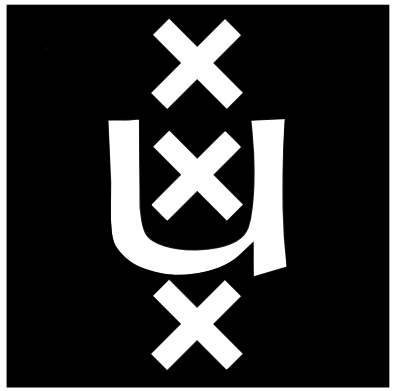
Belastingdienst



Systematic inventory and comparative analysis of formal approaches in legal contexts

The World's First Norm Engineering Conference
28 January 2025 | University of Amsterdam

Sofia Ali

s.ali@uva.nl

s_ali@belastingdienst.nl

About me

- Master’s degrees: Business Information Management (Erasmus University Rotterdam, 2021) and Economics and Management of Innovation and Technology (Bocconi University, Milan, 2020)
- Started at the Dutch Tax Authority in 2021 as Business Analyst.
- Joined the Rule Development and Management Team in 2023 as Rule Analyst.
- Started my PhD at the University of Amsterdam in November 2024.

Agenda

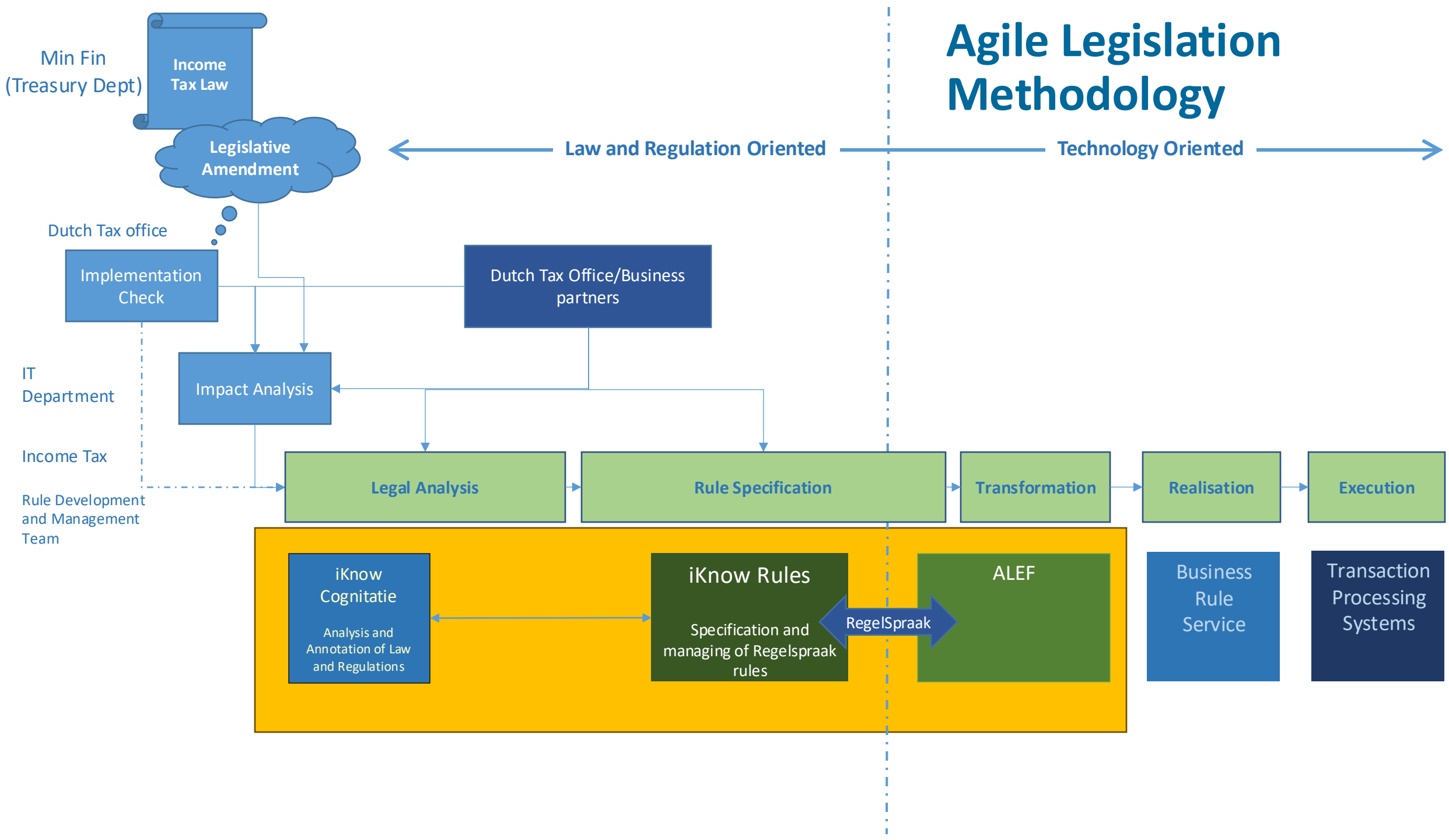
1. My motivation
2. Research Context
3. Formal Languages
4. Background study
5. Methodology
6. Current Status and next steps
7. Q&A

