Consistency checking of legal norms using LegalRuleML

A case study on the consistency between GDPR and BDSG-E

Chun Tian
Scuola di Scienze, Università di Bologna
chun.tian@studio.unibo.it
Numero di matricola: 0000735539

Initial Project Goals

> *"The issues of RuleML consistency and checking is still hot if linked/applied to the GDPR[1]. This is the first EU regulation on data protection and each country (Italy, Germany, France etc) will issues their own GDPR that has to be in harmony with the EU GDPR. Having the EU GDPR written in RuleML and the same for German GDPR will allow to check that consistency among these two regulations."*
> — *Prof. Danilo Montesi, June 28, 2017*

1. Having the EU GDPR written in (Legal)RuleML;
2. The same for German GDPR (BDSG-E);
3. Checking consistency among these two regulations.

1 https://en.wikipedia.org/wiki/General_Data_Protection_Regulation

Outlines

2 Initial Project Goals
4 General Data Processing Regulation (GDPR)
5 The Outline of Regulation (EU) 2016/679
6 BDSG-E
7 The Outline of BDSG-E
8 OASIS LegalRuleML
9 LegalRuleML Sub-standards
10 Structural Authoring in Adobe© FrameMaker
12 The Need of Ontology
13 The Cyc Ontology (cyc.com)
15 Transforming GDPR into LRML
16 Legal Reasoning: the Cognitive Approach
17 Defeasible Reasoning
18 Deontic Logic
19 Overview of Defeasible Reasoning Engines
20 The OSCAR project (John L. Pollock)
21 Defeasible Deontic Logic in OSCAR
22 From LRML to Defeasible Deontic Logic
23 The Definition of Consistency
24 Consistency Checking in OSCAR
25 Consistencies between GDPR & BDSG-E (1)
26 Consistencies between GDPR & BDSG-E (2)
27 Consistencies between GDPR & BDSG-E (4)
28 Semi-automatic tranlations using NLP
29 Conclusions
30 Bibliography

General Data Processing Regulation (GDPR)

- **Made by European Parliament & Council;**
- **Intends to strengthen and unify data protection for all individuals within the European Union;**
- **Address the export of personal data outside the EU;**
- **Replaces the old EU Data Protection Directive (officially Directive 95/46/EC) of 1995.**

» Adopted on 27 April 2016.
» Becomes enforceable from 25 May 2018 after a two-year transition period.
» Published as part of Official Journal L 119

1. Regulation (EU) 2016/679 of the European Parliament and of the Council of 27 April 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing Directive 95/46/EC (General Data Protection Regulation);

2. Directive (EU) 2016/680 of the European Parliament and of the Council of 27 April 2016 on the protection of natural persons with regard to the processing of personal data by competent authorities[1] for the purposes of the prevention, investigation, detection or prosecution of criminal offences or the execution of criminal penalties, and on the free movement of such data, and repealing Council Framework Decision 2008/977/JHA;

3. Directive (EU) 2016/681 of the European Parliament and of the Council of 27 April 2016 on the use of passenger name record (PNR) data for the prevention, detection, investigation and prosecution of terrorist offences and serious crime.

1 (a) any public authority competent for the prevention, investigation, detection or prosecution of criminal offences or the execution of criminal penalties, including the safeguarding against and the prevention of threats to public security; or

(b) any other body or entity entrusted by Member State law to exercise public authority and public powers for the purposes of the prevention, investigation, detection or prosecution of criminal offences or the execution of criminal penalties, including the safeguarding against and the prevention of threats to public security.

The Outline of Regulation (EU) 2016/679

1. General provisions
2. Principles
3. Rights of the data subject
   - 3.1. Transparency and modalities
   - 3.2. Information and access to personal data
   - 3.3. Rectification and erasure
   - 3.4. Right to object and automated individual decision-making
   - 3.5. Restrictions
4. Controller and processor
   - 4.1. General obligations
   - 4.2. Security of personal data
   - 4.3. Data protection impact assessment and prior consultation
   - 4.4. Data protection officer
   - 4.5. Codes of conduct and certification
5. Transfers of personal data to third countries or international organisations
6. Independent supervisory authorities
   - 6.1. Independent status
   - 6.2. Competence, tasks and powers
7. Cooperation and consistency
   - 7.1. Cooperation
   - 7.2. Consistency
   - 7.3. European data protection board
8. Remedies, liability and penalties
9. Provisions relating to specific processing situations
10. Delegated acts and implementing acts
11. Final provisions

(99 articles in total)

BDSG-E

In February 2017, the German Federal Government submitted a draft law (BDSG-E)4 for adapting the GDPR in Germany. The administration wants to push the draft quickly through the Federal Parliament. After a hearing of experts for data protection on the 27th of March, the final enacting is planned for April. The law will likely come into force in May 2017 without significant further changes.

- **Entwurf eines Gesetzes zur Anpassung des Datenschutzrechts an die Verordnung (EU) 2016/679 und zur Umsetzung der Richtlinie (EU) 2016/680 (Datenschutz-Anpassungs- und -Umsetzungsgesetz EU — DSAnpUG-EU)**

or in English:

- **Draft Law on the adaptation of the Data Protection Act into line with Regulation (EU) 2016/679 and implementing Directive (EU) 2016/680 (EU Data Protection Policy and Implementation Act)**

Four parts:

1. Common provisions;
2. Implementing provisions for processing for the purposes of § 2 of Regulation (EU) 2016/679;
3. Provisions for processing as referred to in § 1 (1) of Directive (EU) 2016/680;
4. Specific provisions for processing under activities not covered by the scope of application of Regulation (EU) 2016/679 and Directive (EU) 2016/680

The Outline of BDSG-E

- **Common provisions**

1. Scope and definitions
2. Legal basis of the processing of personal data
3. Data Protection Officer of public authorities
4. The Federal Commissioner for Data Protection and Information Freedom
5. Representation in the European Data Protection Committee, focal point, cooperation of supervisory Visual authorities of the Federal and State Governments in European Union Affairs
6. Appeals

- **Implementing provisions for processing for the purposes of § 2 of Regulation (EU) 2016/679**

1. Legal basis of the processing of personal data
1.1. Processing of special categories of personal data and processing for other purposes
1.2. Special processing conditions
2. Rights of the data subject
3. Duties of the controller and processor
4. Supervisory authority for data processing by non-public institutions
5. Penalties
6. Appeals

- **Provisions For processing as referred to in § 1 (1) of Directive (EU) 2016/680**

1. Scope, definitions and general principles for the processing of personal data
2. Legal basis of the processing of personal data
3. Rights of the data subject
4. Duties of the controller and processor
5. Data transfers to third countries and to international organisations
6. Cooperation of between supervisory authorities
7. Liability and sanctions

- **Specific provisions for processing under activities not covered by the scope of application of Regulation (EU) 2016/679 and Directive (EU) 2016/680**

OASIS LegalRuleML

» XML-based legal rule interchange format proposed by OASIS, which extends RuleML with features specific to legal domain.

