Principles of exchanging legal knowledge

Tom van Engers

Introduction….

- How do we get support from society for our legal system?
- How do we promote transparency (legal knowledge)?
- How do we promote compliance?
- How do we arrange an efficient and effective law enforcement?
- How do we prevent people from going to court?
- How do we get an efficient and effective court system?

The Leibniz Center for Law

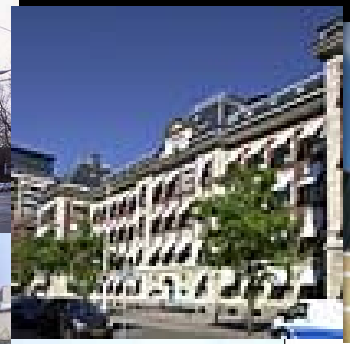
- Operational legal issues
- Support systems
- (Open) standards
- Both fundamental and applied research
- Fundamental research
- Projects!

Legal reasoning, common sense and abstraction

- Input: raw case
- Process: Abstract to legal categories
  - Essentials: common sense knowledge, knowledge of relevant legal concepts and rules for abstraction
- Process: Norm assessment:
  - Essentials: ordered system of norms
- Process: Attribution to a acting person
  - causality, liability concept etc.

eGovernment context





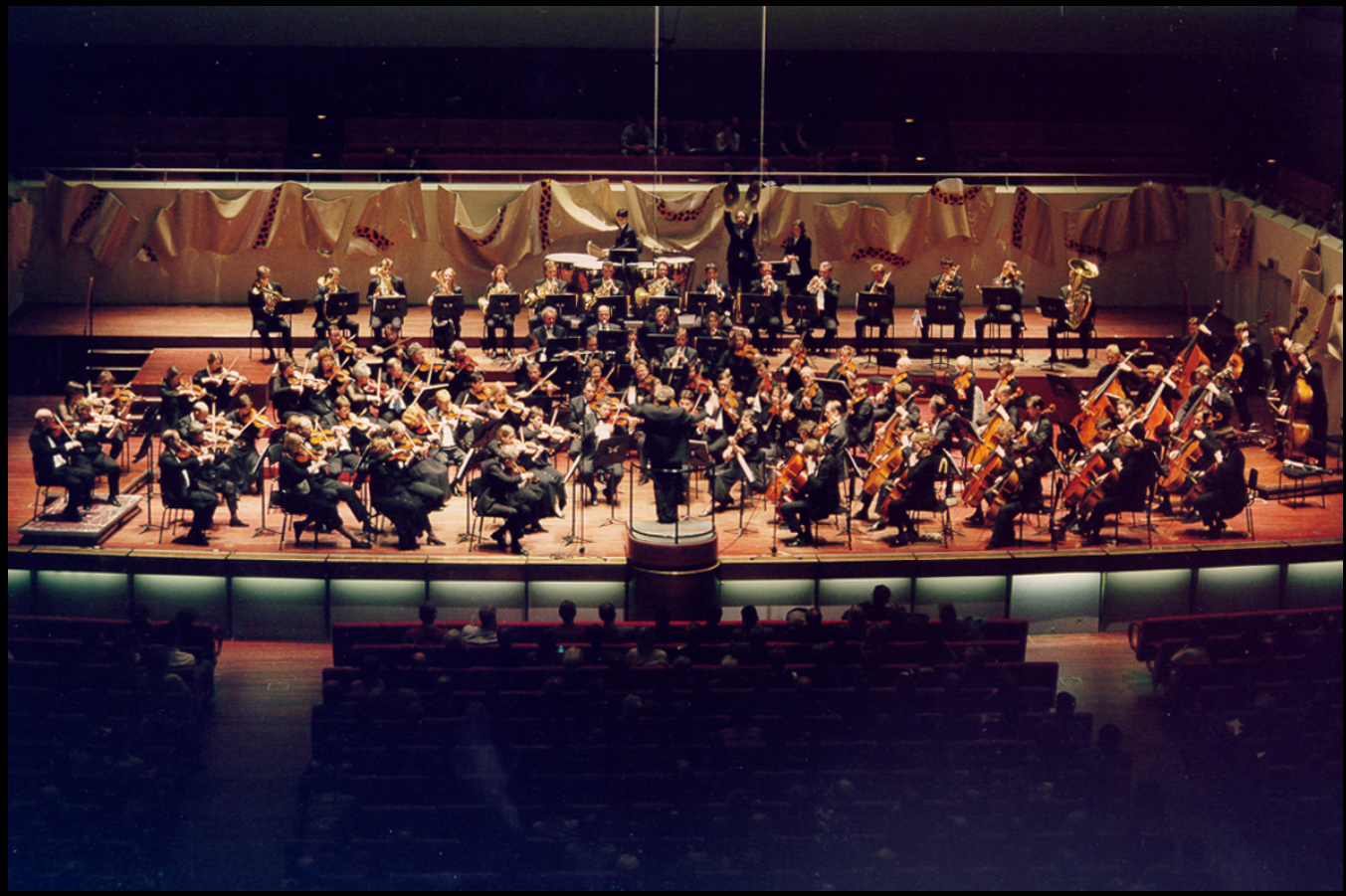



- Legal Pluralism
- Limited process integration
- Conflicting rules
- Poor adaptivity
- High operational costs
- Little trust in the government & non compliant behaviour

Some services have been stable over hundreds of years…

Leibniz Center for Law

Productivity increased over time but not all services became cheaper…



Solution?

