연구보고서

온톨로지 기반 근로감독관 AI 지원시스템 구축 방안 연구

A Study on the Implementation Strategy for an Ontology-based AI Support System for Labor Inspectors

고용노동부
Ministry of Employment and Labor

고용노동부

「온톨로지 기반 근로감독관 AI 지원시스템 구축 방안 연구」

최종보고서

2025. 11. .

주식회사 사미텍

제 출 문

본 보고서를 “온톨로지 기반 근로감독관 AI 지원시스템 구축 방안 연구” 과제의 최종보고서로 출합니다.

2025. 11. 6.

수 탁 기 관 : 주식회사 사미텍

연구책임자 : 김 재 두
연 구 원 : 이 형 용
연구보조원 : 박 미 정

<목 차>

제1장 서론 ····· 1
제1절. 연구 개요 ····· 1
제2절. 소결 및 시사점 ····· 13
제2장 기존 연구 동향 및 분석 ····· 15
제1절. 법령의 구조 및 특징 ····· 15
제2절. 법률의 지식 표현 기술 ····· 23
제3절. 법률정보(지식)시스템 ····· 36
제4절. 소결 및 시사점 ····· 59
제3장 노동법 온톨로지 ····· 61
제1절. 노동 관련 법률 지식 구조화 ····· 61
제2절. 법령 온톨로지 모델 및 데이터화 ····· 87
제3절. 노동법(근로기준법 중심) 온톨로지 구축 ····· 102
제4절. (노동)법령 온톨로지 구축 통계 ····· 132
제5절. 소결 및 시사점 ····· 140
제4장 (노동)법령 온톨로지 활용 ····· 142
제1절. (노동)법령 온톨로지 효과 검증 ····· 142
제2절. (노동)법령 온톨로지 기반 규범 명제 추론 검증 ····· 154
제3절. 소결 및 시사점 ····· 173
제5장 AI 법령서비스 구현 및 비교 평가 ····· 176
제1절. 법령온톨로지 기반 AI 법령시스템 ····· 176
제2절. 소결 및 시사점 ····· 190
제6장 결론 및 향후 과제 ····· 192

<그 림 목 차>

(그림2-1) 법문서의 구조와 부속 구성요소 ····· 17
(그림2-2) 법문서의 구조와 부속 구성요소 ····· 18
(그림2-3) 법문서의 규범적 특징(AI 생성 후 보완 그림) ····· 19
(그림2-4) 법문서의 시간적 효력과 적용(AI 생성 후 보완 그림) ····· 19
(그림2-5) 법문서의 상.하위법 관계 (AI 생성 후 보완 그림) ····· 20
(그림2-6) LegalRuleML main blocks ····· 25
(그림2-7) SHACL 개념 ····· 29
(그림2-8) SHACL 개념 ····· 33
(그림2-10) 법령지식시스템 구성요소 간 데이터 흐름 ····· 50
(그림2-11) 통합 시스템 구성도 ····· 52
(그림3-1) 노동법 관계 법률 체계 ····· 65
(그림3-2) 법률 지식 구조화 프로세스 ····· 66
(그림3-3) 법문 구조 예시 (근로기준법) ····· 67
(그림3-4) 법령 온톨로지 구성 레이어 ····· 69
(그림3-5) 법령온톨로지 전체 구조 ····· 88
(그림3-6) 법령온톨로지 법령위계 설계 구조 ····· 90
(그림3-7) 법령온톨로지의 문서 구조 레이어 ····· 92
(그림3-8) 법령온톨로지의 규범명제 레이어 ····· 94
(그림3-9) 법령온톨로지의 관계/링크 레이어 ····· 97
(그림3-10) 법령온톨로지의 추상클래스 ····· 99
(그림3-11) 법령온톨로지 구축을 위한 파싱 파이프라인 ····· 102
(그림3-11) TopBraid를 이용한 법령 온톨로지 실행 화면 ····· 119
(그림3-12) 법령온톨로지 전체 UML ····· 123
(그림3-13) 온톨로지 추론 엔진 구성 (TopBraid 환경) ····· 128