» Features representing the particularities of the legal normative rules with a rich, articulated, and meaningful markup language:

- **defeasibility of rules and defeasible logic;**
- **deontic operators (e.g., obligations, permissions, prohibitions, rights);**
- **semantic management of negation;**
- **temporal management of rules and temporality in rules;**
- **classification of norms (i.e., constitutive, prescriptive);**
- **jurisdiction of norms;**
- **isomorphism between rules and natural language normative provisions;**
- **identification of parts of the norms (e.g. bearer, conditions);**
- **authorial tracking of rules.**

A bottom-up approach for building LRML documents from legal norms:

1. `<ruleml:Atom>` node as atomic relation predicates;
2. `<ruleml:Var>` and `<ruleml:Ind>` nodes representing rule variables and individuals;
3. Logical connectors (`<ruleml:And>`, `<ruleml:Or>`, `<ruleml:Neg>`, ... for building complex clauses;
4. `<ruleml:Rule>` (with `<ruleml:if>` and `<ruleml:then>`) as the core of legal norms;
5. Deontic operators: `<lrml:Obligation>`, `<lrml:Prohibition>`, `<lrml:Permission>`, ...
6. Defesibility by `<lrml:hasStrength>`;
7. Rule statements (Prescriptive, Constituitive, Factual);
8. Reparation Statements and Penalty Statements;
9. Specificity of statements by `<lrml:Override>`;
10. `<lrml:Context>` connecting statements with all other properties (temporal characteristics, jurisdiction, sources, isomorphism, ...).

LegalRuleML Sub-standards

LRML 1.0 Basic

1. ~~<lrml:hasPrefix>~~
2. No sample LRML documents

LRML 1.0 Compact

1. <ruleml:content> (in <ruleml:Time>, <ruleml:Spatial>, ...)
2. <ruleml:if> and <ruleml:then> are empty elements in flat <ruleml:Rule>; (comparing with LRML Normal)

LRML 1.0 Normal

1. <lrml:hasPrefix>
2. <lrml:hasComment>
3. <lrml:hasLegalReferences>
4. <lrml:hasReferences>
5. <lrml:hasSources>
6. <lrml:hasTimes>
7. <lrml:hasTemporalCharacteristics>
8. <lrml:hasAgents>
9. <lrml:hasFigures>
10. <lrml:hasRoles>
11. <lrml:hasAuthorities>
12. <lrml:hasJurisdictions>
13. <lrml:hasAssociations>
14. <lrml:hasAlternatives>
15. <lrml:hasContext>
16. <lrml:hasStatements>

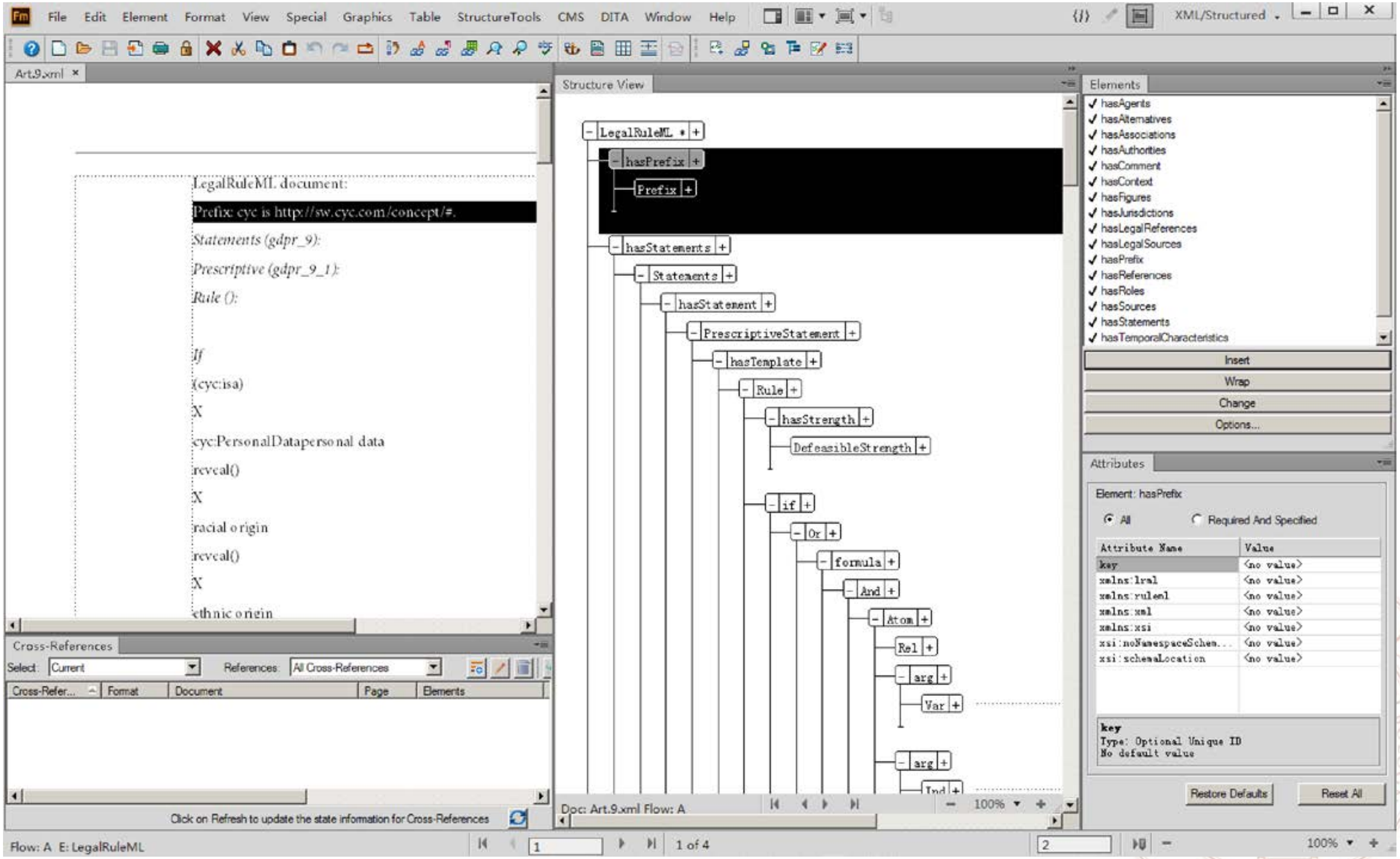
Structural Authoring in Adobe© FrameMaker

