FROM LEGAL SYMBOLIZATION VIA LEGAL FORMALIZATION TOWARDS HUMAN DIGITALITIES

Vytautas Čyras, Friedrich Lachmayer

Table of Contents
A. Textuality in Law................................................................................................................................... 1
B. The Hierarchy of Valid Legal Sentences in Yoshino’s Logical Jurisprudence .......... 3
   I. Legal Sentences............................................................................................................................. 5
   II. Three Primitives: Legal Sentence, Validity and Inference Rule............................. 7
      1. Connections of Legal Sentences .................................................................................. 8
      2. Legal Inference ................................................................................................................. 10
      3. Fundamental Legal Meta-Rule Sentence (FLMRS) ................................................. 11
      4. Basic Legal Meta-Rule Sentence (BLMRS) .......................................................... 12
   III. Comparison of Yoshino’s LJ with Kelsen’s PTL .......................................................... 12
C. One-Arch Bridge Seems Actually not Realistic ................................................................. 13
D. Multi-Arch Bridge Implies Multiphase Transformation.................................................. 16
E. Digital Humanities and Human Digitalities ................................................................. 20
F. Evolution ................................................................................................................................................ 22

A. TEXTUALITY IN LAW

Traditionally, law is textual. Jurists transform texts into texts. There are various kinds of texts: laws, contracts, claims, proceedings, transcripts, judgements, etc. Text transformations require abstracting, extracting, reasoning, deduction and other legal methods. Legal texts contain legal terms and legal sentences (Rechtssätze) that we
may call Kelsen’s legal sentences. Judgements, guidelines and their head notes are formulated in abstract legal terms. Therefore, abstracting and extracting are needed and are performed by jurists and secretaries.

Legal terms that were detached, for example, by Georg Friedrich Puchta in the 18th century, were textual. Thus, the phenomenon was called Begriffsjurisprudenz – the jurisprudence of legal terms. Legal terms are also central in Roman law. Hence, the initial position is that traditionally legal texts are transformed with the aid of the richly formed area of legal terms.

**Multiphase Transformation.** The thesis of this article is that symbolization and visualization precede formalization. We hold that in the transformation “legal text → formalization” the path connecting the legal text to computer implementation requires intermediate steps. In this way, a multibridge is built between legal concepts and their representations in computers; see further Figure 9. Hence, an approach in legal informatics is proposed which is called Multiphase Transformation.

**Digital Humanities and Human Digitalities.** On the right bank of the multibridge, there appears the sphere which is called digital humanities; see further Figure 16. Specifically, Colette Brunschwig brings the digital humanities and digital law closer together, because “both explore visualization, audiovisualization, and multisensoriza-

---


The research subject of digital humanities is the human which is surrounded by digitally-based phenomena. We introduce another sphere – *human digitalities* – a next one on the right bank. Here the research subject is different – the machine – which is structured according to human-centred rules.

**Relation with Yoshino’s Logical Jurisprudence.** The theme under investigation is related with formalizations by Hajime Yoshino. His Logical Jurisprudence (LJ) deals with propositions that are called *legal sentences* (LS). His notation Compound Predicate Formulae is used to represent LSs. Yoshino provides different kinds of legal sentences, such as legal rule sentences (LRS), legal fact sentences (LFS), legal meta-sentences (LMS), etc. These kinds of sentences serve to model hierarchical structure of legal rules. Yoshino also demonstrates the interconnectedness of rules at multiple layers; see the Anzai-Bernard example. A separate study is worth the comparison of Yoshino’s Logical Jurisprudence and Kelsen’s Pure Theory of Law.

**B. THE HIERARCHY OF VALID LEGAL SENTENCES IN YOSHINO’S LOGICAL JURISPRUDENCE**

Yoshino’s Logical Jurisprudence deals with the logical structure of legal science. Legal sentences form a key element in his research. Yoshino aims to make the taxonomy of legal sentences explicit. Therefore, he brings in several layers: LRS, LFS, LOS, etc.

---


7 YOSHINO, 2011, Note 5, p. 121.

Kelsen, with his terminology, stops at texts, whereas Yoshino reaches the logical layer and uses a formal notation.

