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
<table>
<thead>
<tr>
<th>Roadmap</th>
</tr>
</thead>
<tbody>
<tr>
<td>➢ The ASP System DLV</td>
</tr>
<tr>
<td>➢ ASP Development Tools</td>
</tr>
<tr>
<td>➢ Some Industry Level Applications</td>
</tr>
<tr>
<td>➢ Workforce Management in the Gioia Tauro Sea-Port</td>
</tr>
<tr>
<td>➢ The IDUM e-Tourism System</td>
</tr>
<tr>
<td>➢ Ontology-based Data Cleaning in a Medical Environment</td>
</tr>
<tr>
<td>➢ Intelligent Data Extraction</td>
</tr>
<tr>
<td>➢ Autonomous Agents in Business Simulations Games</td>
</tr>
<tr>
<td>➢ Spin-off Companies</td>
</tr>
<tr>
<td>➢ Lessons Learned and Conclusion</td>
</tr>
</tbody>
</table>
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 $\Sigma^P_2$ ($NP^{NP}$)
  - 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:**

\[
\text{col}(X, \text{red}) \lor \text{col}(X, \text{green}) \lor \text{col}(X, \text{blue}) :- \text{state}(X). \]

\[
:- \text{border}(X,Y), \text{col}(X,C), \text{col}(Y,C). \]

}{Guess}

}{Check}
Weak Constraints: *Exams Scheduling*

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

   \[ r_f : \text{assign}(X,s1) \lor \text{assign}(X,s2) \lor \text{assign}(X,s3) :- \text{course}(X). \]

   \[ s_f : \neg \text{assign}(X,S), \text{assign}(Y,S), \text{commonStudents}(X,Y,N), N>0. \]

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

   \[ r_2 : \text{assign}(X,s1) \lor \text{assign}(X,s2) \lor \text{assign}(X,s3) :- \text{course}(X). \]

   \[ w_2 : \neg \text{assign}(X,S), \text{assign}(Y,S), \text{commonStudents}(X,Y,N), N>0. \quad [N:] \]

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

\[
in(I) \lor out(I) :-\ emp(I,Sx,Sk,Sa). \\
:-\ nEmp(N),\ not\ \#\text{count}\{\ I: in(I) \} = N. \\
:-\ nSkill(M),\ not\ \#\text{count}\{\ Sk: emp(I,Sx,Sk,Sa), in(I) \} \geq M. \\
:-\ budget(B),\ not\ \#\text{sum}\{\ Sa, I: emp(I,Sx,Sk,Sa), in(I) \} \leq B. \\
:-\ maxSal(M),\ not\ \#\text{max}\{\ Sa: emp(I,Sx,Sk,Sa), in(I) \} \leq M.
\]
Functions and Lists: Simple Paths

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

A DLV program computing simple paths:

\[
\text{simplePath}([X,Y]) \ :- \ \text{edge}(X,Y), \ X \neq Y.
\]

\[
\text{simplePath}([X|W]) \ :- \ \text{edge}(X,Y), \ \text{simplePath}(W),
\]

\[
\#\text{head}(W,Y), \ \text{not} \ \#\text{member}(X,W).
\]

alternative encoding:

\[
\text{simplePath}([X|[Y|T]]) \ :- \ \text{edge}(X,Y), \ \text{simplePath}([Y|T]),
\]

\[
\text{not} \ \#\text{member}(X,[Y|T]).
\]

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\textsuperscript{DB}
  - TYP files, ODBC import/export
  - Support for Data Integration
% guess the path
%@name = r2
path(X,Y) :- start(X), outPath(X,Y),
    edge(X,Y), reached(X).
%@name = r3
reached(X) :- path(Y,X).
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

```java
class Graph {
    private Set<Arc> arcs = new HashSet<Arc>();
    private Set<String> nodes =
        new HashSet<String>();

    public void addNode(String id) {
        nodes.insert(id);
    }

    public void addArc(String from, String to) {
        arcs.insert(new Arc(from, to));
    }

    public Set<Colored> compute3Coloring() {
        Set<Colored> res = new HashSet<Colored>();

        // in=arcs::arc, nodes::node out=res::col
        col(X, red) v col(X, green)
        v col(X, blue) :- node(X).
        :- col(X, C), col(Y, C), arc(X, Y).

        if_no_answerset { res = null; }
        return res;
    }
}

class Arc {
    public String start; public String end;
}

class Colored {
    public String node; public String color;
}
```
The JDLV Plugin for Eclipse

```java
import java.util.*;
import javax.swing.JOptionPane;

class Graph {
    private Set<String> nodes = new HashSet<String>();
    private Set<Arc> arcs = new HashSet<Arc>();

    public void addNode(String id) {
        nodes.add(id);
    }

    public void addArc(String from, String to) {
        arcs.add(new Arc(from, to));
        addNode(from);
        addNode(to);
    }

    public Set<Colored> compute3Coloring() {
        Set<Colored> res = new HashSet<Colored>();
        for (Arc arc : arcs) {
            Colored node1 = arc.getStart();
            Colored node2 = arc.getEnd();
            if (node1.equals(node2)) {
                throw new RuntimeException("Self-loop error");
            }
            if (node1.equals() == null) {
                res = null;
            }
            catch_error {
                throw new RuntimeException("Invocation error");
            }
            return res;
        }
    }
}
```
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:
  \[ \text{skill(employee, skill)} \]