The legal context

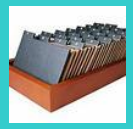
- Due to increased complexity in society we have more and more complex rules
- Regulation is an important steering instrument
- It is very unlikely that the regulatory complexity will diminish
- Therefore we should find a way to cope with this complexity

The world of rule based decision making






Land register



Tax Administration

Wage as known for Income tax

...

Bank

Wage according to the bank

Citizens Admin.

Citizen

Date of birth

Data access authentic sources

Data transformation functions (e.g. date of birth -> age)

Data provision Services

Data Unification Model

WFM

PM

Appl. Data.

Data Mod.

Know. Comp.

KM

Service specification layer: pre- & post conditions Input/output signature etc.

Legal sources

Expert

Operational policy

CMngt

Knowledge sources

The world (context)

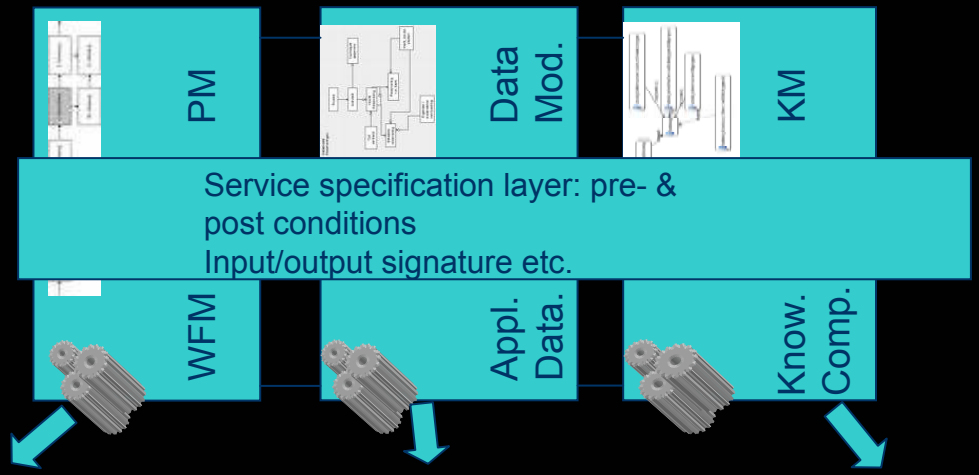
Administrative Representation (eg. in public administrations)

Data Service Layer

Application Service Layer

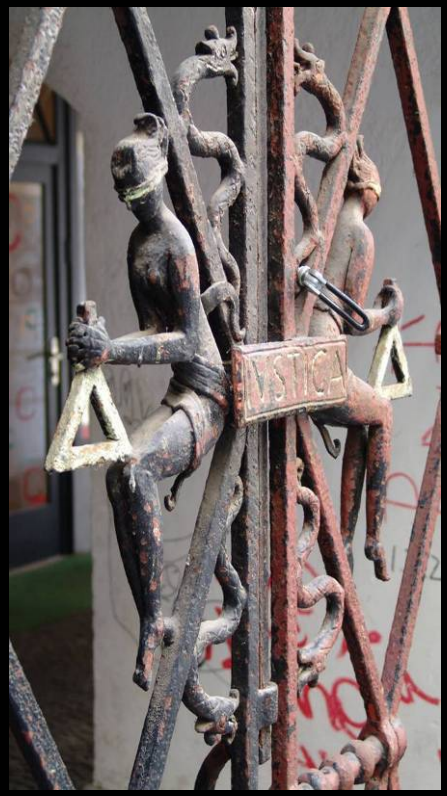
Model Layer

Representation and model types



Actor model	Top & Core ontology	Reasoning layer
Control model	Domain ontology	Epistemic model
Meta comp. model	Domain model	

Dimensions in Legal Knowledge & Reasoning



- True and false
  - (a) Tom crossed the street with a red traffic-light
- Allowed and disallowed
  - (b) Everybody is prohibited to cross with a red traffic light
  - Hence Tom committed a violation, by (a) and (b)
- Known and unknown
  - Public authorities **do not know that** about Tom's violation
  - Hence Tom is not punished, but he should be
- Contigent and necessary
  - Tom was pushed when the traffic-light was red
  - Hence he could not have avoived to cross the street
  - Hence he should not be punished after all.