» Desktop-based XML authoring environment designed for LRML
» Based on FrameMaker and XSLT 2
» Minimized coding efforts
» Freely available to all FrameMaker users
» Major Features:

1. Grammar checking and XML validations;
2. Quickly selecting LRML elements and attributes from candidate list;
3. XML code folding;
4. Automatic insertion of inner elements;
5. Automatic tracking of cross-references.

Steps to build a FrameMaker Structured Application

1. Convert XML Schema into DTDs;
2. Create initial FrameMaker EDD from DTD;
3. Set highest level element in EDD;
4. Create initial document template;
5. Import EDD into document template;
6. Test document template on sample XML code;
7. Create initial read/write rules;
8. Save new DTD from EDD;
9. Import new DTD into EDD;
10. Add text formating rules into EDD;
11. Import EDD into document template;
12. Test again on sample XML code;
13. Identify and create cross-reference elements in EDD;
14. Add more text formating rules into EDD;
15. Repeat steps 11-14, until satisfied.


File Edit Element Format View Special Graphics Table StructureTools CMS DITA Window Help
XML/Structured
Art.9.xml
LegalRuleML document:
Prefix: cyc is http://sw.cyc.com/concept/#.
Statements (gdpr_9):
Prescriptive (gdpr_9_1):
Rule ():
If
(cyc:isa)
X
cyc:PersonalDataperson al data
reveal()
X
racial o rigin
reveal()
X
ethnic o rigin
Cross-References
Select: Current
References: All Cross-References
Cross-Refer...
Format
Document
Page
Elements
Click on Refresh to update the state information for Cross-References
Structure View
LegalRuleML *
hasPrefix
Prefix
hasStatements
Statements
hasStatement
PrescriptiveStatement
hasTemplate
Rule
hasStrength
DefeasibleStrength
if
Or
formula
And
Atom
Rel
arg
Var
arg
Doc: Art.9.xml Flow: A
100%
Elements
hasAgents
hasAlternatives
hasAssociations
hasAuthorities
hasComment
hasContext
hasFigures
hasJurisdictions
hasLegalReferences
hasLegalSources
hasPrefix
hasReferences
hasRoles
hasSources
hasStatements
hasTemporalCharacteristics
Insert
Wrap
Change
Options...
Attributes
Element: hasPrefix
All
Required And Specified
Attribute Name
Value
key
xmlns:lral
xmlns:ruleml
xmlns:xml
xmlns:xsi
xsi:noNamespaceSchem...
xsi:schemaLocation
<no value>
key
Type: Optional Unique ID
No default value
Restore Defaults
Reset All
Flow: A E: LegalRuleML
1
1 of 4
2
100%

The Need of Ontology

» LRML and RuleML elements cannot express all information in any legal sentence;
» Some parts (e. g. Definitions) of legal norms are not rules;
» Relations used in RuleML `<ruleml:Atom>` and `<ruleml:Ind>` are not defined inside RuleML;
» For legal reasoning purposes, it may require more than a vocabulary of unrelated names.
» Ontology as XML namespaces, declared by `<lrml:Prefix>`;
» Ontology as connections to Semantic Web and Knowledge Representation.
» Legal Ontologies.
» "LegalRuleML is independent from any legal ontology and logic framework."

» Examples: (Art. 4 GDPR)

For the purpose of this Regulation:
(1) 'personal data' means ...
(2) 'processing' means ...
(3) 'restriction of processing' means ...
(4) 'profiling' means ...
(5) 'pseudonymisation' means ...
(6) 'filing system' means ...
...

