IndiLegalOnt: An Ontology for the Indian Legal System

Apurv Dube, Joseph Pookkatt, **Raghava Mutharaju**, Vikram Goyal
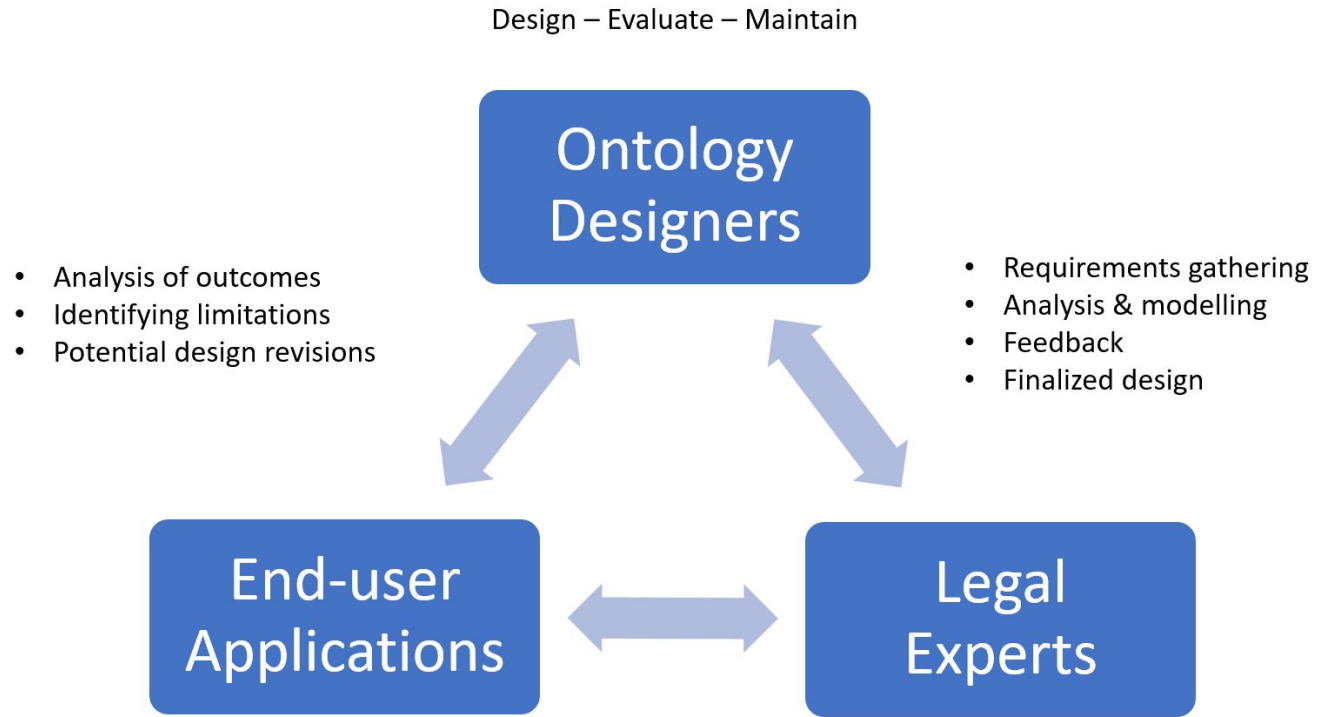
WOP 2024, Baltimore, USA

Motivation

- Law firms need to collect, process & share legal information.

- A consistent vocabulary between clients, vendors and the other businesses helps.

- Structured knowledge facilitates legal information retrieval and decision-making.

- When coupled with other ML models it can help maintain accuracy.
  - For instance, LLMs + KGs

Objectives

- Develop an ontology specific to Indian courts and regulations.