(그림3-14) 온톨로지 추론 성능 검증과 검증 결과 ·····129
(그림3-15) 근로기준법 조문의 온톨로지 인스턴스화 결과 ·····130
(그림3-16) 근로기준법 조문의 온톨로지 인스턴스 그래프(일부) ·····131
(그림4-1) 법령 온톨로지 기반 개정 영향 탐지 체계 ·····142
(그림4-2) 기존 법령정보검색시스템의 검색 결과 (법제처) ·····143
(그림4-3) 주 4.5일제 관련 법령 온톨로지 기반 연관 법령 추론 네트워크 ·····152
(그림4-4) 근로기준법 위반 탐색을 위한 입력 사실 등록 화면 ·····157
(그림4-5) 근로기준법 위반 탐색을 위한 규칙 자동 구성 화면 ·····158
(그림4-6) 근로기준법 위반 탐색을 위한 전방추론 실행 화면 ·····160
(그림4-7) 근로기준법 위반 탐색을 위한 최종 결론도출 화면 ·····162
(그림4-8) 적법성 판단을 위한 시나리오 및 사실관계 입력 화면 ·····165
(그림4-9) 적법성 판단을 위한 시나리오 및 사실관계 입력 화면 ·····166
(그림4-10) 적법성 판단을 위한 규칙 자동 구성 화면 ·····167
(그림4-11) 적법성 판단을 위한 후방추론 실행 화면 ·····170
(그림4-12) 적법성 판단을 위한 최종 결론 출력 화면 ·····172
(그림5-1) LawBrain 아키텍처 ·····177
(그림5-2) LawBrain의 사용자 인터페이스 구성 ·····177
(그림5-3) 질의1 실행 화면 (비교) ·····180
(그림5-4) 질의2 실행 화면 (비교) ·····181
(그림5-5) 질의3 실행 화면 (비교) ·····182
(그림5-6) 질의4 실행 화면 (비교) ·····183
(그림5-7) 질의5 실행 화면 (비교) ·····184
(그림5-8) 질의6 실행 화면 (비교) ·····185

<표 목 차>

(표1-1) 공공서비스에서의 법령의 기대 역할 ····· 5
(표1-2) 법의 디지털 표현 및 저장 ····· 6
(표1-3) 온톨로지가 법령을 적용 가능한 규제 지식으로 만드는 주요 절차 ····· 7
(표2-1) 법체계의 단계와 유형 ····· 16
(표2-2) 법의 연관성 특징 요약 ····· 22
(표2-3) 법령 지식 표현 기준의 장점 및 한계점 ····· 28
(표2-4) 법령 지식 표현 기준의 장점 및 한계점 ····· 31
(표2-5) 규범명제 관점 온톨로지의 장점 및 한계점 ····· 31
(표2-6) 이중 온톨로지 설계의 근거 ····· 33
(표2-7) 법률지식시스템의 발전 단계 ····· 38
(표2-8) LKIF Core 온톨로지 개요 ····· 39
(표2-9) LKIF Core 온톨로지 모듈 구조 ····· 39
(표2-10) LKIF의 Norm 모듈 클래스 구성 ····· 40
(표2-11) 규범 클래스의 의무논리 대응표 ····· 40
(표2-12) 의미 논리 핵심 연산자 ····· 41
(표2-13) 의미 논리 연산자를 이용한 노동법 규범명제의 형식적 표현 예시 ····· 41
(표2-14) 규범 요소의 의무논리 기반 모델링의 시스템적 의의 ····· 41
(표2-15) 대표적인 법률 사례기반 추론 예시 ····· 42
(표2-16) CBR의 추론 프로세스 (4RE Cycle) ····· 42
(표2-17) 법률지식시스템 핵심 구성 요소 ····· 43
(표2-18) 법률지식베이스에서 지식 표현 형식 ····· 45
(표2-19) 규범 추론을 위한 의무논리 핵심 연산자 ····· 46
(표2-20) 의무논리 기반 근로기준법 규범 추론 사례 ····· 46