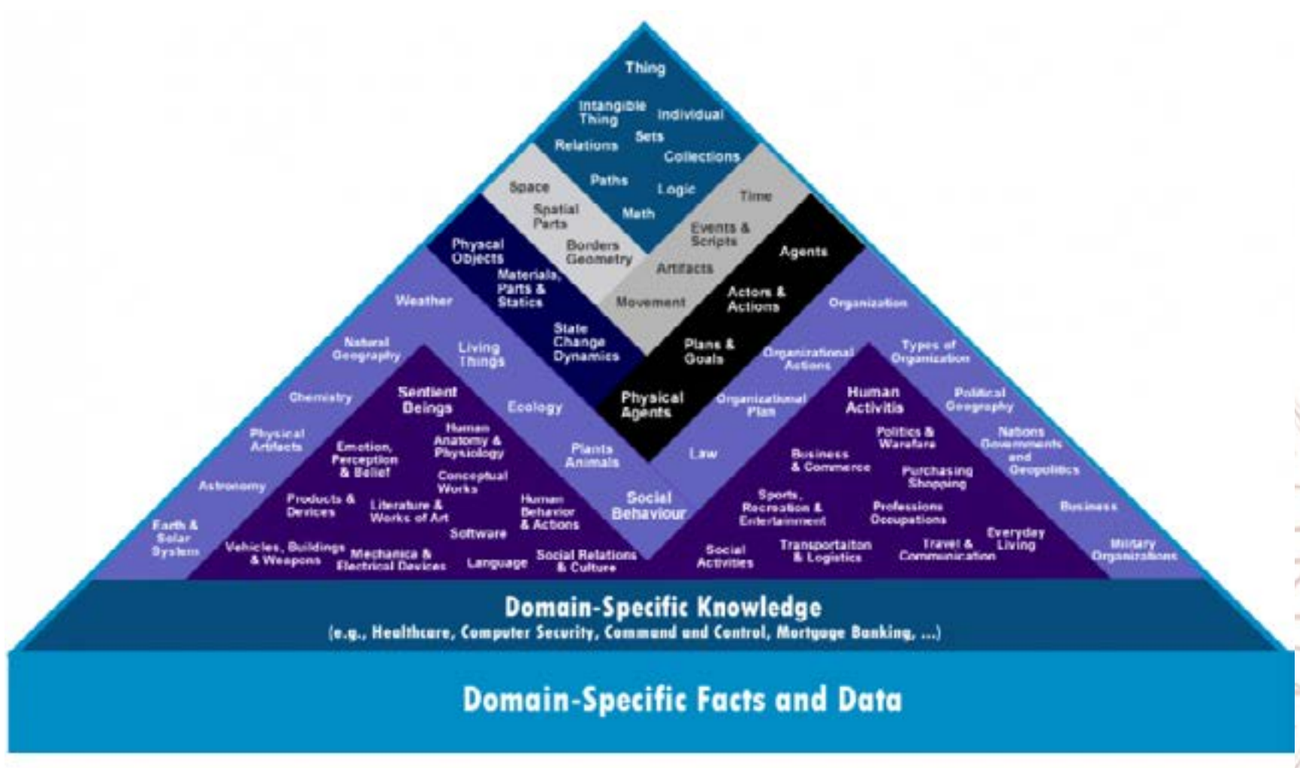
1. Above article are NOT any kind of "rules";
2. They're new concepts used as relations and individuals in legal rules;
3. Any legal norm may contain new concepts not covered by any existing legal ontologies.
4. Relations between concepts may be useful as additional information for legal reasoning purposes.

The Cyc Ontology (cyc.com)

» Started in 1984 by Douglas Lenat at MCC and is developed by the Cycorp company;
» "To counter a then ominous Japanese effort in AI, the 'fifth-generation' project";
» Cyc is an upper ontology with common-sense knowledge;
» Cyc was built by a team of trained ontologies/logicians.
» Released as both OWL file and independent reasoning engine.
» Three versions: EnterpriseCyc, ResearchCyc and OpenCyc (now cancelled):

OpenCyc contains most of the concepts (terms) that are in the full Cyc KB, but only a subset of the information about each of those terms. For instance, OpenCyc will know about birds and animals, will know that birds are animals, will know about wings and the relation anatomicalParts, but OpenCyc does not contain the fact that birds have wings as anatomicalParts.

» Cyc engine is written in Common Lisp, now running on top of Java Virtual Machine.



Permission Search Clear

Assert Compose KE Text Create Documentation History Query

You are: CycAdministrator [Logout]
Server: binghe-mac5:3600
Preferences Tools

Agreement order by : predicate **ontology**
filters : predicate ontology
index view : **inferred** legacy

▶ Browsing ▶ Editing ▶ Advanced

All Assertions *(286)* ▶ open all
▼ via Agreement *(128)* !
- isa *(5)*
- isa arg2 *(5)*
- genls
- genls arg2 *(29)*
- disjointWith
- comment
- arg1Genl arg2
- arg1Isa arg2 *(23)*
- arg2Isa arg2 *(9)*
- argGenl arg3
- argIsa arg3 *(33)*
- prettyString *(14)*
- prettyString-Canonical
- synonymousExternalConcept *(4)*

▶ via PropositionalConceptualWork *(2)*
▶ via ConceptualWork *(3)*
▶ via DevisedPracticeOrWork *(5)*
▶ via IntangibleExistingThing
▶ via AspatialInformationStore *(2)*
▶ via Artifact-NonAgentive *(2)*
▶ via Artifact-Generic *(6)*
▶ via InformationStore *(4)*
▶ via CulturalThing
▶ via IntangibleIndividual
▶ via SomethingExisting *(12)*
▶ via TemporallyExistingThing *(2)*
▶ via TemporalThing *(5)*
▶ via AspatialThing

Collection : Agreement (+)

on the term

isa : ObjectType TemporalStuffType
isa : ClarifyingCollectionType
isa : QAClarifyingCollectionType
isa : Agreement-Topic
genls : Policy
disjointWith : MediaSeriesProduct
comment : "A specialization of Policy (q.v.). Each instance of this collection involves two or more parties who agree that certain propositions (which are the contents of a corresponding Microtheory -- see the shared note #$AgreementNote) should be true. Making the propositions true may require some action or commitment of resources on the part of one or more of the agreeingAgents. Thus, instances of Agreement will usually involve some instances of Obligation. Instances of Agreement and Obligation differ, however, in that an obligatedAgents is responsible for the truth of all of the propositions over which the obligation holds. In an Agreement some agents may not be so responsible. For example, in a loan agreement, the borrower agrees to give the lender back the money, but the borrower is the only obligatedAgents for the repayment. Note that obligatedAgents need not be among the agreeingAgents in the agreement that involves or generates the obligation. Moreover, agreeingAgents aren't always obligatedAgents; e.g., Wanda and Paul may agree that Paul alone is obligated to do some task. Notable specializations of Agreement include PeaceAccord, LegalAgreement, InformalAgreement, BusinessPartnershipAgreement, WorkAgreement, MaintenanceAgreement, Reservation and Appointment. For assertions about what is "supposed to be" true, given some Agreement, see ist-Agreement."
prettyString : "assent" "pacts" "covenants" "covenant" "accords" "accord" "testaments" "testament" "agreements" "agreement" "compacts" "compact" "deals" "deal"
prettyString-Canonical : "pact"
(synonymousExternalConcept Agreement (OWLOntologyFn "http://www.w3.org/2006/03/wn/wn20/instances") "synset-bargain-noun-1")
(synonymousExternalConcept Agreement (OWLOntologyFn "http://www.w3.org/2006/03/wn/wn20/instances") "synset-agreement-noun-1")
(synonymousExternalConcept Agreement (OWLOntologyFn "http://www.w3.org/2006/03/wn/wn20/instances") "synset-accord-noun-4")
(synonymousExternalConcept Agreement (OWLOntologyFn "http://www.w3.org/2006/03/wn/wn20/instances") "synset-accord-noun-2")

Copyright © 1995 - 2017 Cycorp. All rights reserved.