Why Many Dimensions (1) -- Deontic operators

- Existing languages adopted to model leglislation cannot express the distinction between true/false and allowed/disallowed:

  they do not contain deontic operators.

- Deontic operators are needed if we want to reason about situations where violations occur, such as:

  'P' is the case but 'not-P' should be the case ('not-P' is obligatory)

- Without deontic operators, violations can only be represented as contraditions:

  'P' is the case and 'not-P' is the case

  NB: From a contradition everything follows, so no correct reasoning can be performed anymore.

Why Many Dimensions (2) -- Alethic operators

- Suppose Tom did commit a violation

  (represented as: 'P' is the case but 'not-P' is obligatory)

- Suppose it was necessary for Tom to do 'P' (e.g., because he was forced to do 'P' by a situation he happened to be in).

- We need the operator 'necessary' to express:

  'P' is the case, but 'not-P' is obligatory, where 'P' is necessary.

- We can conclude that Tom is not responsible.

KR in Practice

- Industry
  - Various vendor formats
- Computer Science
  - Various formal languages

DLP = Intersection of OWL-DL and Horn Fragment

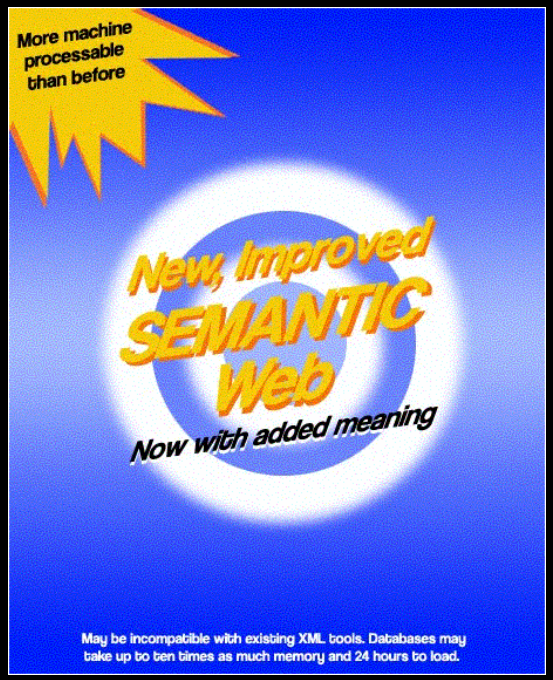
- Function-free Horn Fragment
  - Decidable Subset of first order logic
  - Composed of rules only:
    - Consequent ← Antecedent
    - Only *one positive* literal, e.g. R(x, y), can occur in the consequent
    - Universal quantifier scopes over the implication
- Intersection of OWL-DL and Horn Fragment
  - Some OWL-DL axioms cannot be expressed as rules:
    - E.g., $B(x) \leftarrow (A(y) \leftarrow R(x,y))$
    - As in if you're married with a woman you must be a man
  - Some rules cannot be expressed in OWL-DL:
    - E.g., $B(z) \leftarrow A(x) \wedge R(x, y, z)$

Safe Subset of Horn fragment

- Knowledge Base:
  - T Box and A Box = OWL-DL formulas
  - Program = Rules (i.e. function free Horn fragment)

- Safety Condition on Rules:
  - If a variable occurs in the consequent, then:
    - the same variable should occur in a non OWL-DL atom;
    - this atom should occur in the antecedent of the rule.

LKIF: Context



1. The Semantic Web: distributed models of the law
2. Reusability of models of law outside of original context of use
3. A standard for vendors of legal expert systems (Ruleburst, KnowledgeTools, RuleWise, Beinformed, etc.)
4. A guide for proper modeling of law and legal reasoning

The **L** in LKIF

- Special attention for legal theoretic views on epistemology and argumentation
- Standard legal vocabulary in LKIF ontology

Design Challenges

- The law is always about something else
  - Reuse existing KR standards and ontologies wherever possible
  - Challenge: defining for instance deontics without commitment to a specific theory of action, planning, etc.
- Composition of knowledge items from different sources
  - Especially difficulty to handle defeasibility, which is often used to model for instance burden of proof in adversarial proceedings

