Universidad Rey Juan Carlos

CETINIA
Centre for Intelligent Information Technologies

# Value Awareness Engineering

## At CETINIA URJC

**Joaquín Arias**

**Centre for Intelligent Information Technologies (CETINIA)**

**University Rey Juan Carlos, Madrid (Spain)**

**Workshop on Ethical and Trustworthy AI,**
**Fundación Ramón Areces, Madrid**

*30/01/2024*

# Value Awareness Engineering (VAE)

- General idea:
  - ✓ Ensuring that artificial systems respect, decide and act according to our human values
  - ✓ Develop methods and techniques for a computational approach to value awareness
    - ✓ that allow system to formally reason about values, and the alignemnt of their decisions with respect to those values

# Our work in the context of Value Awareness Engineering

*Formal Models of Value Systems*

- Value alignment, aggregation of values /systems
- Context dependence of alignment and aggregation

*Optimal policies with Value alignment constraints*

*Value correlation in assessing value alignment*

*VAE ontology*

*Learning context-sensitive value systems of agents*

*Computational representations*

Rule-based value system representation (**reasoning: ASP**)

Value systems as optimization problems (**reasoning: Math. Opt.**)

*Queries & explanations: ASP traces*

*Learning: ILP*

*Value alignment in multiagent task allocation and vehicle routing*

*Use Cases*

"Legal" domains

Agricultural cooperatives

Emergency Medical Assistance

*School place assignment*

*Routing of agricultural vehicle fleets*

*Fair allocation of emergency resources*

# Previous work: s(LAW)

- s(LAW) framework for computational legal reasoning:
  - ✓ Based on s(CASP) non-monotonic reasoner: applies top-down evaluation of Answer Set Programs (ASP) with constraints [Arias et al.]
  - ✓ Patterns to translate legal text into ASP
  - ✓ Natural language patterns to allow for human-understandable justifications
- Characteristics of the representation language:
  - ✓ Positive and negative evidence (strong negation)
  - ✓ Exceptions: negation as failure
  - ✓ Even loop: generate alternative models
  - ✓ Constraints: linear equations over rationals/reals

# Previous work: s(LAW)

- Example: Assigning school places in the Region of Madrid
  - ✓ General rules: "for a child to obtain a school place a general (*large family*, *disability*) and a specific requirement (*school proximity*, …) need to be met"
  - ✓ Exceptions: "students coming from non-bilingual public schools, who apply for a place in English language bilingual schools, need to accredit a level of English equivalent to level B1 for 1st/2nd ESO, and to level B2 for 3rd/4th ESO"
  - ✓ Ambiguity: "*school proximity* requires living in the same educational district, unless *force majeure* applies"
  - ✓ Discretion to act: "the school council can add complementary criteria", if the discretion is line with the purpose/intention of the law (promotes diversity) and is not unlawful (e.g. no discrimination)
  - ✓ Absence of information: it may be unclear whether the documents presented accredit a *large family* or not

# s(LAW) framework: school place assignment example

```
%% Obtain a school place if...
obtain_place :-
    met_requirement,
    not exception.
met_requirement :-
    met_common_requirement,
    met_specific_requirement.
%% Common requirements:
met_common_requirement :-
    large_family.

met_common_requirement :-
    recipient_social_benefits.
recipient_social_benefits :-
    renta_minima_insercion.
recipient_social_benefits :-
    ingreso_minimo_vital.

met_common_requirement :-
    disability_status.
disability_status :-
    disabled_parent.
disability_status :-
    disabled_sibling.
%% Specific requirements:
met_specific_requirement :-
    sibling_enroll_center.
met_specific_requirement :-
    legal_guardian_work_center.
met_specific_requirement :-
    relative_former_student.
met_specific_requirement :-
    school_proximity.
school_proximity :-
    same_education_district.
school_proximity :-
    not same_education_district,
    force_majeure.     % Ambiguity

force_majeure :-
    not n_force_majeure.
n_force_majeure :-
    not force_majeure.

%% Exceptions:
exception :-
    come_non_bilingual,
    want_bilingual_section(Course),
    not accredit_english_level(Course).

accredit_english_level('1st ESO') :-
    b1_certificate.
accredit_english_level('2nd ESO') :-
    b1_certificate.
accredit_english_level('3rd ESO') :-
    b2_certificate.
accredit_english_level('4th ESO') :-
    b2_certificate.
%% Discretion To Act:
obtain_place :-
    not met_requirement,
    met_complementary_criterion(CC).
obtain_place :-
    met_requirement, exception,
    met_complementary_criterion(CC).

met_complementary_criterion(CC) :-
    school_criteria(CC),
    purpose(CC), not unlawful(CC),
    not n_met_complementary_criterion(CC).
n_met_complementary_criterion(CC) :-
    not met_complementary_criterion(CC).

purpose(CC) :- promote_diversity(CC).
unlawful(CC) :- sex_discrimination(CC).
unlawful(CC) :- race_discrimination(CC).
unlawful(CC) :- religion_discrimination(CC).

school_criteria(foreign_student) :-
    foreign_student.
school_criteria(specific_etnia) :-
    specific_etnia.

promote_diversity(foreign_student).
promote_diversity(specific_etnia).
race_discrimination(specific_etnia).
```

