ricca

Many advanced features:

Come and See !!!

Industry level applications

- Workforce Management in the Gioia Tauro Sea Port
- The IDUM e-Tourism System
- Ontology-based Data Cleaning in a Medical Environment
- Intelligent Data Extraction
- Autonomous Agents in Business Simulations Games

Workforce Management in the Gioia Tauro Sea Port

Workforce Management in a Sea Port

- The Gioia Tauro seaport
  - the largest transshipment terminal of the Mediterranean
  - main activity: container transhipment
  - recently become an *automobile hub*
- Automobile Logistics by ICO B.L.G. (BLG Logistics Group)
  - several ships of different size shore the port every day
  - transported vehicles are handled, warehoused, technically processed and then delivered to their final destination.
- Management Goal: promptly serve shoring boats!
  - Crucial task: arranging suitable teams of employees
    - teams are subject to many constraints
  - The impossibility of arranging teams
    - → contract violations → pecuniary sanctions for B.L.G.
  - Thus, team building is a crucial management task!

Problem Input (simplified)

- The employees and their skills:

  skill(employee, skill)

- Weekly statistics
  (worked hours per skill and last allocation date):

  wstat(employee, skill, hours, lastTime)

- Absent employees:

  absent(employee)

- Employees excluded by a management decision:

  manuallyExcluded(employee)

- A ‘meta-plan’ specification:

  metaPlan(shift, skill, neededEmployees, duration)

Workforce Management

% Guess the assignment of available employees to shifts in appropriate roles

$$(r)\ \ assign(Em, Sh, Sk) \vee nAssign(Em, Sh, Sk) \coloneq skill(Em, Sk), \text{not } absent(Em), \text{not } manuallyExcluded(Em), metaPlan(Sh, Sk, \_, D), workedHours(Em, Wh), Wh + D \leq 36.$$

% Discards assignments with an wrong number of employees in some skill.

$$(c_1)\ \ \coloneq metaPlan(Sh, Sk, EmpNum, \_), \#count\{Em : assign(Em, Sh, Sk)\} \neq EmpNum.$$

% Avoids that an employee covers two roles in the same shift.

$$(c_2)\ \ \coloneq assign(Em, Sh, Sk1), assign(Em, Sh, Sk2), Sk1 \neq Sk2.$$

Workforce Management

% Implement the tournament of roles.

(c_3) $:\!-$ *wstats*(*Em*1, *Sk*, _, *LastTime*1), *wstats*(*Em*2, *Sk*, _, *LastTime*2),
*LastTime*1 $>$ *LastTime*2, *assign*(*Em*1, *Sh*, *Sk*), not *assign*(*Em*2, *Sh*, *Sk*).

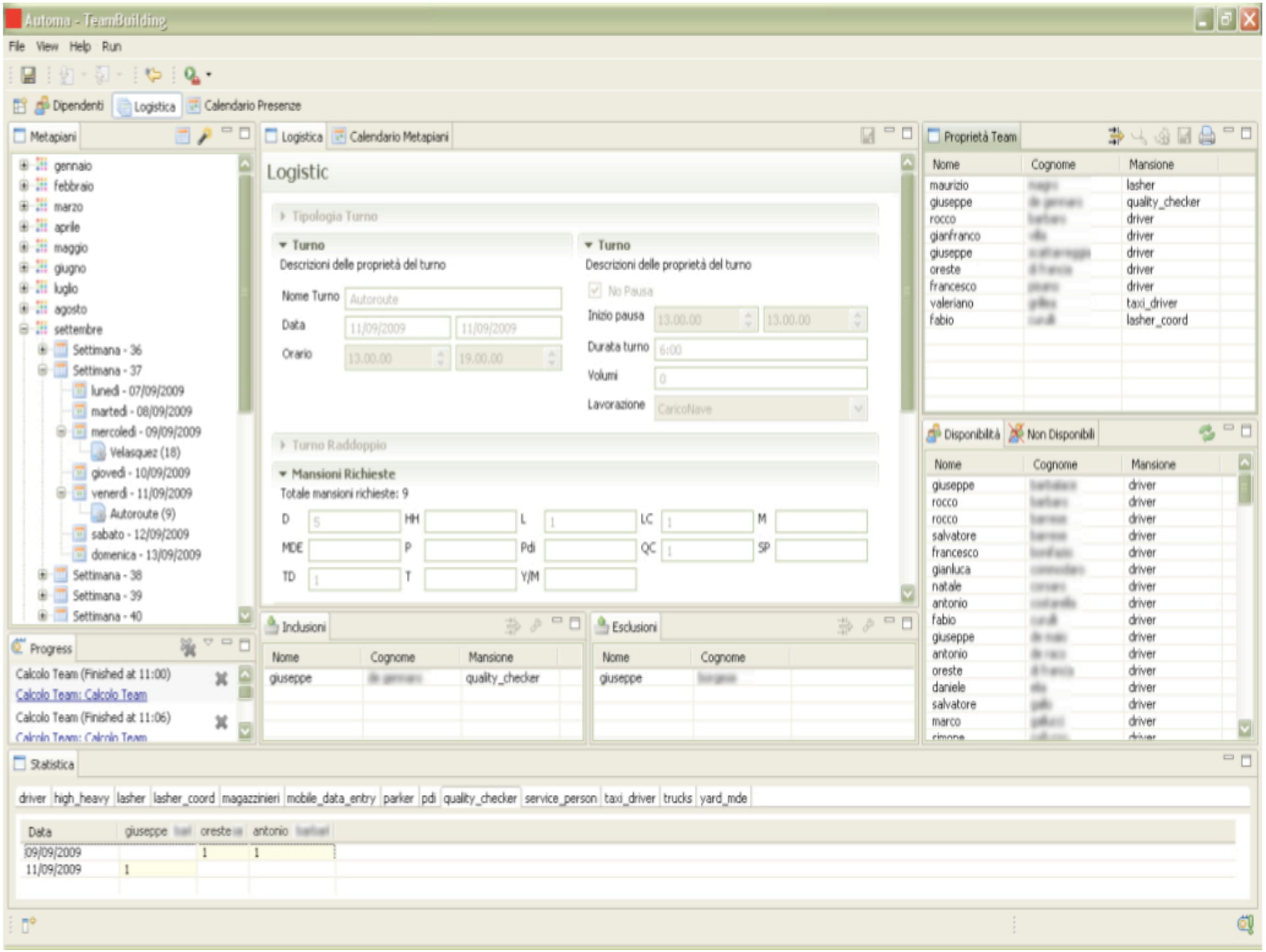
% Guarantees a fair distribution of the workload.

