Datalog Development Tools

O. Febbraro², G. Grasso³, N. Leone¹, K. Reale¹ and **Francesco Ricca**¹

¹ Dipartimento di Matematica, Università della Calabria - 87036 Rende, Italy
 ² DLVSystem s.r.l. - P.zza Vermicelli, Polo Tecnologico, 87036 Rende, Italy
 ³ Department of Computer Science, Oxford University - Parks Road, Oxford, UK







- 日本 - 御子 - 田子 - 田子 - 田子

Acknowledgments. This work has been partially supported by the Calabrian Region under PIA project DLVSYSTEM approved in BURC n. 20 parte III del 15/05/2009 - DR n. 7373 del 06/05/2009.

Datalog 2.0 2012

Francesco Ricca Datalog Development Tools







- ASPIDE: Integrated Development Environment for ASP
- JASP: Integrating ASP with Java
- 5 Conclusion and Related Work

・ロト ・ 同ト ・ ヨト ・ ヨト … ヨ



Disjunctive Datalog under the stable model semantics

[Gelfond-Lifschitz '91]

・ロト ・ 同 ト ・ ヨ ト ・ ヨ ト

-

- Answer Set Programming (ASP)
- Declarative programming paradigm
- Non-monotonic reasoning and logic programming

Idea:

- Logic programs represent computational problems
- Answer sets correspond to solutions
- Use a solver to find solutions

Context (2)

After more than 20 years:

- ASP formal properties are well-understood → ASP is expressive... beyond NP [Eiter et.al '97]
- Product and efficient implementations
 - DLV [Leone et.al '06], Clasp [Gebser et.al '07],
 - CModels [Lierler '05], IDP[Wittocx et al.], etc.
 - continuous improvement witnessed by ASP competitions
- Applications in several fields
 - Artificial Intelligence, Knowledge Representation & Reas.,
 - Information Integration, Data cleaning, Bioinformatics, ...
 - Recently employed for developing industrial applications
 - → Workforce Management [Ricca et al. 2010-tplp]
 - → E-Tourism [Ricca et al. 2010-fi]

ヘロン ヘアン ヘビン ヘビン

Context (2)

After more than 20 years:

- ASP formal properties are well-understood → ASP is expressive... beyond NP [Eiter et.al '97]
- Product and efficient implementations
 - DLV [Leone et.al '06], Clasp [Gebser et.al '07],
 - CModels [Lierler '05], IDP[Wittocx et al.], etc.
 - continuous improvement witnessed by ASP competitions
- Applications in several fields
 - Artificial Intelligence, Knowledge Representation & Reas.,
 - Information Integration, Data cleaning, Bioinformatics, ...
 - Recently employed for developing industrial applications
 - → Workforce Management [Ricca et al. 2010-tplp]
 - → E-Tourism [Ricca et al. 2010-fi]

・ロト ・ 理 ト ・ ヨ ト ・

Context (2)

After more than 20 years:

- ASP formal properties are well-understood → ASP is expressive... beyond NP [Eiter et.al '97]
- Pobust and efficient implementations
 - DLV [Leone et.al '06], Clasp [Gebser et.al '07],
 - CModels [Lierler '05], IDP[Wittocx et al.], etc.
 - continuous improvement witnessed by ASP competitions
- Applications in several fields
 - Artificial Intelligence, Knowledge Representation & Reas.,
 - Information Integration, Data cleaning, Bioinformatics, ...
 - Recently employed for developing industrial applications
 - → Workforce Management [Ricca et al. 2010-tplp]
 - \rightarrow E-Tourism [Ricca et al. 2010-fi]

イロト 不得 とくほ とくほう 二日

Lessons learned

- Viability of the exploitation of ASP in real-world apps.
 - complex business-logic at a lower (implementation) price
 - flexibility, readability, extensibility, ease of maintenance, etc.
- Practical obstacles to ASP-based development:
 - ASP programmers needs an IDE
 - \rightarrow programmers accustomed to Workbenches (e.g. eclipse,...)
 - \rightarrow tools for simplifying development and maintenance
 - \rightarrow graphic tools simplify the approach of novice users
 - ASP is not a full general-purpose language
 - \rightarrow some components better built with O.-O. Programming
 - \rightarrow ASP solutions must be embedded at some point
 - ASP is not integrated in development processes and platforms