Update Comm: Storing Only Agenda: Idle **KB:** 5022 **Local:** (6) System: 10.140388 Learn about ResearchCyc

Transforming GDPR into LRML

» General principles:

1. Identify IF-THEN rules from legal texts expressed in natural language.
2. “Subject-matter and objectives” (Art. 1) as LRML Agents;
3. “Material and territorial scope” (Art. 2-3) as LRML Jurisdications;
4. “Law enforce dates” as LRML Temporal Characteristics;
5. “Definitions” (Art. 4) as Ontology entries defined outside of LRML.
6. “Lawfulness of processing” (Art. 6) as constitutive rules;
7. “Open clauses for Member States Law” as defeasible rules.
8. “Administrative fines” (Art. 83) as LRML Penalty Statements;
9. Most articles as LRML Prescriptive Statements.

» Implementation Issues:

1. Ontology is not ready (but can be built in systems like Cyc as the same time when translating in LRML.
2. The complete XML representations for even single article is huge!
3. There’s no unique “correct” ways for such translations.

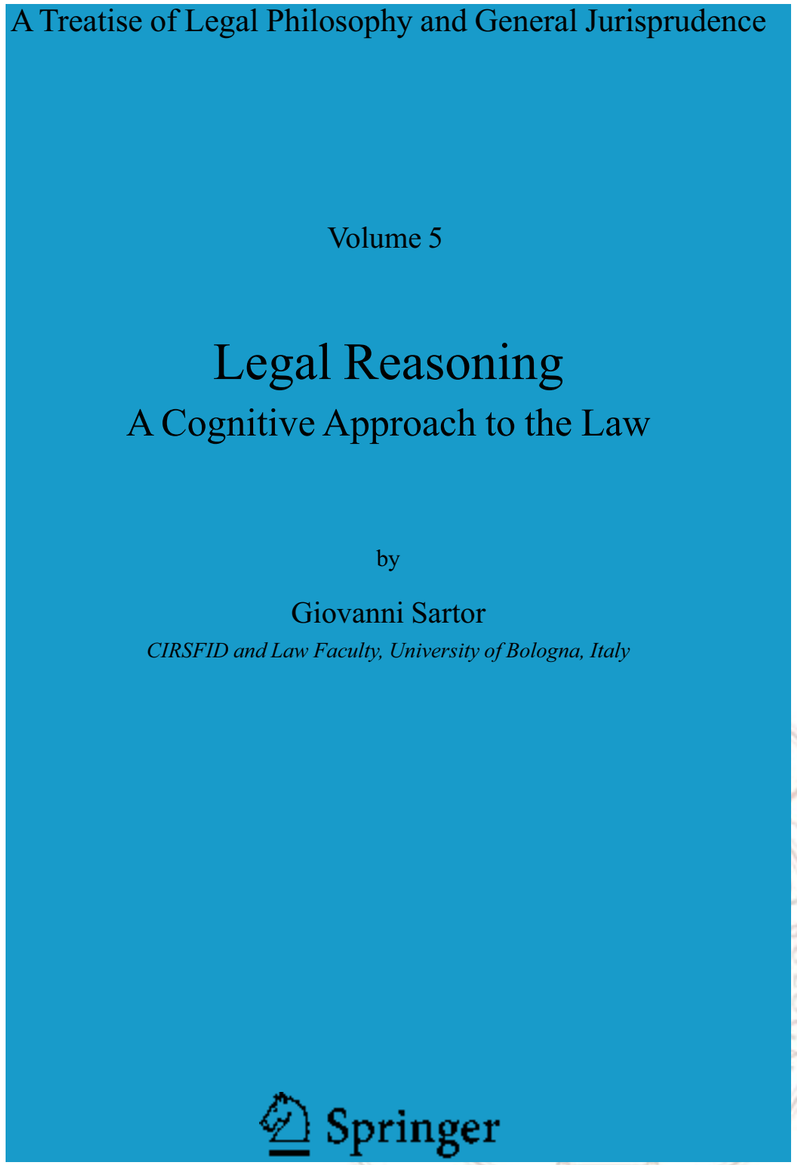
Legal Reasoning: the Cognitive Approach

» Legal reasoning can be viewed as an application of a more general human competence, which we call practical rationality:

- **Implicit Cognition**
  - **Fixed reflexes**
  - **Conditioned reflexes**
- **Explicit Cognition (and Reasoning)**
  - **Epistemic Cognition (and Reasoning)**
    - **Type of cognitive states: Percepts**
    - **Type of cognitive states: Beliefs (A "is a reason for" B)**
  - **Practical Cognition (and Reasoning)**
    - **Legal Reasoning**
    - **Likings (or Preferences)**
    - **Desires (or Goals)**
    - **Intentions**
    - **Wants**
  - **Doxification: transferring into epistemic reasoning the reasoning schemata available for practical reasoning**

» Conclusive and Defeasible Reasoning

» Deontic Notions (Obligation, Prohibition, Permission, ...)



Defeasible Reasoning

» What philosophers call "defeasible reasoning" is roughly the same as non-monotonic reasoning in AI.

» Reasoning proceeds by constructing arguments, where reasons provide the atomic links in arguments.

