Ontology-based Case-Based Reasoning (OntCBR) for Engineering Design

Hyowon Suh & Jae Hyun Lee
KAIST

April 25. 2008

JSME Society of Design Research

Hyowon Suh & Jae Hyun Lee
Guest Researhers
Design Research Group
Manufacturing Engineering Lab (MEL)
NIST, USA

Ontology-Related Research

Contents

- **Ontology-based Mapping between CAD and PDM [2006]**

  Semantic Mapping Based On Ontology and A Bayesian Network and Its Application to CAD and PDM Integration, Min-jung Lee, Min Jung and Hyo-won Suh, ASME International Design Engineering Technical Conferences (DETC) & Computers and Information in Engineering Conference (CIE), September 10-13, 2006, Philadelphia, Pennsylvania, USA.

- **Ontology-based Search for Collaboration [2007]**

  Min Jung, Jahyun Lee, Juhoon Nam and Hyowon Suh, "Ontology Mapping-based Search with Multi-dimensional Similarity and Bayesian Network", Design Engineering Workshop/7th IJCC Japan-Korea CAD/CAM Workshop, July 26-27, 2007, RCAST, The University of Tokyo, Komaba, Meguro, Japan, p39-44.

- **Ontology-based Multi-layered Knowledge Framework [2006]**

  Jae-Hyun Lee and Hyo-Won Suh, 2006, submitted "Ontology-based Multi-layered Knowledge Framework for Product Lifecycle Management", Concurrent Engineering-Research and Application (to be appear)

- **Ontology-based Multi-layered Knowledge Framework for Robot Context [2007]**

  Il Hong Suh, Gi Hyun Lim, Wonil Hwang, Hyowon Suh,, Jung-Hwa Choi, Young-Tack Park, "Ontology-based Multi-layered Robot Knowledge Framework (OMRKF) for Robot Intelligence", 2007 IEEE/RSJ International Conference on Intelligent Robots and Systems, Oct 29 - Nov 2, 2007, Sheraton Hotel, San Diego, CA, USA

- **OWL–based Product Ontology (POWL) [2007]**
  ASME ROBERT E. FULTON EIM BEST PAPER AWARD

  Jae-Hyun Lee and Hyowon Suh, "OWL-based Product Ontology (POWL) Architecture and Representation for Sharing Product Knowledge Sharing on A Web", Proceedings of IDETC / CIE 2007 (ASME 2007 International Design Engineering Technical Conferences Computers and Information in Engineering Conference), Las Vegas, Nevada, USA, September 4-7, 2007.

- **Ontology-based Case-Based Reasoning [2008]**

Contents

Ontology - Definition

a formal and explicit specification of a shared conceptualization of a domain of interest.

From T.R. Gruber

T. R. Gruber, “What is an ontology?,” http://www-ksl.stanford.edu/kst/what-is-an-ontology.html, Stanford University, 1993.

Ontology – Term’s definition

Term Sharing

“Employee”

Informally Semantic Sharing

“Employee” **a person who works at organization**

“An employee is a person who is paid to work for an organization or for another person [Naver] “

An employee contributes labour and expertise to an endeavour. Employees perform the discrete activity of economic production. Of the three factors of production, employees usually provide the labour.
Specifically, an **employee** is any person hired by an employer to do a specific "job". In most modern economies the term employee refers to a specific defined relationship between an individual and a corporation, which differs from those of customer, or client. [Wikipedia]

Formally Machine Understandable Sharing

DL: $\text{Employee} \equiv \text{person} \sqcap \exists \text{workAt.Organization}$

FOL: $\forall x\ \text{Employee}(x) \Leftrightarrow \text{person}(x) \wedge \exists y\ \text{workAt}(x,y) \wedge \text{organization}(y)$

DL: description logic, FOL: first-order logic

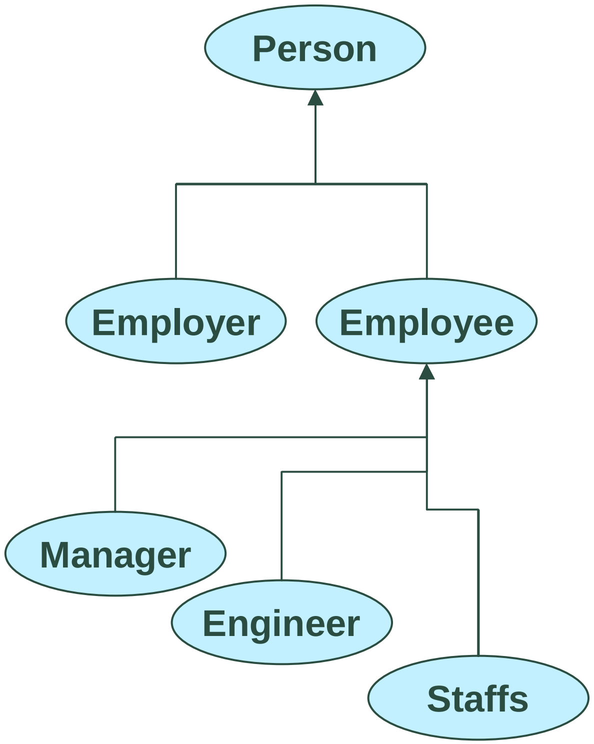
DL: F. Baader, D. Calvanese, D.L. McGuinness, D. Nardi and P.F. Pater-Schneider, “The Description Logic Handbook: Theory, Implementation, and Applications”, Cambridge University Press, 2003.

FOL: Russell and P. Norvig, “Artificial Intelligence”, Prentice Hall, 1995.

Ontology – Mapping/Similarity

Matching	"Employee"	"Worker"
Thesaurus	"Employee" : worker, labourer, workman, staff member, member of staff, hand, wage-earner, white-collar worker.	"Worker" : employee, hand, labourer, workman, craftsman, artisan, tradesman
Dictionary	"Employee" a person who is paid to work for an organization or another person	"Worker" a person who is employed in industry or business and who is not a manager.
Logic	Employee ≡ person ⊓ ∃workAt.Organization	Worker ≡ person ⊓ ∃employedIn.Business

Ontology - Taxonomy



Person

Employee ≡ person ⊓ ∃workAt.Organization

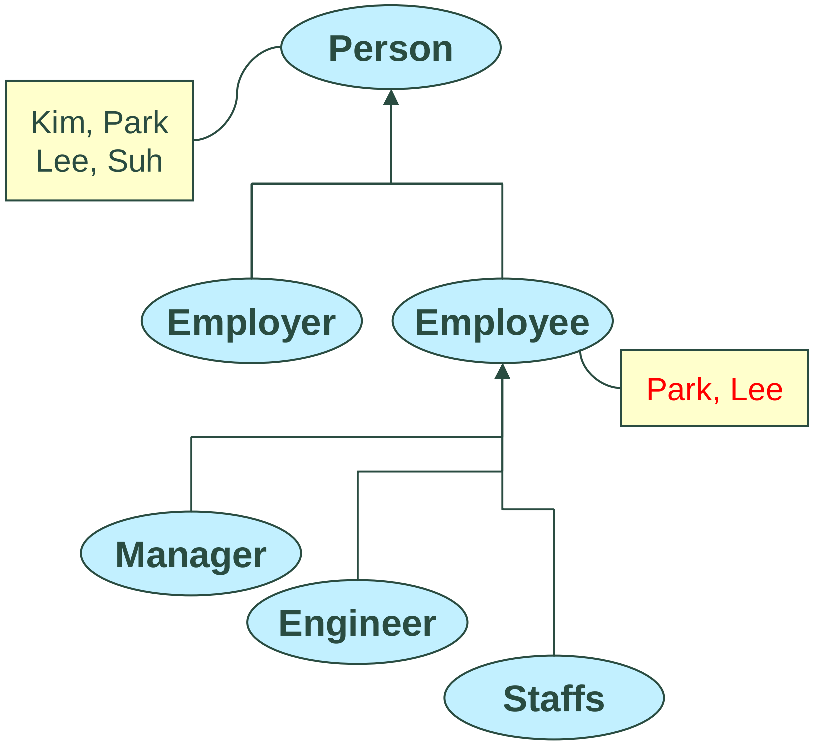
⊂ person

Manager ≡ Employee ⊓ ∃manage.Engineer

⊂ employee
⊂ person

DL: F. Baader, D. Calvanese, D.L. McGuinness, D. Nardi and P.F. Pater-Schneider, "The Description Logic Handbook: Theory, Implementation, and Applications", Cambridge University Press, 2003.

Taxonomy



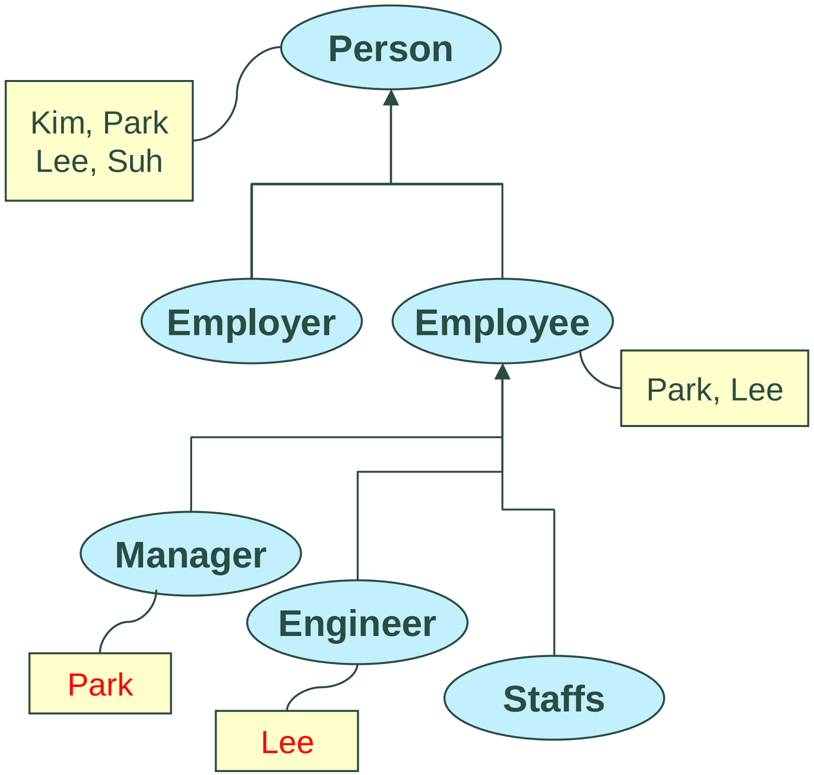
Person (Park)
Person (Lee)
Organization(Samsung)
workAt (Park, Samsung)
workAt (Lee, Samsung)

Employee ≡ person ⊓ ∃workAt.Organization

Employee (Park)
Employee (Lee)

DL: F. Baader, D. Calvanese, D.L. McGuinness, D. Nardi and P.F. Pater-Schneider, "The Description Logic Handbook: Theory, Implementation, and Applications", Cambridge University Press, 2003.

Taxonomy



Person (Park)
Person (Lee)
Organization(Samsung)
workAt (Park, Samsung)
workAt (Lee, Samsung)

Employee ≡ person ⊓ ∃workAt.Organization

Employee (Park)
Employee (Lee)
manage (Park, Lee)
hasJob(Lee, Design)
Engineering(Design)

Engineer ≡ employee ⊓ ∃hasJob.Engineering
Engineer (Lee)

Manager ≡ Employee ⊓ ∃manage.Engineer
Manager (Park)

DL: F. Baader, D. Calvanese, D.L. McGuinness, D. Nardi and P.F. Pater-Schneider, "The Description Logic Handbook: Theory, Implementation, and Applications", Cambridge University Press, 2003.

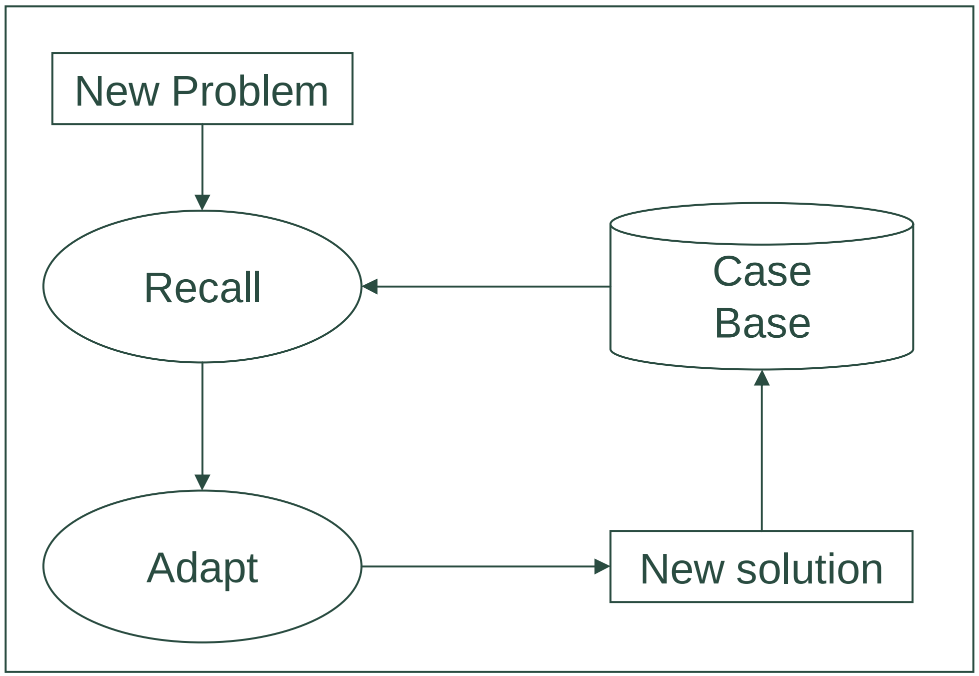
Contents

1. Introduction

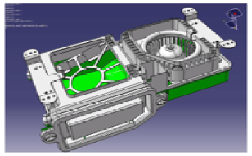
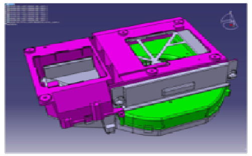
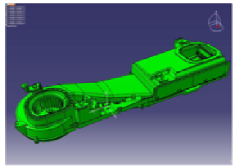
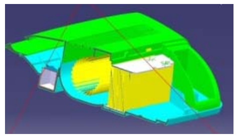
Case-Based Reasoning

New Problem

Simple model of case-based reasoning



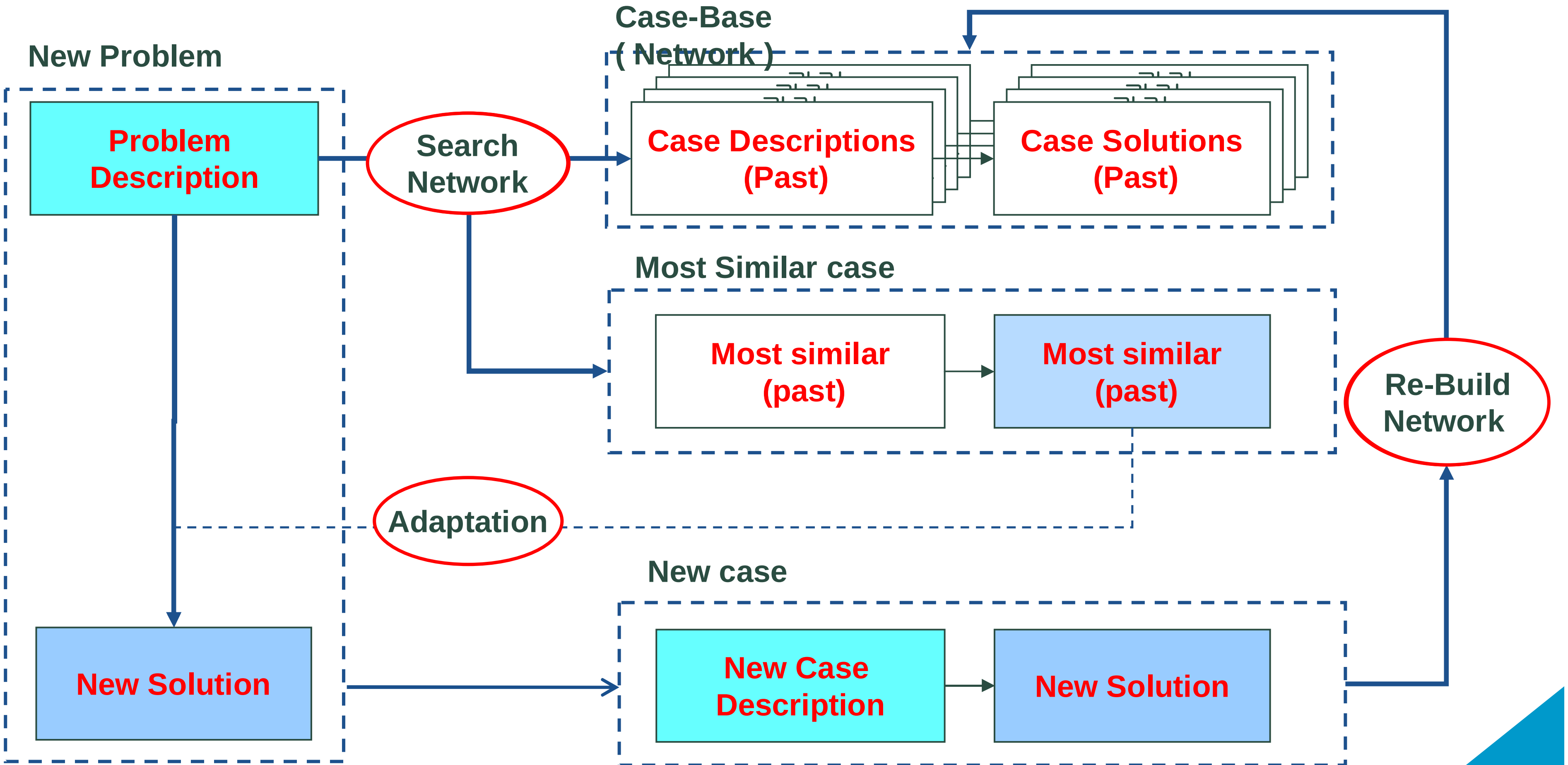
Case-base

Cases	Solutions
Previous case 1	
Previous case 2	
Previous case 3	
Previous case 4	

Maher, M.L., Balachandran, M.B. and Zhang, D.M., "Case-based reasoning in design", Lawrence erlbaum associates, 1995.