(c_4) $:\!-$ *workedHours*(*Em*1, *Wh*1), *workedHours*(*Em*2, *Wh*2), *threshold*(*Tr*),
*Wh*1 + *Tr* $<$ *Wh*2, *assign*(*Em*1, *Sh*, *Sk*), not *assign*(*Em*2, *Sh*, *Sk*).

% Computes the total number of worked hours per employee.

(r_{aux}) *workedHours*(*Em*, *Wh*) $:\!-$ *skill*(*Em*, _),
$\#\mathtt{count}\{H, Em : wstats(Em, \_, H, \_)\} = Wh.$

% ...many other constraints were developed, tuned and tested!


Automa - TeamBuilding
File View Help Run
Dipendenti
Logistica
Calendario Presenze
Metapiani
gennaio
febbraio
marzo
aprile
maggio
giugno
luglio
agosto
settembre
Settimana - 36
Settimana - 37
lunedì - 07/09/2009
martedì - 08/09/2009
mercoledì - 09/09/2009
Velasquez (18)
giovedì - 10/09/2009
venerdì - 11/09/2009
Autoroute (9)
sabato - 12/09/2009
domenica - 13/09/2009
Settimana - 38
Settimana - 39
Settimana - 40
Logistica
Calendario Metapiani
Logistic
Tipologia Turno
Turno
Descrizioni delle proprietà del turno
Nome Turno
Autoroute
Data
11/09/2009
11/09/2009
Orario
13.00.00
19.00.00
Turno
Descrizioni delle proprietà del turno
No Pausa
Inizio pausa
13.00.00
13.00.00
Durata turno
6:00
Volumi
0
Lavorazione
CaricoNave
Turno Raddoppio
Mansioni Richieste
Totale mansioni richieste: 9
D
5
MH
L
1
LC
1
M
MDE
P
Pdi
QC
1
SP
TD
1
T
Y/M
Proprietà Team
Nome
Cognome
Mansione
maurizio
lasher
giuseppe
quality_checker
rocco
driver
gianfranco
driver
giuseppe
driver
oreste
driver
francesco
driver
valeriano
taxi_driver
fabio
lasher_coord
Disponibilità
Non Disponibili
Nome
Cognome
Mansione
giuseppe
driver
rocco
driver
rocco
driver
salvatore
driver
francesco
driver
gianluca
driver
natale
driver
antonio
driver
fabio
driver
giuseppe
driver
antonio
driver
oreste
driver
daniele
driver
salvatore
driver
marco
driver
Inclusioni
Nome
Cognome
Mansione
giuseppe
quality_checker
Esclusioni
Nome
Cognome
giuseppe
Progress
Calcolo Team (Finished at 11:00)
Calcolo Team: Calcolo Team
Calcolo Team (Finished at 11:06)
Statistica
driver
high_heavy
lasher
lasher_coord
magazzinieri
mobile_data_entry
parker
pdi
quality_checker
service_person
taxi_driver
trucks
yard_mde
Data
giuseppe
oreste
antonio
09/09/2009
1
1
11/09/2009
1