» Basic concepts:

- **Conclusive reasons are reasons that are not defeasible. Conclusive reasons logically entail their conclusions.**
- **Those that are not conclusive are prima facie reasons. Prima facie reasons create a presumption in favor of their conclusion, but it can be defeated.**
- **Considerations that defeat prima facie reasons are defeaters.**
  - **Rebutting defeaters: reasons for denying the conclusion of prima facie reasons.**
  - **Undercutting defeaters: attacking the connection between the premises and the conclusion.**

» Example of undercutting defeaters:

- **'x looks red to me' is a prima facie reason for an agent to believe 'x is red';**
- **'x is illuminated by red lights and red lights can make things look red when they are not' is a defeater;**
- **But it is not a reason for thinking that x is not red (because red things also looks red under red lights), so it is not a rebutting defeater.**
- **Instead, it attacks the connection between 'x looks red to me' and 'x is red', giving us a reason for doubting that x wouldn't look red unless it were red.**

» Rebutting defeaters are more common in legal norms;

» Undercutting defeaters are not explicitly supported in LRML but can be encoded as special <lrml:hasStrength> elements.

Deontic Logic

The branch of philosophical logic that is specifically concerned with obligations, prohibitions and permissions (the term deontic derives from the Greek verb deomai which means being due or obligatory).

» Deontic Operators:

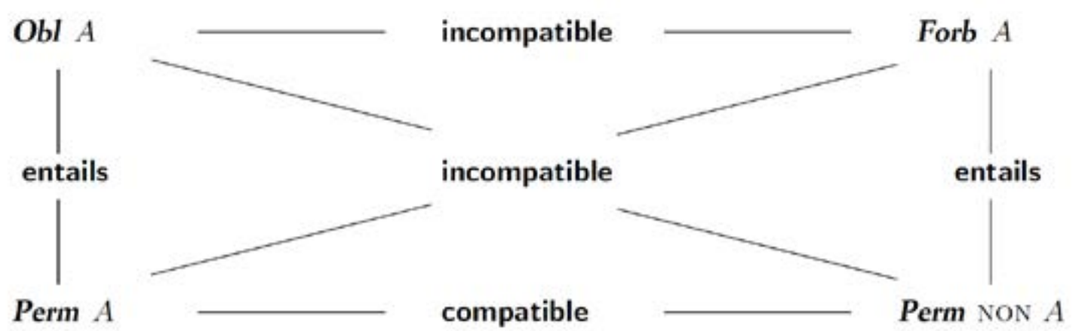
- **Obl: to say that an action is obligatory is to say that the action is due, has to be held, must be performed, is mandatory or compulsory.**
- **Forb: being forbidden or prohibited is the status of an action that should not be performed.**
- **Perm: the doxification of a may-intention, by which we mean one's intention that something may be done.**
- **Facult: an action A is facultative when both A and A's omission are permitted.**

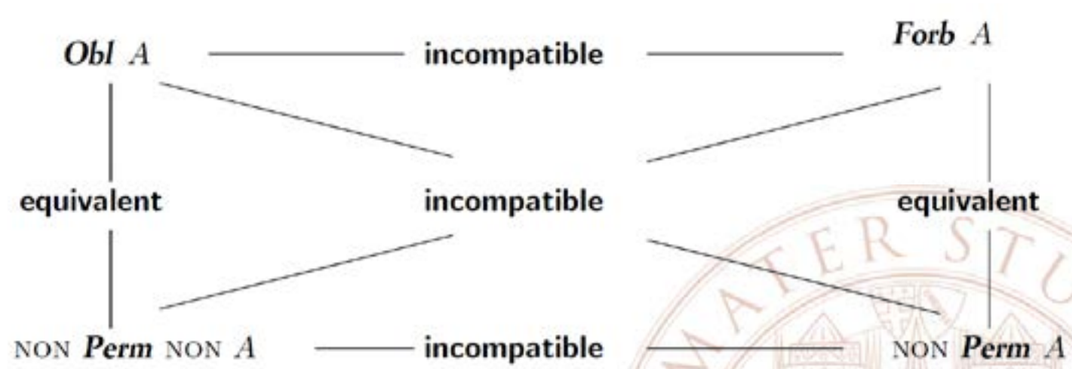
» Deontic Logic Axioms:

Deontic logic

FON: **Forb** $A \equiv$ **Obl** (NON A)
OP: **Obl** $A \vdash$ **Perm** A
PNF: **Forb** $A \equiv$ NON (**Perm** A)
JO: (**Obl** A AND **Obl** B) $\vdash$ **Obl** (A AND B)

- **Connections between Obligation and Prohibition and Permission:**





Facult $A \equiv$ (**Perm** A) AND (**Perm** NON A)
(⌈*it is facultative that A*⌉ *is equivalent to* ⌈*it is permitted that A* AND *it is permitted that* NON *A*⌉)

Overview of Defeasible Reasoning Engines

System Name (ref)	Reasoning	Lang Supported	Technology	Interface	Complexity Analysis	Empirical Eval	Extra Notes
Nathan ([47])	Defeasible	First order	Prolog meta	Command line	None	Simple benchmarks	None
d-Prolog ([55])	Defeasible	First order	Prolog meta	Command line	None	Simple benchmarks (See [51] and [4])	None
EVID ([16])	Defeasible	First order	Prolog meta	Command line	None (although discussed)	Simple benchmark	*howdefeatit* query
OSCAR ([60])	Defeasible	First order	LISP program	Command line	None (although considered)	Unavailable	Used within "rational agent" implementation
Deimos ([51, 62])	Defeasible	Propositional	Haskell	Web and Command line	Undertaken (available in seperate paper)	Comprehensive benchmarking with DTScale	Only supports query answering
Delores ([51])	Defeasible	Propositional	C	Command line	Undertaken (but not detailed)	Comprehensive benchmarking with DTScale	Only supports total answering and requires preprocessing of theory
Phobos ([64, 65])	Plausible	Propositional	Haskell	Web and Command line	Undertaken (but not included)	Comprehensive benchmarking with custom tools	None
DeLP ([40])	Defeasible (dialectic argumentation)	First order	Prolog/JAM	Web, Agent component and command line	See [20]	Simple benchmarks and anecdotal evidence	None
DR-PROLOG ([4])	Defeasible	First order	Prolog meta	Web and command line	None	Comprehensive benchmarking with DTScale	RuleML support and ambiguity propogation
IACAS ([74])	Argumentation	First order	LISP program	Command line	None	Simple benchmarks	Chisholms Theory of Knowledge support
AS ([76])	Argumentation	First order	Ruby	Web (command line)	None	Benchmarks	Produces diagram of argument structure
Vreeswijk's Admissible Defence Sets ([75])	Argumentation	First order	Ruby	Web (command line)	Detailed complexity analysis	Comprehensive examples	Discussion of empirical complexity analysis
Argue tuProlog ([12])	Argumentation	First order	Java	Software Component and GUI	None	None (although discussed)	API to allow integration
CaSAPI ([37])	Argumentation	First order	Prolog meta	Command line	None	Simple examples	Translation (by-hand) from a given formalism possibly required

The OSCAR project (John L. Pollock)

» OSCAR[1] is a complex architecture for the construction of artificial rational agents.