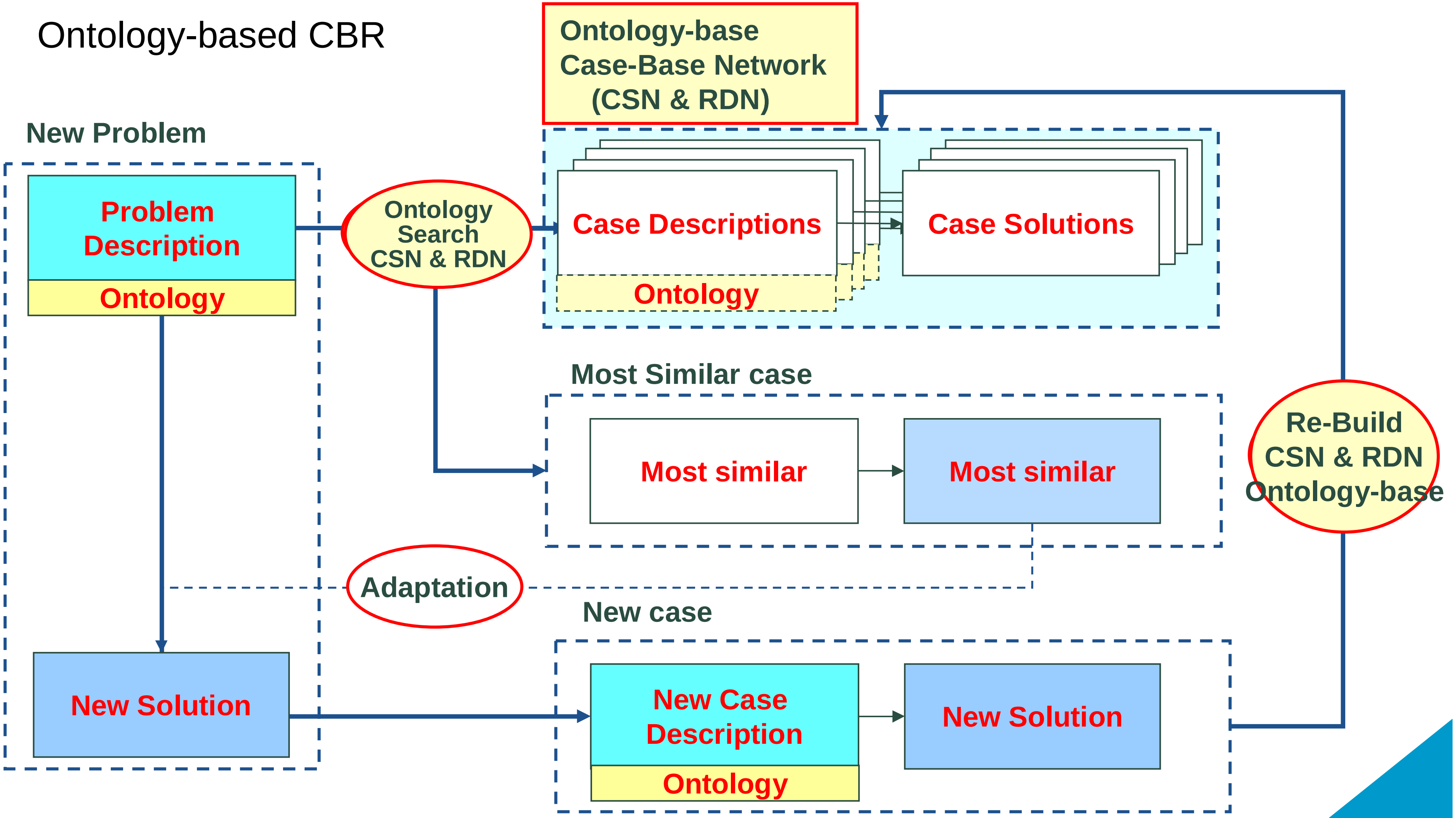
1. Introduction

Case-Based Reasoning



1. Introduction

Ontology-based CBR



2. Previous Research

Prev. Research	Problem description Case Rep.	Cases Re-Organizing	Similarity Calculation	Reasoning Method
Archie-II (1997)	Text	List of shared attr.	Attr-val Sim.	-
JULIA (1991)	Attr-value	Hierarchy str.	Attr-val Sim.	Constraint satisfaction
DDIS (1991)	Attr-value	Hierarchy Str.	Attr-val Sim.	Rule-based reasoning
CADRE (1992)	Graph	-	Attr-val Sim.	Constraint satisfaction, Rule-based reasoning
KRITIK (1989)	Text	List of shared attr.	Attr-val Sim.	Qualitative reasoning
CADET (1989)	Graph	List of shared attr.	Graph Sim	Qualitative reasoning
Roseman and *et al.* (1991)	Object-oriented	-	-	instantiation
CYRUS (1984)	Attr-value	RDN	Attr-val Sim.	-
Memoire (2006)	Ontology (OWL)	-	-	Axiom reasoning
Wriggers and *et al*. (2007)	Ontology (OWL)	Concept vector list	Graph Sim	-
Wang and *et al*. (2003)	Ontology	-	Graph Sim	-
OntCBR (Proposed)	Ontology (DL & HL)	CSN* & RDN	Ontology-based Graph Sim.	Taxonomy reasoning Axiom reasoning

DL: Description Logic
HL: Horn Logic
CSN: Concept-Structure Network
RDN: Redundant Discrimination Network

2. Previous Research

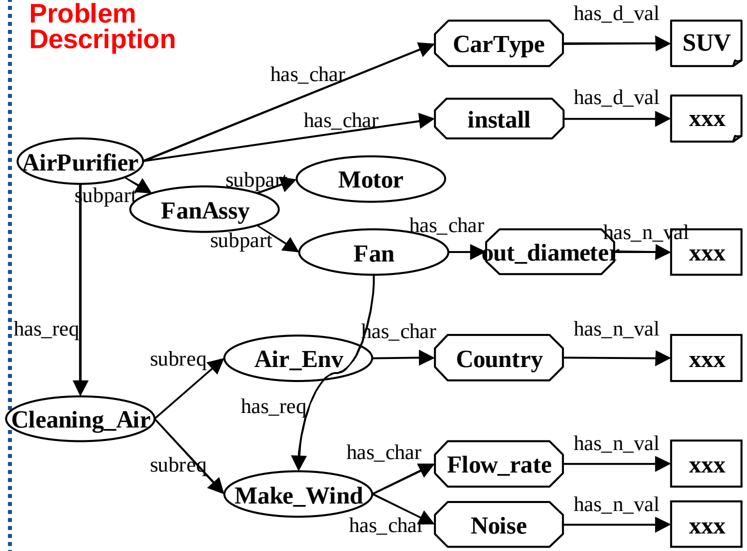
CBR

- Case Representation
  - Text
  - Attribute-value list
  - Object-oriented
  - Graph
  - Graph + Ontology
- Case-base organization method
  - For attribute-value list reprsentation
    - Flat structure
    - List structure of shared attribute
    - Hierarchy structure of shared attributes
    - Redundant discrimination network (RDN)
  - For graph representation
    - Concept vector list structure
    - Concept structure network (CSN)
- Case similarity evaluation method
  - For attribute-value list representation
    - attribute and value term's similarity
  - For graph representation
    - concept vector similarity
    - graph similarity
    - ontology-based graph similarity

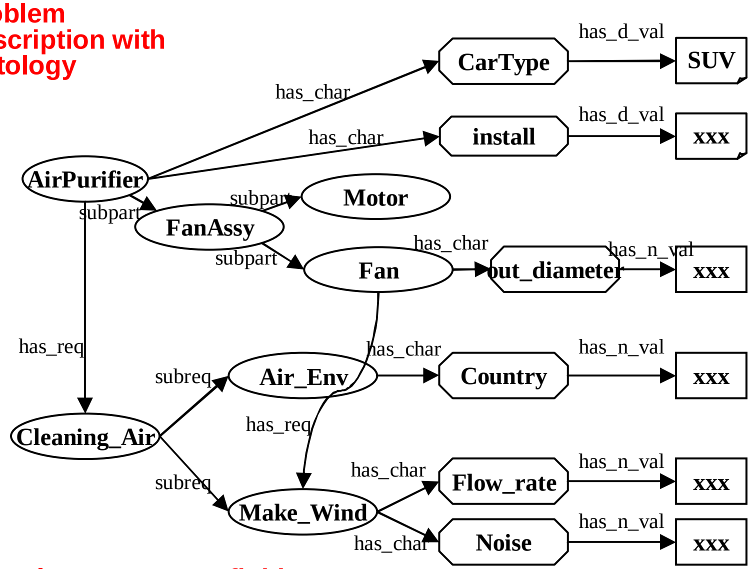
3. Proposed Approach

Problem Description + Ontology

Problem Description



Problem Description with Ontology



Ontology: Terms Definitions

Axioms

[DL] has-subpart⁺

[HL] x,y,z has_subpart(x,y) ∧ has_subpart(y,z) → has_subpart(x,z)

…

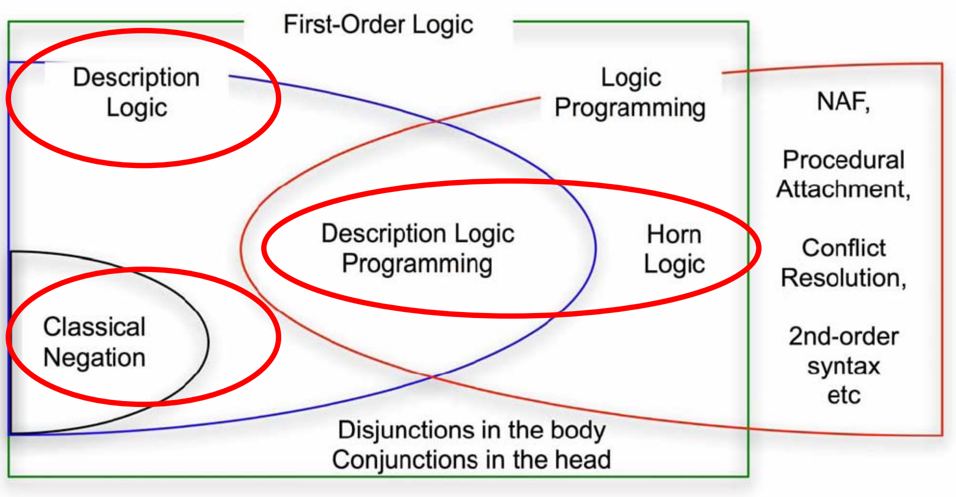
Definitions

[DL] install ≡ location ⊓ ∃ is_char_of.airpurifier
⊓ ∃ has_d_value.(inside_car ⊔ trunk)

[DL] SUV ≡ CarTypeValue
⊓ ∃ hasCarPurpose.NormalRoad
⊓ ∃ hasCarPurpose.RoughRoad
⊓ ∃ hasDisplacement.MoreThan2000LessThan3000
⊓ ∃ hasCarPurpose.Leisure.

3. Proposed Approach

Logic representation



Horn logic [HL]

a disjunction of literals with at most 1 positive.

$$(p \wedge q \wedge \ldots \wedge r \rightarrow u) \equiv (\neg p \vee \neg q \vee \ldots \vee \neg r \vee u)$$

All logical variables are universally quantified at the outer level.

An inference algorithm that derives only entailed sentences (not entailed) is called sound (not)or truth-preserving.
An inference algorithm that can derive any sentence (not all) that is entailed is complete (not).

3. Proposed Approach

Logic representation

Description logic [DL]

-. manipulate complex predicates
-. concept & Individual representation
-. taxonomy reasoning.

TBox
Man ≡ Person ⊓ ¬Woman
Father ≡ Man ⊓ ∃ hasChild.Person
…

ABox
Father(PETER)
hasChild(PETER; HARRY)

…

F. Baader, W. Nutt

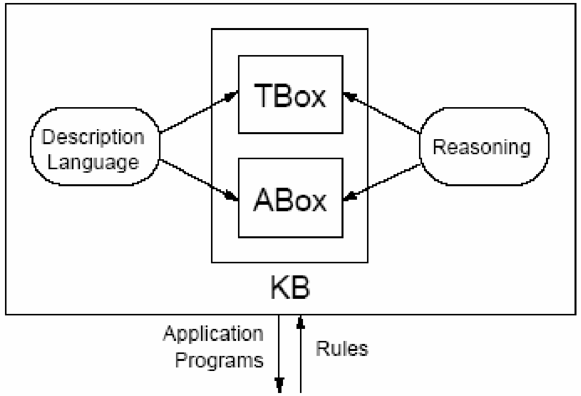


Fig. 2.1. Architecture of a knowledge representation system based on Description Logics.

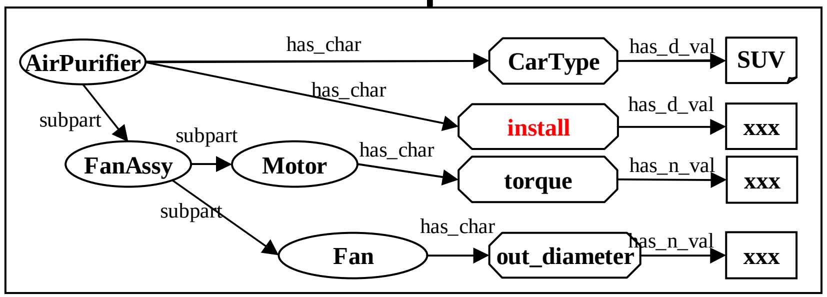
$$
\begin{aligned}
\top^{\mathcal{I}} &:= \Delta^{\mathcal{I}}, \\
\bot^{\mathcal{I}} &:= \emptyset, \\
(\neg A)^{\mathcal{I}} &:= \Delta^{\mathcal{I}} \setminus A^{\mathcal{I}}, \\
(\geq n\, r)^{\mathcal{I}} &:= \{d \mid card\{r^{\mathcal{I}}(d)\} \geq n\}, \\
(\leq n\, r)^{\mathcal{I}} &:= \{d \mid card\{r^{\mathcal{I}}(d)\} \leq n\}, \\
(a_1 \cdots a_k \downarrow b_1 \cdots b_l)^{\mathcal{I}} &:= \{d \mid (a_1 \cdots a_k)^{\mathcal{I}} \text{ and } (b_1 \cdots b_l)^{\mathcal{I}} \text{ are defined} \\
&\qquad \text{on } d \text{ and } (a_1 \cdots a_k)^{\mathcal{I}}(d) = (b_1 \cdots b_l)^{\mathcal{I}}(d)\}, \\
(C \sqcap D)^{\mathcal{I}} &:= C^{\mathcal{I}} \cap D^{\mathcal{I}}, \\
(C \sqcup D)^{\mathcal{I}} &:= C^{\mathcal{I}} \cup D^{\mathcal{I}}, \\
(\neg C)^{\mathcal{I}} &:= \Delta^{\mathcal{I}} \setminus C^{\mathcal{I}}, \\
(\forall R.C)^{\mathcal{I}} &:= \{d \mid R^{\mathcal{I}}(d) \subseteq C^{\mathcal{I}}\}, \\
(\forall a.C)^{\mathcal{I}} &:= \{d \mid a^{\mathcal{I}}(d) \subseteq C^{\mathcal{I}}\}, \\
(\exists r.C)^{\mathcal{I}} &:= \{d \mid r^{\mathcal{I}}(d) \cap C^{\mathcal{I}} \neq \emptyset\}.
\end{aligned}
$$

3. Proposed Approach

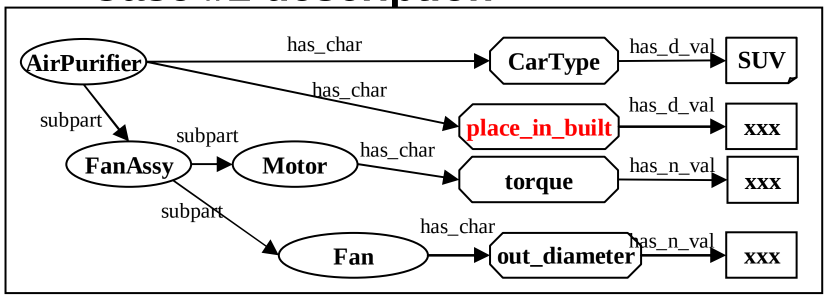
Ontology: Term’s definition

Problem Description	Case Descriptions
Ontology	Ontology

Problem description



Case #2 description



$$\text{install} \equiv \text{location} \sqcap \exists\, \text{is\_char\_of.airpurifier} \sqcap \exists\, \text{has\_d\_value.}(\text{inside\_car} \sqcup \text{trunk})$$

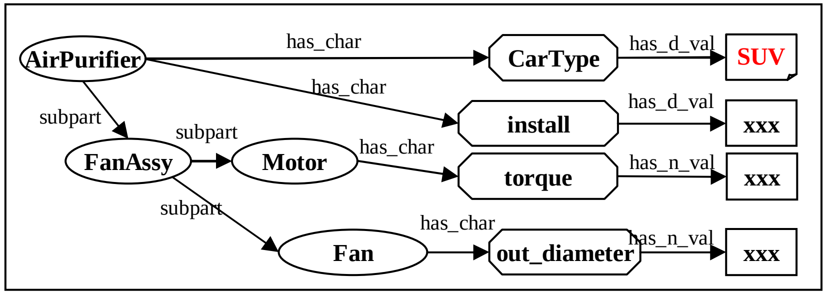
$$\text{place\_in\_built} \equiv \text{location} \sqcap \exists\, \text{is\_char\_of.airpurifier} \sqcap \exists\, \text{has\_d\_value.}(\text{inside\_car} \sqcup \text{trunk})$$

3. Proposed Approach

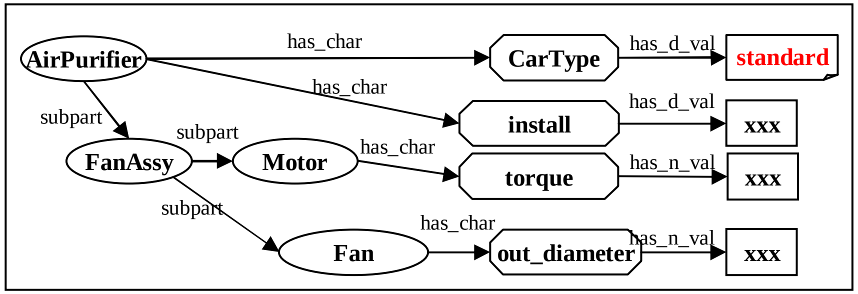
Ontology: Term's definition

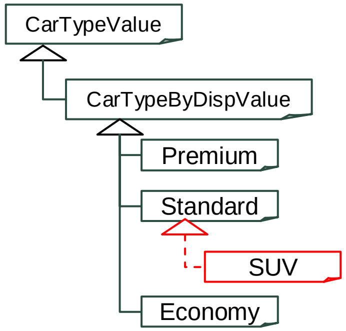
Problem Description	Case Descriptions
Ontology	Ontology

Problem description



Case #1 description





Standard $\equiv$ CarTypeByDispValue
$\wedge$ $\exists$hasDisplacement.MoreThan2000LessThan3000

SUV $\equiv$ CarTypeValue
$\sqcap$ $\exists$ hasCarPurpose.NormalRoad
$\sqcap$ $\exists$ hasCarPurpose.RoughRoad
$\sqcap$ $\exists$ hasDisplacement.MoreThan2000LessThan3000
$\sqcap$ $\exists$ hasCarPurpose.Leisure.