Lessons learned

- Viability of the exploitation of ASP in real-world apps.
 - complex business-logic at a lower (implementation) price
 - flexibility, readability, extensibility, ease of maintenance, etc.
- Practical obstacles to ASP-based development:
 - ASP programmers needs an IDE
 - \rightarrow programmers accustomed to Workbenches (e.g. eclipse,...)
 - \rightarrow tools for simplifying development and maintenance
 - \rightarrow graphic tools simplify the approach of novice users
 - ASP is not a full general-purpose language

 \rightarrow some components better built with O.-O. Programming

→ ASP solutions must be embedded at some point

ASP is not integrated in development processes and platforms

Lessons learned

- Viability of the exploitation of ASP in real-world apps.
 - complex business-logic at a lower (implementation) price
 - flexibility, readability, extensibility, ease of maintenance, etc.
- Practical obstacles to ASP-based development:
 - ASP programmers needs an IDE
 - \rightarrow programmers accustomed to Workbenches (e.g. eclipse,...)
 - \rightarrow tools for simplifying development and maintenance
 - \rightarrow graphic tools simplify the approach of novice users
 - ASP is not a full general-purpose language
 - \rightarrow some components better built with O.-O. Programming
 - \rightarrow ASP solutions must be embedded at some point
 - ASP is not integrated in development processes and platforms

Development Tools for ASP

() ASPIDE: IDE for ASP... the most comprehensive

- Cutting-edge editing tool
 - \rightarrow textual/graphical (assisted) composition of programs
- Development tools
 - \rightarrow debugging, profiling, testing, run configuration, output-handling
- Application configuration and deployment tools
 - \rightarrow DBMS access, solver execution configuration, ...
- Extensible with plugins

a framework integrating ASP with Java

- The hybrid language \mathcal{JASP}
 - simply embed ASP code in a Java program
 - \rightarrow bilateral interaction between ASP and Java
- The Eclipse plug-in JDLV
 - compiler from *JASP* to Java

・ロト ・ 同 ト ・ ヨ ト ・ ヨ ト

Development Tools for ASP

ASPIDE: IDE for ASP... the most comprehensive

- Cutting-edge editing tool
 - \rightarrow textual/graphical (assisted) composition of programs
- Development tools
 - \rightarrow debugging, profiling, testing, run configuration, output-handling
- Application configuration and deployment tools
 - \rightarrow DBMS access, solver execution configuration, ...
- Extensible with plugins

a framework integrating ASP with Java

- The hybrid language *JASP*
 - simply embed ASP code in a Java program
 - \rightarrow bilateral interaction between ASP and Java
- The Eclipse plug-in JDLV
 - compiler from *JASP* to Java

イロト 不得 とくほ とくほう 二日

Development Tools for ASP

ASPIDE: IDE for ASP... the most comprehensive

- Cutting-edge editing tool
 - \rightarrow textual/graphical (assisted) composition of programs
- Development tools
 - \rightarrow debugging, profiling, testing, run configuration, output-handling
- Application configuration and deployment tools
 - \rightarrow DBMS access, solver execution configuration, ...
- Extensible with plugins

A framework integrating ASP with Java

- $\bullet~$ The hybrid language \mathcal{JASP}
 - simply embed ASP code in a Java program
 - \rightarrow bilateral interaction between ASP and Java
- The Eclipse plug-in JDLV
 - compiler from \mathcal{JASP} to Java

イロト 不得 とくほ とくほう 二日

ASP Syntax and Semantics



- Semantics:
 - Consider the Ground Instantiation $P = ground(\pi)$
 - Apply the Gelfond-Lifschitz semantics on P

Informal meaning:

"At least one a_i is true if b_1, \ldots, b_k are true and b_{k+1}, \ldots, b_m are false".