» This architecture provides the framework for a system of epistemic and practical cognition.

» OSCAR implements a sophisticated system of defeasible reasoning that enables it to deal defeasibly with:

- **Perception**
- **Change and persistence**
- **Causation**
- **Probabilities**
- **Plan construction and evaluation**

» User must implement specific kinds of reasoning within the architecture to support legal reasoning.

» We have ported[2] OSCAR from MCL to modern Common Lisp platforms.

The
OSCAR
Manual

Operating manual and annotated code for OSCAR — the general-purpose defeasible reasoner and architecture for a rational agent.

John L. Pollock
Department of Philosophy
University of Arizona
Tucson, Arizona 85721
(e-mail: pollock@ccit.arizona.edu)

1 http://johnpollock.us/ftp/OSCAR-web-page/oscar.html
2 https://github.com/binghe/OSCAR

Defeasible Deontic Logic in OSCAR

```
(def-forwards-reason FON1
    :forwards-premises “(Forb A)” :conclusions “(Obl ~A)” :variables A)

(def-forwards-reason FON2
    :forwards-premises “(Obl ~A)” :conclusions “(Forb A)” :variables A)

(def-forwards-reason OP
    :forwards-premises “(Obl A)” :conclusions “(Perm A)” :variables A)

(def-forwards-reason PNF1
    :forwards-premises “(Forb A)” :conclusions “~(Perm A)” :variables A)

(def-forwards-reason PNF2
    :forwards-premises “~(Perm A)” :conclusions “(Forb A)” :variables A)

(def-forwards-reason JO
    :forwards-premises “(Obl A)” “(Obl B)” :conclusions “(Obl A & B)”
    :variables A B :defeasible? t)

(def-backwards-reason i-FON1
    :backwards-premises “(Forb A)” :conclusions “(Obl ~A)” :variables A)

(def-backwards-reason i-FON2
    :backwards-premises “(Obl ~A)” :conclusions “(Forb A)” :variables A)

(def-backwards-reason i-OP
    :backwards-premises “(Obl A)” :conclusions “(Perm A)” :variables A)

(def-backwards-reason i-PNF1
   :backwards-premises “(Forb A)” :conclusions “~(Perm A)” :variables A)

(def-backwards-reason i-PNF2
   :backwards-premises “~(Perm A)” :conclusions “(Forb A)” :variables A)

(def-backwards-reason i-JO
    :backwards-premises “(Obl A)” “(Obl B)” :conclusions “(Obl A & B)”
    :variables A B :defeasible? t)
```

Demo

```
Problem #6
The Alternative Definition of Permission
Given premises:
Ultimate epistemic interests:
     (all A)((Perm (Does A)) <-> ~(Obl ~(Does A)))    interest = 1.0


================== ULTIMATE EPISTEMIC INTERESTS ==================
  Interest in (all A)((Perm (Does A)) <-> ~(Obl ~(Does A)))
  is answered affirmatively by node 13
------------------------------------------------------------

Elapsed time = 0.004 sec

========================================================================
ARGUMENT #1
This is a deductive argument for:
      (all A)((Perm (Does A)) <-> ~(Obl ~(Does A)))
 which is of ultimate interest.

7.  (Obl ~(Does c0)) supposing { (Obl ~(Does c0)) }     REDUCTIO-SUPPOSITION
9.  (Forb (Does c0)) supposing { (Obl ~(Does c0)) }     I-FON2 from { 7 }
10.  ~(Perm (Does c0)) supposing { (Obl ~(Does c0)) }     I-PNF1 from { 9 }
11.  ((Perm (Does c0)) -> ~(Obl ~(Does c0)))   CONTRA-CONDITIONALIZATION from { 10 }
1.  ~(Perm (Does c0)) supposing { ~(Perm (Does c0)) }     REDUCTIO-SUPPOSITION
4.  (Forb (Does c0)) supposing { ~(Perm (Does c0)) }     I-PNF2 from { 1 }
5.  (Obl ~(Does c0)) supposing { ~(Perm (Does c0)) }     I-FON1 from { 4 }
6.  (~(Obl ~(Does c0)) -> (Perm (Does c0)))     CONTRA-CONDITIONALIZATION from { 5 }
12.  ((Perm (Does c0)) <-> ~(Obl ~(Does c0)))     bicondit-intro from { 6 , 11 }
13.  (all A)((Perm (Does A)) <-> ~(Obl ~(Does A)))     UG from { 12 }

========================================================================


Cumulative size of arguments = 10
Size of inference-graph = 13 of which 1 were unused suppositions.
76% of the inference-graph was used in the argument.
17 interests were adopted.
5 suppositions were made.
```

From LRML to Defeasible Deontic Logic

(Taken ideas from existing research[1])

» General idea:

- **LegalRuleML implements defeasibility as within the law where the precedent of a rule is satisfied by the facts of a case, then assumably the conclusion of the rule holds, but not necessarily.**
- **The business logic can be transformed into a target language of a rule-based system to execute.**

» Translation of LRML statements:

- **Norm Statements (prescriptive or constitutive):**

IF-THEN rules can be translated directly.

body -> head

- **Factual Statements:**

Logical predicates.

P x y z ...

- **Override Statements:**

Justifications (ranged from 0 to 1 in OSCAR) or relative specifity (DL) of statements.

- **Violation-Reparation Statements:**

They're the "ELSE" part of IF-THEN rules.

body -> head ⊗ reparation

Note: the formula (P ⊗ Q) (read "It is false that P would not be true unless Q were true", or abbreviated as "P does not guarantee Q").

» Transforms can be implemented in XSLT 2.0 as a portable solution.