- Reuse and extend existing ontologies.
  - SALI-LMSS, GeoNames

- Start with the Indian court system and the Negotiable Instruments Act (NIA).
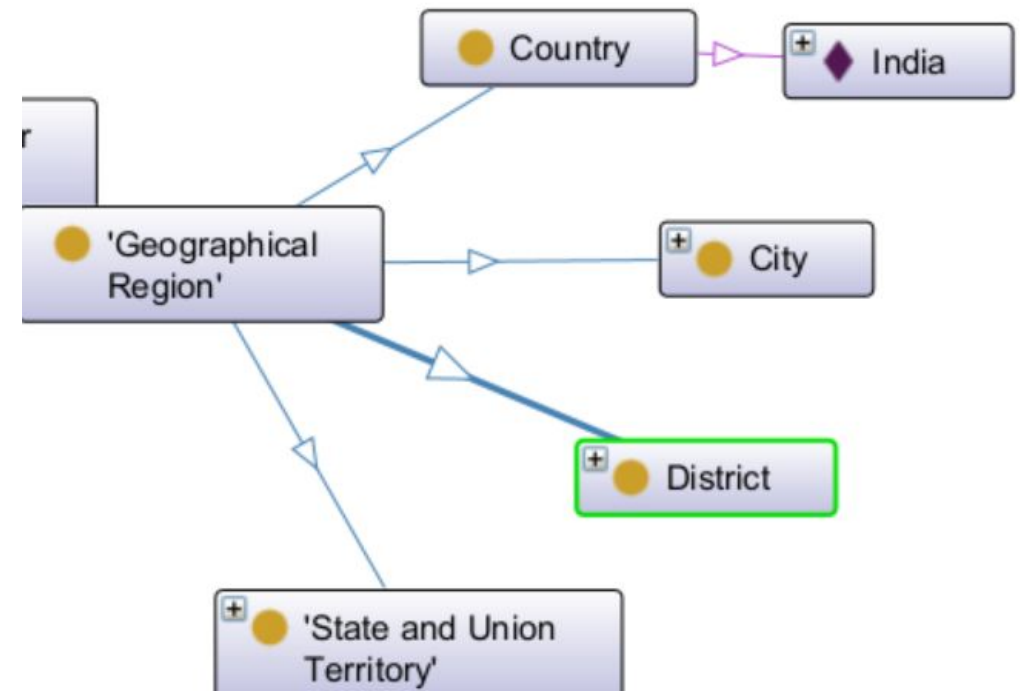
Methodology

- Competency questions (examples):
  - List all district courts in a given state.
  - Which court(s) have authority over a given court?
  - When is a check payment of a bad check?
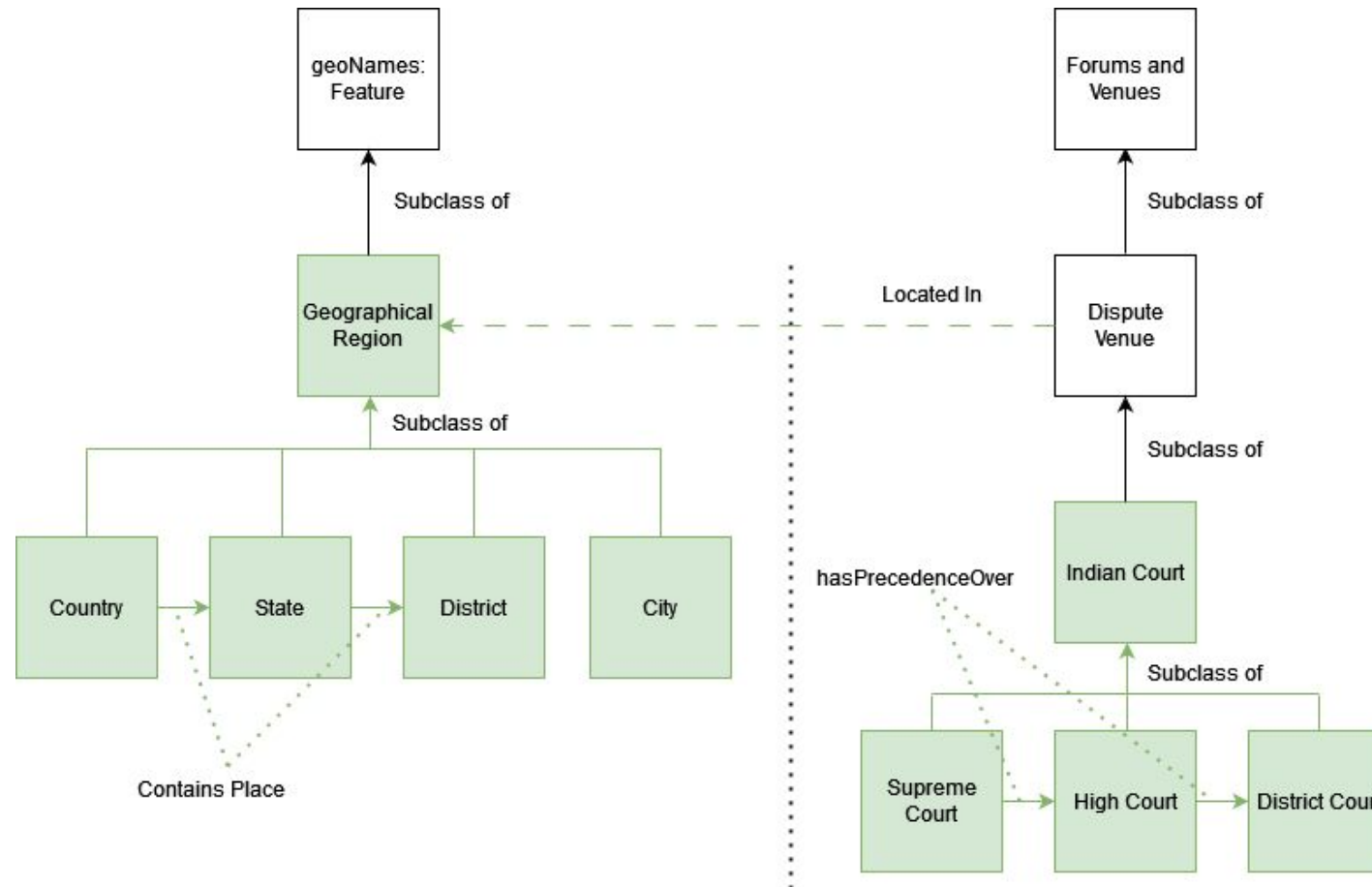 - What is the transaction status of a given check payment?