Guido Tsuno investigates the legal lexicon and steps outside of the streams above. Kelsen mentioned the concept of the modally indifferent substrate in the General Theory of Norms, which appeared rather late. Thus, the terms do not form a specific issue for Kelsen. In contrast, Tsuno investigates legal terms, namely, in their historical dimension. He starts with medieval historical dissertations and law dictionaries. Legal terms allow the testing of these works. They also appear in titles and headlines, and hence, in immanent metadata.

---

Yoshino notes that there have been various theories of the systematization of law. For instance, when speaking about natural law theory, Pufendorf tried to construct a deductive system of law. Pufendorf classifies entities into entia physica and entia moralia and speaks about impositio. Thus, the Is/Ought terminology is used. Kelsen’s basic norm appears on the meta-level (Figure 1). The constitution, statutes and individual decisions form a hierarchy. Further, the elements of Yoshino’s LJ are explained. Logical Jurisprudence tries to explain the whole legal system.

I. Legal Sentences

Logical Jurisprudence starts from legal sentences, not from norms. Legal sentences are composed of three alternative types:

- Legal rule sentences (LRS) and legal fact sentences (LFS)
- Legal elementary sentences (LES) and legal complex sentences (LCS)
- Legal object sentences (LOS) and legal meta-sentences (LMS)

We start by observing the following kinds of entities: LRS, LFS and LOS (Figure 2). First, legal rule sentences are in the world of Logical Jurisprudence, but describe legal rules that are in Ought. Secondly, legal fact sentences describe legal facts that are in Is. Thirdly, legal object sentences also describe legal objects that are in Is. Here, obligations deserve special attention.

Legal sentence is a syntactic entity and serves as a description. Validity, i.e., a truth value, is concerned with legal meta-sentences. LRS and LFS correspond to rule and fact in terms of logic programming. An example of an LRS structure is: \( a(X) \leftarrow b(X), c(X,Y) \). Read: For all \( X \), \( X \) becomes effective if \( X \) is an offer and \( X \) reaches the offeree \( Y \).

Legal rule sentences have the syntactic structure of rule as a hypothetical proposition:

\[ \text{CISG}^{12} \text{ Article 15(1): An offer becomes effective when it reaches the offeree.} \]

---


11 YOSHINO, 2011, Note 5, p. 121.

Figure 2. The three primary kinds of legal sentences in Logical Jurisprudence

This is represented by

\[ \text{become_effective(offer(X,A), T) :- reach(offer(X,A), offeree(B,X), T)} \]

Legal fact sentences have, for example, the following structure: \( b(x_1), c(x_1,y_1) \). Read, for example, \( x_1 \) is \( A \)'s offer and it reaches \( B \) on 5 April. Legal fact sentences have the syntactic structure of fact as a categorical proposition, e.g., \( A \)'s offer reaches offeree \( B \) on 5 April. This is represented by \( \text{reach(offer(o1,anzai), offeree(bernard,o1), April-05)} \).

An LOS describes the obligations of a person, e.g., \textit{It is obligatory for } A \textit{ to deliver the goods to } B.\textit{ }

A \textit{legal elementary sentence} is the smallest unit in legal sentences. An example is CISG Article 15(1) cited above. Another example is, “One must drive a car at less than 100 km/hour on a highway.” LESs play the role of atoms.
II. Three Primitives: Legal Sentence, Validity and Inference Rule

Logical Jurisprudence works on the vertical stage of science; see Figure 1, right hand side. It aims, with a minimal number of elements, to explain the whole legal system, which is depicted on the left hand side of Figure 1. LJ starts with three primitives:

1. **Legal sentence.** Logical Jurisprudence considers that norm as a meaning in Ought does not exist. Thus, LJ starts from sentences.

2. **Validity** of legal sentences.

3. **Inference rule.** The modus ponens rule is used: \( P, P \rightarrow Q \mid - Q \). This is used for deduction from valid legal sentences (in the sense of LJ).

The validity concept in Logical Jurisprudence is treated as scientific validity, i.e., a truth value in the world of science, \( \text{is\_valid(sentence1, goal1, time1)} \); see Figure 3. It is not the same as Kelsen’s legal validity concept that refers to Ought. Yoshino speaks about the validity of legal sentences as truth. The legal domain can allow such a formalization to a certain degree, namely, in formalising the deduction in law. The reason is that statements in the legal domain can be defeated. Arguments can also be assigned different weights. Hence, both the concept of validity (Kelsen’s Ought) and the concept of truth are inherent to law.