- Weekly statistics
  (worked hours per skill and last allocation date):
  \[ \text{wstat(employee, skill, hours, lastTime)} \]

- Absent employees:
  \[ \text{absent(employee)} \]

- Employees excluded by a management decision:
  \[ \text{manuallyExcluded(employee)} \]

- A ‘meta-plan’ specification:
  \[ \text{metaPlan(shift, skill, neededEmployees, duration)} \]
% Guess the assignment of available employees to shifts in appropriate roles

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

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

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

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

(c_2) \leftarrow \text{assign}(Em, Sh, Sk1), \text{assign}(Em, Sh, Sk2), Sk1 \neq Sk2.
% Implement the tournament of roles.

\[(c_3) \leftarrow \text{wstats}(Em_1, Sk, _, \text{LastTime}_1), \text{wstats}(Em_2, Sk, _, \text{LastTime}_2), \]
\[\text{LastTime}_1 > \text{LastTime}_2, \text{assign}(Em_1, Sh, Sk), \text{not assign}(Em_2, Sh, Sk).\]

% Guarantees a fair distribution of the workload.

\[(c_4) \leftarrow \text{workedHours}(Em_1, Wh_1), \text{workedHours}(Em_2, Wh_2), \text{threshold}(Tr), \]
\[Wh_1 + Tr < Wh_2, \text{assign}(Em_1, Sh, Sk), \text{not assign}(Em_2, Sh, Sk).\]

% Computes the total number of worked hours per employee.

\[(r_{aux}) \text{workedHours}(Em, Wh) \leftarrow \text{skill}(Em, _), \]
\[\#\text{count}\{H, Em : \text{wstats}(Em, _, H, _)\} = Wh.\]

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

Ø Some needs of travel agencies:
  Ø Automatic classification of touristic packages
    Ø Make incoming offers immediately available
  Ø Advanced search of the best-fitting package
    Ø “intelligent” match of user’s needs and offer

Ø The IDUM System
  Ø An e-tourism system based on ASP
    Ø Ontologies + logic programs for holiday-package classification and intelligent search
The IDUM System

- Emails from Tour Operators
- Automatic Classification (Hilex system)
- User Preferences
- Touristic offers
- Geographic Information
- Tourism Ontology (OntoDLV)
- Reasoning Services (OntoDLV)
- IDUM User Interface
Tourism Ontology

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

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

relation CustomerPrefersTrip ( cust: Customer, kind: TripKind ).
...

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

Input Table
Reference Ontology
Data Transformation Functions

Cleaner

Clean data
Suggested Repairs
Unresolvable conflicts
Ontology-based Data Cleaning in a Biomedical Application Context

- Automatic generation of datalog programs

\[
\begin{align*}
(1) & \quad \text{tuple_ok}(K, J, F) \leftarrow \text{tab}(K, J, F), \text{dic}(J, NJ).
(2) & \quad \text{tuple_not_ok}(\bar{K}, \bar{J}, \bar{F}) \leftarrow \text{tab}(\bar{K}, \bar{J}, \bar{F}), \neg \text{tuple_ok}(\bar{K}, \bar{J}, \bar{F}).
(3) & \quad \text{in_dic}_1(\bar{K}, J_1) \leftarrow \text{tuple_not_ok}(\bar{K}, J_1, \ldots, \bar{F}), \text{dic}(J_1, \ldots, \bar{NJ}).
& \quad \ldots
(4) & \quad \text{in_dic}_m(\bar{K}, J_m) \leftarrow \text{tuple_not_ok}(\bar{K}, \ldots, J_m, \bar{F}), \text{dic}(\ldots, J_m, \bar{NJ}).
(5) & \quad \text{inconst_in_dic}(\bar{K}, \bar{J}) \leftarrow \text{in_dic}_1(\bar{K}, J_1), \ldots, \text{in_dic}_m(\bar{K}, J_m).
(6) & \quad \text{inconst_out_dic}(\bar{K}, \bar{J}) \leftarrow \text{tuple_not_ok}(\bar{K}, \bar{J}, \bar{F}), \neg \text{inconst_in_dic}(\bar{K}, \bar{J}).
(7) & \quad \text{attr_out_dic}_1(\bar{K}, J_1) \leftarrow \text{inconst_out_dic}(\bar{K}, \bar{J}), \neg \text{in_dic}_1(\bar{K}, J_1).
& \quad \ldots
(8) & \quad \text{attr_out_dic}_m(\bar{K}, J_m) \leftarrow \text{inconst_out_dic}(\bar{K}, \bar{J}), \neg \text{in_dic}_m(\bar{K}, J_m).
\end{align*}
\]
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