Design Challenges

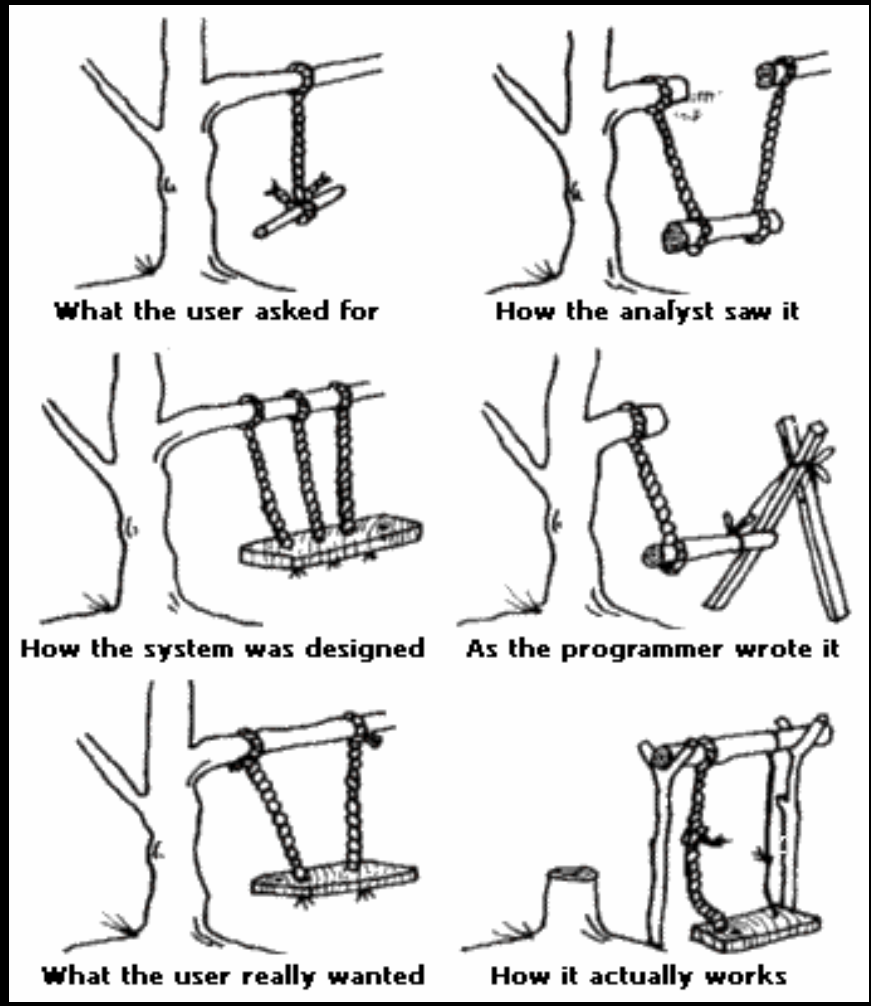
- No sense of control over when the law changes
  - Affects horizon for ROI negatively, and
  - Mostly because of this limits the market for legal expert systems
- Need for review and update of systems is caused by fairly frequent outside events:
  - a change in the law, or
  - its interpretation by courts
- Efficient maintenance is critical

Leibniz Center for Law

Design Challenges

- Concurrent version control: rules to be applied in decision making depend on
  1. when a decision is/was made (Stichtag),
  2. when the relevant events to be judged happen(ed), and
  3. when we look (Sichttag) at the decision.
- Examples:
  - Retroactive application of law refers to the events, not to decision making
  - decision making based on future law (i.e. Stichtag in the future) is often simulated, while the law can still be changed before its enactment.

Requirements of LKIF



Technical LKIF requirements

- R1:

  The language should be based on a logic and have a defined formal semantics, so that it can support correct automatic reasoning and the consistency of models can be checked.

- R2:

  If a language (equipped with a formal semantics) is meant to support mechanized reasoning, its expressiveness should be limited in order to preserve the tractability of the language.