E-Tourism

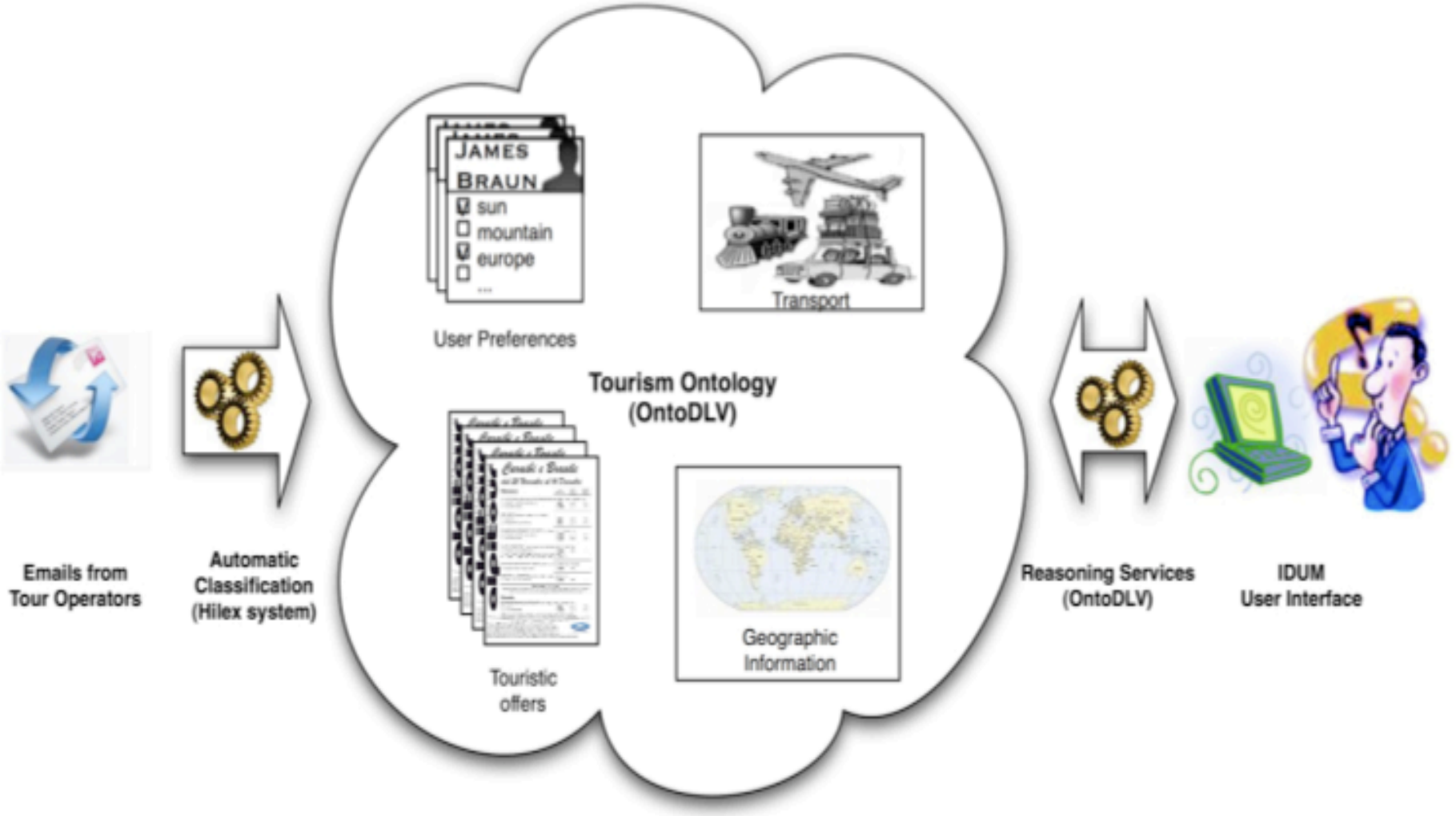
IDUM - Scenario

- New strategies for the tourism industry
  - Tour operators exploit web portals and e-mails
    - can reach directly the client
    - produce a huge volume of packaged offers
- Traditional travel agencies loose competitiveness
  - A large amount of offers
    - Travel agents cannot be aware of all of them properly
  - Clients are more “exigent”
    - finding proper packages is more difficult
  - Employee turn-over
    - The knowledge of clients is often missing

Motivation

- Some needs of travel agencies:
  - Automatic classification of touristic packages
    - Make incoming offers immediately available
  - Advanced search of the best-fitting package
    - “intelligent” match of user’s needs and offer
- The IDUM System
  - An e-tourism system based on ASP
    - Ontologies + logic programs for holiday-package classification and intelligent search

The IDUM System



Tourism Ontology

- An OntoDLP ontology for tourism:
  - offers, transportation, geographic information, user profiles

```
class Customer (name: string, birthDate: Date, status: string, childNumber: integer, job: Job).

relation CustomerPrefersTrip ( cust:Customer, kind: TripKind ).
relation CustomerPrefersMeans ( cust:Customer, means: TransportationMean ).
...
class Place ( description:string ).
relation PlaceOffer( place: place, kind: tripKind ).
relation SuggestedPeriod ( place:place, period: positive integer ).

intentional relation Contains ( pl1:place, pl2:place )
 {
  Contains(P1,P2) :- Contains(P1,P3), Contains(P3,P2).
  Contains('Europe', 'Italy'). Contains('Italy', 'Sicily').
  Contains('Sicily', 'Palermo'). ...
  }

class TouristicOffer( start: Place, destination: Place, kind: TripKind, means: TransportationMean, cost: integer,
   fromDay: Date, toDay: Date, maxDuration: integer, deadline: Date, uri: string, tourOperator: TourOperator ).

...
```