A legal meta-sentence describes the validity of a legal sentence, for example:

‘It is obligatory for A to deliver the goods to B’ is valid on 2010-05-01.

Note that this is scientific validity – in the world of science (i.e., LJ), and not validity in Ought, which is Kelsen’s validity concept (Figure 3).

---

13 Yoshino writes: “The concept of validity is to be conceived of as a truth concept. That a legal sentence is valid means that it is true in the legal discourse of the world. If a legal sentence which describes a legal state of affairs is valid, it means that the legal state of affairs exists in the legal world.”; see YOSHINO, 2011, Note 5, p. 122.
Formalising the concept of validity is central in Yoshino’s research. The validity of a legal sentence (i.e., whether it is true or false) is assigned through a legal meta-sentence. This is a natural way to model the world of discourse. The representation of a model of a world in formal logic includes a set of assertions which are held to be true. Then formal inference rules step in to model legal reasoning. Yoshino introduces also the terms legal meta-rule sentence (LMRS) and legal meta-fact sentence (LMFS). These describe the validity of an LMS or an LFS, respectively.

1. **Connections of Legal Sentences**

There are four kinds of connections of legal sentences:

1. ‘and’ (\&) – conjunction

2. Connection into a legal complex sentence (LCS)

3. Connection between LOS and LMS

4. Connection between LMS and LMS

Each connection is described below.

**Connector and (\&).** It connects two legal sentences, e.g., $LS_1 \& LS_2$. A structure of legal sentences can be built in this way. The structure is treated as a group of legal sentences and can be assigned a unique name.

**Connection into a Legal Complex Sentence (LCS).** An LCS is composed of legal sentences, namely, a group of LSs; but the way they are connected is not specified. Hence, an LCS is treated as an aggregate. An LCS has a unique name, such as, “A–B Contract”, “The UN Convention”, a part of statute, etc. (Figure 4a).
An important feature of LCSs is that if an LCS is valid, every LS is valid. This formalizes the meaning of abstract concepts like contracts, judgements, administrative orders, statutes, conventions, etc. We note that a legal complex sentence is treated as an aggregate (or a formula with the connector &), e.g., \( p_1 \& p_2 \& p_3 \).

**Connection between Legal Object Sentence (LOS) and Legal Meta-Sentence (LMS).** An LOS describes an object. An example of an object type is an obligation. Examples of LOSs are, “It is obligatory for A to deliver machinery to B,” or “B must pay A the price of $58,000.” The modus of a legal object, for example, obligation or right, is expressed by a distinct predicate: \( s_1: \text{is\_obligatory}(A, \text{deliver}(A,B,\text{machinery})) \). The time for the obligation to become valid and to terminate is also represented with predicates, e.g., \( s_2: \text{become\_valid}(s_1, 2010-04-09) \), \( s_3: \text{is\_terminated}(s_1, 2010-05-01) \).

Yoshino treats the accrual of the validity of a legal object sentence by exercising the right as a fundamental meta-rule sentence (FLMRS); see next sub-section.

\[ r_{3aa2}: \text{If } A \text{ has a right to require } B \text{ to do } Z \text{ at time } T \text{ and } A \text{ requires } B \text{ to do } Z \text{ at } T, \]

\[ \text{then a legal sentence “It is obligatory for } B \text{ to do } Z \text{” becomes valid at } T. \]

The above rule asserts that a duty which is represented by a legal object sentence follows from a rule which represents the right. Yoshino analyses the concept of right in terms of legal meta-sentences. The “dynamic systematization of law”\(^\text{14}\) is an aim.

---

**Connection between LMS and LMS.** A legal meta-sentence describes the validity of legal sentences. Some LMSs describe the validity of other LMSs. A connection is LMS–LMS (Figure 4b). The following are two examples of legal meta-rule sentences:

*CISG Article 1(1):* This Convention applies to contracts of sale of goods between parties whose places of business are in different States...

*CISG Article 23:* A contract is concluded at the moment when an acceptance of an offer becomes effective...

Positive legal meta-rule sentences, LMRSs, assist a fundamental meta-rule sentence (FLMRS; see further) as its sub-rule sentences to decide on the fulfilment of each requirement of the FLMRS. Hence, a connection between LMRS and LMRS is produced. If the validity of a positive LMRS is regulated by other positive LMRSs, the latter belong to a *higher meta-level* than the former.