Technical LKIF requirements

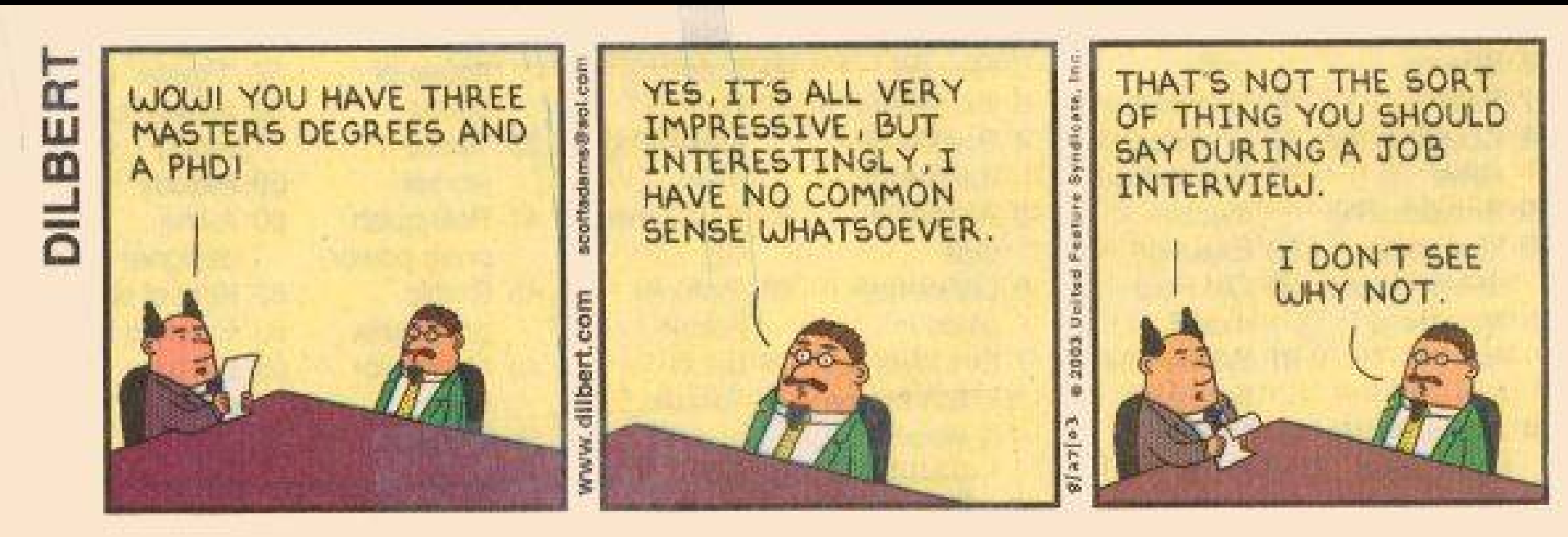
- R3: the language should contain a layer that can model common-sense and technical definitions of terms used in the legislation. The definitional layer (ontology) should be separated from the layer expressing legal knowledge strictu sensu and it should be reusable.

Ontology and order in the legal system

- Ontology
  - (formal) specification of a conceptualisation
    - Conceptualisation: a systematic description of what exists
  - Subsumption between concepts
    - Vehicle -> motor vehicle etc
  - Necessary and sufficient requirements

    Dutch examples:
    - Is “loon” in Wet Loonbelasting zelfde als “loon” in CSV?
    - Bestuursorgaan = public body? Diefstal = theft?

Common Sense Knowledge



Functional LKIF requirements

- R4:

  the language should contain a set of deontic operators (associated with a precise semantics), provided that the language will be used to perform deontic reasoning, and/or it should contain a device to keep track of deontic expressions occurring in the legislation.

Specific deontic operators e.g. for exceptions

- Pieces of legislation often contain syntactic constructions like the following, where unless is an exception-marker:

  Citizens complying with requirement X shall be entitled to Y, unless Z holds.

- An explicit syntactic treatment of exception can be avoided by rephrasing the above conditional norm as follows:

  Citizens complying with requirement X and not with Z shall be entitled to Y.

- But this expression has a different semantics!

Functional LKIF requirements

- R5: the language should contain an operator to express exceptions, e.g., unless, and/or a device to keep track of which norms are exception to which other norms.

Exceptions and Norm Conflicts

- Conflicting Norms.

  Exceptions can be seen as resolved conflicts between norms:

  - One norms says that X is the case and the other says that the negation of X is the case, where the legislation states that the latter is an exception to the former.

Priorities in the legal system

- Lex superior,

  i.e., pieces of legislation promulgated by higher legislative sources take priority (e.g., laws whose source is the parlament takes priory over a Royal decree, because the parlament is deemed superior to the Queen);
- Lex posterior,

  i.e., newer pieces of legislation take priority over older ones;
- Lex specialis,

  i.e., more specific pieces of legislation take priority over more general ones.

Functional LKIF requirements

- R6: the language should contain a priority relation symbol between norms, and/or it should contain a device to formulate the principles of lex superior and the like.

Cross-referencing Between Norms

- When we declare that a norm takes priority over another one, we implicitly introduce a twist in the syntactic status of norms.
- On the one hand, norms are sentences:

  e.g., if X complies with Y, X shall be Z,

- But when they are compared, norms become (complex) expressions:

  e.g., the norm that if X complies with Y, X shall be Z.