(표2-21) 법률 온톨로지 계층 ····· 46
(표2-22) Neural vs Symbol vs Neuro-Symbol 통합 비교 ····· 53
(표2-23) 통합 시스템 계층도 주요 구성 요소와 역할 ····· 53
(표2-24) 국가 운영 공공 법률정보시스템 현황 ····· 55
(표2-25) 민간 운영 법률정보시스템 현황 ····· 58
(표3-1) 온톨로지 구축을 위한 3대 기본 구조 ····· 66
(표3-3) 법문 구조 메타데이터 체계 요약 ····· 67
(표3-4) 온톨로지 구성 레이어별 구성 현황* ····· 71
(표3-5) 근로기준법 조문의 지식구조 통계 ····· 115
(표3-6) 근로기준법 시행령 조문의 지식구조 통계 ····· 117
(표5-1) LawBrain 핵심 특성 ····· 176
(표5-2) 비교 시스템 구성 ····· 179
(표5-3) 평가 질의 의도 설계 ····· 179
(표5-4) 질의1 비교 분석 ····· 180
(표5-5) 질의2 비교 분석 ····· 181
(표5-6) 질의3 비교 분석 ····· 182
(표5-7) 질의4 비교 분석 ····· 183
(표5-8) 질의5 비교 분석 ····· 184
(표5-9) 질의5 비교 분석 ····· 185
(표5-10) 질의7 비교 분석 ····· 186
(표5-11) 질의별 정량적 평가 결과 ····· 187
(표5-12) 종합 평가 지표 ····· 187
(표5-13) 기존 RAG 한계와 온톨로지 RAG기반 극복 ····· 189

제1장 서론

제1절. 연구 개요

1. 연구의 배경과 필요성

가. 법의 보편성과 사회적 기능

법은 인간 사회가 지속되기 위해 필요한 보편적 규범이자 공동체의 지식으로서, 법치국가에서 국가 운영의 근간을 이룬다. 법은 국민의 권리와 의무를 규정하고 사회 질서를 유지하며, 분쟁을 예방·해결하는 기준을 제공한다는 점에서 개인의 삶과 분리될 수 없다. 따라서 법이 실질적인 규범으로 기능하기 위해서는 국민 누구나 법을 확인하고, 자신의 상황에 어떤 규범이 적용되는지 이해하며, 필요한 경우 적절히 활용할 수 있어야 한다. 그러나 법은 본질적으로 전문적 언어와 복잡한 규정 체계를 기반으로 구성되어 있어 일반 국민에게는 접근과 이해가 쉽지 않다는 문제가 지속적으로 제기되어 왔다. 법률용어의 난해함뿐 아니라 조문 구조, 예외·특례·준용, 타 법령·타 조문에 대한 참조 등 다층적 관계가 얽혀 있어, 단순히 조문을 읽는 것만으로 "누가 무엇을 어떤 조건에서 해야 하는가"를 파악하기 어렵다.

더욱이 법은 고정된 텍스트가 아니라 사회 변화에 대응하여 제정·개정·폐지되며 시간에 따라 계속 변화한다. 특히 우리나라 법체계는 개정이 빈번하게 이루어지는 경향이 강하며, 개정이 발생할 때마다 공포 및 후속 정비를 위해 변경된 부분을 식별하고 체계적으로 관리해야 한다. 이 과정은 조·항·호 등 법문서의 구조적 체계와 문장 규칙에 대한 정확한 이해를 전제로 하며, 단순 문서 편집을 넘어 법규범의 의미와 관계를 함께 다루어야 한다. 그럼에도 현실에서는 법령이 텍스트 문서 형태로 제공·관리되는 경우가 많아, 법령 간 연관성, 조문 간 참조 구조, 개정 이력 및 시간적 효력(시행·경과·적용례)과 같은 중요한 요소가 체계적으로 표현·활용되지 못하는 한계가 존재한다.