# Explainability in s(Law)

- s(LAW) models are (partially) "**self-explainatory**": ASP proof trees
- Example: school place assignment with s(LAW)

Case description (student 1):

```
come_non_bilingual.
want_bilingual_section('2nd ESO').

evidence(large_family).
evidence(renta_minima_insercion).
evidence(sibling_enroll_center).
evidence(same_education_district).
evidence(b1_certificate).
-evidence(foreign_student).
-evidence(specific_etnia).
```

Query: `? Obtain_place`

Result (model fulfilling the query):

```
{ obtain_place, large_family, sibling_enroll_center, come_non_bilingual,
  want_bilingual_section(2nd ESO), b1_certificate }
```

Justification:

```
1  s/he may obtain a school place, because
2      a common requirement is met, because
3          s/he is part of a large family.
4      a specific requirement is met, because
5          s/he has siblings enrolled in the center.
6      there is no evidence that an exception applies, because
7          s/he came from a non-bilingual public school, and
8          s/he wish to study 2nd ESO in the Bilingual Section, and
9          s/he accredit required level of English for 2nd ESO, because
10             in the four skills certificate level b1.
```

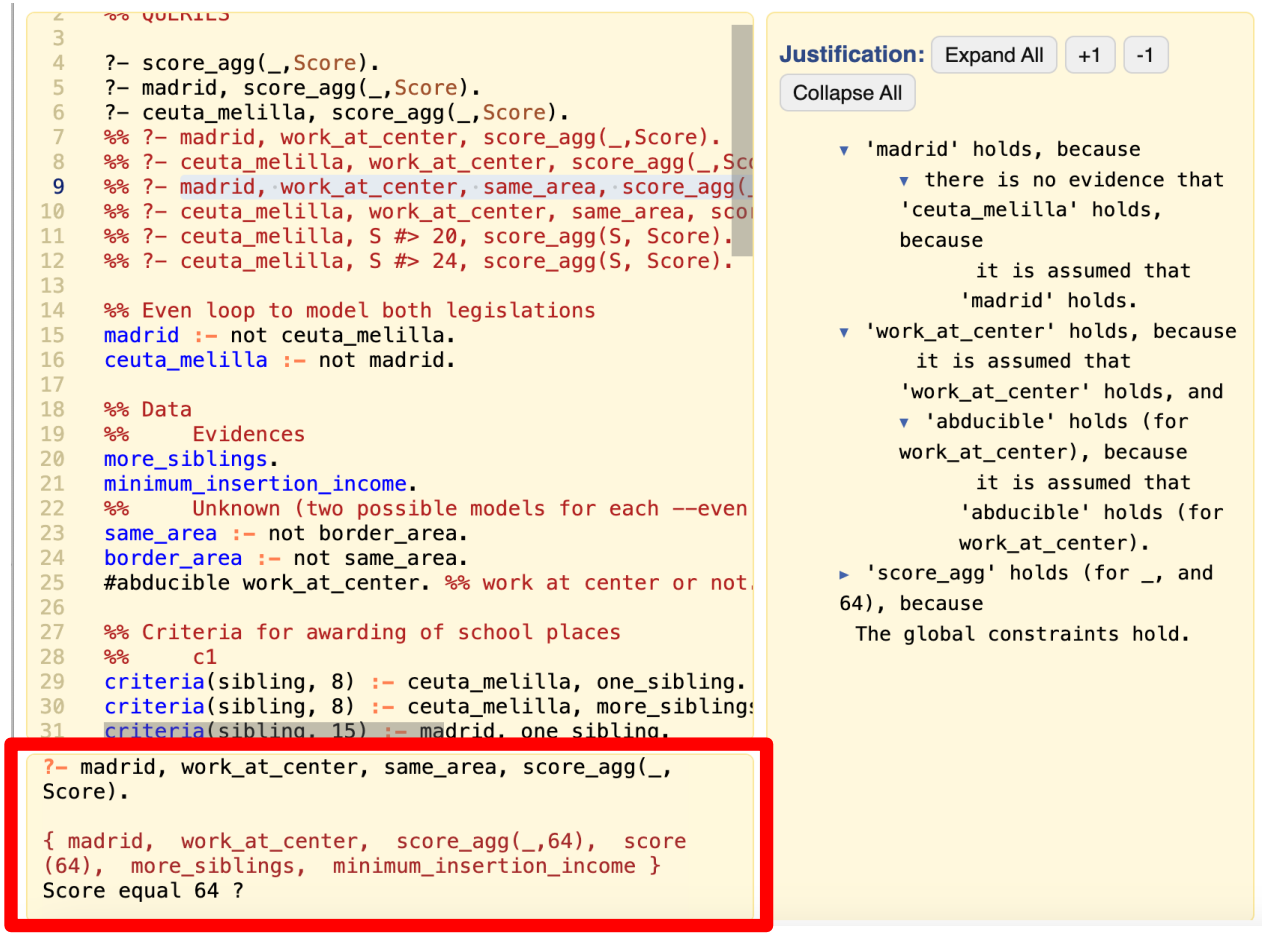
# Current work: Comparing school place assignment models