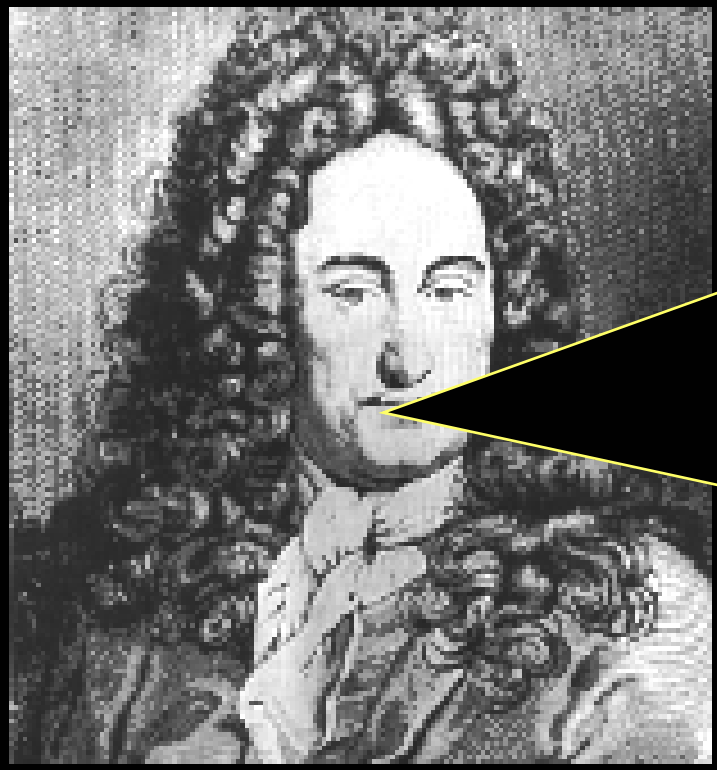
In other words:

- on the one hand, norms are statements,
- and on the other hand they are concepts or objects.
- This twist shows up in legal texts by means of the heavy cross-referencing between norms.

Functional LKIF requirements

- R7: the language should contain a device to look at the same norm both as a normative statement (sentence) and a normative concept (expression), e.g., by assigning a number to each norm, and/or it should have a device to keep track of cross referencing relations.

Leibniz



"Once the characteristic numbers of most notions are determined, the human race will have a new kind of tool, a tool that will increase the power of the mind much more than optical lenses helped our eyes, a tool that will be as far superior to microscopes or telescopes as reason is to vision."
Leibniz, *Philosophical Essays*

Time and Space

- Often obligations, rights and alike legal notions are based on time and space. For example:
  - one is entitled to complain for a ticket issued by a policeman, provided she notifies her complaint to the competent authorities within 30 days.
- But references to time can be crucial for matters concerning the application and validity of norms:
  - A norm may be valid when it is promulgated, but it may also have a retrospective validity.
- Examples of norms depending on space are contained in traffic regulations, e.g.,
  - depending on the area a vehicle is on, the speed limit varies.
- Other examples can be found in urban planning and zoning,
  - e.g., houses shall be built in area X, but not in area Z.

Space & time

- Similarly to the case of deontic notions, one may claim that a special treatment of time and space is unnecessary.
- Once again, the issue is the one of performing reasoning about time and space. For example:
  - if a complaint can be forwarded within 30 days, it follows that it can also be forwarded within 2 days;
  - or if hunting is prohibited in area X, then it will be prohibited also in a smaller area included in area X.
- These apparently trivial inferences cannot be made by a machine, unless a precise semantics for space and time is defined.

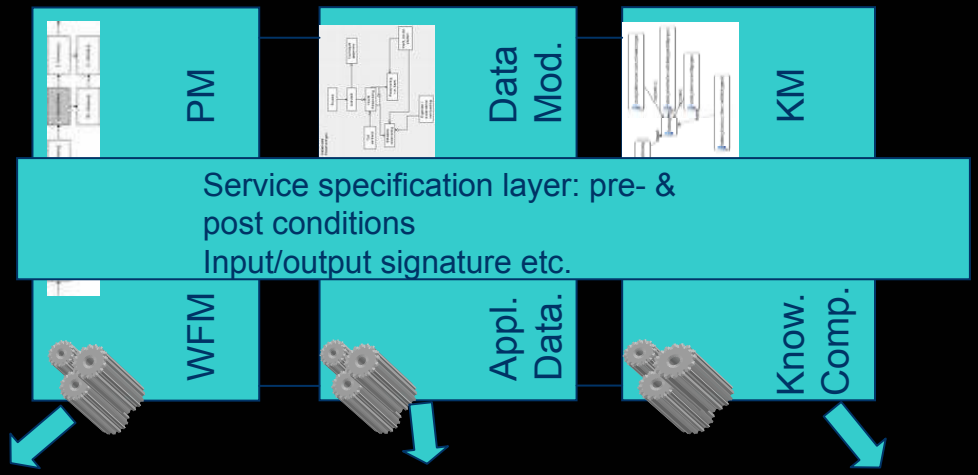
Functional LKIF requirements

- R8: the language should express time and space (associated with a precise semantics), provided that the language will be used to perform reasoning about time and space, and/or it should contain a device to keep track of when and where norms are valid.