**2. Legal Inference**

In Logical Jurisprudence, the *modus ponens* rule stands for the main inference rule. LRSs are deduced from the existence of legal rules in Ought; see arrow (1) in Figure 5. The LRSs obtained, and legal fact sentences, LFSs, are used to deduce legal object sentences, LOSs; see arrow (2). The LOSs obtained point to legal objects on the Is stage; see arrow (3).

Legal sentences are developed through the process of legal reasoning. Yoshino concentrates on two types of legal reasoning:

1. *Reasoning of legal justification.* This is based on deduction through *modus ponens*.

2. *Reasoning of legal creation* (or discovery). This is further divided into two parts:
   - (a) *abduction*, where the inference rule *modus tollens* is applied, and (b) *induction*.

The legal sentence of a judgement may not be deduced from statutes and facts alone, but from the whole body of legal sentences, including legal principles, cases, theories and implicit legal common sense. Therefore, Yoshino sees two ways of legal reasoning:

---

“propose a system of analysis that recognizes the inherent hierarchy between a right on the meta-level and a duty on the object level language.” ibid., p. 305.
1. **Concretization.** Statutory terms are made tangible by creating LRSs which describe inclusion relations.

2. **Systematization.** The sentences of legal principles are created and these enable us to bring mere collections of LSs into a unified, coherent deductive system.

3. **Fundamental Legal Meta-Rule Sentence (FLMRS)**

   A fundamental legal meta-rule sentence (FLMRS) is implicitly taken for granted for all regulations. The following is an example:

   \[ \text{ro: A legal sentence is valid at time } T, \text{ if and only if a legal sentence becomes valid before } T \text{ and it is not the case that the sentence is terminated before } T. \]

   Yoshino notes that event calculus provides a hint for this rule, and the rule is treated as the most fundamental FLMRS. It is represented as follows:

   A legal sentence \( S \) is valid at the time \( T \) \( \iff \)

   \[ (S \text{ becomes valid at time } T_1 \text{ before } T) \& \quad \text{// First requirement} \]

   \[ \text{not (} S \text{ is terminated after } T_1 \text{ and before } T) \quad \text{// Second requirement} \]
Yoshino holds that all positive FLMRSs regulate the fulfilment of the first requirement \((S \text{ becomes valid})\) or the second requirement \((S \text{ is terminated})\) of \(ro\) above.

4. **Basic Legal Meta-Rule Sentence (BLMRS)**

The validity of the final, highest legal sentence, whose validity cannot be deduced through the application of legal meta-rule sentences, is called the *basic legal meta-rule sentence*; see Figure 6. (Again, validity is considered as truth – in the world of science). For example, in international law, LMRSs that regulate the validity of conventions must be assigned to the highest level.

![Diagram of the hierarchy of legal sentences](image)

**Figure 6. Basic Legal Meta-Rule Sentence (BLMRS)**

### III. Comparison of Yoshino’s LJ with Kelsen’s PTL

First, there is a similar solution at the top of the system: Yoshino’s BLMRS and FLMRS correspond to Kelsen’s basic norm (Figure 7).

Secondly, Yoshino’s hierarchy of legal sentences corresponds to Kelsen’s hierarchy of norms. Thirdly, Yoshino’s validity as truth corresponds to Kelsen’s legal validity in Ought. Fourthly, Yoshino uses predicate logic while Kelsen reasons in textual form. Yoshino allows a differentiation between juridical sentences. Yoshino departs from logic, and goes through Logical Jurisprudence – LMS, LRS, LOS, LFS – to expert systems which can contribute to legal objects in Is. Legal Object Sentences refer to
legal objects on the Is stage of individual rights and duties. Fifthly, Yoshino’s work is
directed at future computer applications while Kelsen’s work is directed at jurists.
Legal ontologies can be a further supplement.

Figure 7. A similar solution at the top of the system

C. ONE-ARCH BRIDGE SEEMS ACTUALLY NOT REALISTIC

The premise of this article is that it seems not very realistic to proceed directly in one
step from legal texts to their formalization (Figure 8). Intermediate steps are needed
(Figure 9). In other words, we hold that a one-arch bridge is not realistic and advocate
a multi-arch bridge of some kind.