- Principle of educational equality: independence of wealth, race, religion, etc.
- Different school place assignment procedures:
  - ✓ zoning, open enrolment, lottery, reservations, …
  - ✓ Different procedures (i.e., the corresponding legislation) promotes different values
    - ➢ *Zoning*: promotes equality (avoids segregation / "ghettos")
    - ➢ *Single district*: promotes liberty (freedom of choice) / quality (competition)
- Example: assignment procedures in Spain
  - ✓ Nationwide score system: different "calibrations"
  - ✓ Madrid: Single district / Ceuta & Melilla: Zoning

| PRIORITY CRITERIA | Ceuta and Melilla | Madrid |
|---|---|---|
| Existence of siblings enrolled | 8 | 15 30 (two or more siblings) |
| Proximity to the home or place of work of a parent or legal guardian: Family home located within the catchment area in which the requested center is located | 10 | 12 (within the same municipality) + 1 (In the municipality of Madrid, if the family domicile is in the same municipal district) |
| Proximity to the home or place of work of a parent or legal guardian: Family home located within the catchment area in which the requested center is located | 8 | 12 (within the same municipality) + 1 (In the municipality of Madrid, if the family domicile is in the same municipal district) |
| Proximity of the domicile or place of work of any of the parents or legal guardians: If any of them is located in the areas bordering the area of influence in which the requested center is located | 2 | 8 (municipality other than the one in which the school is located) |
| Per capita income of the family unit: Income equal to or less than the minimum interprofessional salary | 4 | 0 |
| Per capita income of the family unit: Income between one and two times the minimum wage | 2 | 0 |
| Per capita income of the family unit: Fathers, mothers or legal guardians receiving the Minimum Insertion Income (excludes the previous two) | 6 | 12 (minimum insertion income or minimum vital income) |
| **COMPLEMENTARY CRITERIA** | **Ceuta and Melilla** | **Madrid** |
| Fathers, mothers or legal guardians working at the center | 4 | 10 |
| Concurrence of disability (student, siblings, parents or legal guardians): Disability in the student him/herself from 33 % | 2 | 7 (max 7 points for the disability section, no distinction is made in the following cases) |
| Concurrence of disability (student, siblings, parents or legal guardians): Disability in the student him/herself from 33 % | 1 | |
| Status as a victim of gender violence. | 1 | 2 |
| Status as a victim of terrorism | 1 | 2 |
| Transfer of the family unit due to the forced mobility of any of the parents or legal guardians. | 1 | [Excluded, preferentially attributed] |
| Legal status as a large family: Special status | 2 | 11 (computes conceived, unborn) |
| Legal status as a large family: Special status | 1 | 10 (computes conceived, unborn) |
| Single-parent family | 1 | 3 |
| Foster care status of students. | 1 | 3 |
| Students born of multiple births: Birth of two children. | 1 | 3 (max 3 points for multiple births) |
| Students born of multiple births | 1 (+1 per child) | |
| Consideration of the student as a high-level or high-performance athlete: High-level athlete. | 2 | 0 |
| Former student status of the student himself/herself, parents, legal guardians or any of the applicant's siblings, in the center for which he/she is applying for a place. | [Criteria decided by each center] | 4 |