◆□▶ ◆□▶ ★ □▶ ★ □▶ → □ → の Q ()

ASP Example

Example (3-col)

Input: a Graph represented by node(_) and edge(_, _).
 Problem: assign one color out of 3 colors to each node such that two adjacent nodes have always different colors.

% guess a coloring for the nodes (r) $col(X, red) \lor col(X, yellow) \lor col(X, green) := node(X)$. % discard colorings where adjacent nodes have the same color (c) := edge(X, Y), col(X, C), col(Y, C).

ASPIDE: Integrated Development Environment for Answer Set Programming

 O. Febbraro, K. Reale, F. Ricca
 "ASPIDE: Integrated Development Environment for Answer Set Programming."
 In: Proc. of LPNMR 2011 (2011)

ヘロン ヘアン ヘビン ヘビン

Motivation

Diffused programming languages come with SDKs

- \rightarrow tools for simplifying development and maintenance
- \rightarrow programmers accustomed to Workbenches (e.g. eclipse,...)
- ightarrow graphic tools simplify the approach of novice users
- Integrated Development Environment (IDE)
 - tools for the entire life-cycle of development
 → from (assisted) programs editing to application deployment
- No IDE \rightarrow usage of ASP may be discouraged

・ロト ・ 同ト ・ ヨト ・ ヨト … ヨ

Motivation

Diffused programming languages come with SDKs

- \rightarrow tools for simplifying development and maintenance
- \rightarrow programmers accustomed to Workbenches (e.g. eclipse,...)
- ightarrow graphic tools simplify the approach of novice users

Integrated Development Environment (IDE)

- tools for the entire life-cycle of development
 → from (assisted) programs editing to application deployment
- No IDE \rightarrow usage of ASP may be discouraged

◆□▶ ◆□▶ ◆三▶ ◆三▶ ● ○ ○ ○

Motivation

Diffused programming languages come with SDKs

- \rightarrow tools for simplifying development and maintenance
- \rightarrow programmers accustomed to Workbenches (e.g. eclipse,...)
- ightarrow graphic tools simplify the approach of novice users

Integrated Development Environment (IDE)

tools for the entire life-cycle of development
 → from (assisted) programs editing to application deployment

• No IDE \rightarrow usage of ASP may be discouraged

・ロト ・ 同ト ・ ヨト ・ ヨト … ヨ



• **Goal:** An Integrated Development Environment supporting the entire life-cycle of ASP development

• ASPIDE

- Cutting-edge editing tool
 - \rightarrow textual/graphical (assisted) composition of programs
- Development tools
 - → debugging, profiling, testing,
 - ightarrow run configuration, output-handling
 - ightarrow visual representation of results, plug-ins
- Application configuration and deployment tools
 - \rightarrow DBMS access, execution configuration, ...

ヘロン ヘアン ヘビン ヘビン



• **Goal:** An Integrated Development Environment supporting the entire life-cycle of ASP development

ASPIDE

- Cutting-edge editing tool
 - \rightarrow textual/graphical (assisted) composition of programs
- Development tools
 - \rightarrow debugging, profiling, testing,
 - \rightarrow run configuration, output-handling
 - \rightarrow visual representation of results, plug-ins
- Application configuration and deployment tools
 - \rightarrow DBMS access, execution configuration, ...

・ロト ・ 同 ト ・ ヨ ト ・ ヨ ト

э

Main Features (1)

Workspace management

organize projects and related resources

Advanced Program Editing

- Text Editor
 - Text coloring, automatic completion, refactoring, code templates, code checking, errors highlighting, quick fixes,etc.
- Visual Editor with Reverse-engineering
 - Drawing logic programs in a QBE-like style
- Outline navigation
- Dependency graph visualizer
- Run Configuration and Presentation of Results
 - Setup execution & friendly browsing of results

・ロト ・ 理 ト ・ ヨ ト ・

Main Features (2)

Debugger, and Profiler

- GUI for the debugging tool Spock [Brain-etal-07]
- GUI for the DLV Profiler [Calimeri-etal-09]

• Unit Testing for ASP

- a framework in the style of JUnit
- a new language for unit tests in ASP
- graphical interface for creating unit tests