- **It's hard to figure out "what" to transform;**
- **It's easy (pure engineering issue) to actually write the working XSL sheets.**

» Position of LegalRuleML in the project:

- **LRML as a rule interchange format;**
- **Use "LRML normalized" format for easy processing;**
- **Same transform approach, different logics;**
- **Shared code (for all reasoners) in XSLT.**

1 Lam, Ho-Pun, Mustafa Hashmi, and Brendan Scofield. "Enabling reasoning with LegalRuleML." International Symposium on Rules and Rule Markup Languages for the Semantic Web. Springer International Publishing, 2016.

The Definition of Consistency

» Consistency in Propositional Logic and First-Order Logic:

One cannot derive a contradiction from a set of propositions.

» (Weakened) "**deductive consistency**" in Defessible Logic:

A set of propositions is deductively consistent if and only if it does not have an explicit contradiction as a deductive consequence.

» What's the "consistency" between an EU regulation and its adoption in a Member State? We think:

- **An EU regulation is consistent (in a Member State) if all its conclusions are not retracted by its adoptions in that member state (e.g., Germany), unless the regulation clearly states that it's defeasible.**
- **Any conclusion made by a member state law, should be doubly checked in related EU regulations, and the member state law is consistent (within EU) only when the EU regulation makes the same conclusion or no conclusion.**

» But how to simplify above two guidelines into single query in a defeasible reasoner?

Consistency Checking in OSCAR

» My proposal to the consistency checking problem, assuming:

- **Two legal norms (say, L1 and L2) are already correctly transformed in LegalRuleML;**
- **The same vocabulary (Ontology) are used;**
- **It's feasible for a XLST-based solution to transform LegalRuleML into any native logic representations;**
- **The defeasible reasoner is FOL, that is, universal and existential quantifiers are available.**
- **Given infinite time and enough information, the automatic reasoning can find the answer to any question.**

» Steps for concsistency checking:

1. Encode all rules with additional scope information:

e.g. A reason "P -> Q" in L1 should be encoded into "P & (under L1) -> Q" or "(under L1) & P -> Q".

2. Send all rules to the reasoner and ask if there exists at least one legal proposition *p* and one situation *C*, such that C entails p under L1 while C entails ~p under L2:

```
(some p)(some C)
   ((C & (under L1) ||-> p) &
    (C & (under L2) ||-> ~p))
```

» Observations:

- **If above "ultimate epistemic interests" is not answered, then we have reasons to believe that L1 and L2 are consistent.**
- **If it's answered, a reasoner like OSCAR will actually give the found value of *p* and *C*, together with the "proof steps", and this will lead us to fix L1 or L2 for the found inconsistency.**
- **In theory we can repeat this process until all inconsistencites were identified and fixed.**

Consistencies between GDPR & BDSG-E (1)

(Based on Internet Blog[1]: «Data Protection: Does the German Implementation Act (BDSG-E) undermine the GDPR?»)

» Case 1: Administrative fines and Penalties

"Art. 83 GDPR provides for the opportunity to impose fines up to 20 million Euros or up to 4 % of the total worldwide annual turnover of the preceding financial year. While the fines primarily apply to companies, the member states can determine 'different penalties'. The German legislator makes use of this opportunity by creating rules also allowing for the sanctioning of individuals, leading to a risk of liability for managers, employees and in-house data protection officers. Section 42 BDSG-E even foresees a punishment of up to three years imprisonment. Thus, an effective and well organised compliance-structure is more crucial than ever."

1 http://privacylawblog.fieldfisher.com/2017/data-protection-does-the-german-implementation-act-bdsg-e-undermine-the-gdpr/

» There seems no inconsistencies:

1. If a Germany-located data controller has violated its obligations on personal data protecting, according to GDPR as a company it should be punished with administrative fines (but GDPR doesn't forbidden any imprisonment);

2. According to BSDG-E, administrative fines apply first, then the person who is actually responsible for the infringements may be further punished for up to two or three) years of imprisonment.

» We've made partial LRML translations for and as part of the sample code of Structural Application in FrameMaker.

Consistencies between GDPR & BDSG-E (2)

» Case 2: Change of purpose of processing

"Section 22 BDSG-E states a number of exceptions for the lawful processing of special categories of personal data according to Art. 9 GDPR, e.g. in the area of health and social services. Those exceptions are safeguarded by a number of requirements regarding appropriate and specific measures on a technical and organizational level.

Interestingly, the high level of protection for special categories of personal data experiences another dilution: In case of scientific or historic research, even special categories of personal data can be processed without consent of the data subject, provided it is necessary for the relevant purpose and no overriding interests of the data subject are given. Although the controller has to provide appropriate measures to safeguard the data subjects interests (without further specification), this seems to be a quite generous exemption that allows for a flexible interpretation."

» Again, there seems no explicit consistency issues

- **Art. 9 GDPR is extremely flexible for Member States**
- **We've made sample translations of Art. 9 GDPR in LRML.**
- **Section 22 BDSG-E has been translated in English, no further actions in translating to LRML.**

Consistencies between GDPR & BDSG-E (4)

» Case 4: Data Protection Officer

"With regard to the obligation to appoint a data protection officer, the draft law keeps the current provisions of the German Data Protection Act and obliges every company with at least ten persons employed with the automatic processing of personal data to appoint a data protection officer. The GDPR only obliges companies to do so in exceptional cases. The controller or the processor will also have to appoint a data protection officer if they deal with processing subject to a data protection impact assessment according to Article 35 GDPR. The same applies if personal data is processed for the purpose of commercial transfer of data or for marketing and market research purposes."

» This is so far the only inconsistency issue we found: a company with at least ten persons, under 'non-exceptional' cases (C):

- **According to GDPR, it's (conclusively) not obligate to appoint a data protection officer (p):**

```
C & (under GDPR) -> ~p
```

- **According to BDSG-E, it's (conclusively) obligate to appoint a data protection officer:**

```
C & (under BDSG-E) -> p
```

» It remains to actually confirm it in OSCAR, once the related legislation articles/sections were translated into LRML.

Semi-automatic tranlations using NLP

Conclusions

Bibliography