Design – Evaluate – Maintain

Ontology Designers

- Analysis of outcomes
- Identifying limitations
- Potential design revisions

- Requirements gathering
- Analysis & modelling
- Feedback
- Finalized design

End-user Applications

Legal Experts

Addition of location data

- Extracting information from GeoNames* web API:
  - Administrative divisions at level 1 (states and union territories)
  - Level 2 (districts)
  - Selected cities

- Integrating GeoNames location instances with SALI India.
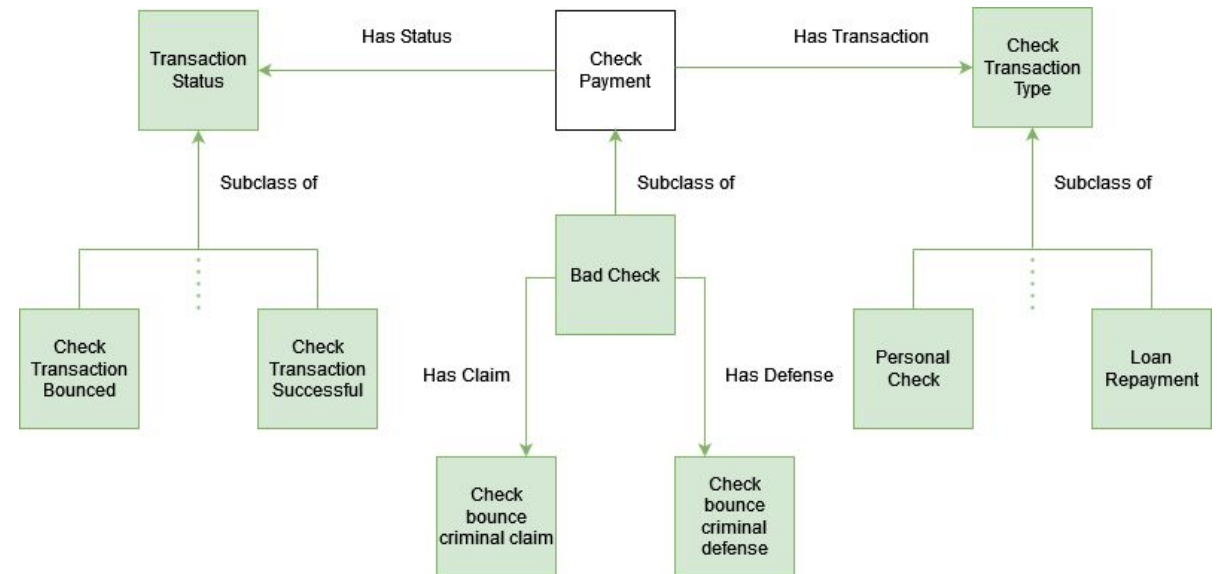
Location data & court structure

Note:
- Additions are shown in green.
- "Located In" property connects court instances to location instances.