Annotations for ASP programs

- annotations start with @ (eg. @name, @schema, ...)
- rule names, predicate schemas and database specification

ヘロア 人間 アメヨア 人口 ア

Main Features (3)

• User-defined Plugins \rightarrow SDK free for download

- **()** Input data is not encoded in ASP \rightarrow Input Plugin
 - needs to be translated
- 2 Rules might need "refactorings" \rightarrow **Rewriting Plugin**
 - performance, input languages etc.
- Specific output formats → Output Plugin
 - solver result as set of ASP facts

Database Access

- Connect to DBMSs via JDBC, DLV ^{DB} TYP files, ODBC import/export
- Attribute names and data types for predicates
- Data Integration scenario
- Implemented as an Input Plugin

イロト 不得 とくほ とくほう 二日



DEMO

- Create/Modify/Test Programs
- Create a program that runs on a RDBMS

http://www.mat.unical.it/ricca/aspide

Integrated Development Environment for Answer Set Programming								
Overview	Usage example	Manual	Download					
Aspide	Manual.	Download ASPIDE.						
	Text coloring	Control of the second s	And a					
	Th	e ASPIDE Team						
	Dott. Onofrio Febbraro,	Dott. Kristian Reale and Dott.	Francesco Ricca					

Francesco Ricca

Datalog Development Tools

JASP: integrating Java with Answer Set Programming

 O. Febbraro, N. Leone, G. Grasso, F. Ricca
 "JASP: A Framework for Integrating Answer Set Programming with Java."
 In: Proc. of KR 2012 (2012)

ヘロン ヘアン ヘビン ヘビン

ASP and Object-Oriented Programming

• Existing Application Programming Interfaces (API)

- the DLV Wrapper [Ricca 2003], OntoDLV API [Gallucci '07]
 - libraries for interacting with an ASP solver from an Java program
- control the execution of an external solver and
- convert data from logic-based to Java representations

• Shortcomings:

the programmer has the burden of the integration

 → repetitive and time-consuming ad-hoc procedures
 no support from programming tools and workbenches
 no support for enterprise applications standards
 → persistency of complex object-oriented domain models

 Developers are hindered from adopting a poorly-supported non standard technology

ヘロア 人間 アメヨア 人口 ア

ASP and Object-Oriented Programming

• Existing Application Programming Interfaces (API)

- the DLV Wrapper [Ricca 2003], OntoDLV API [Gallucci '07]
 - libraries for interacting with an ASP solver from an Java program
- control the execution of an external solver and
- convert data from logic-based to Java representations

Shortcomings:

- the programmer has the burden of the integration
 - \rightarrow repetitive and time-consuming ad-hoc procedures
- In a support from programming tools and workbenches
- on support for enterprise applications standards
 - \rightarrow persistency of complex object-oriented domain models
- Developers are hindered from adopting a poorly-supported non standard technology

・ロト ・ 理 ト ・ ヨ ト ・

ASP and Object-Oriented Programming

• Existing Application Programming Interfaces (API)

- the DLV Wrapper [Ricca 2003], OntoDLV API [Gallucci '07]
 - libraries for interacting with an ASP solver from an Java program
- control the execution of an external solver and
- convert data from logic-based to Java representations

Shortcomings:

- the programmer has the burden of the integration
 - \rightarrow repetitive and time-consuming ad-hoc procedures
- In a support from programming tools and workbenches
- o support for enterprise applications standards
 - \rightarrow persistency of complex object-oriented domain models

Developers are hindered from adopting a poorly-supported non standard technology

ヘロン ヘアン ヘビン ヘビン

The needs that inspired \mathcal{JASP}

• Needs:

- Seamless embedding of ASP
- Integration in standard software processes/technologies
- Support for development tools/workbenches
- A framework integrating ASP with Java
 - 1) The hybrid language \mathcal{JASP}
 - simply embed ASP code in a Java program
 - \rightarrow bilateral interaction between ASP and Java
 - exploit standard ORM technologies
 - 2 An implementation of \mathcal{JASP}
 - compiler from *JASP* to Java
 - plugin for the Eclipse platform