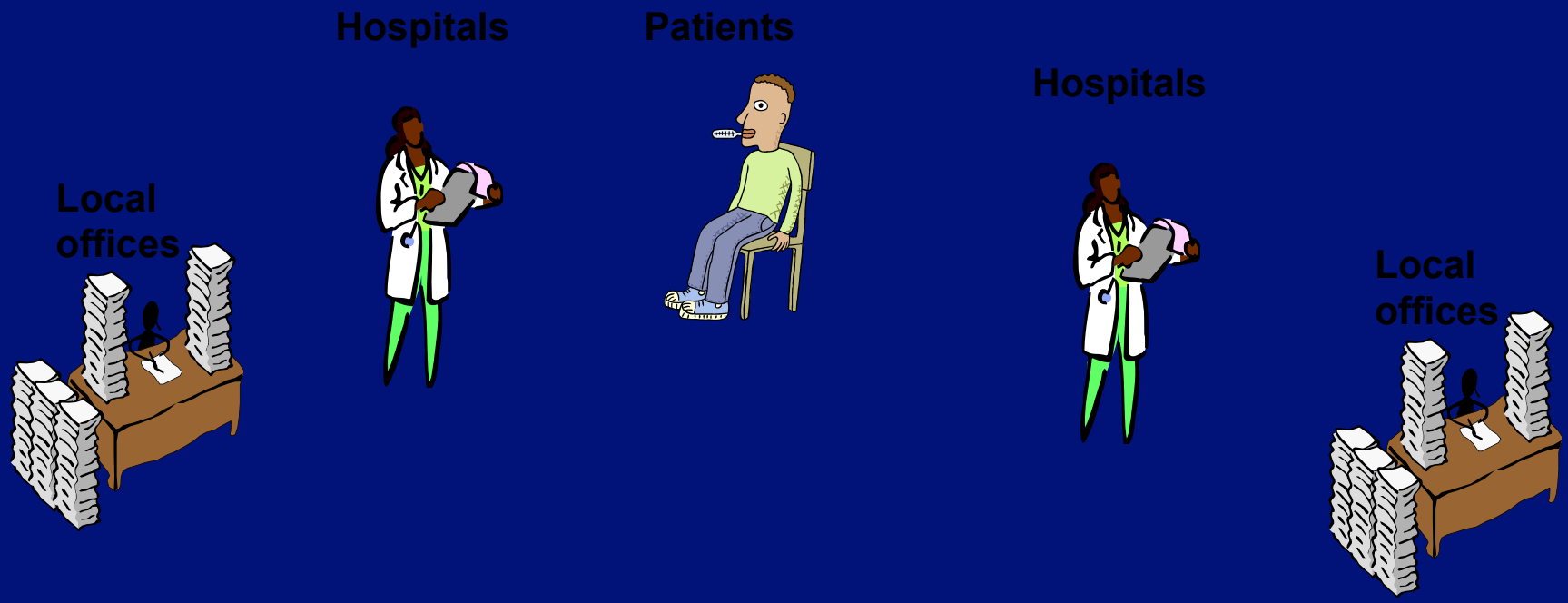
Personalized search

- Search of best-fitting holiday packages made simpler
- “Simulate” the deductions made by travel-agents
  - information modeled in the tourism ontology exploited by using ASP programs
- Key concepts:
  - where, when, how, budget
- Search combines:
  - user’s current desires
  - available offers
  - geographic Information
  - agent’s knowledge and User Profile

Ontology-based Data Cleaning in a Biomedical Application Context

Ontology-based Data Cleaning in a Biomedical Application Context

- Application for a regional medical system
- Information generated by several autonomous and distributed offices, dispersed over the country