Figure 8. The task as a one-bridge approach

In the transformation “legal text → formalization” we expect a formalization in the
form of a technical notation (language) or a specification that further allows computer
implementation. Hence, the target is “legal text → program” transformation where
‘program’ means a technical language – computer code referring to software and
hardware. A one-bridge approach metaphor can be used to specify the task.
One-Bridge Formalizations. There are many approaches to formalizations in the legal domain. Here, various formalisms, notations, logics and modeling techniques are used. As a one-bridge approach, Ilmar Tammelo\textsuperscript{15} addressed logic-based representation. He was successful in representing short legal texts in the prefix notation of binary operators. However, such formal notation was not easy to read. Ota Weinberger\textsuperscript{16} employed conventional logic representation. Marek Sergot et al.\textsuperscript{17} employed logic programming while representing the British Nationality Act as a logic program. Grabmair and Ashley\textsuperscript{18} examined two transformations: Firstly, the statute text is transformed into an Intermediate Norm Representation, and then to a rulebase.

Whilst the transformation is feasible in the case of a clear statement, difficulties arise with complex texts and a scalability problem is faced. Hence, the quality of transformation is acceptable for small texts only. However, the quantity (scalability) is not acceptable. The problem is the application of logic to reproduce the whole text and a corpus of texts. The selective application of logic can be compared with a scalpel. You cannot use a scalpel to chop as you would an axe. Logic is a precise tool and its limitations should be considered. Similarly, a bridge from San Francisco to Hawaii is impossible. However, nearly eight kilometres Öresund Bridge between Sweden and Denmark was built in 2000. Here, the early attempts of artificial intelligence research

\textsuperscript{15} TAMMELO, ILMAR, Modern Logic in the Service of Law, 1978, Springer, Vienna.

\textsuperscript{16} WEINBERGER, OTA, Rechtslogik, 1989, Duncker & Humblot, Berlin.


on understanding natural language can be recalled. You can succeed in a world of toy blocks, but it is hardly feasible to represent the meaning in the general case. Moreover, “if not most, legal inference schemes are non-deductive.”

Symbol level representations of legal teleology are extensively used in artificial intelligence and law; see, e.g., Bench-Capon’s work on legal teleologies. We find the knowledge level representation a true challenge for legal informatics. In this way, a knowledge engineer meets a problem of understanding the meaning of fundamental legal concepts, such as the nature of law, a legal method, legal interpretation, justice, value, etc. Though, jurists know the meanings. The concepts are studied at different levels of abstraction in specific branches of law, legal dogmatics, legal theory, jurisprudence, and legal philosophy.

**Ken Satoh’s JUF Theory.** Ken Satoh’s the Japanese Presupposed Ultimate Fact Theory (the JUF theory) is also a ‘one-bridge’ approach. Satoh et al. translated the JUF theory into logic programming. Satoh’s one bridge can be divided into two phases. The first is ‘formalising the JUF theory’ (see his Section 2), where the JUF tree is an intermediate result. The second step is ‘translation into logic programming’ (see his Section 3) – the result of which is a Prolog program (Figure 10).

Satoh’s work concerns the *non liquet* (unknown truth value) problem in civil law. Jurists are familiar with assumption rules and the burden of persuasion – who has to prove what. The formalization is an attempt to invoke computer science to solve the

---


23 Ibid., pp. 165–167.
non liquet truth value. We would also identify the next steps: ‘developing the PROLEG system’ (see Satoh et al.’s Section 4) and ‘using the JUF theory’, in particular by judges.

D. MULTI-ARCH BRIDGE IMPLIES MULTIPHASE TRANSFORMATION

The building of a bridge is continued with the observation that legal knowledge representation is needed as an intermediate step. Hence, the input/output chain is Legal text → Legal knowledge representation → Program (Figure 11).

Next, Legal knowledge representation is decomposed into three intermediate stages: textual microcontent, symbolization/visualization, and formalization (Figure 12).

The four steps in Figure 12 are specified by “input → output” pairs:
Step 1. Microcontenting: legal text → textual microcontent
Step 2. Visualising: textual microcontent → symbolization/visualization
Step 3. Formalising: symbolization/visualization → formalization
Step 4. Implementing: formalization → program

In Step 4, the program is implemented in a programming language. This program can be comprised in a complicated information system.