・ロト ・ 同 ト ・ ヨ ト ・ ヨ ト

The needs that inspired \mathcal{JASP}

• Needs:

- Seamless embedding of ASP
- Integration in standard software processes/technologies
- Support for development tools/workbenches
- A framework integrating ASP with Java
 - 1 The hybrid language \mathcal{JASP}
 - simply embed ASP code in a Java program
 - \rightarrow bilateral interaction between ASP and Java
 - exploit standard ORM technologies
 - 2 An implementation of \mathcal{JASP}
 - $\bullet~$ compiler from \mathcal{JASP} to Java
 - plugin for the Eclipse platform

イロト 不得 とくほ とくほう 二日

Common Scenario

Develop an application: (e.g., Map coloring tool)

- Adopt (say) Java
 - \rightarrow libraries, programming tools, O-O programming...
- Complex problem to solve (e.g., 3-Col)
 - Write a complex backtracking procedure
 - Write a two-rule ASP program!
- Developer Desiderata:
 - I do not want to write ASP to Java glue code
 - I need to know an API
 - Repetitive tasks: Variables \rightarrow Facts + Build ASP Rules \rightarrow
 - \rightarrow Call solver + Parse Answer Sets \rightarrow Variables

I do not want to change my programming environment

(e.g., Eclipse)

Common Scenario

Develop an application: (e.g., Map coloring tool)

- Adopt (say) Java
 - \rightarrow libraries, programming tools, O-O programming...
- Complex problem to solve (e.g., 3-Col)
 - Write a complex backtracking procedure
 - Write a two-rule ASP program!
- Developer Desiderata:
 - I do not want to write ASP to Java glue code
 - I need to know an API
 - Repetitive tasks: Variables \rightarrow Facts + Build ASP Rules \rightarrow
 - \rightarrow Call solver + Parse Answer Sets \rightarrow Variables

I do not want to change my programming environment

(e.g., Eclipse)

Common Scenario

Develop an application: (e.g., Map coloring tool)

- Adopt (say) Java
 - \rightarrow libraries, programming tools, O-O programming...
- Complex problem to solve (e.g., 3-Col)
 - Write a complex backtracking procedure
 - Write a two-rule ASP program!
- Developer Desiderata:
 - I do not want to write ASP to Java glue code
 - I need to know an API
 - Repetitive tasks: Variables \rightarrow Facts + Build ASP Rules \rightarrow
 - \rightarrow Call solver + Parse Answer Sets \rightarrow Variables

I do not want to change my programming environment

(e.g., Eclipse)

Common Scenario

Develop an application: (e.g., Map coloring tool)

- Adopt (say) Java
 - \rightarrow libraries, programming tools, O-O programming...
- Complex problem to solve (e.g., 3-Col)
 - Write a complex backtracking procedure
 - Write a two-rule ASP program!
- Developer Desiderata:
 - I do not want to write ASP to Java glue code
 - I need to know an API
 - Repetitive tasks: Variables \rightarrow Facts + Build ASP Rules \rightarrow
 - \rightarrow Call solver + Parse Answer Sets \rightarrow Variables

I do not want to change my programming environment

(e.g., Eclipse)

Common Scenario

Develop an application: (e.g., Map coloring tool)

- Adopt (say) Java
 - \rightarrow libraries, programming tools, O-O programming...
- Complex problem to solve (e.g., 3-Col)
 - Write a complex backtracking procedure
 - Write a two-rule ASP program!
- Developer Desiderata:
 - I do not want to write ASP to Java glue code
 - I need to know an API
 - Repetitive tasks: Variables \rightarrow Facts + Build ASP Rules \rightarrow
 - \rightarrow Call solver + Parse Answer Sets \rightarrow Variables

I do not want to change my programming environment

(e.g., Eclipse)

Common Scenario

Develop an application: (e.g., Map coloring tool)

- Adopt (say) Java
 - \rightarrow libraries, programming tools, O-O programming...
- Complex problem to solve (e.g., 3-Col)
 - Write a complex backtracking procedure
 - Write a two-rule ASP program!
- Developer Desiderata:
 - I do not want to write ASP to Java glue code
 - I need to know an API
 - Repetitive tasks: Variables \rightarrow Facts + Build ASP Rules \rightarrow
 - \rightarrow Call solver + Parse Answer Sets \rightarrow Variables