My motivation

The Rule Development and Management Team:

- 20 professionals (mix of tax specialists and rule analysts).
- Goal: Ensure unambiguous and consistent tax rules.

Agile Legislation Methodology



My motivation

Optimizing the rule generation process

Research Context

Why this research?

Formal Languages are essential for transforming legal texts into **clear, machine-readable**, **executable rules,** ensuring consistency and traceability.

Current evaluation methods for formal languages are inconsistent and often lack completeness, making it challenging to assess the practical utility of these languages

Research Goal:

Develop a framework to assess the operational viability of formal languages in regulatory and legal contexts.

Formal Language at the Dutch Tax Authority

- **What is RegelSpraak?**
  - RegelSpraak is based on an enhanced version of RuleSpeak.
  - Domain-Specific Language (DSL) and Controlled Natural Language (CNL), developed in Dutch.
  - Used to translate complex tax laws into machine-readable rules.
  - RegelSpraak rules are transformed into executable code by Agile Law Execution Factory (ALEF).
- **Purpose:**
  - Reduce ambiguity and ensure consistent application.
  - Reduce the resources needed to create applications.

Background study

- Three key studies —Chun et al. (2024), Kaptijn & Klaver (2024), and Parvizimosaed et al. (2022)— compared various formal languages, such as Catala, Symboleo, and Logical English.
- Each study evaluated specific formal languages based on different aspects.
- Some overlap existed in the aspects analyzed, but they were often named differently across studies.
- Certain formal languages appeared in multiple studies.
- Identified 31 formal languages and 31 unique components for evaluation.
- This analysis forms the foundation for the next steps in my research

Overview of other formal languages

Logical English (LE):

- Developed by Kowalski et al. in 2019-2020.
- Connects legal language with logic programming.
- Resembles normal English, making it accessible for non-technical users.
- Capable of handling complex legal rules and processes while remaining human-readable.
- Used in areas such as contracts, legal reasoning, and smart contracts.

Catala:

- Developed by Merigoux et al. in 2021, France.
- Transforms legal texts into executable code.
- Focuses on laws involving calculations and decisions, such as tax laws, pensions and social benefits.
- Facilitates collaboration between legal experts and programmers.

Symboleo:

- Developed in 2020 by Parvizimosaed et al., University of Ottawa, Canada.
- Formal language for legal contracts.
- Focuses on obligations and powers to ensure compliance.
- Applications in supply chain, insurance, and energy sectors.
- Includes tools for validation and monitoring.
- Ensures contracts are clear, consistent, and legally valid.