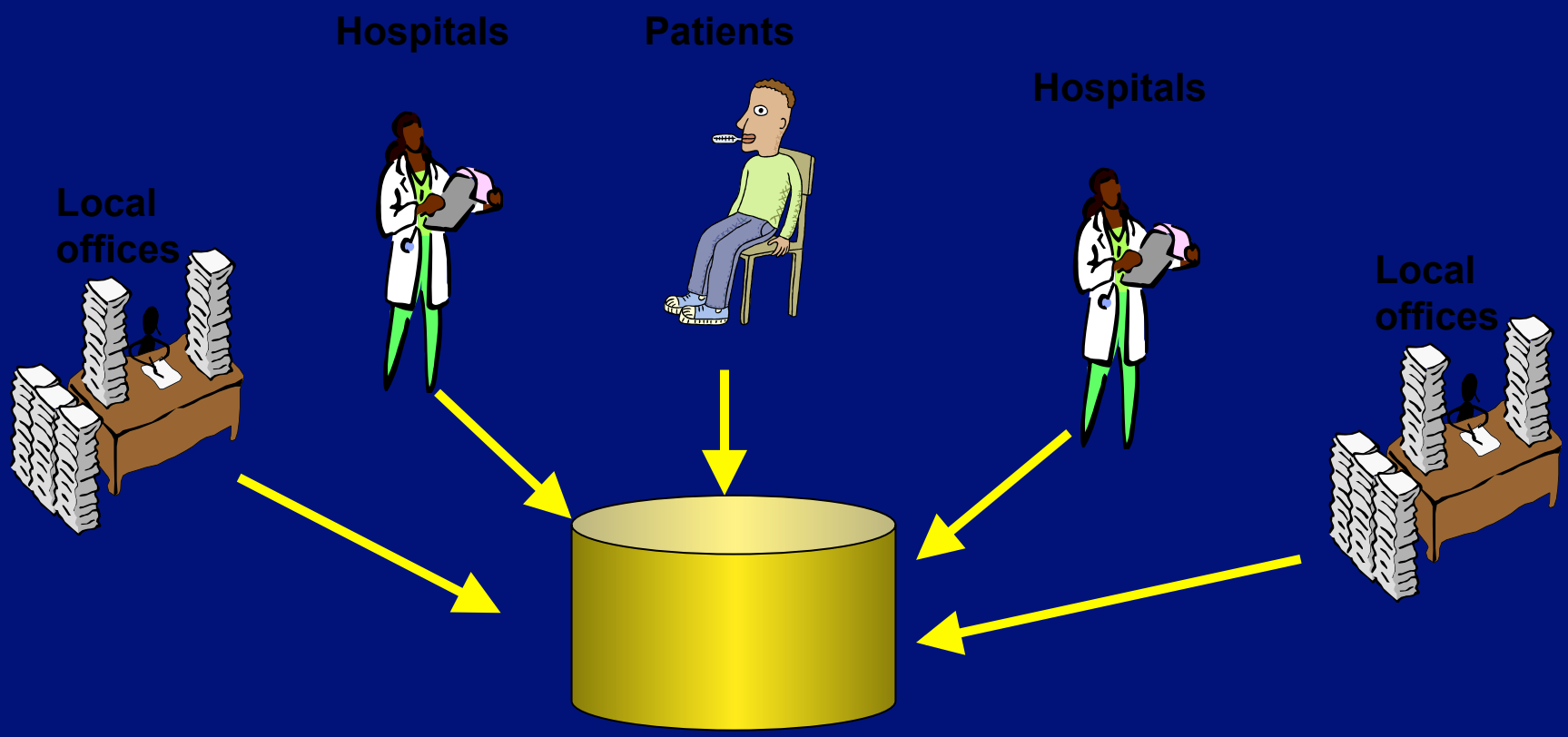


Ontology-based Data Cleaning in a Biomedical Application Context

- Information generated in different formats, often unstructured, and handwritten
- No common structure for the database
- No common data domains

Ontology-based Data Cleaning in a Biomedical Application Context

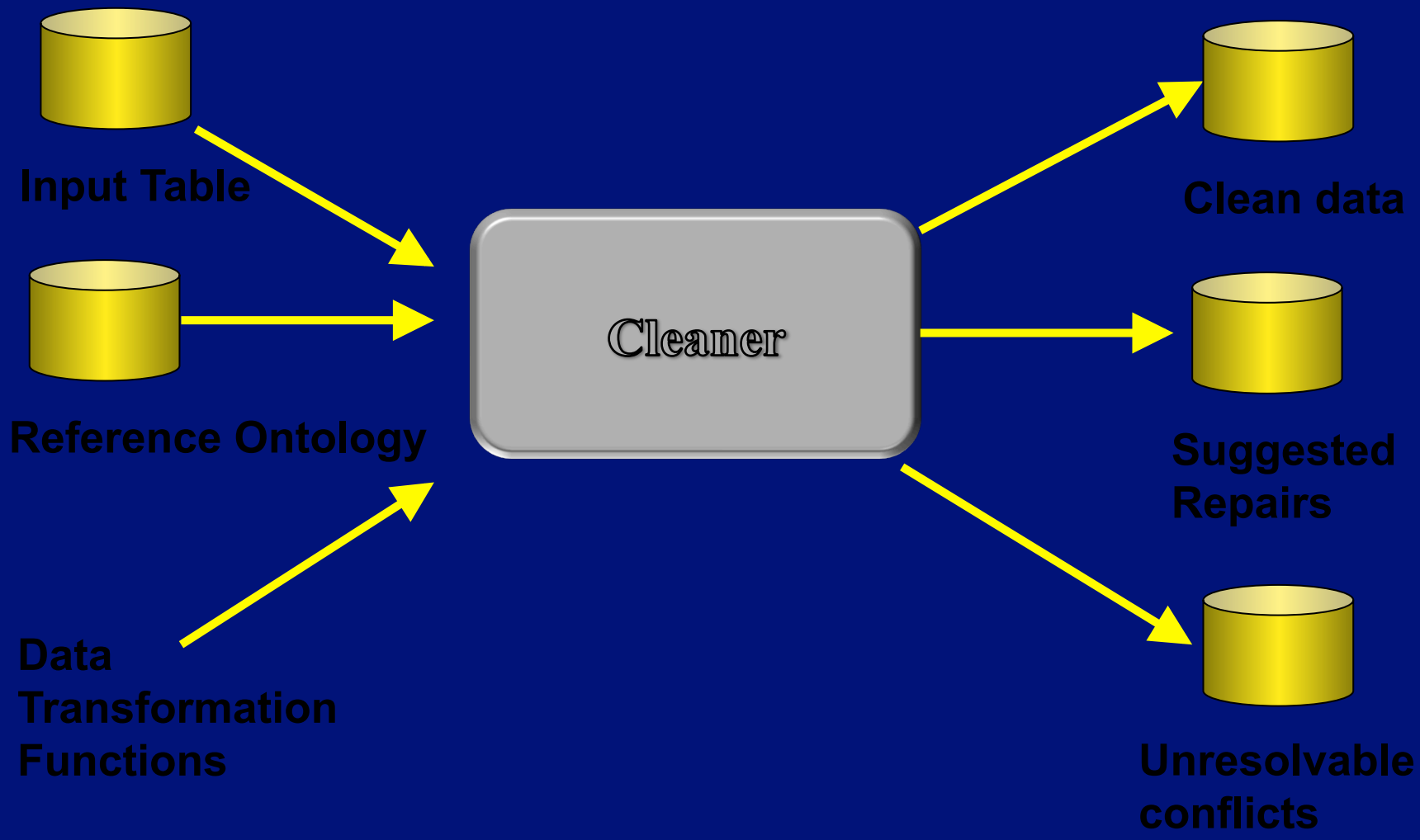
- Strong need to integrate and clean available information