# 1. Automate the allocation of school places

- ✓ Given a **score system** and (possibly partial) information on **student characteristics**
- ✓ automate the process of awarding places, i.e. determine the **student's scores**



```
%% QUERIES

?- score_agg(_,Score).
?- madrid, score_agg(_,Score).
?- ceuta_melilla, score_agg(_,Score).
%% ?- madrid, work_at_center, score_agg(_,Score).
%% ?- ceuta_melilla, work_at_center, score_agg(_,Sc
%% ?- madrid, work_at_center, same_area, score_agg(
%% ?- ceuta_melilla, work_at_center, same_area, sco
%% ?- ceuta_melilla, S #> 20, score_agg(S, Score).
%% ?- ceuta_melilla, S #> 24, score_agg(S, Score).

%% Even loop to model both legislations
madrid :- not ceuta_melilla.
ceuta_melilla :- not madrid.

%% Data
%%     Evidences
more_siblings.
minimum_insertion_income.
%%     Unknown (two possible models for each --even
same_area :- not border_area.
border_area :- not same_area.
#abducible work_at_center. %% work at center or not.

%% Criteria for awarding of school places
%%     c1
criteria(sibling, 8) :- ceuta_melilla, one_sibling.
criteria(sibling, 8) :- ceuta_melilla, more_siblings
criteria(sibling, 15) :- madrid, one_sibling.
```

```
?- madrid, work_at_center, same_area, score_agg(_,
Score).

{ madrid,  work_at_center,  score_agg(_,64),  score
(64),  more_siblings,  minimum_insertion_income }
Score equal 64 ?
```

**Justification:** Expand All +1 -1 Collapse All

- ▾ 'madrid' holds, because
  - ▾ there is no evidence that 'ceuta_melilla' holds, because
    - it is assumed that 'madrid' holds.
- ▾ 'work_at_center' holds, because
  - it is assumed that 'work_at_center' holds, and
  - ▾ 'abducible' holds (for work_at_center), because
    - it is assumed that 'abducible' holds (for work_at_center).
- ▸ 'score_agg' holds (for _, and 64), because
  - The global constraints hold.

# 1. Automate the allocation of school places

- ✓ Obtaining intervals of possible scores, depending on available evidence
  - ➢ Augmented transparency and explainability

```
?- ceuta_melilla, work_at_center, same_area, score_agg
(_,Score).

{ ceuta_melilla,  work_at_center,  score_agg(_,
Score | {Score #>= 20,Score #=< 28}),  score(28),
more_siblings,  same_area,
minimum_insertion_income,  score(20),  border_area }
Score greater or equal 20, and less or equal 28 ?
```

```
?- madrid, score_agg(_,Score).

{ madrid,  score_agg(_,Score | {Score #>= 54,Score
#=< 64}),  score(64),  more_siblings,
minimum_insertion_income,  work_at_center,  score
(54) }
Score greater or equal 54, and less or equal 64 ?
```

# 2. Compare the value alignment of different norms

- Given:
  - ✓ Various score systems
  - ✓ Assignments of students to schools for those systems
  - ✓ Grounding of relevant values on outcomes:
    - ✓ Non-segregation: distribution of low-income students among the schools (e.g., Gini index)
    - ✓ Freedom of choice: proportion of students assigned to the desired school
- Determine
  - ✓ Which system is better aligned with respect to the different values

- We are trying to get real data (but administrations are reluctant to support)

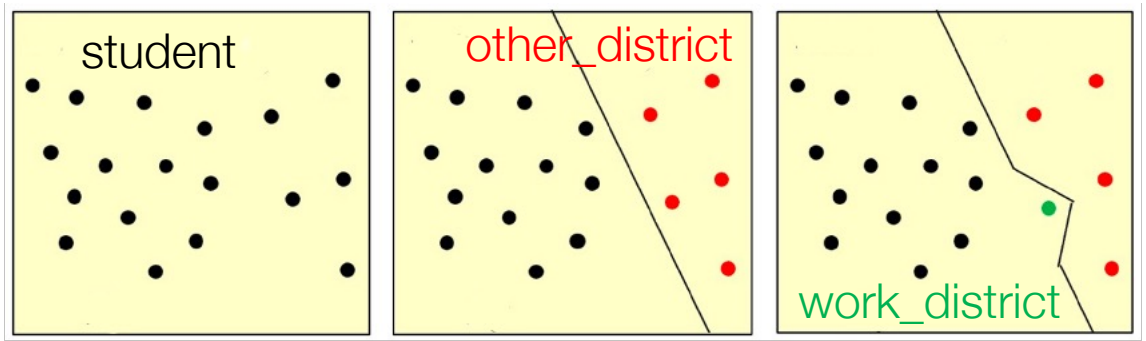
# 3. Adapt norms according to desired values