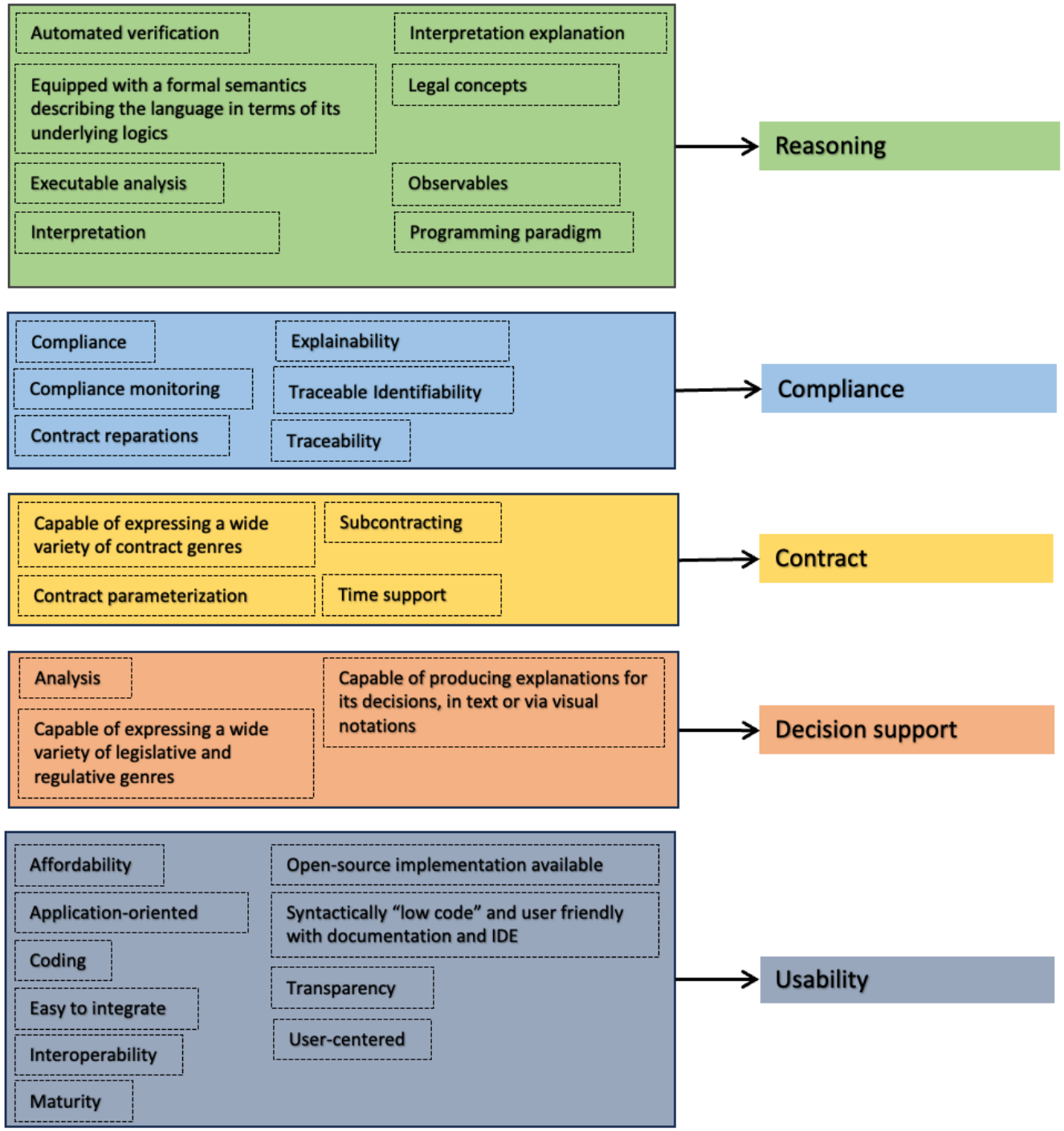
Components	Catala	Symboleo	Logical English
Equipped with a formal semantics describing the language in terms of its underlying logics	X	X	X
Capable of expressing a wide variety of contract genres		X	X
Capable of expressing a wide variety of legislative and regulative genres	X		X
Open-source implementation available	X	X	X
Syntactically “low code” and user friendly with documentation and integrated development environment (IDE) support			X
Capable of producing explanations for its decisions, in text or via visual notations			X