The transformation is also illustrated in Figure 9 with the bridge metaphor that connects the legal domain and informatics. Here, symbolization precedes strict mathematical formalizations; the latter are set aside for the future. The multi-arch bridge metaphor is a form of knowledge visualization, too.

Formalization and symbolization have to be distinguished. Symbolization means to be a symbol of something. The four steps above can be worded in other ways or even divided into smaller steps. Intermediaries, such as draft/detailed design and testing, can be taken into account. The boundaries of the intermediate steps depend on engineering methodologies.

The transformation above is introduced at the parole level. The transformation can also be viewed in the langue meta-level context. Therefore, the intermediaries are viewed in the parole and langue levels (Figure 13).

Figure 13. The Multiphase Transformation in the parole and langue context

---

Next, two more levels are added: an ontology level and language reference level. The ontology level is comprised of the core ontology – how to build domain ontologies – and legal ontologies. Therefore, the transformations are both multiphase and multilevel (Figure 14).

![Diagram showing elements of multiphase multilevel transformations]

Figure 14. Elements of multiphase multilevel transformations

The illustrations in this article are experimental – a proof of the concept. They are intentionally not drawn very formally and take the form of representation, which is targeted at human beings rather than computers.

**Intermediate Phases.** Intermediate structures serve as semantic bridges and are necessary. A one-arch bridge is impossible to build when the gap is too wide. Another case is when the “banks” are blurred.

Four steps (phases) above are called Microcontenting, Visualising, Formalising, and Implementing. Step 1 leads from the legal text to textual microcontent; see Figures 9

---

and 12. In this step, the linear legal text is divided into smaller portions that are organised, but not necessarily sequentially.

Step 2 takes the textual microcontent to symbolization/visualization. This can be achieved by structuring the legal text. Symbolization is a good intermediary. For the sake of formality, the text can be made more structured. The subject index, the lists (of references, tables, figures, etc.), abstractions, hypernyms/hyponyms, abbreviations, and other means can be used. Next, getting free of sentences and their grammar allows operating with symbols directly, and not with words as in the Greco–Roman tradition. Hence, a symbol and not a sentence becomes a communication unit. A grammar of symbols can be introduced, cf. traffic signs and Olympic pictograms. Structural legal visualization aims at representing legal meaning.\(^{27}\)

Step 3 leads from symbolization/visualization to formalization. This phase can be decomposed additionally. Step 4 leads from formalization to knowledge representation/implementation in a computer. From engineer’s view this step is a complex task that can also be divided into sub-phases. Various software engineering methodologies can be employed.

---

Figure 15. Stages in the development of knowledge-based systems. This diagram emphasizes the development of knowledge in KBS from a product-oriented point of view

**Systems Development Life-Cycle in Software Engineering.** Intermediate steps in the legal text-to-program transformation can follow the life-cycle stages of knowledge-based systems. The model of Mark Stefik in Figure 15 identifies five stages: Identification, Conceptualization, Formalization, Implementation, and Testing.\(^{28}\) This model is a variation of the waterfall model, Requirements–Design–Implementation–Verification–Maintenance.

**E. DIGITAL HUMANITIES AND HUMAN DIGITALITIES**

In this section the right bank of the multibridge is studied (Figure 16). There appears the sphere which is called *digital humanities*. It “refers to new modes of scholarship and institutional units for collaborative, transdisciplinary, and computationally-engaged research, teaching and publication.”\(^{29}\) A field map of digital humanities can be proposed and comprises such methods as “enhanced critical curation”, “augmented editions and fluid textuality”, “visualization and data design”, “the animated archive”, etc.\(^{30}\) Digital humanities “place a primacy on experiential navigation, epistemologies of representation, and the rhetorics of visualization.”\(^{31}\) The human is the research subject of digital humanities, and this human is surrounded by digitally-based phenomena, including information and communication technologies (ICT).

![Figure 16. Digital humanities and human digitalities](image)

---


\(^{30}\) Ibid., pp. 30–31.

\(^{31}\) Ibid., p. 46.
Next comes the sphere which we call human digitalities. Its research subject is the machine, which is structured according to human-centred rules. The term may seem too pretentious. However, we have coined it to articulate machines, which are required to act in conformance with the law – analogously as human beings ought to act.

**Digital Humanities.** Brunschwig starts noting that “both the digital humanities and what might (as a working hypothesis) be called digital law explore visualization, audiovisualization, and multisensorization.”\(^3^2\) She brings the digital humanities and digital law closer together. We, however, further examine human digitalities.