- Given:
  - ✓ A general framework for assigning school places
    - ✓ Scoring criteria
  - ✓ Examples of desired outcomes
    - ✓ “Value aligned” assignments of students
  - ✓ Grounding of relevant values on outcomes:
    - ✓ Non-segregation, Freedom of choice, ...
- Determine
  - ✓ The scores that would lead to the desired outcomes

# 3. Adapt norms according to desired values

- Provides possibility to find admissible score ranges wrt. admissible value alignment:
  - ✓ e.g.: “the number students with low-income in a school” should not exceed 20%
- Looking into **ILP** to learn or adjust normative systems
  - ✓ Exploiting **existing domain knowledge**
    - ✓ **Given** general rules …
    - ✓ …identify exceptions that increase value alignment

```
1   obtain_place(X) :- student(X), not exception(X).
2   exception(X) :- ohter_district(X), not district_exception(X).
3   district_exception(X) :- tutor(X,Y), work_district(Y).
```



# Current work: “Forgetting what we want to forget”

- s(LAW) models are (partially) “**self-explainatory**” (ASP proof trees):

```
1  s/he may obtain a school place, because
2      a common requirement is met, because
3          s/he is part of a large family.
4      a specific requirement is met, because
5          s/he has siblings enrolled in the center.
6      there is no evidence that an exception applies, because
7          s/he came from a non-bilingual public school, and
8          s/he wish to study 2nd ESO in the Bilingual Section, and
9          s/he accredit required level of English for 2nd ESO, because
10             in the four skills certificate level b1.
```

- However: Justifications may expose sensitive information (e.g., data on gender violence).
- Solution: Manipulate the justifications and/or apply forgetting
  - a syntactic transformation that forgets predicates in ASP programs

# Current work: “Forgetting what we want to forget”

- Implementation of an algorithm that:
  - ✓ Eliminates “sensitive predicates” from an ASP program without affecting its semantic
  - ✓ Example:

Justifications for the query `?- s.`

Initial program

```
% Model {s,p}
s :-
  p :-
    not q :-
      not r :-
        chs(s).
      neg_a :-
        chs(not q).
```

Forgetting `p` and `q`

```
% Model {s}
s :-
  not r :-
    chs(s).
  not neg_b :-
    neg_a :-
      proved(not r),
      chs(not neg_b).
```

# Current work: “Forgetting what we want to forget”

- Forgetting can also improve explainability in ILP:

Given a school allocation database, the algorithm FOLD-R++ learns:

```
obtain_p(yes) :- large_f(yes), not ab3, not ab1.
ab1 :- come_non_b(yes), want_b_s(yes), not b1_c(yes).
ab2 :- same_education_d(yes), not ab1.
ab3 :- not sibling_enroll_c(yes), not ab2.
```

After forgetting the predicates ab1, ab2 and ab3, we obtain:

```
obtain_p(yes) :- large_f(yes), sibling_enroll_c(yes), not come_non_b(yes).
obtain_p(yes) :- large_f(yes), sibling_enroll_c(yes), not want_b_s(yes).
obtain_p(yes) :- large_f(yes), sibling_enroll_c(yes), b1_c(yes).
obtain_p(yes) :- large_f(yes), same_education_d(yes), not come_non_b(yes).
obtain_p(yes) :- large_f(yes), same_education_d(yes), not want_b_s(yes).
obtain_p(yes) :- large_f(yes), same_education_d(yes), b1_c(yes).
obtain_p(yes) :- large_f(yes), same_education_d(yes), not come_non_b(yes), b1_c(yes).
obtain_p(yes) :- large_f(yes), same_education_d(yes), not want_b_s(yes), not come_non_b(yes).
obtain_p(yes) :- large_f(yes), same_education_d(yes), b1_c(yes), not want_b_s(yes).
```

Universidad Rey Juan Carlos

CETINIA
Centre for Intelligent Information Technologies

# Value Awareness Engineering

## At CETINIA URJC

**Work in collaboration with**

Members of the AI group at URJC