3. Proposed Approach

Ontology: Term's definition

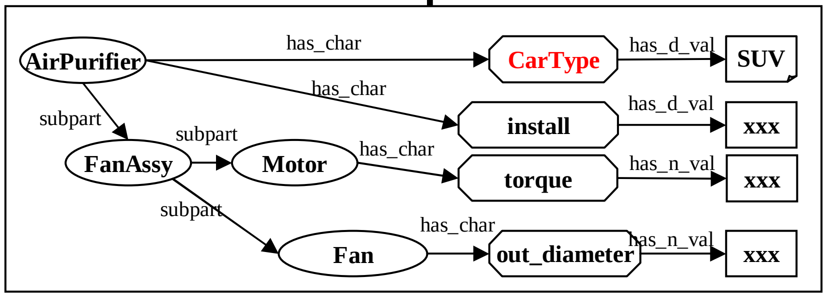
Problem Description

Ontology

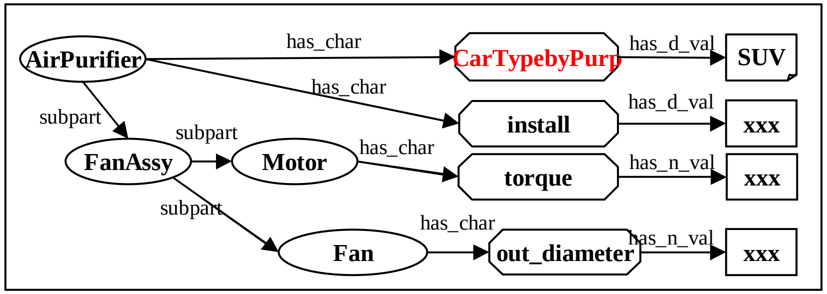
Case Descriptions

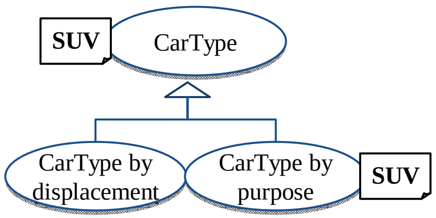
Ontology

Problem description



Case #1 description





$$\text{CarType} \equiv \text{attr\_concept} \sqcap \forall\ \text{has\_d\_value.CarTypeValues} \sqcap \exists\ \text{has\_d\_value.CarTypeValues}$$

$$\text{CarType\_by\_displacement} \equiv \text{CarType} \sqcap \forall\ \text{has\_d\_value.CarTypeByDispValues} \sqcap \exists\ \text{has\_d\_value.CarTypeByDispValues}$$

$$\text{CarType\_by\_purpose} \equiv \text{CarType} \sqcap \forall\ \text{has\_d\_value.CarTypeByPurpValues} \sqcap \exists\ \text{has\_d\_value.CarTypeByPurpValues}$$

…

3. Proposed Approach

Ontology: Term's definition

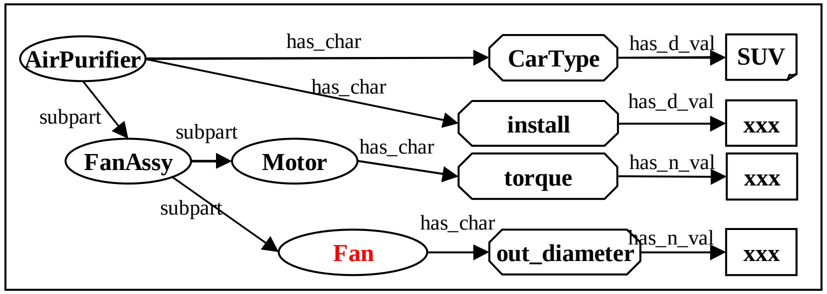
Problem Description

Ontology

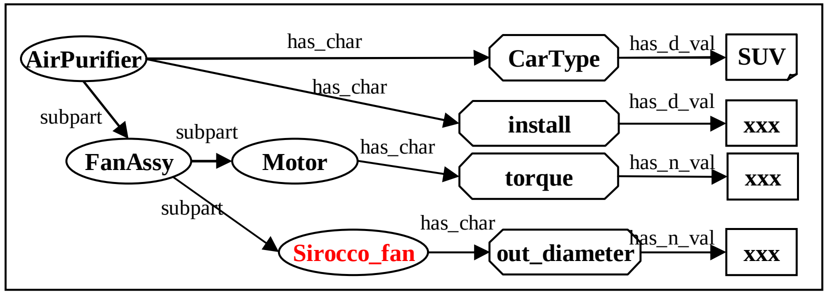
Case Descriptions

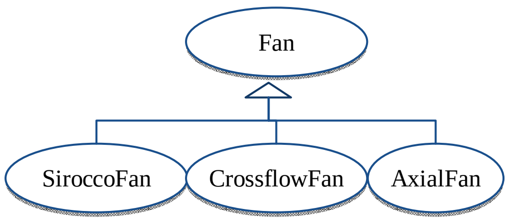
Ontology

Problem description



Case #1 description





$\text{fan} \sqsubseteq \text{part} \sqcap \exists\, \text{has\_superpart.AirPurifier}$

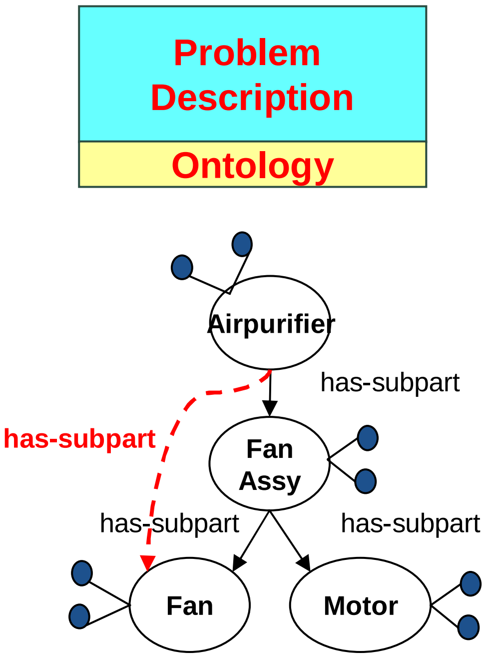
$\text{sirocco\_fan} \equiv \text{fan} \sqcap \exists\, \text{has\_req.(make\_wind}$
$\quad \sqcap \exists\, \text{has\_char.(flow\_rate} \sqcap \exists\, \text{has\_q\_val.\{fast\})}$
$\quad \sqcap \exists\, \text{has\_char.(noise} \sqcap \exists\, \text{has\_q\_val.\{mid\})).}$

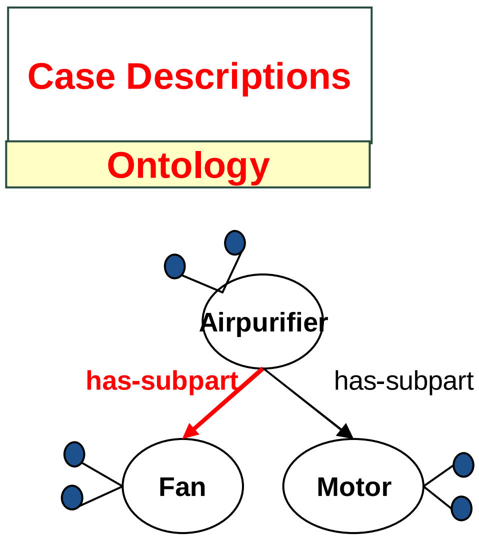
$\text{crossflow\_fan} \equiv \text{fan} \sqcap \exists\, \text{has\_req.(make\_wind}$
$\quad \sqcap \exists\, \text{has\_char.(flow\_rate} \sqcap \exists\, \text{has\_q\_val.\{normal\})}$
$\quad \sqcap \exists\, \text{has\_char.(noise} \sqcap \exists\, \text{has\_q\_val.\{low\})).}$

...

3. Proposed Approach

Ontology: Property's axiom





has-subpart$^+$: transitive property

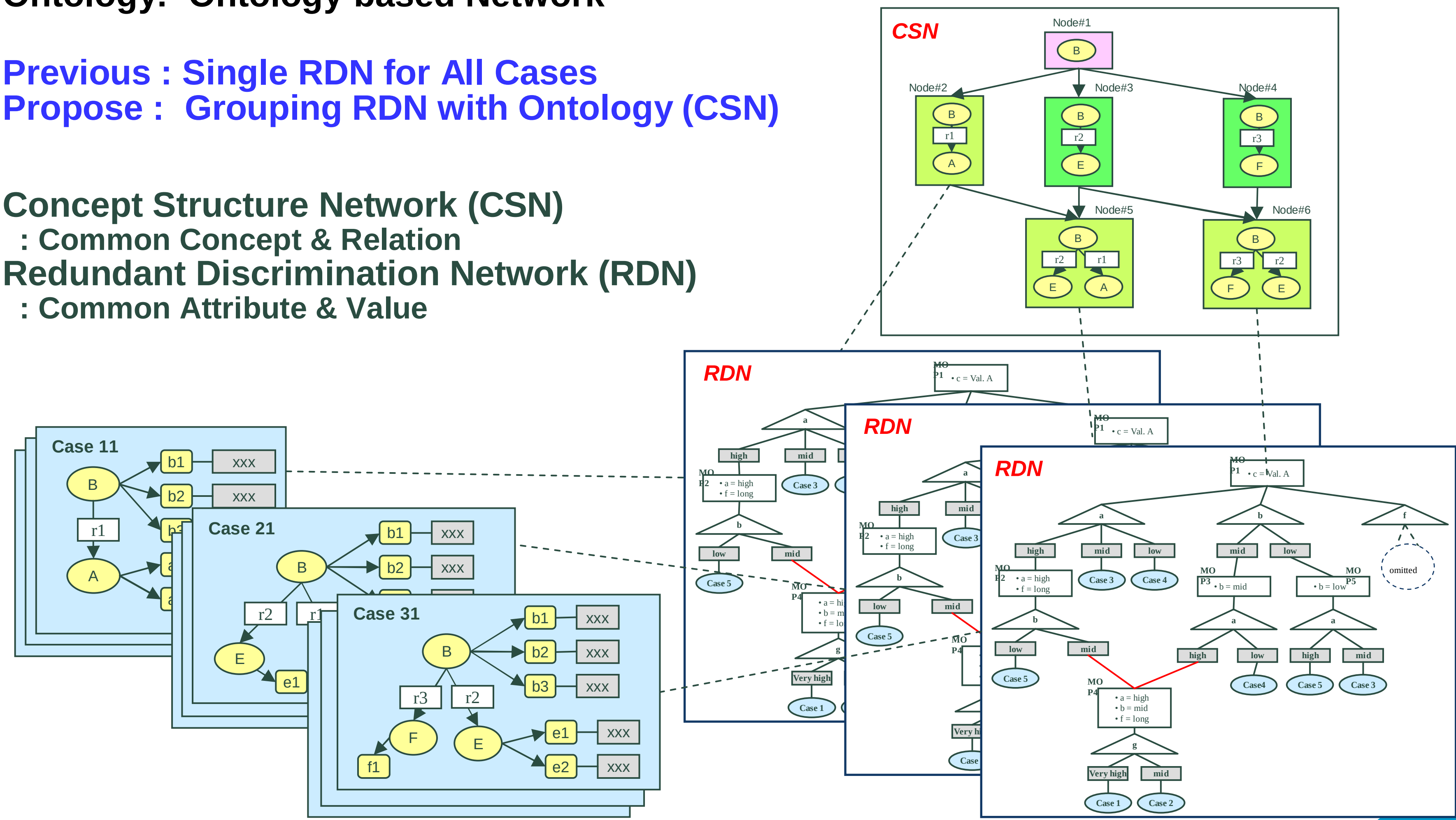
$\forall x,y,z \; has\_subpart(x, y) \land has\_subpart(y, z) \rightarrow has\_subpart(x, z)$

3. Proposed Approach

Ontology: Ontology-based Network

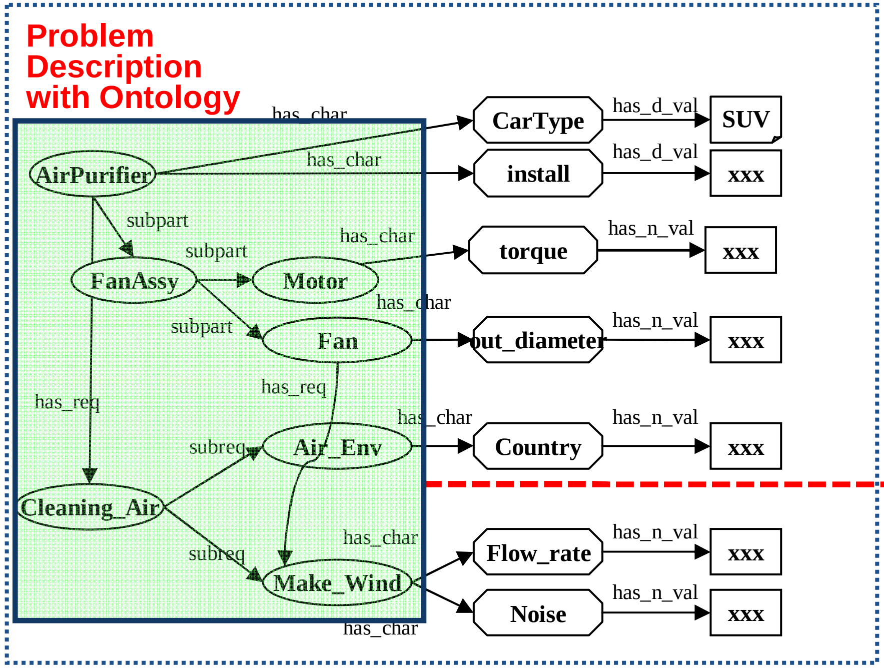
Previous : Single RDN for All Cases
Propose : Grouping RDN with Ontology (CSN)

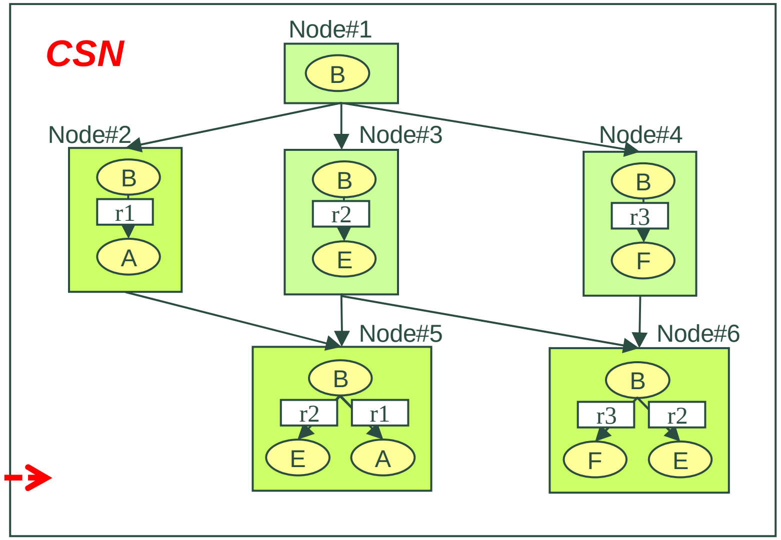
Concept Structure Network (CSN)
: Common Concept & Relation
Redundant Discrimination Network (RDN)
: Common Attribute & Value



3. Proposed Approach

Ontology: CSN Search





Ontology: Term's Definition & Axioms

Term's Definition

fan ⊑ part ⊓ ∃ has_superpart.AirPurifier

sirocco_fan ≡ fan ⊓ ∃ has_req.(make_wind

⊓ ∃ has_char.(flow_rate ⊓ ∃ has_q_val.{fast})

⊓ ∃ has_char.(noise ⊓ ∃ has_q_val.{mid})).