Bad check crime modelling

- Indian legal system is flooded with check-bouncing cases.
  - 8% of the overall pending criminal cases.

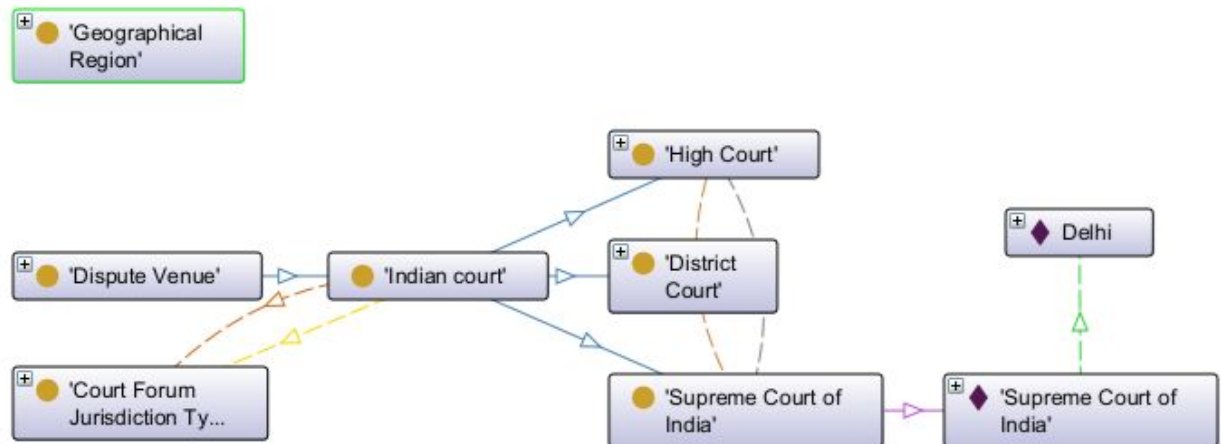- Automate data labelling and identify precedents.

Axioms

- SupremeCourt ⊑ ∃ hasPrecedenceOver.HighCourt

- BadCheck ≡ CheckPayment ⊓ ∃ hasStatus.CheckTransactionBounced

- hasPrecedenceOver ≡ precededBy⁻

- hasPrecedenceOver⁺ ⊑ hasPrecedenceOver

Automating bulk additions to OWL ontology

- A general purpose Java library using OWL API was written for the purpose of making these additions.
  - Creating class structures
  - Object, data & annotation properties
  - Searching etc.
- Data for about 25 high courts and 688 district courts was added.
- 104 government ministries and departments
- Integrating location data including states, districts and cities in India.

Protégé was used for only for visualization & for adding some axioms.

Conclusions & Future work

- **Long-Term Project:** Expanding the ontology gradually for broader use in the legal industry.

- **Applications:** Useful for tagging cases, analyzing documents, and supporting decisions.

- **Next Steps:**
  - LLM integration – Knowledge graph grounding reduces AI hallucinations.
  - Exploring ways to semi-automate ontology updates by learning from text, with expert supervision.

Code and queries

SPARQL query example using GraphDB

- **IndiLegalOnt**
  - GitHub Repository: https://github.com/kracr/indian-legal-ontology

  - Includes OWL ontology, Java library, SPARQL queries and documentation.

```
1   # In which state is a given district court instance located?
2
3   PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
4   PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
5   PREFIX ex: <http://lmss.sali.org/>
6   PREFIX geoNames: <http://www.geonames.org/ontology#>
7   PREFIX schema: <http://schema.org/>
8
9   SELECT DISTINCT ?stateLabel
10  WHERE {
11      # Approximate string matching for district court
12      ?districtCourt rdfs:label ?courtLabel .
13      ?districtCourt rdf:type ?districtCourtType .
14      ?districtCourtType rdfs:label "District Court" .
15
16      # Find the location of the district court
17      ?districtCourt geoNames:locatedIn ?district .
18
19      # Find the state that contains the district
20      ?state schema:containsPlace ?district .
21      ?state rdfs:label ?stateLabel .
22
23      # Replace "XYZ District Court" with your district court name
24      FILTER (regex(?courtLabel, "Prakasam District Court", "i"))
25  }
```

Run

keyboard shortcuts

⬇ Download as ∨

Filter query results   Compact view ☐ Hide row numbers ☐

Showing results from 0 to 1 of 1. Query took 0.2s, minutes ago.

	stateLabel	⬍
1	"Andhra Pradesh"	