Analysis of Articles and Components

Methodology

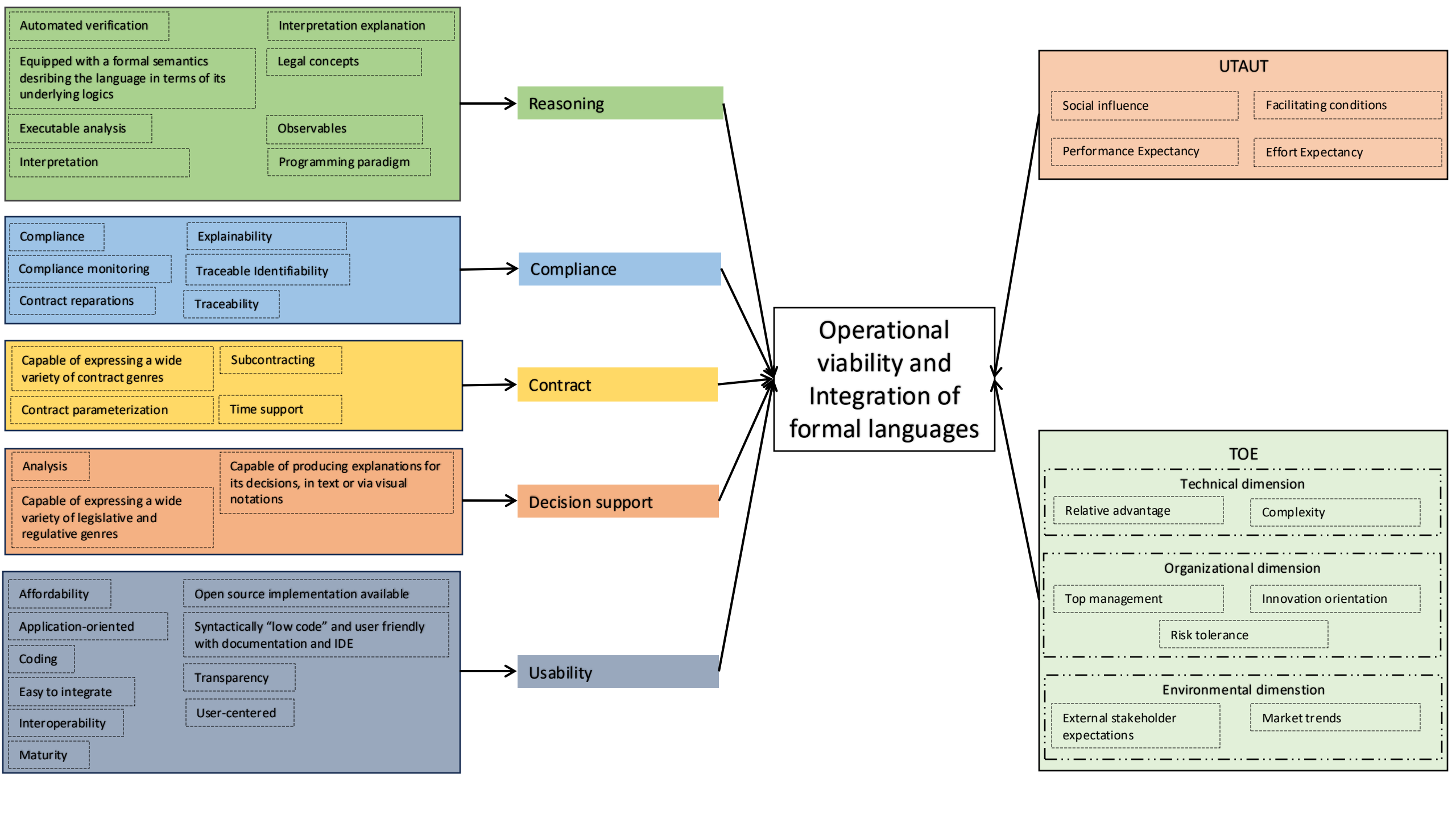
- Grounded Theory Approach used to develop the evaluation framework.
- Collected and analyzed aspects identified in key studies.
- Merged these aspects into categories.
- Resulted in five key categories:
  - **Reasoning:** How well the language handles logic and legal analysis.
  - **Compliance:** Ensures legal standards and validation mechanisms are met.
  - **Contract:** Supports flexibility in creating and managing contracts.
  - **Decision Support:** Helps with clear decision-making and resolving disputes.
  - **Usability:** Makes the language practical and accessible for users.

Key Categories Derived from Grounded Theory



Methodology

- **Purpose:** Fill gaps in current frameworks by including external factors that influence adoption.
- **Key Components:**
  - **Grounded Theory:** Five main categories (reasoning, compliance, contract management, decision support, usability)
  - **External adoption factors such as….**
    - **Unified theory of acceptance and use of technology (UTAUT) model:** User-centered adoption factors
    - **Technology-Organization-Environment (TOE) framework:** Technological, organizational and environmental dimensions


Automated verification
Interpretation explanation
Equipped with a formal semantics desribing the language in terms of its underlying logics
Legal concepts
Executable analysis
Observables
Interpretation
Programming paradigm
Reasoning
Compliance
Explainability
Compliance monitoring
Traceable Identifiability
Contract reparations
Traceability
Compliance
Capable of expressing a wide variety of contract genres
Subcontracting
Contract parameterization
Time support
Contract
Analysis
Capable of producing explanations for its decisions, in text or via visual notations
Capable of expressing a wide variety of legislative and regulative genres
Decision support
Affordability
Open source implementation available
Application-oriented
Syntactically "low code" and user friendly with documentation and IDE
Coding
Transparency
Easy to integrate
User-centered
Interoperability
Maturity
Usability
Operational viability and Integration of formal languages
UTAUT
Social influence
Facilitating conditions
Performance Expectancy
Effort Expectancy
TOE
Technical dimension
Relative advantage
Complexity
Organizational dimension
Top management
Innovation orientation
Risk tolerance
Environmental dimenstion
External stakeholder expectations
Market trends

Current status and next steps

Status:

Framework is under development

Next Steps:

Refine the framework based on feedback and observations

Apply the framework to formal languages (e.g., Regelspraak, Catala, Symboleo, eFLINT, Logical English)