Axioms

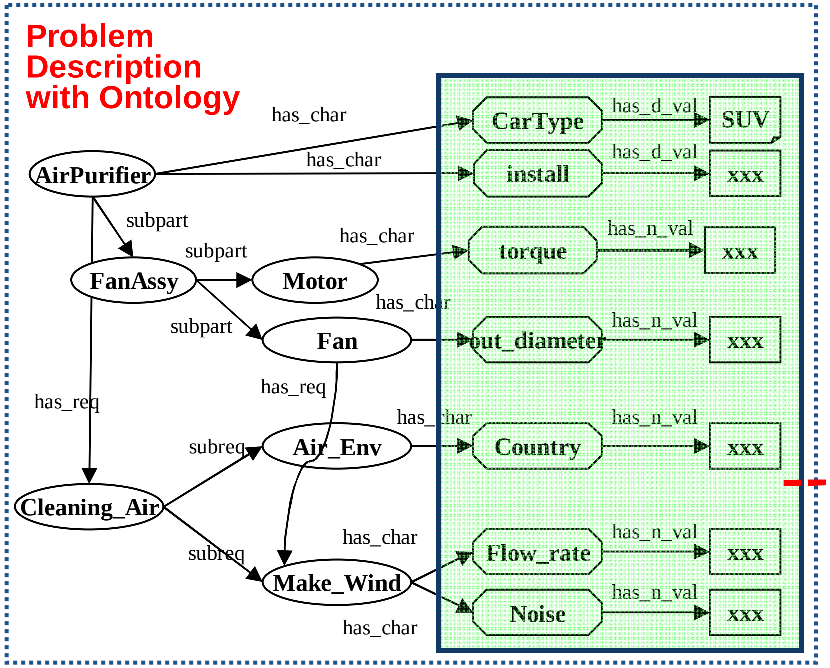
has_superpart = has_subpart $^{-}$

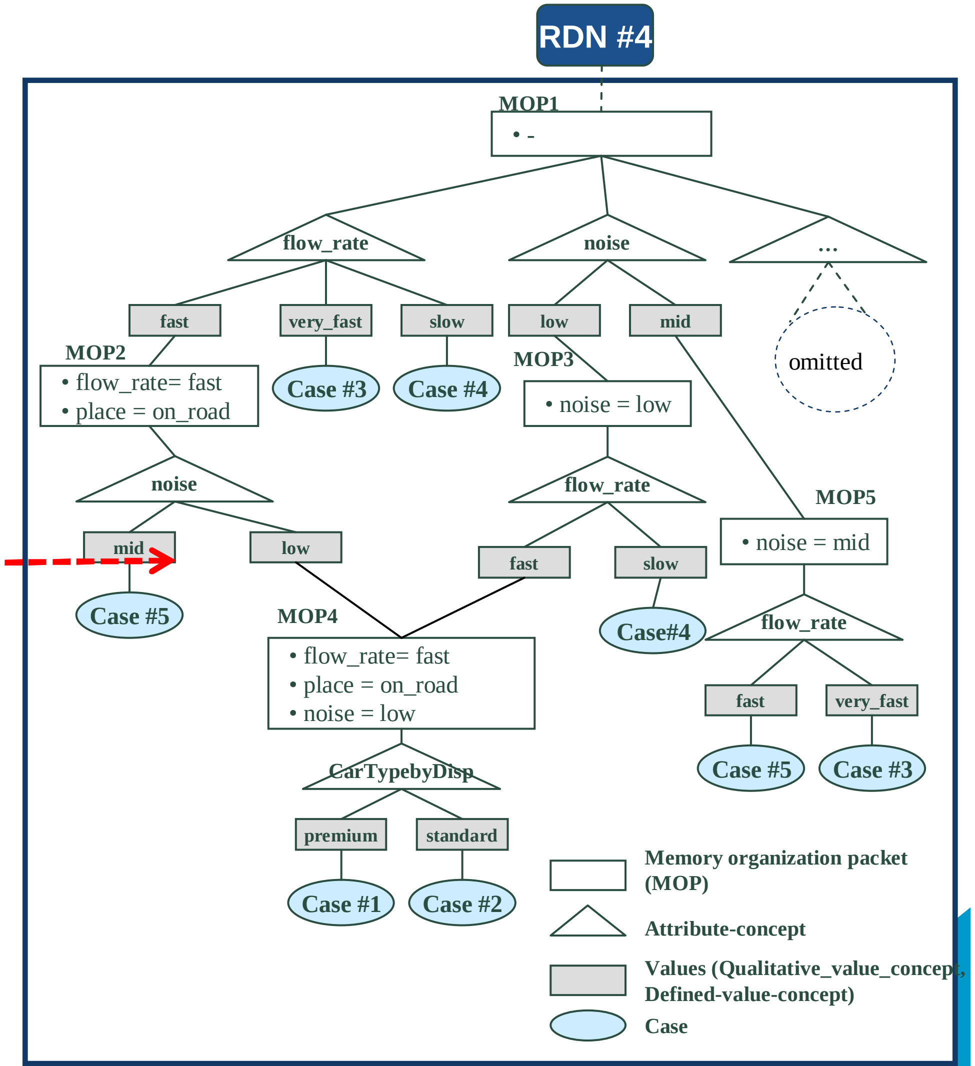
∀x,y,z has_subpart(x, y) ∧ has_subpart(y, z) → has_subpart(x, z).

…

3. Proposed Approach

Ontology: RDN Search





Ontology: Term's Definition & Axioms

Term's Definition

CarType ≡ attr_concept ⊓ ∀ has_d_value.CarTypeValues
⊓ ∃ has_d_value.CarTypeValues

CarType_by_displacement ≡ CarType
⊓ ∀ has_d_value.CarTypeByDispValues
⊓ ∃ has_d_value.CarTypeByDispValues

…

Axioms

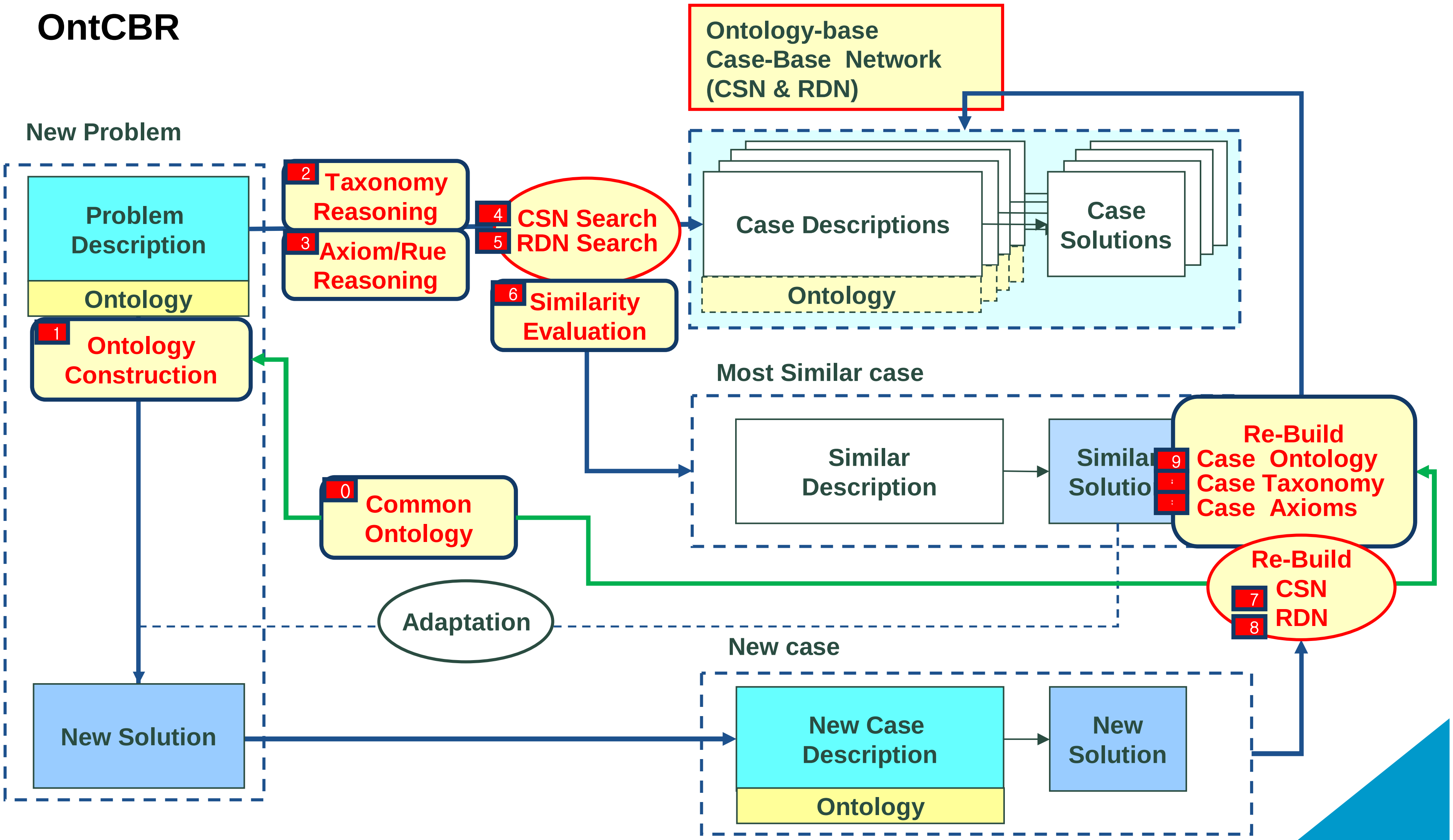
…

3. Proposed Approach

- Ontology Construction
  - Term's definitions
  - Property axioms
- Reasoning
  - Taxonomy Reasoning
  - Axiom/Rule Reasoning
- Network Construction
  - CSN
  - RDN
- Similarity Evaluation
  - Term's similarity
  - Graph structure similarity
- Case Search with Networks
  - Search Method (CSN & RDN)
- Update Ontology and Case-base

3. Proposed Approach

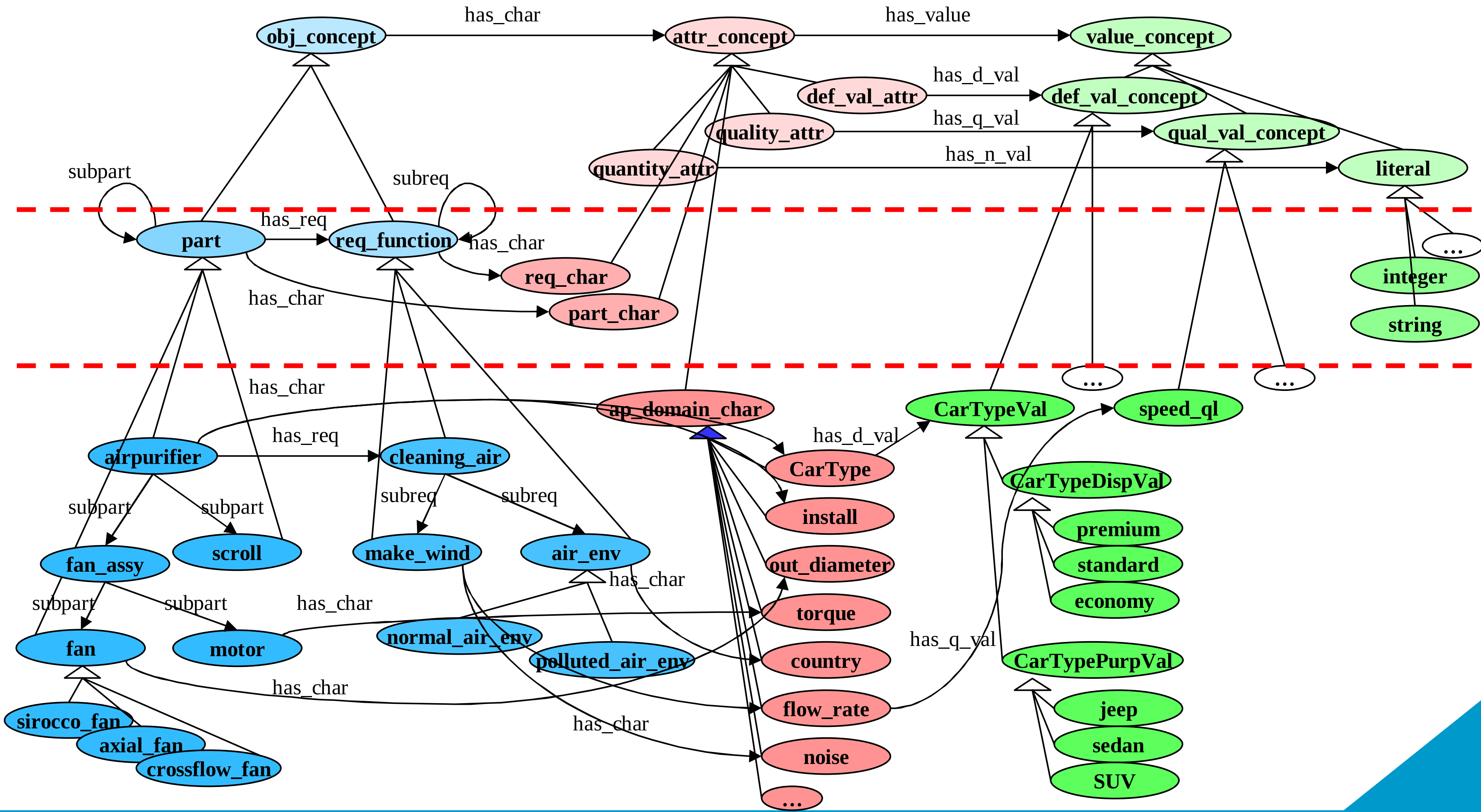
OntCBR



4. Ontology Construction

0 Common Ontology

4.1. Common Case Ontology (Multi-Layered Ontology [Lee & Suh 2008])

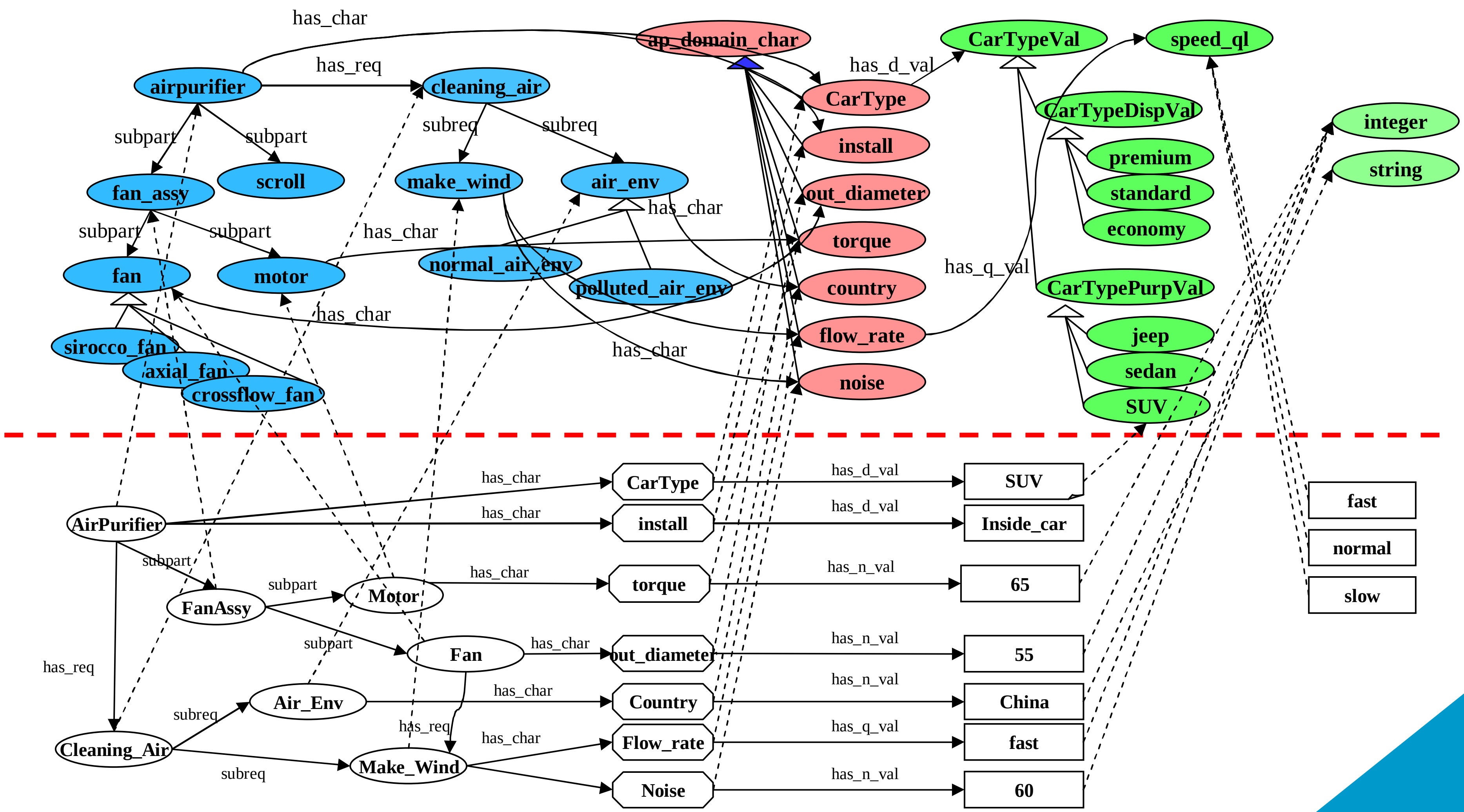


Jae-Hyun Lee and Hyo-Won Suh, 2006 (accepted),
"Ontology-based Multi-layered Knowledge Framework for Product Lifecycle Management", Concurrent Engineering-Research and Application.