LKIF Design Principles

- All LKIF is OWL = RDF = XML
- Layers:
  1. LKIF Ontology of basic legal concepts
     - Terminological, therefore monotonicity of entailment
     - Standard OWL DL semantics
  2. LKIF Rules
     - Special operators: unless, assuming
     - Presumptive argument
     - Defeasible “argumentation-theoretic” semantics
  3. LKIF Arguments
  4. LKIF Cases

Representation and model types

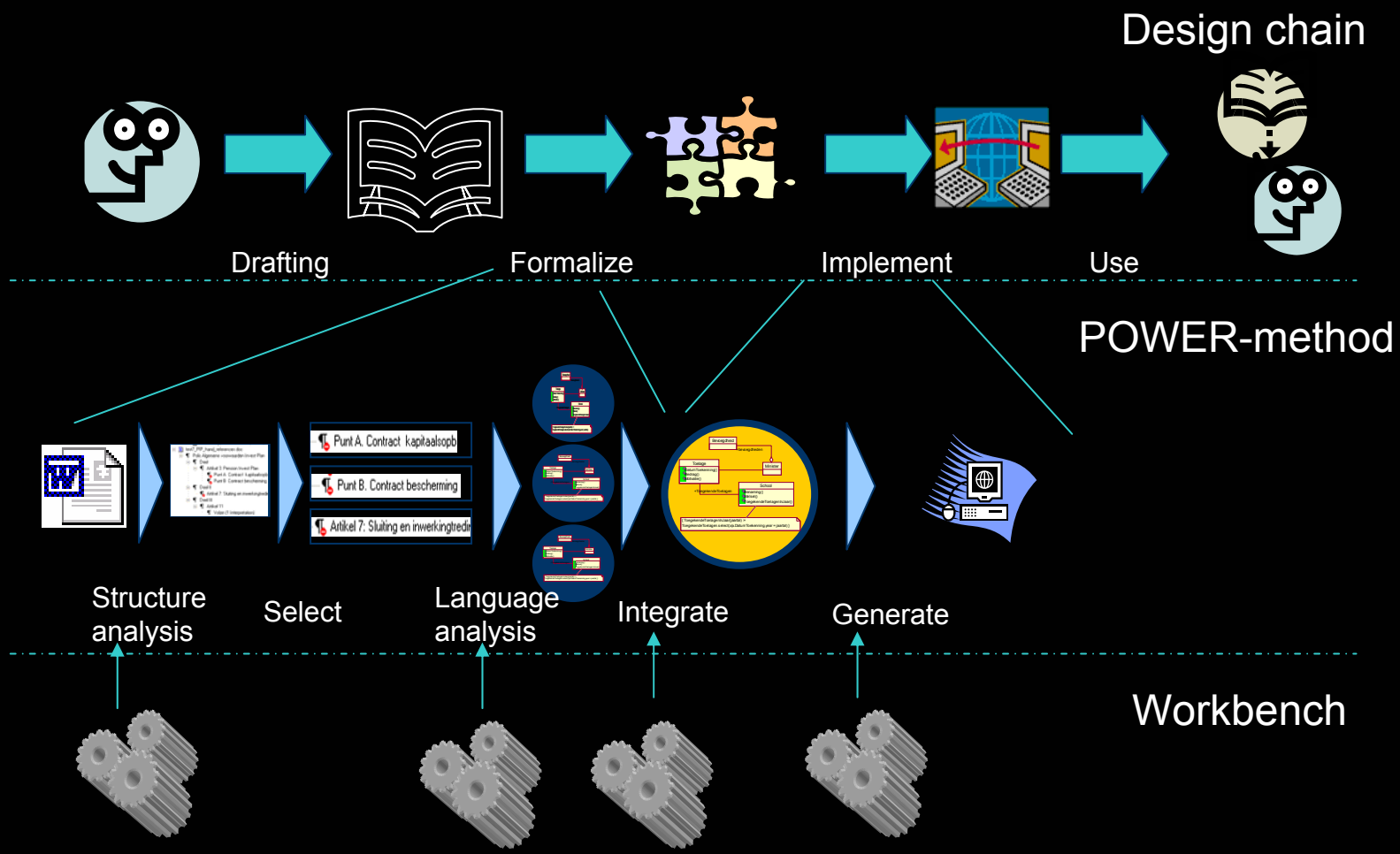


Actor model	Top & Core ontology	Reasoning layer
Control model	Domain ontology	Epistemic model
Meta comp. model	Domain model	