I do not want to change my programming environment

(e.g., Eclipse)

Common Scenario

Develop an application: (e.g., Map coloring tool)

- Adopt (say) Java
 - \rightarrow libraries, programming tools, O-O programming...
- Complex problem to solve (e.g., 3-Col)
 - Write a complex backtracking procedure
 - Write a two-rule ASP program!
- Developer Desiderata:
 - I do not want to write ASP to Java glue code
 - I need to know an API
 - Repetitive tasks: Variables \rightarrow Facts + Build ASP Rules \rightarrow
 - \rightarrow Call solver + Parse Answer Sets \rightarrow Variables

2 I do not want to change my programming environment

(e.g., Eclipse)

イロト 不得 とくほ とくほう 二日

\mathcal{JASP} -core by example

```
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>();
  <# in=arcs::arc,nodes::node out=res::col</pre>
11
      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; }
   return res; }
19 public class Arc {
   public String start; public String end; }
21 public class Colored {
   public String node; public String color;}
```

Java World

・ロト ・ 同ト ・ ヨト ・ ヨト … ヨ

\mathcal{JASP} -core by example



\mathcal{JASP} -core by example



\mathcal{JASP} -core by example



\mathcal{JASP} -core by example



Francesco Ricca

Default ORM Mapping

Map Java objects to ASP facts (and vice-versa)

- Impedance Mismatch problem [Maier '90,Keller et al. '93]
 - \rightarrow shared with object persistence frameworks
- Basic Object-Relational Mapping (ORM) Strategy
 - For classes with:
 - no-arguments constructor, non-recursive type definition
 - getters and setters, one-to-one associations only
 - Intuitively:
 - one attribute per (simple) field
 - several fields for one-to-one associations
 - compound key of all class attributes + embedded value
 - attributes are filled according to the declaration order of fields

・ロト ・ 同ト ・ ヨト ・ ヨト … ヨ

• ... custom mappings with Java Persistence API (JPA)

Basic ORM Example

```
public class Arc {
    private String start:
    private String end:
    public Arc(String from, String to) {
        start = from; end = to; }
    public String getStart() {
        return start; }
    public void setStart(String start) {
        this.start = start; }
    public String getEnd() {
        return end: }
    public void setEnd(String end) {
        this.end = end; }
    3
```

・ロト ・ 理 ト ・ ヨ ト ・

-

Semantics

```
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; }
```

```
public Set<Colored> compute3Coloring() {
 Set<Colored> res = new HashSet<Colored>();
 Module module = new Module();
 module.program.add("col(X, red) OR
       col(X,green) OR col(X,blue) :- node(X).
     :- col(X,C), col(Y,C), arc(X,Y).");
 module.buildFacts("arc", arcs);
 module.buildFacts("node", nodes);
 if ( ! module.nextAnswerset()) {return null:
 else do 
       res = (Set<Colored>)
          module.createObject("col", res);
       break:
       }while(module.nextAnswerset());
 return res; }
```

ヘロン ヘアン ヘビン ヘビン

-

Enterprise Applications

Enterprise Applications:

- AkA: "information systems" or "data processing"
- They involve:
 - persistent data... a lot of data
 - a lot of user interface screens
 - complex business (il)logic
- Widely employed technologies
 - Relational Database Systems
 - Persistence via ORM Tools (e.g., Hibernate, TopLink)
 - Component based Frameworks: (e.g., EJB, Spring)
 - Web-based Interfaces
 - Fully-featured Workbenches (e.g., Eclipse)

・ロト ・ 理 ト ・ ヨ ト ・

Enterprise Applications Architecture



◆□ ▶ ◆□ ▶ ◆ □ ▶ ◆ □ ▶ ● □ ● ● ● ●

The \mathcal{JASP} Language

• The basic language *JASP*-core

 \rightarrow is ok in most cases

 \rightarrow improved for easing (enterprise) application development

• *JASP* (full) features:

- Java Persistence API (JPA) Mappings
- Named Non-positional Notation
- Dynamic Composition of Modules
- Access Java variables in logic rules
- Database Access

・ロト ・ 同ト ・ ヨト ・ ヨト … ヨ

JPA Mappings

Custom ORM mappings according with Java Persistence API package it.unical.mat.teambuilding.persistence; import java.util.ArravList: @Entity public class Shift implements Comparable<Shift>{ @Td Integer id; @Column String name; @OneToMany(cascade=CascadeType.ALL) List<NeededEmployee> neededEmployees=new ArrayList<NeededEmployee>(): @Column Date date; @Column Integer duration: @OneToMany(cascade=CascadeType.ALL) List<Employee> exludedEmployees=new ArrayList<Employee>(); public Shift(){}

//...

◆□▶ ◆□▶ ★ □▶ ★ □▶ → □ → の Q ()

Supported JPA Annotations

Annotation	Summary			
@Entity	Indicates a class with mapping.			
	Class name is the predicated name.			
@Table (name="pred-	In conjunction with @Entity, to re-			
name")	name the default predicate name.			
@Column	Identifies a class member, to be in-			
	cluded in the mapping.			
@Id	Marks a class member as identifier			
	(key) of the relative table.			
@OneToMany	On class members to denote associ-			
@ManyToOne	ations multiplicity			
@ManyToMany				
@OneToOne				
@JoinTable	In conjunction with @OneToMany			
(name="pred-name")	or @ManyToOne to specify a map-			
	ping realized through an associative predicate			

Non-positional Notation & Dynamic Composition

- O Access attributes by name → compact and intuitive rules
- Compose ASP programs at runtime

```
public void createTeam(boolean forceMixG) {
   List<Person> personList = loadPeople();
   <#+ (m1) in=personList::people</pre>
      inTeam(X,G) v outTeam(X,G) :-
                    people(name: X, gender:G).
      :- #count{X: inTeam(X,G)} >5.
   if(forceMixG) {
    + <#+ (m1)</p>
      :- inTeam(X,GX), not inTeam(Y, GY),
       people(name: Y, gender: GY), GX != GY.
     #>}
   Set<Team> res = new HashSet<Team >();
   <# (m1) out=res::inTeam #>
   for each answerset
   //do something with res
   } }
```

Access Java variables



Database Access

- Support data intensive applications \rightarrow database access
- Interact directly with DBMSs, possibly exploit DLV^{DB}

```
<# fromDB=Arc::arc</pre>
```

reachable(X,Y) :- arc(X,Y).
reachable(X,Y) :- arc(X,U), reachable(U,Y).
reaches(Y) :- reachable("Rome",Y).
#>

• Access tables fromTable = Table@DBURL :: predname

イロト 不得 とくほ とくほう 二日

Database Access

- Support data intensive applications \rightarrow database access
- Interact directly with DBMSs, possibly exploit DLV^{DB}

```
<# fromDB=Arc::arc
```

reachable(X,Y) := arc(X,Y).
reachable(X,Y) := arc(X,U), reachable(U,Y).
reaches(Y) := reachable("Rome",Y).
#>

• Access tables fromTable = Table@DBURL :: predname

・ロト ・ 同ト ・ ヨト ・ ヨト … ヨ

Implementation

• The Jdlvc compiler

- generates plain Java classes from \mathcal{JASP} files
- support ASP-core and DLV syntax
- Implemented in Java, based on the DLV Wrapper API

• The JDLV Eclipse plug-in

- Editing Features
 - automatic completion
 - dynamic code checking and errors highlighting
 - outline view
- Includes Jdlvc
 - *automatic generation of Java code*, by means of our *Jdlvc* compiler.

イロト 不得 とくほ とくほう 二日

• Java and ASP on the same environment!