<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>City</th>
<th>ZIP</th>
<th>Tel</th>
</tr>
</thead>
<tbody>
<tr>
<td>101</td>
<td>Smith</td>
<td>Remde</td>
<td>87036</td>
<td>+39….</td>
</tr>
<tr>
<td>102</td>
<td>Doh</td>
<td>Cosenza</td>
<td>87036</td>
<td>+39….</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>City</th>
<th>ZIP</th>
<th>State</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cosenza</td>
<td>87100</td>
<td>Italia</td>
</tr>
<tr>
<td>Rende</td>
<td>87036</td>
<td>Italia</td>
</tr>
</tbody>
</table>
Ontology-based Data Cleaning in a Biomedical Application Context

- Solution of errors:
  - Attribute level: Matching function

Table

<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>City</th>
<th>ZIP</th>
<th>Tel</th>
</tr>
</thead>
<tbody>
<tr>
<td>101</td>
<td>Smith</td>
<td>Remde</td>
<td>87036</td>
<td>+39....</td>
</tr>
<tr>
<td>102</td>
<td>Doh</td>
<td>Cosenza</td>
<td>87036</td>
<td>+39....</td>
</tr>
</tbody>
</table>

Dictionary

<table>
<thead>
<tr>
<th>City</th>
<th>ZIP</th>
<th>State</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cosenza</td>
<td>87100</td>
<td>Italia</td>
</tr>
<tr>
<td>Rende</td>
<td>87036</td>
<td>Italia</td>
</tr>
</tbody>
</table>

Hamming dist < 2
Ontology-based Data Cleaning in a Biomedical Application Context

- Solution of errors:
  - Attribute level: Matching function

**Dictionary**

<table>
<thead>
<tr>
<th>City</th>
<th>ZIP</th>
<th>State</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cosenza</td>
<td>87100</td>
<td>Italia</td>
</tr>
<tr>
<td>Rende</td>
<td>87036</td>
<td>Italia</td>
</tr>
</tbody>
</table>

**Table**

<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>City</th>
<th>ZIP</th>
<th>Tel</th>
</tr>
</thead>
<tbody>
<tr>
<td>101</td>
<td>Smith</td>
<td>Rende</td>
<td>87036</td>
<td>+39....</td>
</tr>
<tr>
<td>102</td>
<td>Doh</td>
<td>Cosenza</td>
<td>87036</td>
<td>+39....</td>
</tr>
</tbody>
</table>

Hamming dist < 2?
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?

<table>
<thead>
<tr>
<th>City</th>
<th>ZIP</th>
<th>State</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cosenza</td>
<td>87100</td>
<td>Italia</td>
</tr>
<tr>
<td>Rende</td>
<td>87036</td>
<td>Italia</td>
</tr>
</tbody>
</table>

Two alternatives:

<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>City</th>
<th>ZIP</th>
<th>Tel</th>
</tr>
</thead>
<tbody>
<tr>
<td>101</td>
<td>Smith</td>
<td>Rende</td>
<td>87036</td>
<td>+39….</td>
</tr>
<tr>
<td>102</td>
<td>Doh</td>
<td>Cosenza</td>
<td>87036</td>
<td>+39….</td>
</tr>
</tbody>
</table>
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?

Two alternatives:

Table

<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>City</th>
<th>ZIP</th>
<th>Tel</th>
</tr>
</thead>
<tbody>
<tr>
<td>101</td>
<td>Smith</td>
<td>Rende</td>
<td>87036</td>
<td>+39.....</td>
</tr>
<tr>
<td>102</td>
<td>Doh</td>
<td>Cosenza</td>
<td>87100</td>
<td>+39.....</td>
</tr>
</tbody>
</table>

Dictionary

<table>
<thead>
<tr>
<th>City</th>
<th>ZIP</th>
<th>State</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cosenza</td>
<td>87100</td>
<td>Italia</td>
</tr>
<tr>
<td>Rende</td>
<td>87036</td>
<td>Italia</td>
</tr>
</tbody>
</table>
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?

Two alternatives:

<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>City</th>
<th>ZIP</th>
<th>Tel</th>
</tr>
</thead>
<tbody>
<tr>
<td>101</td>
<td>Smith</td>
<td>Rende</td>
<td>87036</td>
<td>+39….</td>
</tr>
<tr>
<td>102</td>
<td>Doh</td>
<td>Rende</td>
<td>87036</td>
<td>+39….</td>
</tr>
</tbody>
</table>
Intelligent Data Extraction
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\textsuperscript{\textregistered} 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).
```
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

make decisions → evaluate results → learn
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**

```prolog
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