4. Ontology Construction

Common Ontology

4.1. Common Case Ontology



Jae-Hyun Lee and Hyo-Won Suh, 2006 (accepted),
"Ontology-based Multi-layered Knowledge Framework for Product Lifecycle Management", Concurrent Engineering-Research and Application.

4. Ontology Construction

Common Ontology

4.1. Common Case Ontology

Term's Definitions:

-. attr_concept $\equiv$ concept $\sqcap \exists$ has_char^{-}.object_concept.

-. def_val_attr $\equiv$ attr_concept $\sqcap \exists$ has_d_val.def_val_concept
$\sqcap \forall$ has_d_val.def_val_concept

-. sirocco_fan $\equiv$ fan $\sqcap \exists$ has_req.(make_wind
$\sqcap \exists$ has_char.(flow_rate $\sqcap \exists$ has_q_val.{fast})
$\sqcap \exists$ has_char.(noise $\sqcap \exists$ has_q_val.{mid})).

-. crossflow_fan $\equiv$ fan $\sqcap \exists$ has_req.(make_wind
$\sqcap \exists$ has_char.(flow_rate $\sqcap \exists$ has_q_val.{normal})
$\sqcap \exists$ has_char.(noise $\sqcap \exists$ has_q_val.{low})).z

Property Axioms

subpart + => 'subpart' relation is transitive.

subreq + => 'subreq' relation is transitive

$\forall$x,y,z req_function(x) $\wedge$ req_function(y) $\wedge$ subreq(x,y)
$\wedge$ part(z) $\wedge$ has_req(z, x) $\rightarrow$ has_req(z,y).

4. Ontology Construction

Common Ontology

4.1. Common Case Ontology

Rules:

Rule#01

$\forall$x,y,v1,v2 Fan(x) $\wedge$ flow_rate(y) $\wedge$ has_char(x,y)
$\wedge$ has_n_value(y, v1) $\wedge$ (120 < v1) $\wedge$ has_q_value(y, v2)
$\rightarrow$ (v2 = fast)

(if quantity of flow_rate is more than 120, its qualitative value is very_fast)

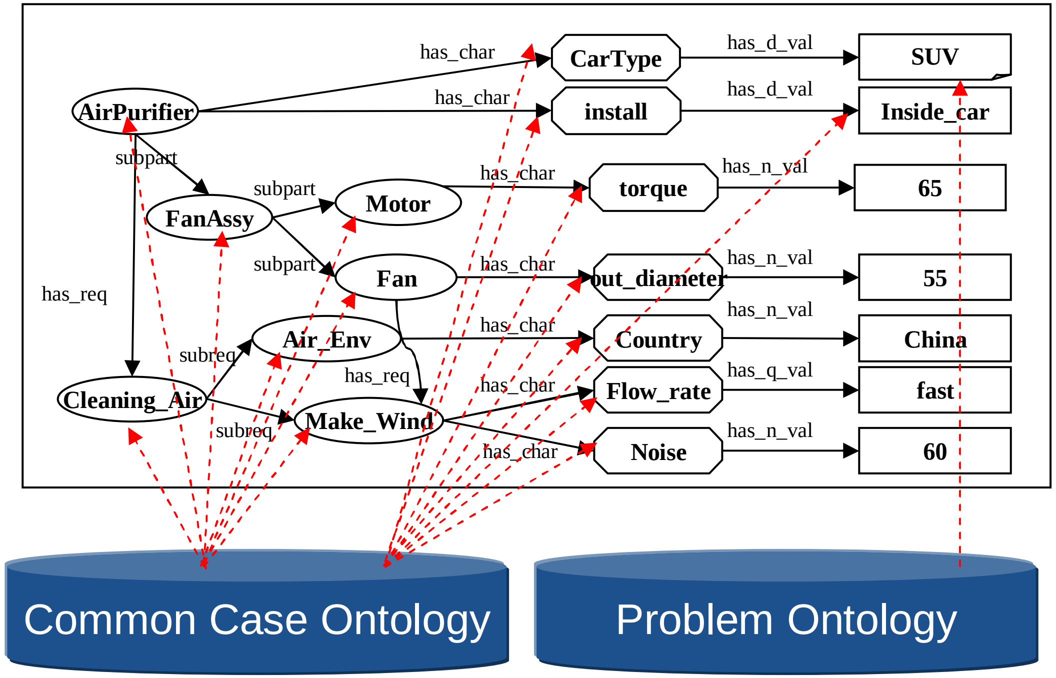
Rule#02

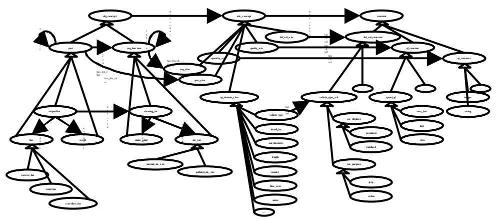
$\forall$x,y,v1,v2 Fan(x) $\wedge$ noise(z) $\wedge$ has_char(x,z)
$\wedge$ has_n_value(y, v1) $\wedge$ (83 < v1) $\wedge$ has_q_value(y, v2)
$\rightarrow$ (v2 = mid)

(if quantity of noise is more than 83, its qualitative value is mid)

4. Ontology Construction

4.2. Problem ontology construction





Def#03

SUV ≡ CarTypeValue

⊓ ∃ hasCarPurpose.NormalRoad

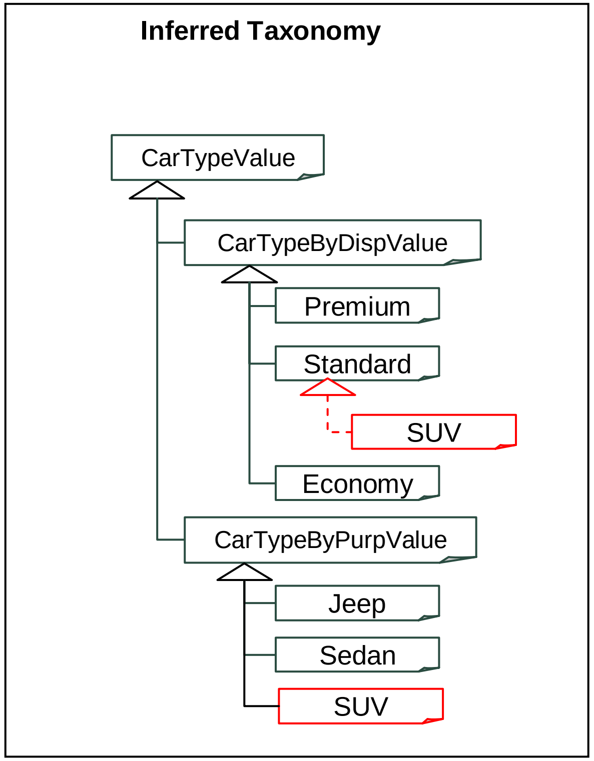
⊓ ∃ hasCarPurpose.RoughRoad

⊓ ∃ hasDisplacement.MoreThan2000LessThan3000

⊓ ∃ hasCarPurpose.Leisure.

5. Ontology Reasoning

5.1 Taxonomy Reasoning



Definitions in Common Case Ontology

CarTypeByDispValue ≡ CarTypeValue
⊓ ∃hasDisplacement.DisplacementRange
Premium ≡ CarTypeByDispValue
∧ ∃hasDisplacement.MoreThan3000LessThan4000
Standard ≡ CarTypeByDispValue
∧ ∃hasDisplacement.MoreThan2000LessThan3000
Economy ≡ CarTypeByDispValue
∧ ∃hasDisplacement.LessThan1500
Jeep≡ CarTypeByPurpValue
∧ ∃ hasCarPurpose.DrivingRoughRoad.
…

Newly defined value's definition

Def#03
SUV ≡ CarTypeValue
∧ ∃ hasCarPurpose.NormalRoad
∧ ∃ hasCarPurpose.RoughRoad
∧ ∃ hasDisplacement.MoreThan2000LessThan3000
∧ ∀hasCarPurpose.Leisure.

Taxonomy Inference Results

(by Def#03)
KB_for_new_prob |= SUV ⊑ Standard.

5.1 Taxonomy Reasoning - Tableau Algorithm

The $\rightarrow_{\sqcap}$-rule
Condition: $\mathcal{A}$ contains $(C_1 \sqcap C_2)(x)$, but it does not contain both $C_1(x)$ and $C_2(x)$.
Action: $\mathcal{A}' = \mathcal{A} \cup \{C_1(x), C_2(x)\}$.

The $\rightarrow_{\sqcup}$-rule
Condition: $\mathcal{A}$ contains $(C_1 \sqcup C_2)(x)$, but neither $C_1(x)$ nor $C_2(x)$.
Action: $\mathcal{A}' = \mathcal{A} \cup \{C_1(x)\}$, $\mathcal{A}'' = \mathcal{A} \cup \{C_2(x)\}$.

The $\rightarrow_{\exists}$-rule
Condition: $\mathcal{A}$ contains $(\exists R.C)(x)$, but there is no individual name z such that $C(z)$ and $R(x, z)$ are in $\mathcal{A}$.
Action: $\mathcal{A}' = \mathcal{A} \cup \{C(y), R(x, y)\}$ where y is an individual name not occurring in $\mathcal{A}$.

The $\rightarrow_{\forall}$-rule
Condition: $\mathcal{A}$ contains $(\forall R.C)(x)$ and $R(x, y)$, but it does not contain $C(y)$.
Action: $\mathcal{A}' = \mathcal{A} \cup \{C(y)\}$.

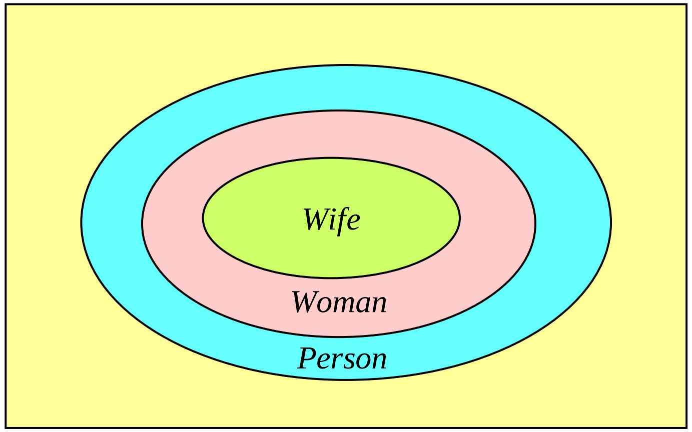
The $\rightarrow_{\geq}$-rule
Condition: $\mathcal{A}$ contains $(\geqslant n\, R)(x)$, and there are no individual names $z_1, \ldots, z_n$ such that $R(x, z_i)$ $(1 \leq i \leq n)$ and $z_i \not\doteq z_j$ $(1 \leq i < j \leq n)$ are contained in $\mathcal{A}$.
Action: $\mathcal{A}' = \mathcal{A} \cup \{R(x, y_i) \mid 1 \leq i \leq n\} \cup \{y_i \not\doteq y_j \mid 1 \leq i < j \leq n\}$, where $y_1, \ldots, y_n$ are distinct individual names not occurring in $\mathcal{A}$.

The $\rightarrow_{\leq}$-rule
Condition: $\mathcal{A}$ contains distinct individual names $y_1, \ldots, y_{n+1}$ such that $(\leqslant n\, R)(x)$ and $R(x, y_1), \ldots, R(x, y_{n+1})$ are in $\mathcal{A}$, and $y_i \not\doteq y_j$ is not in $\mathcal{A}$ for some $i \neq j$.
Action: For each pair y_i, y_j such that $i > j$ and $y_i \not\doteq y_j$ is not in $\mathcal{A}$, the ABox $\mathcal{A}_{i,j} = [y_i/y_j]\mathcal{A}$ is obtained from $\mathcal{A}$ by replacing each occurrence of y_i by y_j.

Tableau Algorithm – Preliminary

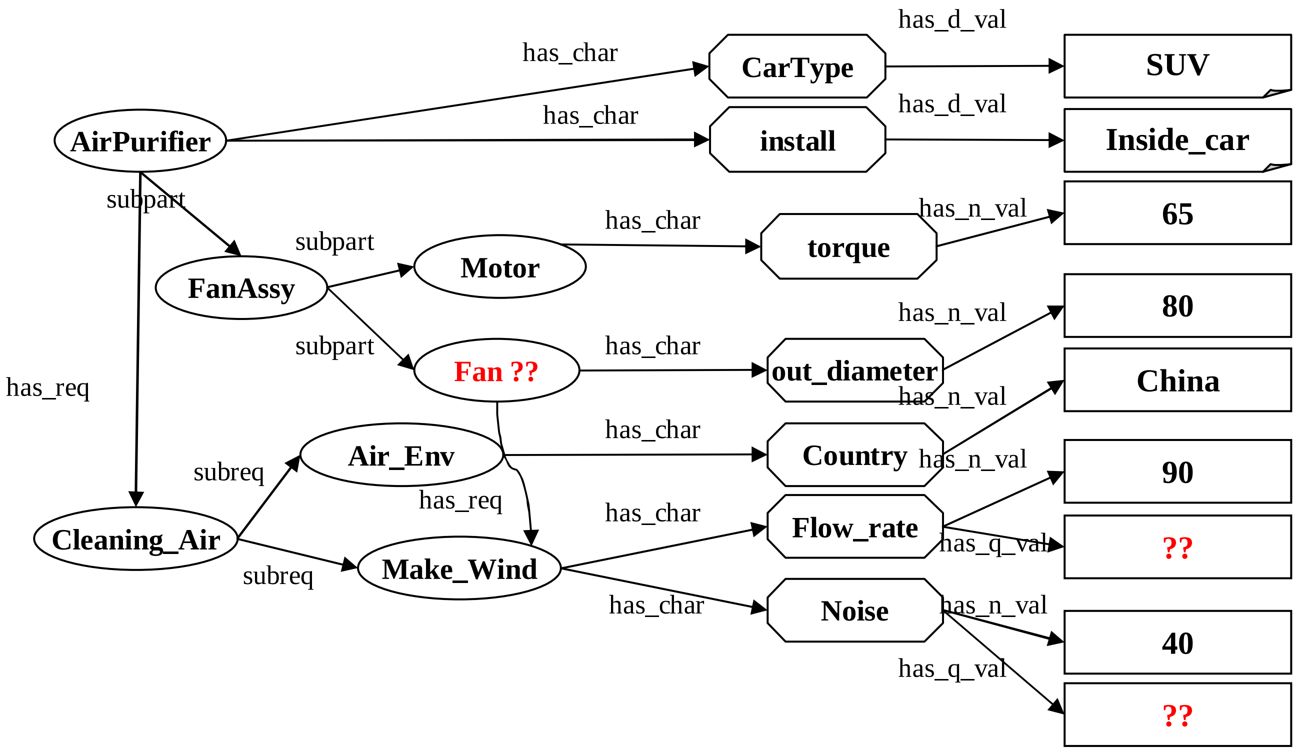
- **Given : Wife ⊑ Woman, Woman ⊑ Person**
  Query : Wife ⊑ Person

- **Reasoning process**
  - Test if there is an individual that is a Woman but not a Person, i.e. test the satisfiability of concept
    C_0= (Wife ⊓ ¬ Person)
  - C_0 (x) -> Wife(x), (¬ Person) (x)
  - Wife(x) -> Woman(x)
  - Woman(x) -> Person(x)
  - Conflict!
  - C_0 is unsatisfiable, therefore Wife ⊑ Person is true with the given ontology.

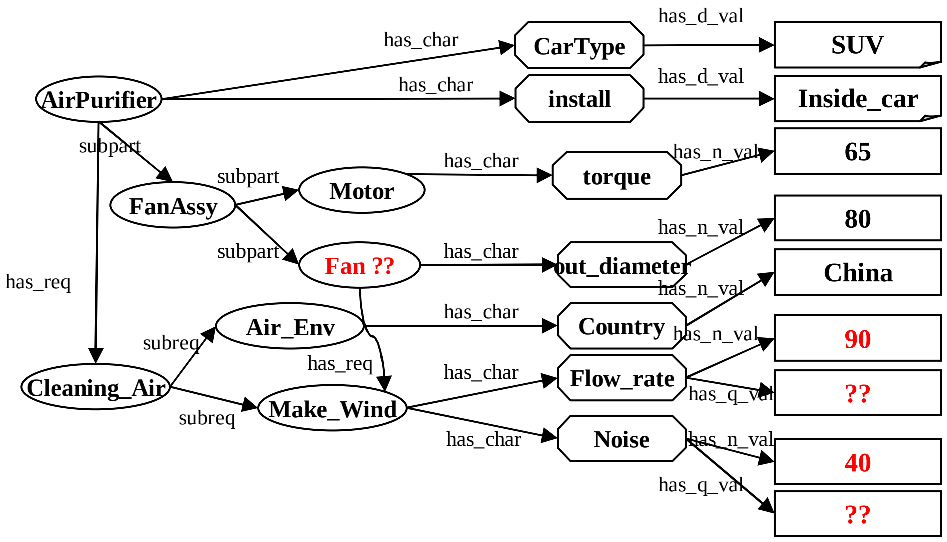


5. Ontology Reasoning

5.2 Axiom Reasoning - concept



5.2 Axiom Reasoning - concept



KB_for_new_prob
fan(fan_new_prob).
flow_rate(flow_rate_new_prob)
noise(noise_new_prob).
make_wind(make_wind_new_prob)
SUV(suv_1).
has_req(fan_new_prob, make_wind_new_prob).
has_char(fan_new_prob, flow_rate_new_prob).
has_char(fan_new_prob, noise_new_prob).
has_char(fan_new_prob, veh_type_new_prob)
has_n_value(flow_rate_new_prob, 90).
has_n_value(noise_new_prob, 40).
has_d_value(veh_type_new_prob, suv_1)
…

Concept Definition:
Def#02
crossflow_fan ≡ fan ⊓ has_req.(make_wind ⊓ has_char.(flow_rate ⊓ has_q_val.{normal})
⊓ has_char.(noise ⊓ has_q_val.{low})).