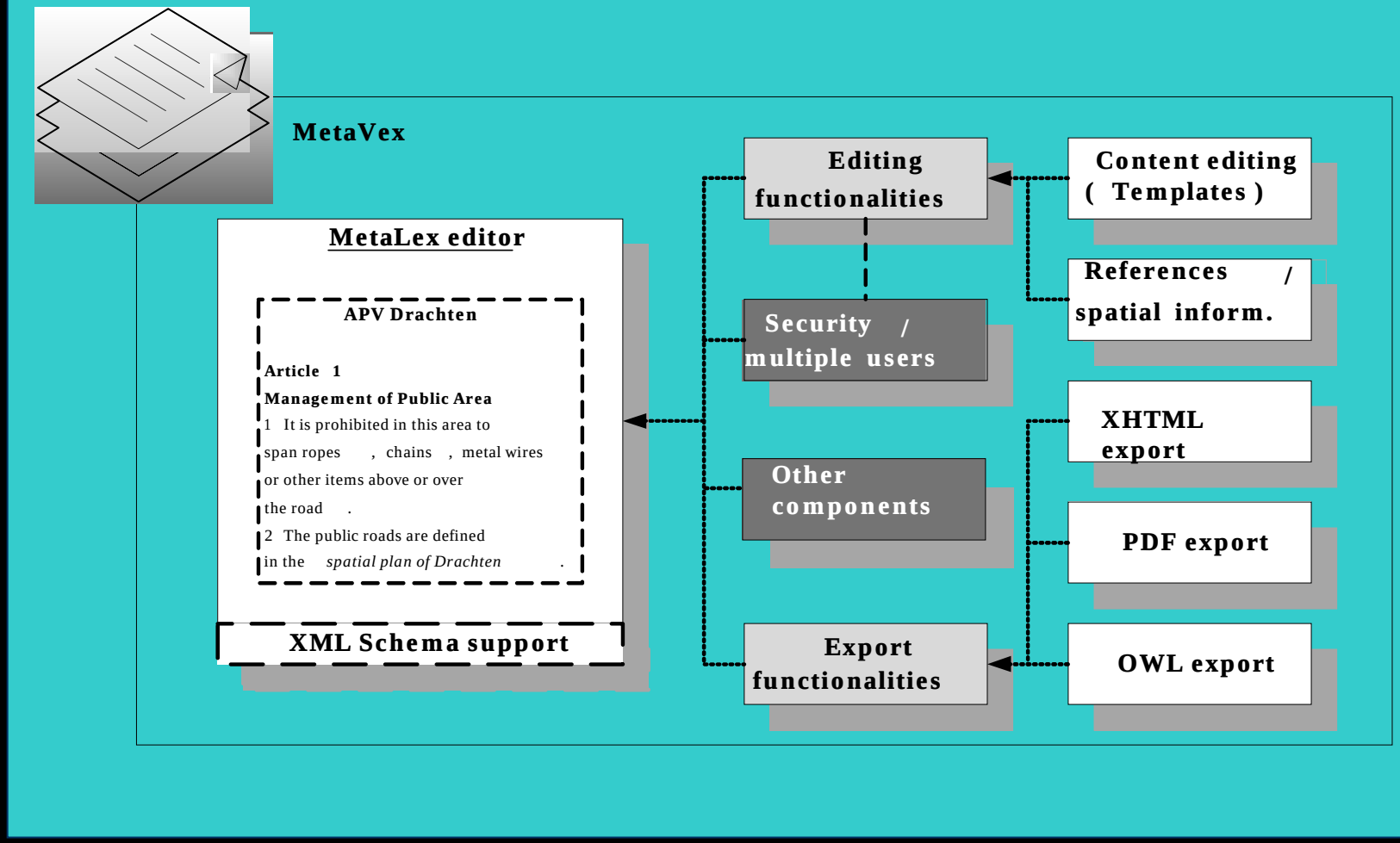
The past and the future

- (E-)POWER
  - Aimed at formalizing legislation in order to support the drafters and the implementers
  - Resulted in the first Metalex, and XML standard for legal sources (now a CEN standard)
- Seal
  - Aimed at supporting legislation drafters and policy makers (parliamentarians, drafters etc.) and facilitate cooperation
- Estrella
  - Aimed at creating LKIF
- …

The POWER approach



Seal: MetaVex



LKIF

- Work in progress
- We will test LKIF in three different settings
- LKIF is created in the ESTRELLA project financed under the 7th framework programme

The near future…

- A tool that provides insight in the legal sources
- The processes, concepts, actors/roles terms/concepts…
- Services, forms…
- Versioning
- ….

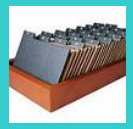
- Dashboard for decision makers
- Design Tool

The world of rule based decision making






Land register



Tax Administration

Wage as known for Income tax

...

Bank

Wage according to the bank

Citizens Admin.

Citizen

Date of birth

Data access authentic sources

Data transformation functions (e.g. date of birth -> age)

Data provision Services

Data Unification Model

WFM

PM

Appl. Data.

Data Mod.

Know. Comp.

KM

Service specification layer: pre- & post conditions Input/output signature etc.

Legal sources

Expert

Operational policy

CMngt

Knowledge sources

The world (context)

Administrative Representation (eg. in public administrations)

Data Service Layer

Application Service Layer

Model Layer

Questions?

vanEngers@uva.nl

www.LeibnizCenter.org