

A TYPOLOGY OF LEGAL TECHS

A METHOD

Mireille Hildebrandt, FBA

PI **COHUBICOL** ERC ADG project

A TYPOLOGY OF LEGAL TECHS

A MINDSET

Mireille Hildebrandt, FBA

PI **COHUBICOL** ERC ADG project

A TYPOLOGY OF LEGAL TECHS

A RESOURCE

Mireille Hildebrandt, FBA

PI **COHUBICOL** ERC ADG project

What's next?

- The Typology
- Code-driven Legal Techs
- Rules as Code
- The Typology: WhiteBoxing Legal Techs

What's next?

- **The Typology**
- Code-driven Legal Techs
- Rules as Code
- The Typology: WhiteBoxing Legal Techs

By who?

L. Diver, P. McBride, M. Medvedeva, A. Banerjee, E. D'hondt, T. Duarte, D. Dushi, G. Gori, E. van den Hoven, P. Meessen, M. Hildebrandt

- Law Team: Diver, Van den Hoven, Gori, Duarte, McBride, Dushi, Banerjee – Hildebrandt (Vrije Universiteit Brussel)
- CS Team: Meessen, D'hondt, Medvedeva (Radboud)

For whom?

Legal practice and academia

- Courts
- Law firms
- Legislature
- Public Administration
- Regulators
- Bar Associations
- Law Schools
 - Legal education (Tel Aviv course)
 - Legal research

CS practice and academia

- Developers of legal technologies
- Computer Science

Natural and legal persons

- Citizens
- NGOs in advocacy or activism around fundamental rights
- Companies

For whom?

Legal practice and academia

- Courts
- Law firms
- **Legislature**
- **Public Administration**
- **Regulators**
- Bar Associations
- Law Schools
 - Legal education (Tel Aviv course)
 - Legal research

CS practice and academia

- **Developers of legal technologies**
- Computer Science

Natural and legal persons

- **Citizens**
- **NGOs in advocacy or activism around fundamental rights**
- **Companies**

What

- Mapping
- Comparing
- Assessing

What

Assessing:

Let's not be naïve and investigate:

- Upstream design decisions, downstream effects
- Intended purpose, claimed functionality and other foreseeable use

- Design
- Deployment
- Effect on legal effect

What's next?

- The Typology
- Code-driven Legal Techs
- Rules as Code
- The Typology: WhiteBoxing Legal Techs

Code-driven

- Legal Expert Systems
- Automated Decision Systems
- Rules as Code

SHOWING 13 TECHS

RESET FILTER

END-USERS: Any | FUNCTIONALITY: Any | CODE/DATA-DRIVEN: Code-driven | TYPE OF SYSTEM: Any (selected), App, Dataset, Paper

Akoma Ntoso Legislation Search	Blawx Legislation	Catala ADM Legislation
DataLex LES Contract Legislation	Iubenda Compliance Contract	Legislation Editing Open Software (LEOS) Legislation
Polisis LES Compliance	RegelSpraak ADM Legislation	The British Nationality Act as a logic program (Sergot et al. 1986) LES Legislation
Traffic Regulation Application and Comparison System (TRACS) LES Legislation	TreeAge LES Litigation	Truendo Compliance
Westlaw Edge Litigation Search		








What's next?

- The Typology
- Code-driven Legal Techs
- **Rules as Code**
- The Typology: WhiteBoxing Legal Techs

SHOWING 7 TECHS

 RESET FILTERS

END-USERS	FUNCTIONALITY	CODE/DATA-DRIVEN	TYPE OF SYSTEM			
<input type="text" value="Any"/>	<input type="text" value="Legislation: represent"/>	<input type="text" value="Code-driven"/>	<input checked="" type="radio"/> Any	<input type="radio"/> App	<input type="radio"/> Dataset	<input type="radio"/> Paper


Akoma Ntoso Legislation Search 	Blawx Legislation 	Catala ADM Legislation 
DataLex LES Contract Legislation 	RegelSpraak ADM Legislation 	The British Nationality Act as a logic program (Sergot et al. 1986) LES Legislation 
Traffic Regulation Application and Comparison System (TRACS) LES Legislation 		


SHOWING 4 TECHS


 RESET FILTERS


END-USERS FUNCTIONALITY CODE/DATA-DRIVEN TYPE OF SYSTEM

Any Legislation: executior Code-driven Any App Dataset Paper

Blawx
Legislation 

Catala
ADM Legislation 



RegelSprak
ADM Legislation 

The British Nationality Act as a logic program (Sergot et al. 1986)
LES Legislation 

SHOWING 2 TECHS

 RESET FILTERS

END-USERS	FUNCTIONALITY	CODE/DATA-DRIVEN	TYPE OF SYSTEM			
<input type="text" value="Any"/>	<input type="text" value="Legislation: drafting"/>	<input type="text" value="Code-driven"/>	<input checked="" type="radio"/> Any	<input type="radio"/> App	<input type="radio"/> Dataset	<input type="radio"/> Paper

<p>Akoma Ntoso</p> <p>Legislation Search </p>	<p>Legislation Editing Open Software (LEOS)</p> <p>Legislation </p>
---	---

Rules as code

- Representation (enabling to find and search binding legal text)
- Execution (enabling to build enforcement and/or compliance tools)
- Drafting (enabling to foresee contradictions and implications)

Rules as code

- Representation (enabling to find and search binding legal text)
- Execution (enabling to build enforcement and/or compliance tools)
- Drafting (enabling to foresee contradictions and implications)

- Code-driven
 - Representing and/or drafting legislation

- Markup language

- AKN is infrastructure, much is built on it (e.g. LEOS)
 - The toothpaste analogy
 - The force of law differs from the force of technology
 - What if applications are patented?

POTENTIAL LEGAL IMPACT

- Akoma Ntoso's default set of concepts may be inappropriate or insufficient for a specific [jurisdiction](#) or context of application. If left untailored, this could have implications if these are relied upon by other computational law systems that rely on document structure for their functionality (e.g. search).
- Maintenance of Akoma Ntoso documents will be required to ensure that any relevant changes in the natural language meaning of a document (perhaps due to [judicial interpretation](#)) are reflected in the machine-readable tags embedded in it.
- Maintenance may be required to ensure cross-references between documents are kept up to date, for example following legislative amendments or repeals. This is especially true where other computational systems rely on Akoma Ntoso tagging as an authoritative statement of the relationships between legal texts.

- Code-driven
 - Representing legislation
 - AKN based, **automating the mark-up process**

- 1. Making legislation machine readable, easy to find and navigate
- 2. Allowing for cross-referencing, visualisation of interacting legal norms
- 3. Thus visualising complexity (foreseeing contradictions, increasing agency)

What's next?

- The Typology
- Code-driven Legal Techs
- Rules as Code
- The Typology: WhiteBoxing Legal Techs

The Typology: WhiteBoxing Legal Techs

- Legal norms actually depend on case law
 - The complexity is endless
- We need qualitative assessment
- Quantitative assessment is meaningless (cannot replace quality)
 - Visualisation of complexity invites gaming the system
- Imagine patenting a more advanced version, giving deep pockets a headstart



m.e.menair