Compiler and Generated Code Performance









Compiler Performance

File name	Size(KB)	Size (LOC)	DLV calls	Compile time
SokobanDecision.jdlv 🖉	9	205	1	70 ms
WeightAssignmentTree.jdlv 🔗	4	123	1	30 ms
TwentyCalls.jdlv 🗗	47	1509	20	51 ms
Huge.jdlv 🔗	7456	188938	1	4871 ms

ヘロト ヘワト ヘビト ヘビト

э

Use-case: Workforce Management via ASP

• The Gioia Tauro seaport

- the largest transshipment terminal of the Mediterranean Sea
- main activity: container transshipment [Vacca et. al]
- recently become an *automobile hub*

Automobile Logistics by ICO B.L.G. (subsidiary of 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 (contract, required skills, fair distribution of load)

Designed a Team Building system with ASP

ヘロア 人間 アメヨア 人口 ア

\mathcal{JASP} ORM Mapping

```
@Entity public class Employee
{
   GId Integer id;
    String name;
   GOneToMany
   GJoinTable(name="hasSkill")
    Set<Skill> skills;
}
```

```
@Entity
public class AllocationDiary
{
    @Id Integer id;
    @OneToOne Employee employee;
    @OneToOne Skill skill;
        Date date;
}
```

```
@Entity
public class Allocation
{
  @Id Integer id;
  @OneToOne Employee employee;
  @OneToOne Shift shift;
  @OneToOne Skill skill;
```

@Entity public class Calendar
{
@Id Integer id;
@OneToOne Employee employee;
 Date date;
 Boolean isAbsent;
 Integer dayHours,weekHours,
 weekOvertime;

@Entity public class Skill
{
 @Id String name;
 Boolean isCrucial,isHeavy;
}

```
@Entity public class Shift
```

@Id Integer id; Date date; Integer duration;

@OneToMany
Map<Skill,Integer> neededEmp;
@OneToMany
@JoinTable(name="excluded")
Set<Employee> excluded;
}

イロン 不良 とくほう 不良 とうせい

\mathcal{JASP} Encoding

```
public Allocation computeTeam(Shift s,
      Set<Employee> e, Integer maxGap) {
Set<Allocation> teams = Sets.newHashSet();
<#+(teambuilding)
   fromDB=AllocationDiary::lastAllocation,
         Calendar:0
   in=s::shift, e::employee
   out=teams::assign
 assign(employee:Em, shift:Sh, skill:Sk)
      v nAssign(Em,Sh,Sk) :-
                  canBeAssigned(Em,Sh,Sk).
 canBeAssigned(Em,Sh,Sk) :-
                [e.hasSkill] (Em,Sk),
                [s.neededEmp](Sh,Sk,_),
               not exceedTimeLimit(Em,Sh),
  not absent(Em,Sh), not [s.excluded](Sh,Em).
 absent(E,Sh) := shift(id:Sh, date:D),
   Calendar(date:D, employee:E, absent:true).
#>
```

イロト 不得 トイヨト イヨト 二日 二



DEMO

- Application to Workforce Management
- 2 Modify a simple program

Conclusion

On the field experience \rightarrow Development Tools

- ASPIDE: The complete IDE for ASP
 - Advanced editing features... extensible via plugins
 - \rightarrow textual/graphical editing tool, debugging, testing, profiling, etc.
 - ightarrow supports the entire life-cycle of logic programs development
 - http://www.mat.unical.it/ricca/aspide

A framework for integrating Java and ASP

- The \mathcal{JASP} language
 - \rightarrow ASP code in-lined within Java programs
 - \rightarrow complies with application development standards (i.e., JPA)
- JDLV : a plugin for the Eclipse platform
 - \rightarrow http://www.dlvsystem.com.it/jdlv

・ロト ・ 同 ト ・ ヨ ト ・ ヨ ト

Related Work

API-Based approaches

• JSetL [Rossi et al. '07], ILOG, Jess, ...

• Multi-paradigm Languages [Hailpern 1986; Placer 1991]

- Alma-0 [Apt et al. '98], DJ [Zhou '99],
- Oz [Roy 2005], PROVA [Koslenkov et al. '06],
- etc. [Spinellis 1994]

ASP and Java

- DLVWrapper [Ricca 2003]
- OntoDLV API [Gallucci and Ricca 2007]
- Recent proposal of an hybrid language [Oetsc et al. 2011]

イロト 不得 とくほ とくほう 二日