**Human Digitalities.** The latter focuses on the concept of legal machines.\(^3^3\) After the text-to-program transformation, the program becomes law and is treated as a legal machine. A legal machine can be defined as a machine in a system whose actions have legal importance and legal consequences. There are simple legal machines, such as traffic lights, barriers and vending machines, and complex ones, such as the electronic forms that are used for taxes and finance. Legal machines contribute to law enforcement, and their software implements legal rules. Legal machines are purposed at regulation by computer code. Here, we confer to the concept “[computer] code is law”\(^3^4\) in the context of cyberspace regulation.

---

\(^3^2\) Brunschwig, Note 4, p. 1.


Animals–human beings–machines is a kind of evolution line (Figure 17). Animals are the ancestors of human beings and human beings produce machines. Digital humanities belongs to the world of human beings, whereas human digitalities belongs to the world of machines.

The theme of evolution leads further than legal formalization and brings indeed to human digitalities. Wagner/Haag\textsuperscript{35} write about the step from symbolization towards formalization. They use the term symbolization (Symbolisierung → Formalisierung). However, symbolization is currently interpreted in a broad sense, i.e., comprising both the symbolization-in-the-narrow-sense and visualization. Hence, we go further from Wagner/Haag and investigate the evolution step which results in machines for digital processing.

\textbf{Lyubimskii’s Community of Programs.} Eduard Lyubimskii\textsuperscript{36} writes about the infosphere and introduces the concept of community of programs. The community of programs is comprised by the world of machines. Lyubimskii sees the problem “to ensure the coexistence and cooperative operation of programs.” Lyubimskii’s has two

\textsuperscript{35} Wagner, Heinz/Haag, Karl, Die moderne Logik in der Rechtswissenschaft, 1970, Gehlen Verlag, Bad Homburg vor der Höhe, Berlin, Zürich.

messages to the creators of human digitalities. The first message is that the interaction of programs should be ruled by human laws: “the structure of the community of programs and the means of their interaction are largely similar to the structure and means of interaction in human society.” The second message warns us that “the human society becomes increasingly dependent on the community of programs, while the community of programs gradually becomes less dependent on the participation of people in its activity.”37 The infosphere will be a community of programs organised on the same principles as the human society. The worldwide computer network is the habitat of software.

Conclusions. The article starts with the view that symbolization/visualization precede formalization on the way to legal knowledge representation. The context is the changeover from a text culture to a machine culture in law. Formalising legal meanings reveals additional goals and makes them explicit. Specifically, open-texture in the law can be decreased.

Intermediaries are important, as additional goals can be revealed. It is complicated to focus concurrently on multiple, probably conflicting, goals. Intermediaries appear within multistep and multilevel transformations: ontology – langue – parole – referential areas. Intermediaries can be explained by the roles of communication, tertium comparationis, and abstraction in the process of cognition.

Yoshino’s Logical Jurisprudence was created about seven decades after the first appearance of Kelsen’s PTL. Currently, the computational modelling of legal reasoning also needs a system of legal terms, such as a thesaurus. Artificial Intelligence & Law requires formal methods, and here Logical Jurisprudence serves as a means. Yoshino’s Compound Legal Formula representation and Prolog remind us of the high expectations of knowledge engineers in the 1980s. However, a critical perspective on the nature of law has to be taken by both the academic legal community and informatics.38

37 Ibid. p. 2.
Yoshino is concerned with both legal reasoning and systematization. This article aims to visualise the architecture of LJ. Logical Jurisprudence is within a traditional research of legal logic and is an achievement on the level of a model. However, it does not succeed in implementing the whole interconnectedness of norms. Nowadays, we would suggest expanding these concerns with, first, legal ontologies and, second, words. We view the latter within the granularity of word-phrase-sentence-text. Different methods of legal informatics are applied to different units. Here, we also refer to the research of Tsuno\textsuperscript{39} on the legal lexicon.

The term \textit{human digitalities} is introduced intentionally to contrast with the term \textit{digital humanities}. The research subject of human digitalities is machines, which are required to act in conformance with the law. Human digitalities is viewed in the context of evolution animals–human beings–machines.

\textsuperscript{39} Tsuno, Repertorium Aureum, Note 9.