Ontology-based Data Cleaning in a Biomedical Application Context

- Integration of procedural cleaning techniques (e.g. string matching functions) with declarative repairing techniques
- Automatic generation of repairing programs, based on dictionaries

Ontology-based Data Cleaning in a Biomedical Application Context



Ontology-based Data Cleaning in a Biomedical Application Context

- Automatic generation of datalog programs

$$(1)\ tuple\_ok(K, J, F) \leftarrow tab(K, J, F), dic(J, NJ).$$
$$(2)\ tuple\_not\_ok(\bar{K}, \bar{J}, \bar{F}) \leftarrow tab(\bar{K}, \bar{J}, \bar{F}), not\ tuple\_ok(\bar{K}, \bar{J}, \bar{F}).$$

$$(3)\ in\_dic\_J_1(\bar{K}, J_1) \leftarrow tuple\_not\_ok(\bar{K}, J_1, \_, \ldots, \_, \bar{F}), dic(J_1, \_, \ldots, \_, \bar{NJ}).$$

$\ldots$

$$(4)\ in\_dic\_J_m(\bar{K}, J_m) \leftarrow tuple\_not\_ok(\bar{K}, \_, \_, \ldots, J_m, \bar{F}), dic(\_, \_, \ldots, J_m, \bar{NJ}).$$

$$(5)\ inconst\_in\_dic(\bar{K}, \bar{J}) \leftarrow in\_dic\_J_1(\bar{K}, J_1), \ldots, in\_dic\_J_m(\bar{K}, J_m).$$
$$(6)\ inconst\_out\_dic(\bar{K}, \bar{J}) \leftarrow tuple\_not\_ok(\bar{K}, \bar{J}, \bar{F}), not\ inconst\_in\_dic(\bar{K}, \bar{J}).$$

$$(7)\ attr\_out\_dic\_J_1(\bar{K}, J_1) \leftarrow inconst\_out\_dic(\bar{K}, \bar{J}), not\ in\_dic\_J_1(\bar{K}, J_1).$$

$\ldots$

$$(8)\ attr\_out\_dic\_J_m(\bar{K}, J_m) \leftarrow inconst\_out\_dic(\bar{K}, \bar{J}), not\ in\_dic\_J_m(\bar{K}, J_m).$$

Ontology-based Data Cleaning in a Biomedical Application Context

- Identification of errors:
  - Attribute level: the value is not in the dictionary
  - Tuple level: values are singularly in the dictionary, but not in the right configuration

Dictionary

City	ZIP	State
Cosenza	87100	Italia
Rende	87036	Italia

Table

ID	Name	City	ZIP	Tel
101	Smith	Remde	87036	+39….
102	Doh	Cosenza	87036	+39….

Ontology-based Data Cleaning in a Biomedical Application Context

- Solution of errors:
  - Attribute level: Matching function

Dictionary

City	ZIP	State
Cosenza	87100	Italia
Rende	87036	Italia

Hamming dist < 2

Table

ID	Name	City	ZIP	Tel
101	Smith	Remde	87036	+39….
102	Doh	Cosenza	87036	+39….

Ontology-based Data Cleaning in a Biomedical Application Context

- Solution of errors:
  - Attribute level: Matching function

Dictionary

City	ZIP	State
Cosenza	87100	Italia
Rende	87036	Italia

Hamming dist < 2 ?

Table

ID	Name	City	ZIP	Tel
101	Smith	Rende	87036	+39….
102	Doh	Cosenza	87036	+39….

Ontology-based Data Cleaning in a Biomedical Application Context

- Solution of errors:
  - Tuple level: can the substitution of one single attribute "repair" the entire tuple?

Dictionary

City	ZIP	State
Cosenza	87100	Italia
Rende	87036	Italia

Two alternatives:

Table

ID	Name	City	ZIP	Tel
101	Smith	Rende	87036	+39….
102	Doh	Cosenza	87036	+39….

Ontology-based Data Cleaning in a Biomedical Application Context

- Solution of errors:
  - Tuple level: can the substitution of one single attribute "repair" the entire tuple?

Dictionary

City	ZIP	State
Cosenza	87100	Italia
Rende	87036	Italia

Two alternatives:

Table

ID	Name	City	ZIP	Tel
101	Smith	Rende	87036	+39….
102	Doh	Cosenza	87100	+39….

Ontology-based Data Cleaning in a Biomedical Application Context

- Solution of errors:
  - Tuple level: can the substitution of one single attribute "repair" the entire tuple?

Dictionary

City	ZIP	State
Cosenza	87100	Italia
Rende	87036	Italia

Two alternatives:

Table

ID	Name	City	ZIP	Tel
101	Smith	Rende	87036	+39….
102	Doh	Rende	87036	+39….