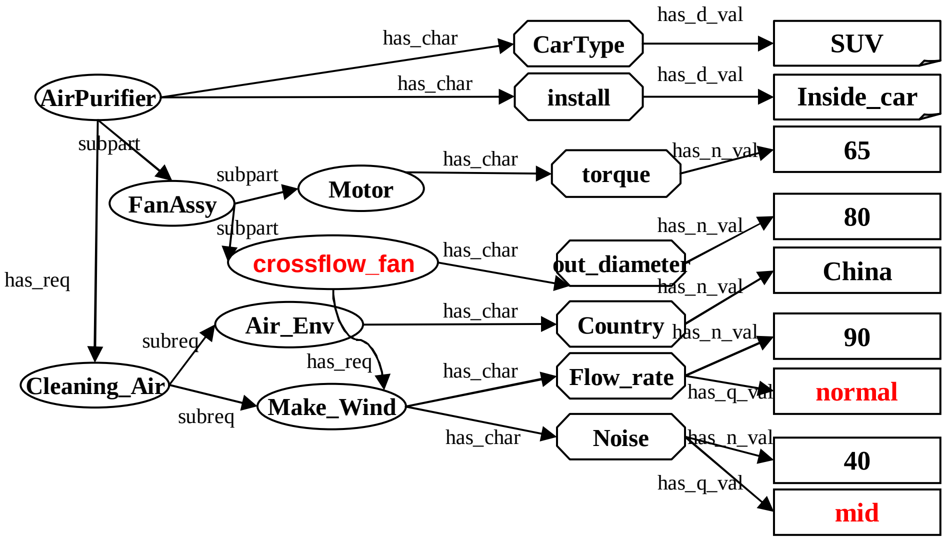
Rule#03 ∀x,y,v1,v2 Fan(x) ∧ flow_rate(y) ∧ has_char(x,y) ∧ has_n_value(y, v1) ∧ (80 < v1 < 120)
→ has_q_value(y, 'normal')

Rule#04 ∀x,y,v1,v2 Fan(x) ∧ noise(z) ∧ has_char(x,z) ∧ has_n_value(y, v1) ∧ (30 < v1 < 80) →
has_q_value(y, 'low')

5. Ontology Reasoning

3 **Axiom Reasoning**

5.2 Axiom Reasoning - concept



KB_for_new_prob
fan(fan_new_prob).
flow_rate(flow_rate_new_prob)
noise(noise_new_prob).
make_wind(make_wind_new_prob)
SUV(suv_1).
has_req(fan_new_prob, make_wind_new_prob).
has_char(fan_new_prob, flow_rate_new_prob).
has_char(fan_new_prob, noise_new_prob).
has_char(fan_new_prob, veh_type_new_prob)
has_n_value(flow_rate_new_prob, 90).
has_n_value(noise_new_prob, 40).
has_d_value(veh_type_new_prob, suv_1)
…

Inference Results

(*by Rule#03*) KB_for_new_prob |= has_q_value(flow_rate_new_prob, 'normal').
(*by Rule#04*) KB_for_new_prob |= has_q_value(noise_new_prob, 'low').
(*by Def#02*) KB_for_fan_v1 |= crossflow_fan(fan_new_prob).

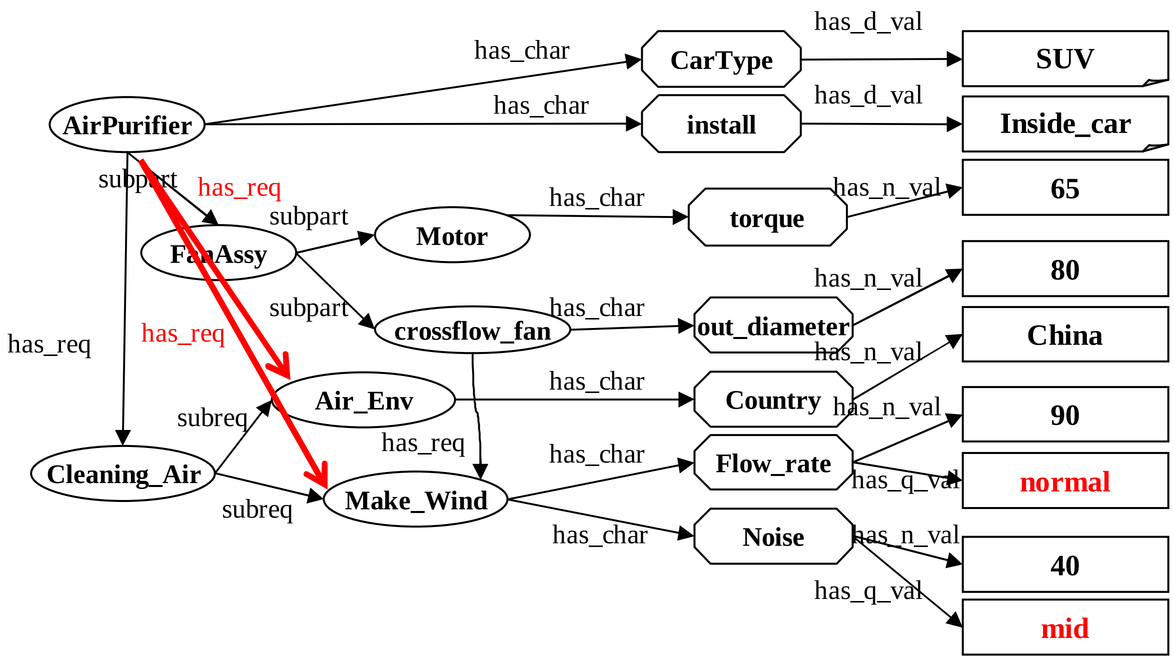
Def#02

crossflow_fan ≡ fan ⊓ has_req.(make_wind ⊓ has_char.(flow_rate ⊓ has_q_val.{normal})
⊓ has_char.(noise ⊓ has_q_val.{low})).

5. Ontology Reasoning

3 Axiom Reasoning

3) Axiom Reasoning - relation



KB_for_new_prob
airpurifier(airpurifier_np).
cleaning_air(cleaning_air_np).
air_env(air_env_np).
make_wind(make_wind_np).
has_req(airpurifier_np, cleaning_air_np).
subreq(cleaning_air_np, air_env_np).
subreq(cleaning_air_np, make_wind_np).
…

Property Axioms

∀x,y,z req_function(x) ∧ req_function(y) ∧ subreq(x,y) ∧ part(z) ∧ has_req(z, x) → has_req(z,y).

Inference Results
(by Rule#05)
KB_for_new_prob |=
has_req(airpurifier_np, air_env_np) ∧
has_req(airpurifier_np, make_wind_np)

6. Ontology-based Case-Base Network

Re-Build
7 CSN
8 RDN

CSN and RDN

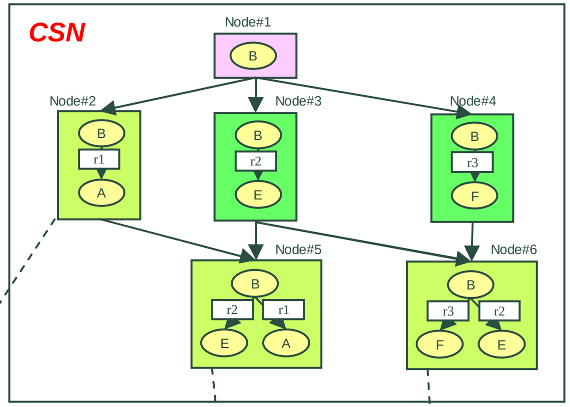
1) Case Structure Network (CSN)

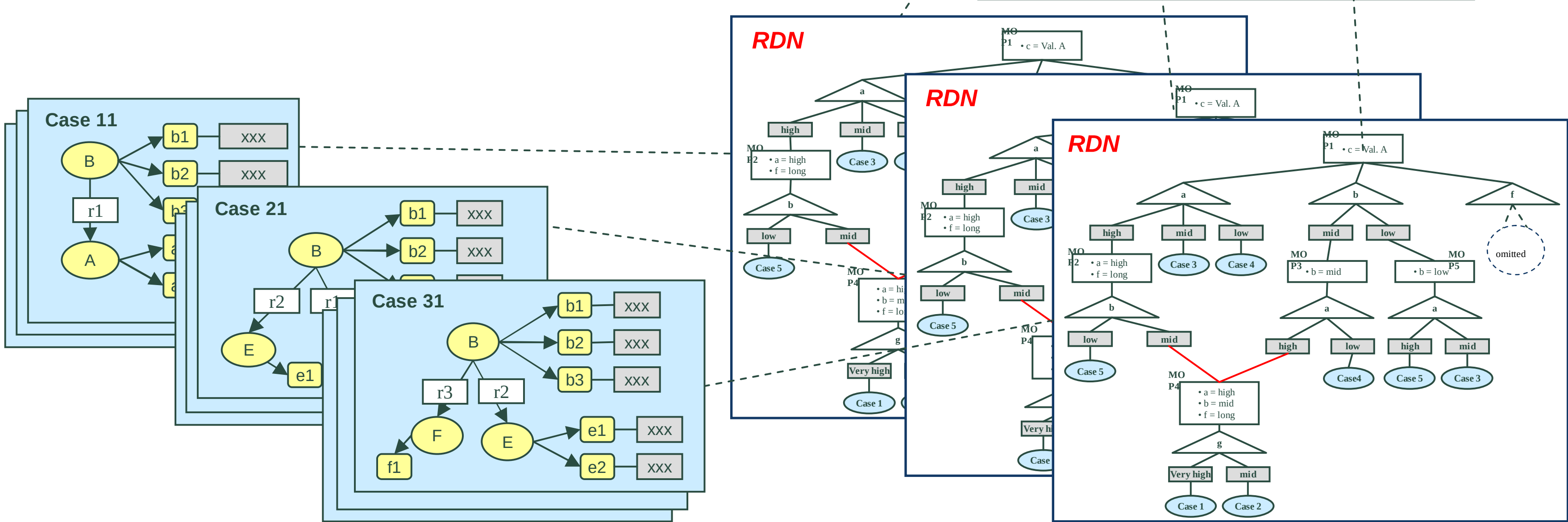
: to cluster cases according to their similar concept structure

it is built by cases' concept structures and their sub-structure

2) Redundant Discrimination Network (RDN)

: to discriminate cases according to their attribute-values





6. Ontology-based Case-Base Network

Re-Build
7 CSN
8 RDN

6.1 CSN Construction

CSN := (Node, Direct arc)

Node := Structure of Object-concepts

Node types := top node, case structure node, case sub-structure node

- Top node := key-object-concept only.
- Case structure node := Structure of object-concepts of some cases
- Case sub-structure node := Sub-structure of some case structure nodes

Direct arc:= direct_link(*Node_A*, *Node_B*), inferred_link(*Node_A*, *Node_B*)

Node_A of ‘direct_link’ has sub_graph of *Node_B*

Node_A of ‘inferred_link’ is sub-graph of *Node_B* and includes inferred structure

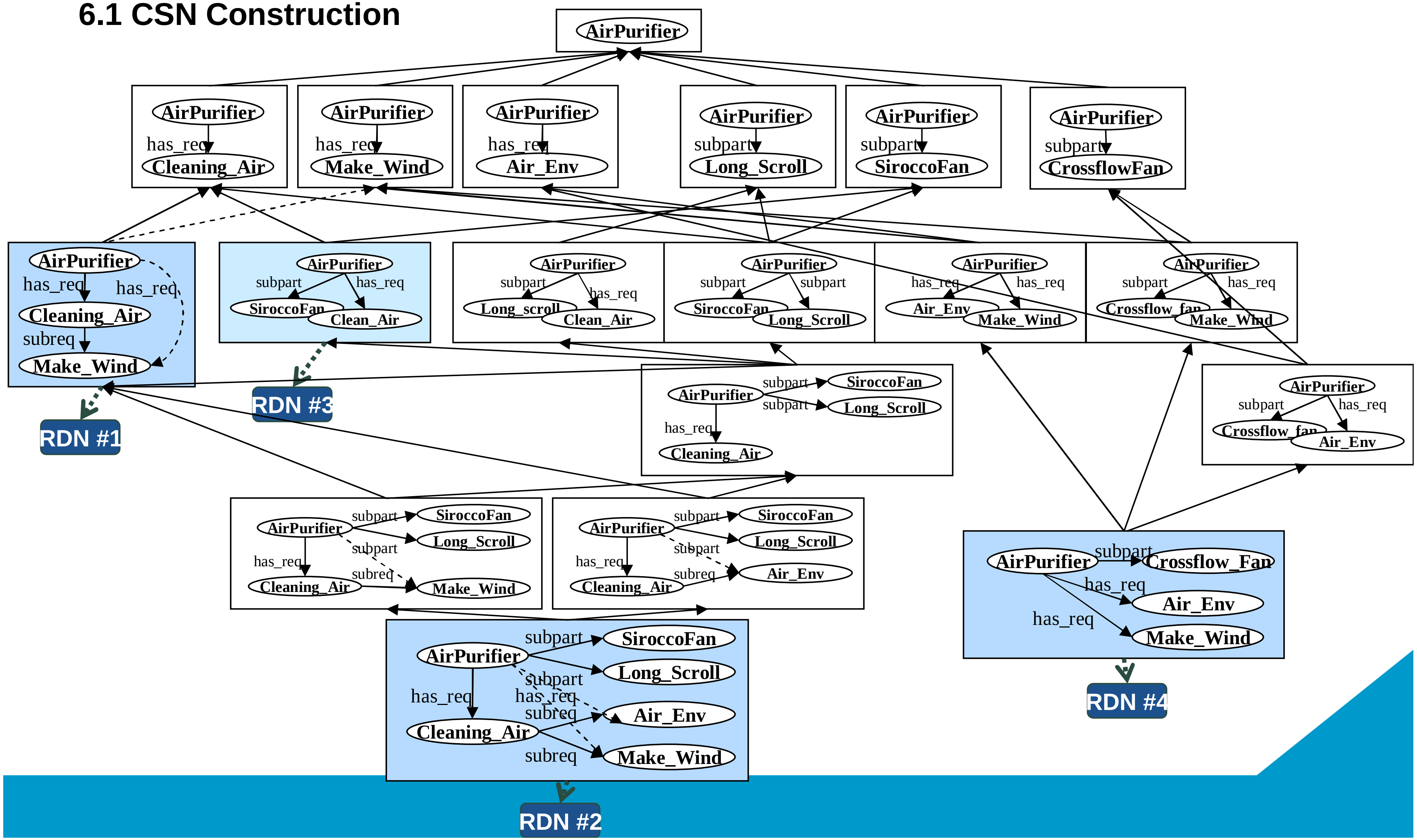
*. Case structure node has reference address of RDN which classifies identical structure cases.

*. Case sub-structure node is necessary for problem structure and partial match.

6. Ontology-based Case-Base Network

Re-Build
7 CSN
8 RDN

6.1 CSN Construction



6. Ontology-based Case-Base Network

Re-Build
7 CSN
8 RDN

6.2 RDN (Redundant Discrimination Network) Construction

RDN := (Node, Direct arc)

Node := (MOP, Attribute-Concept, Value, Case)

- MOP (Memory Organization Packet) Node :=
  Nodes that has information of common attributes and values of sub-cases
- Attribute-Concept Node:= Attribute-concepts that cases have.
- Value Node := Value_concepts that cases have.
- Case Node := each case.