Intelligent Data Extraction

DIADEM

ERC Advanced Grant @Oxford - G. Gottlob

- Domain-centric, Intelligent, Automated Data Extraction
  - fully automated extraction from domain-specific websites
    - no per site training, no user input other than the domain model
  - main target: websites with structured records
  - based on extensive domain knowledge
    - web form understanding
    - result page analysis (records, attributes)
    - navigation blocks classification (next page link, detail pages)
  - Template language on Datalog$^{\neg,Agg}$ rules compiled to DLV, plus Gazetteers, GATE annotation&regex, ML classifiers

Web Form Understanding with OPAL

Ontology-based Pattern Analysis with Logic

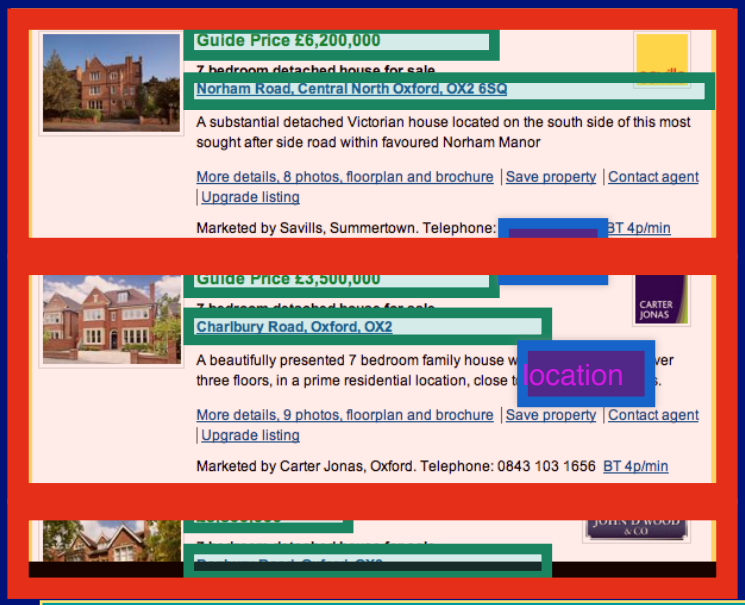
- Recognizes and labels groups of fields + classifies them w.r.t. the domain ontology
  - Reasoning on structural & visual patterns + annotations

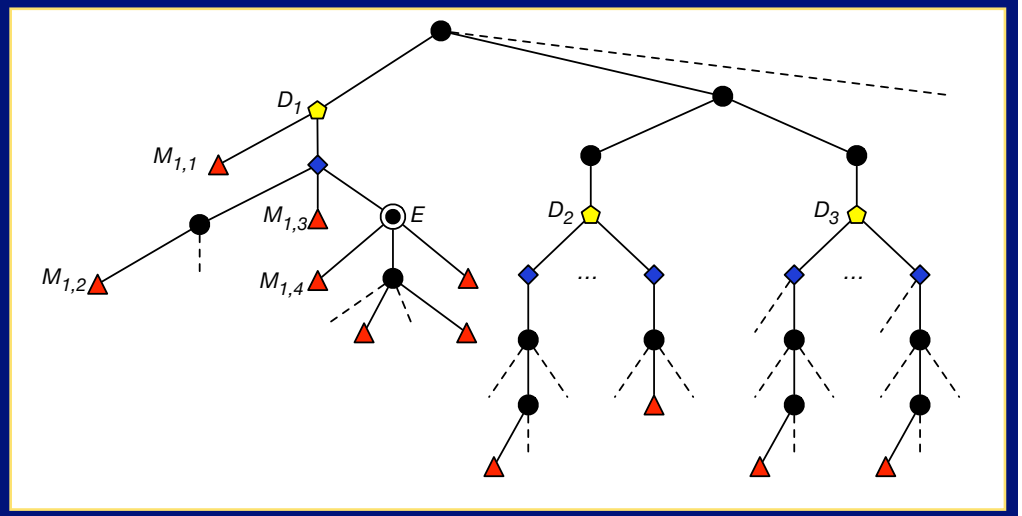
```
group(Es) :
   similarFieldSequence(Es),
   leastCommonAncestor(A,Es),
   not hasAdditionalField(A,Es).

leastCommonAncestor(A,Es) :-
    commonAncestor(A,Es),
    not ( child(C,A),
          commonAncestor(C,Es) ).
partOf(E,A) :-
        group(Es),
    member(E,Es),
    leastCommonAncestor(A,Es).
```

Result Page Analyses with AMBER

Adaptable Model Based Extraction of Result Pages

- Reasoning on annotations and page structure to identify records & attributes






```
consistent_cluster_members(C, N1, N2, N3) :- pivot(N1), pivot(N2), ...
  similar_depth(N1, N2), similar_depth(N2, N3), similar_depth(N1,N3),
  similar_tree_distance(N1, N2, N3).
```