Direct arc := direct_link(Node_A, Node_B)

direct_link(MOP node, Attribute-Concept nodes)

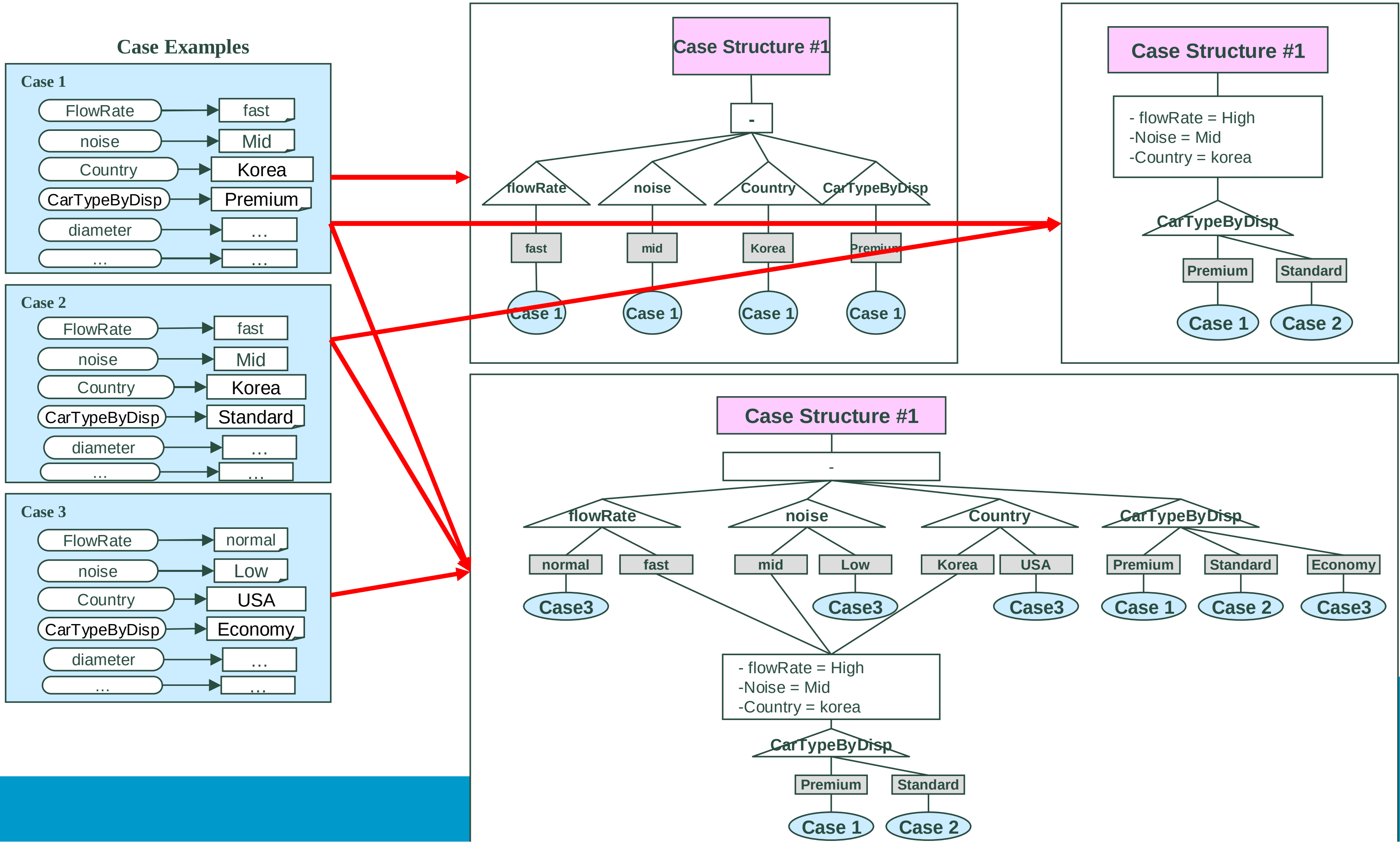
direct_link(Attribute-Concept node, Value nodes)

direct_link(Value node, [MOP node | Case node])

6. Ontology-based Case-Base Network

Re-Build
7 CSN
8 RDN

6.2 RDN (Redundant Discrimination Network) Construction

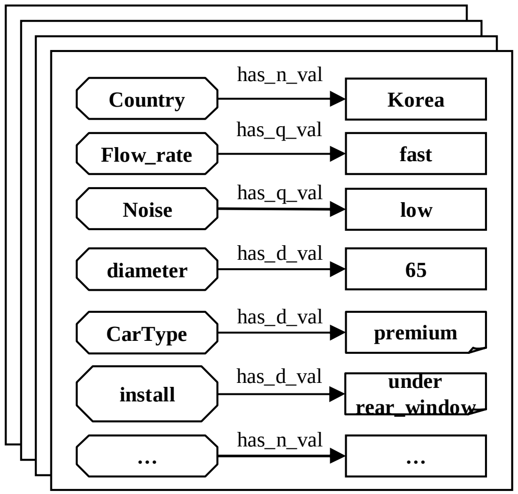


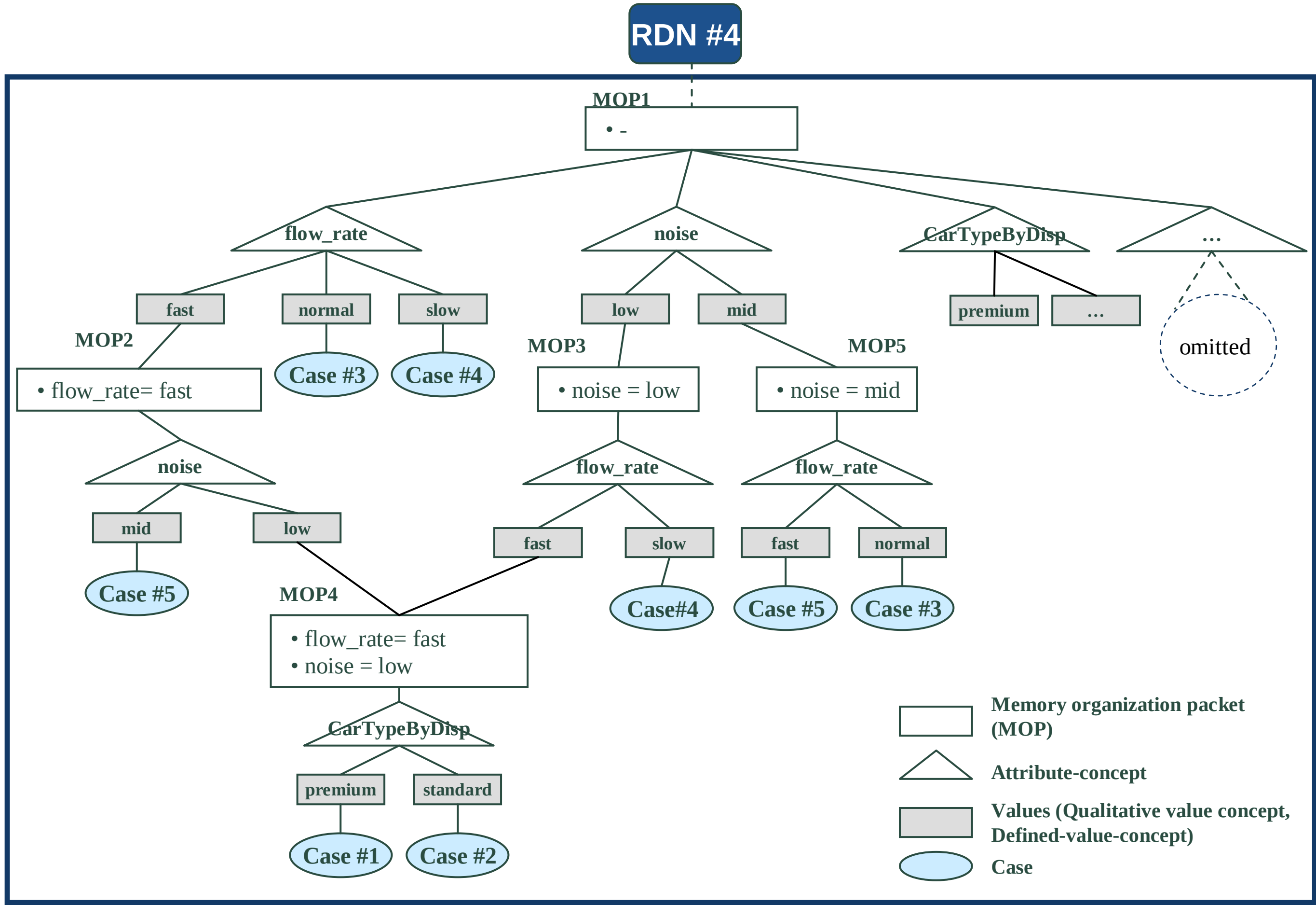
6. Ontology-based Case-Base Network

Re-Build
7 CSN
8 RDN

6.2 RDN (Redundant Discrimination Network) Construction

Cases in RDN#4





7. Similarity Evaluation

Character similarity

Definition similarity

Taxonomy similarity

Dependency similarity

M

M. Jung, H.W. Suh, "Ontology Mapping-based Search with Multi-dimensional Similarity and Bayesian Network', 7th IJCC Japan-Korea CAD/CAM Workshop RCAST, Tokyo, Japan, July 26-27, 2007 Japan .

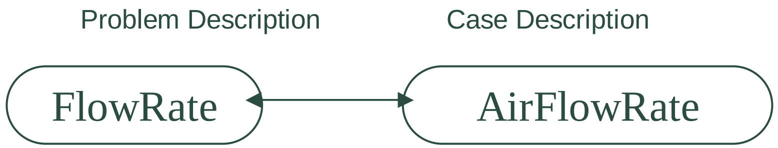
P. Mitra, N. F. Noy and A. R. Jaiswal., "OMEN: A Probabilistic Ontology Mapping Tool," Workshop on Meaning coordination and negotiation at the Third International Conference on the Semantic Web (ISWC-2004), Hisroshima, Japan, 2004.

7. Similarity Evaluation

7.1 Term’s Similarity Calculation

1) Character Similarity by *Jaro* metrics

$$Jaro(s,t) = \frac{1}{3} \cdot \left(\frac{|s'|}{|s|} + \frac{|t'|}{|t|} + \frac{|s'| - T_{s',t'}}{2|s'|} \right)$$



JaroSim(FlowRate, AirFlowRate) =
1/3 * (8/8 + 8/11 + (8 – 0)/(2*8)) **= 0.74**

<Jaro Metric>

s =WILLLAIM |s| = 8
t = WILLIAM |t| = 7
s' = WILLAM |s'| = 6
t' = WILLIM |t'| = 6
Transposition 수 = 1

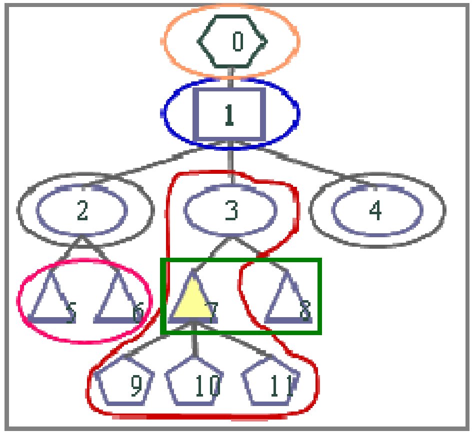
i	s' vs t'
1	W
2	I
3	L
4	L
5	A≠L
6	M

Jaro(s,t) = (1/3)*(6/8 + 6/7 + (6-1)/2*6)
≒ 0.84

7. Similarity Evaluation

7.1 Term’s Similarity Calculation

2) Taxonomy similarity by positions in the taxonomy



Direct hierarchy relation: 0.8

Sibling relation: 0.7

Grand parent relation: 0.5

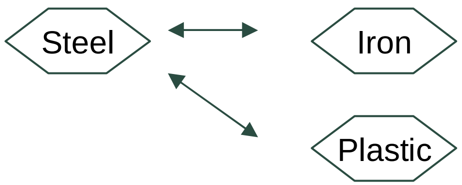
Uncle relation: 0.3

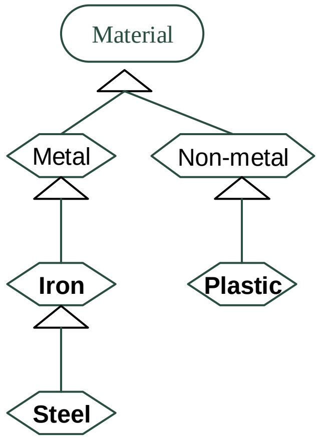
Ancestor: 0.2

Other: 0

Problem Description

Case Description





TaxSim(Steel, Iron) **= 0.8**

TaxSim(Steel, Plastic) **= 0.2**

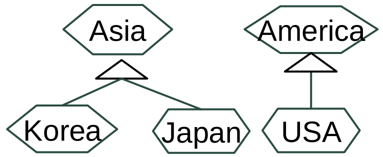
7. Similarity Evaluation

7.1 Term's Similarity Calculation

3) Definition Similarity by similarity of terms in the definitions

$$DefSim(c_1, c_2) = \frac{N_R}{N_{c_1} + N_{c_2}} + \sigma \times \frac{\tilde{N}_R}{N_{c_1} + N_{c_2}}, \qquad \sigma < 1$$

Problem Description: **Geo_location** ⟷ Case Description: **Country**



Geom_location $\equiv$ *geographical_place*
$\wedge \exists$ *in_which_used.AirPurifier*
$\wedge \forall$ *hasDvalue.(Asia* $\vee$ Europe $\vee$ *America)*

Country $\equiv$ *PoliticalUnit*
$\wedge \forall$ *hasDvalue.(Korea* $\vee$ *China* $\vee$ *USA)*

Total concept = Nc1 + Nc2

Similar (same) concept with same (similar) Role = Nr

Similar (same) concept with different (not similar) Role = Nr'

DefSim(Geo_location, Country) **=** 6/9 + $\sigma \times$ 0 **= 0.66**

7. Similarity Evaluation

7.1 Term's Similarity Calculation

4) Total Term's Similarity

a. Concept Similarity

For concept c_1 and c_2,

$ConSim(c_1, c_2) = Max[\ JaroSim(c_1, c_2), TaxSim(c_1, c_2), DefSim(c_1, c_2)]$

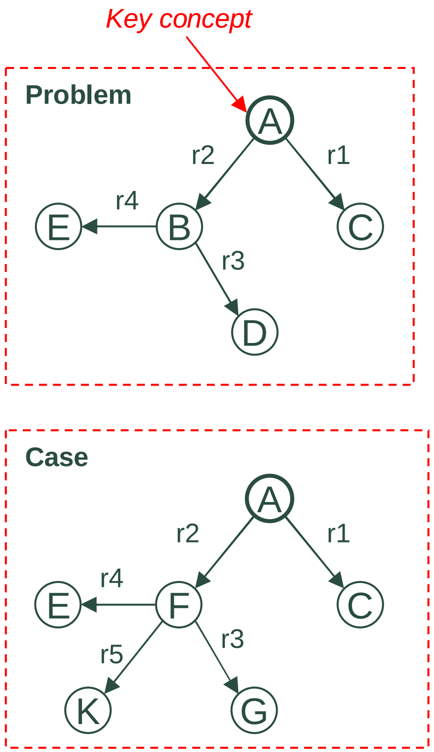
b. Relation Similarity

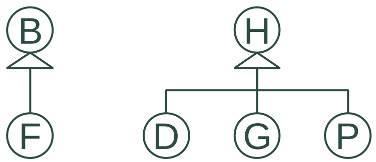
For relation r_1 and r_2,

$RelSim(r_1, r_2) = Max[\ JaroSim(r_1, r_2), TaxSim(r_1, r_2), DefSim(r_1, r_2)]$

7. Similarity Evaluation

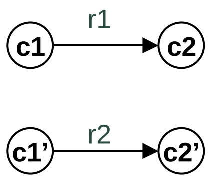
7.2 Graph Structure's Similarity Calculation





Axiom; D ≠ P

Problem \ Case	A –r2→ F	A –r1→ C	F –r4→ E	F –r5→ K	F –r3→ G	Max
A –r2→ B	0.9	0	0	0	0	0.9
A –r1→ C	0	1	0	0	0	1
B –r4→ E	0	0	0.9	0	0	0.9
B –r3→ D	0	0	0	0	0.54	0.54



If ConSim(c1, c1') > 0,
SubNetSim(Net1, Net2) =
RelSim(r1, r2) * [α * ConSim(c1, c1') + β * ConSim(c2, c2')] / 2
If ConSim(c1, c1') = 0
SubNetSim(Net1, Net2) = 0

**** α and β are matching weight value for concepts of c1 and c2, respectively.***
0 < α and β ≤ 1

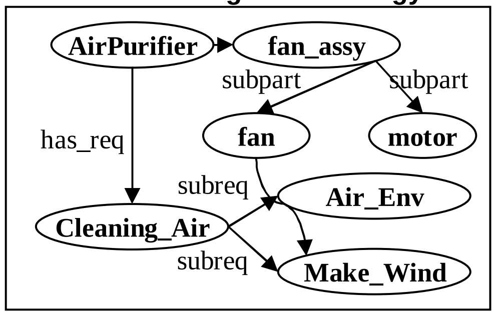
***NetSim*(Problem, Case)** = **Average**(0.9, 1, 0.9, 0.54) = **0.817**

8. Case Search with Networks

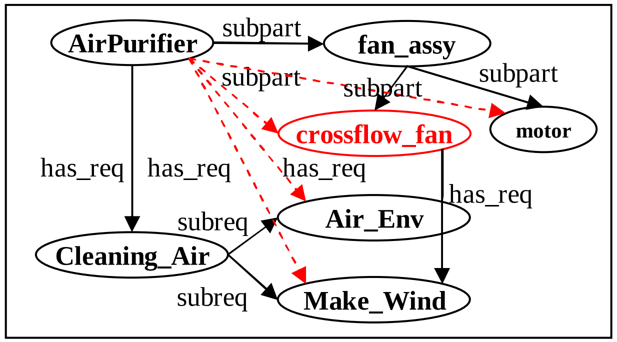
8.1 Searching CSN

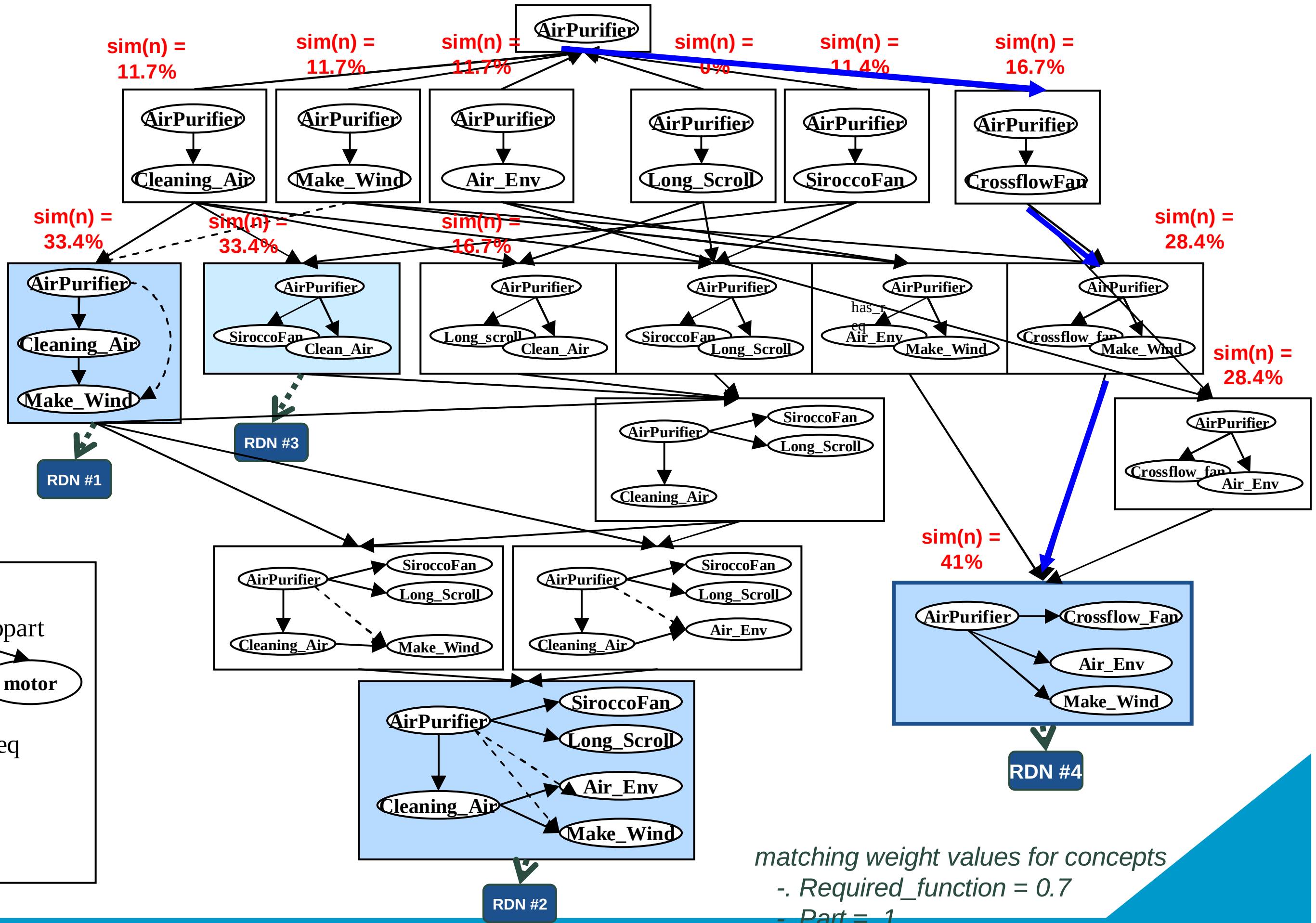
Searching algorithm: *Best-first search with a threshold*

Concept structure before reasoning with ontology



Concept structure after reasoning with ontology.

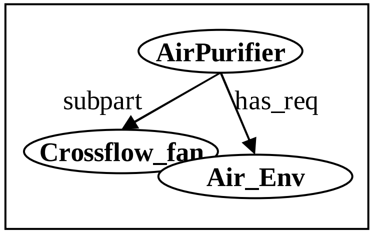


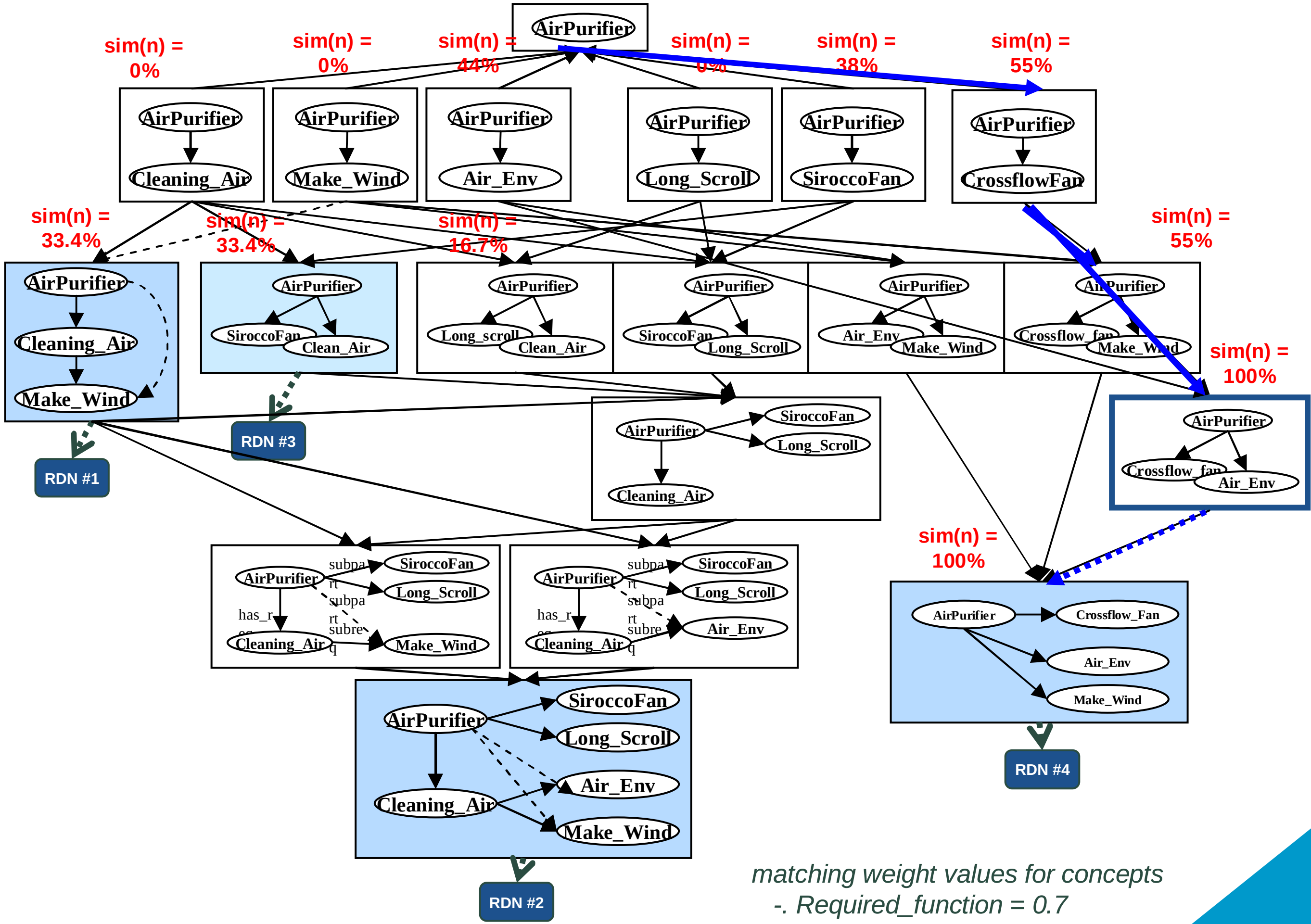


matching weight values for concepts
-. Required_function = 0.7
-. Part = 1

8.1 Searching CSN

Concept structure of a new prob.





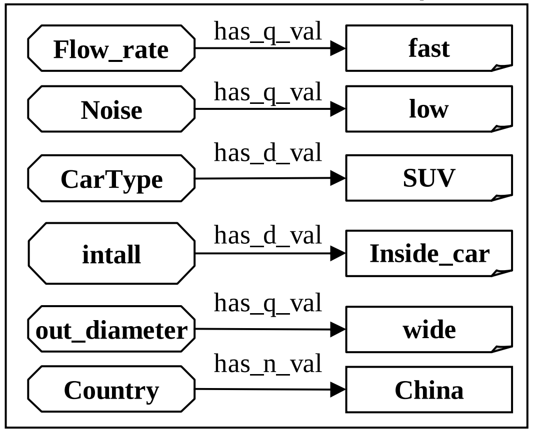
matching weight values for concepts
-. Required_function = 0.7
-. Part = 1

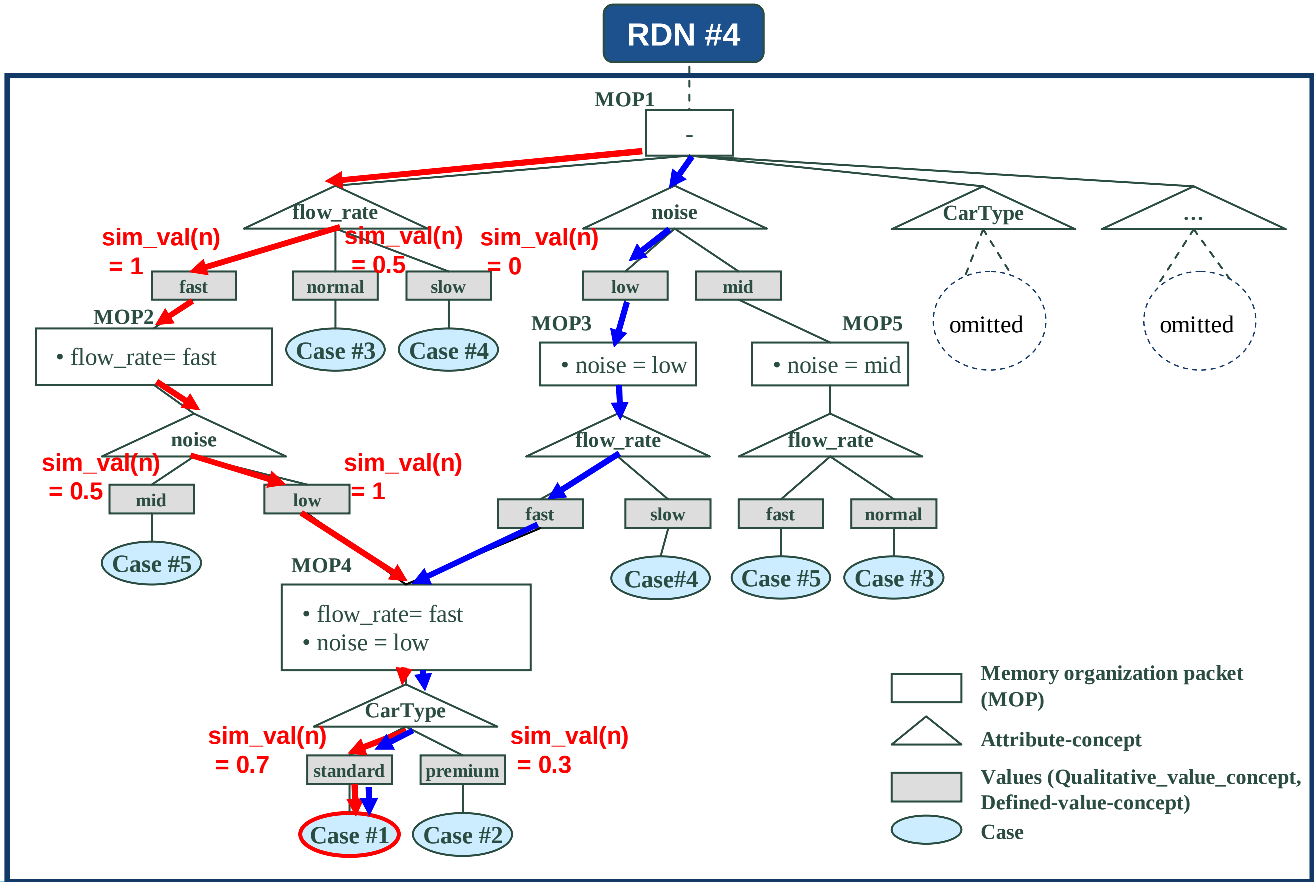
8. Case Search with Networks

8.2 Searching RDN

1) Use parallel searching for RDN connected with the case-structure-node
2) Find a union set of cases after parallel searching
3) Find the most similar case through full partial match with the found cases.

attributes and values of a new prob.





9. Discussion

Representation	**Limitations**
Text	Is hard to calculate similarity. NLP is required
Attribute-value list	can not represent relations in a data model
Object-oriented	can not represent a case beyond a class model
Graph	Graph similarity calculation is required.
Graph + Ontology	Ontology construction is required.

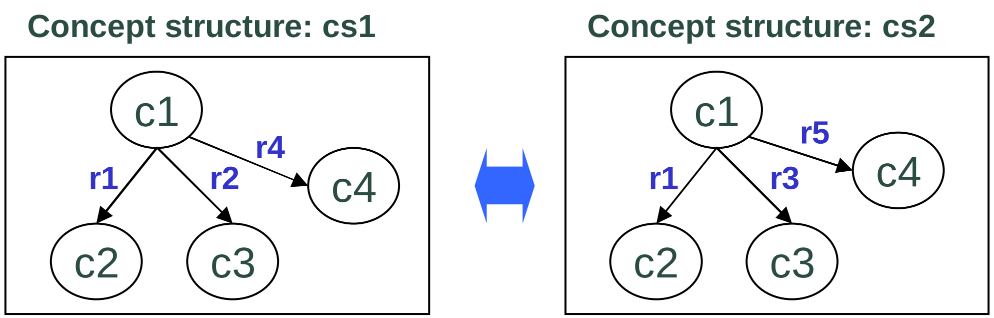
Why the Graph + Ontology representation is better than Graph representation?

1) The ontology can be used to infer taxonomy relations of new terminology.
2) The ontology can be used to infer new relations between concepts.

9. Discussion

Why CSN is effective to search a similar concept structure?

If we only compare concepts in concept structures,
we can miss the meanings of relations in the concept structure.



Axiom
∀x,y r4(x,y) → r5(x,y)

Concept vector similarity (FEM paper)
CBS(cs1, cs2) = 100%

CBS(v1, v2) = (v1 * v2) / ||v1|| * ||v2||

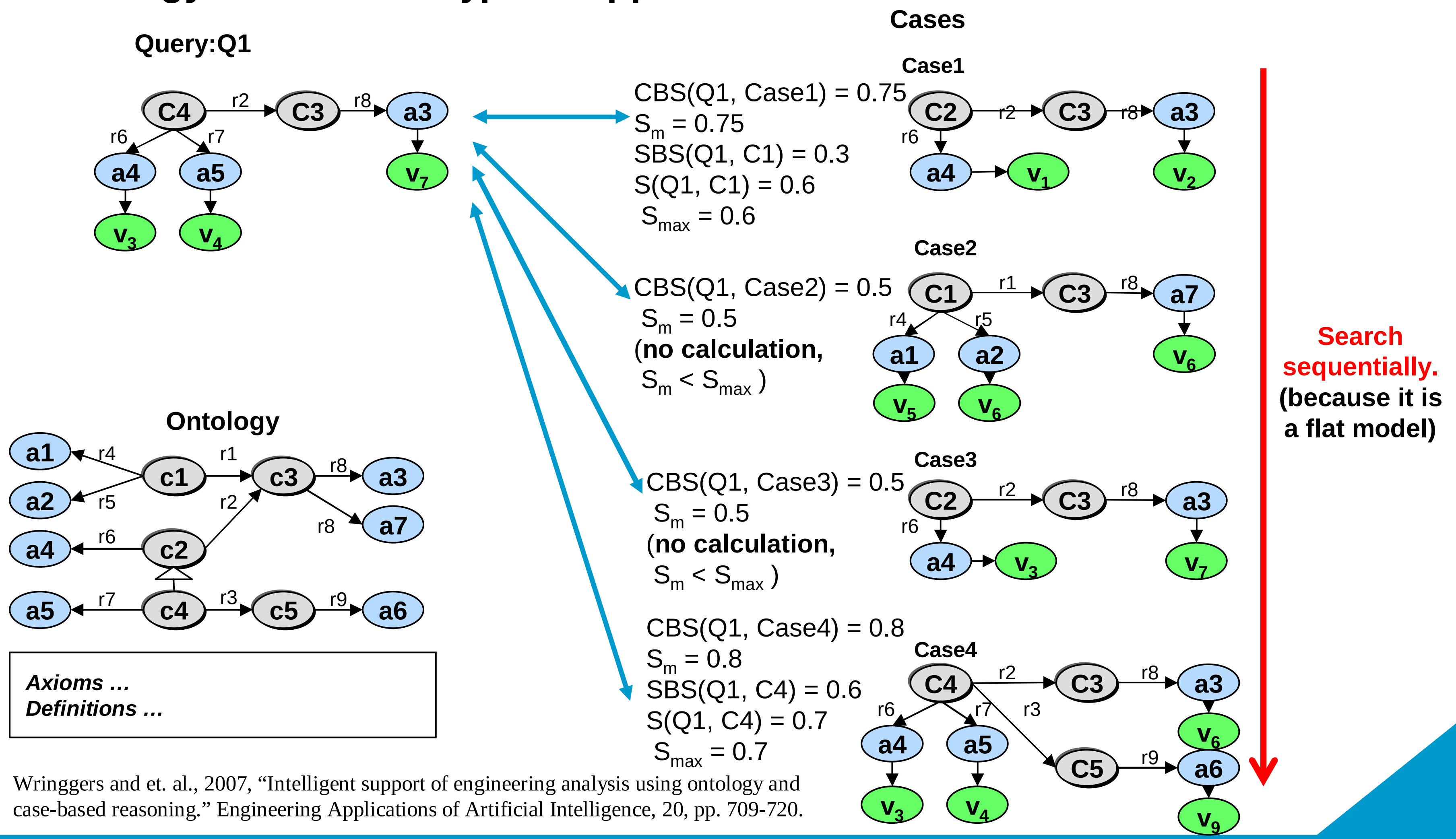
v1=[1, 1, 1, 1]
v2=[1, 1, 1, 1]

CSN similarity (Our approach)
NetSim(cs1, cs2) = 56.6%

NetSim(cs1, cs2) =
(rel_sim(r1, r1) * con_sim(c2, c2)
+ rel_sim(r2, r3) * con_sim(c3, c3)
+ rel_sim(r4, r5) * con_sim(c4, c4))
/ 3
= (1*1 + 0*1 + 0.7*1)/3 = 1.7/3 = 0.566

9. Discussion

Ontology-based CBR Typical Approach



Wringgers and et. al., 2007, "Intelligent support of engineering analysis using ontology and case-based reasoning." Engineering Applications of Artificial Intelligence, 20, pp. 709-720.

10. Conclusion

We propose the Architecture of Ontology-based CBR approach.

Problems

1. How to overcome ‘providing ontology definition’
2. How to overcome ‘reasoning overhead’
3. How to construct ‘network’ considering domain’
4. How to develop ‘more efficient search method’