Autonomous Agents in Business Simulations Games

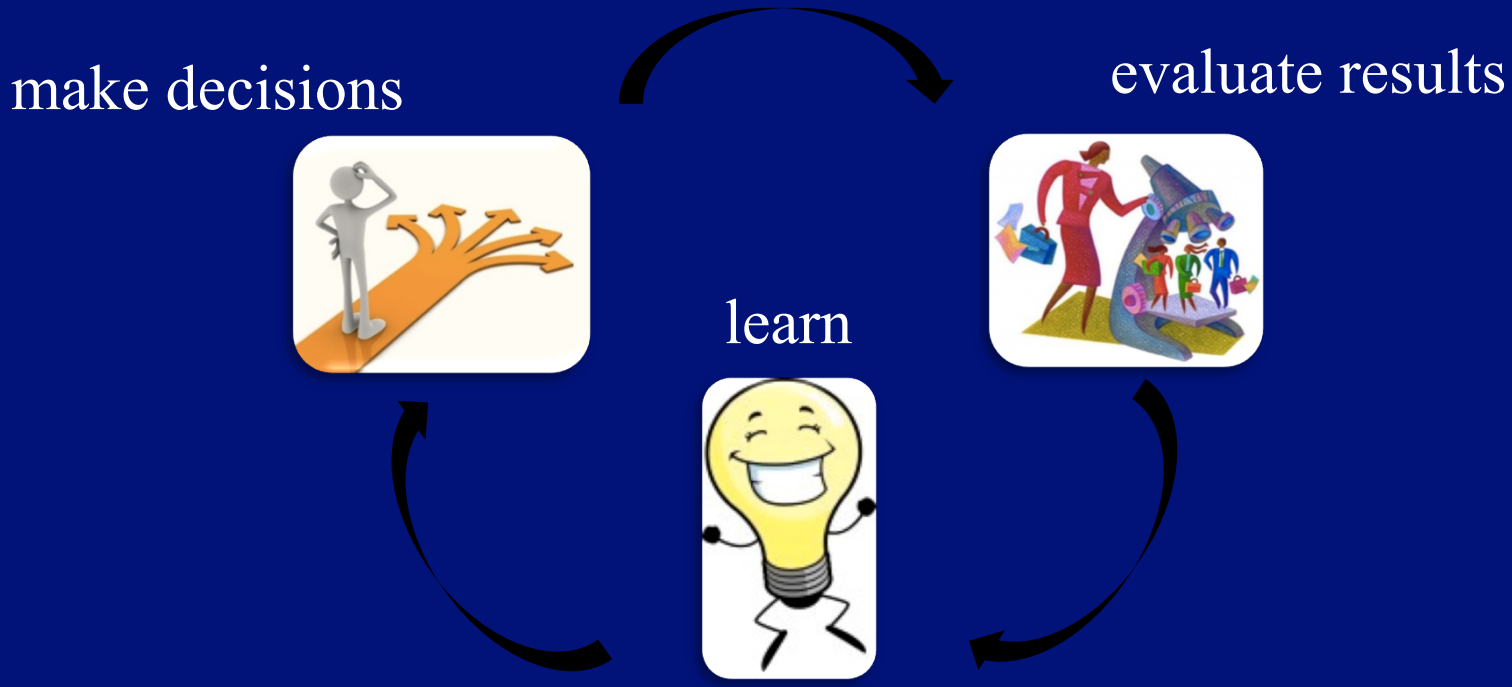
Business Simulation Games (BSGs)

- Simulation games for business training/analysis.
- Learning objectives include: strategic thinking, financial analysis, market analysis, operations, teamwork, leadership

Business Simulation Games (ctd.)

Similar to role-play simulations, to some extent:

- A scenario is played out in a simulated environment
- The player makes individual or team-based decisions
- Feedback on outcomes is provided



Artémat Business Game Studio™

- Commercial Business Simulation Game architecture
- Formal representation and actual implementation of all components of a generic business game
- Novel framework for modeling different economic scenarios
- Web-based architecture and interface
- Conceived for multiplayer

- Ideal for business classes

…but:

- Similarly to other commercial products, artificial players are inadequate for serious learning classes, thus forcing the actual participation of teachers

Intelligent ASP Agents as skilled BSG players

DLVSystem ltd. has realized a framework for specifying autonomous BSG players

- Declarative approach, based on Answer Set Programming
- Relies on DLV system
- Allows for easily defining the behavior of the artificial players, thus creating different profiles/ business strategies

ASP BSG players: some details

```
specifyBehavior("Price", low, costLeadership).
specifyBehavior("Price",high,differentiation).
[…]
output(Lever, Value, Company) :-
   specifyBehavior(Lever, LeverStrategy, Company),
   editable(Lever),
   coherentAction(Lever, Value, LeverStrategy, Turn).
```

- Rich Ontology describing all relevant aspects
  - variable types and range, performance gauges, typical what-if rules, …
- Proper rules for deciding numerical and fuzzy values to be played

Spin Off Companies

- DLVSystem
  - DLV engineering and maintenance
  - Consulting for the development of ASP-based applications
  - 4 permanent employees
- Exeura
  - Consulting on exploiting ASP (/DLV) for KM
  - Also working on Data Mining
  - 30 permanent employees
- Idum
  - Industrial distribution of the e-travel system
  - Strong interest of a Venture Capitalist

Lessons Learned and Conclusion

- The high expressiveness of ASP language is a relevant competitive advantage over other technologies
  - Executable specifications
  - The Gioia Tauro experience
  - Refining user specifications “on site”
- Flexibility, Readability, Ease of maintenance
- Building an efficient NMR system is not enough
  - Often efficiency is not the main issue in real world applications
    - Ease of use
    - Robustness
    - Development tools
    - Programming methodology
    - Testing environments

Lessons Learned and Conclusion

- (In our experience) Typical successful applications
  - Employ Ontologies and Reasoning
  - Exploit Nonmonotonicity
  - Deal with large data sets
  - Do not necessarily involve combinatorial tasks
  - If combinatorial tasks are involved, use an in-depth complexity analysis to single out tractable cases, and isolate the hard kernel !
- Our experience confirms that ASP and, in general, NMR have a high potential for developing innovative applications and could succeed on the market, if the appropriate domains/tasks are chosen, and some technological limitations are overcome