한편 인터넷과 웹 기반 정보 환경의 확산으로 다양한 검색 서비스가 발전하면서, 국민이 법령을 확인할 수 있는 기반은 크게 개선되었다. 실제로 웹 기반 법령 검색 서비스는 국민이 자신에게 필요한 법령을 찾고 법적 정보를 획득하기 위한 필수적인 공공 인프라로 자리 잡았다. 그러나 "찾을 수 있게

하는 것”과 “이해할 수 있게 하는 것” 사이에는 여전히 간극이 크다. 법령은 조문 단위가 서로 긴밀하게 연결되고, 규범의 적용은 조건·절차·예외에 의해 달라지며, 상·하위 법령의 위임 구조를 통해 세부 기준과 서식이 하위 규범에 분산되어 규정되는 경우가 많다. 이런 특성 때문에 키워드 기반 검색과 단순 열람만으로는 사용자의 구체적 상황에 실제로 적용되는 규범을 식별하기 어렵고, 법령 정보는 “접근 가능하지만 이해하기 어려운 지식”으로 남는 경우가 많다.

또한 정보기술(IT)과 인공지능(AI)의 발전은 법 영역에 새로운 과제를 제기한다. 전자계약, 전자정부, 개인정보보호 등 IT 환경에서 발생하는 새로운 사회 현상은 규율 대상과 법적 쟁점을 확장시키며, 실무 차원에서는 리걸테크(Legal-Tech) 확산을 통해 상담·검색·분석 등 법무 처리 방식 자체가 빠르게 변화하고 있다. 이러한 변화는 법을 단순히 텍스트로 보관·열람하는 수준을 넘어, 법령이 갖는 구조·관계·시간적 효력과 규범 요소를 기계가 처리 가능한 형태로 표현하고, 그 결과를 설명 가능하게 제공하는 방향의 고도화를 요구한다. 다시 말해 법령 정보화는 단순한 문서 관리나 검색을 넘어, 법령을 구성하는 정의(개념), 규범(주체-행위-객체-요건-효과), 규범적 양상(의무·금지·허용 등), 수정관계(예외·특례·준용), 위임·참조 구조, 그리고 시행·경과·적용례와 같은 시간 규칙을 함께 구조화하여 지식으로 전환하는 단계로 발전할 필요가 있다.

나. 법령 정보화의 성과와 근본적 한계

대한민국은 1990년대 후반부터 법령 정보화를 추진하여 국가법령정보센터를 중심으로 법령의 전자적 제공 체계를 구축해 왔다. 이러한 노력은 법령에 대한 접근성을 획기적으로 개선하였으나, 법이 사회적으로 수행하는 다양한 역할과 기능-예컨대 규범의 적용대상 판정, 요건 충족 여부 판단, 예외 및 특례 적용, 준수 진단 및 제재 여부 판단-을 정보시스템 차원에서 충분히 제공하는 데에는 구조적 제약이 존재한다.

현행 법령 정보서비스는 주로 텍스트 기반 검색과 조문 단위 열람에 최적화되어 있다. 그러나 키워드 검색은 법령의 의미적 연결(semantic linkage)과 논리적 구조(logical structure)를 충분히 드러내기 어렵고, 조문 단위의 분절적 제공은 법령이 내재하는 유기적 관계—정의, 본칙, 예외, 준용, 위임 간의 연결—및 체계적 맥락을 단절시키는 경향이 있다. 그 결과 비전문가가 법령의 정확한 의미를 이해하고 적용하기 위해서는 여전히 법률 전문가의 해석과 자문에 대한 의존성이 높게 유지된다.

더 근본적으로, 국가 법령은 사회·경제 환경 변화에 따라 지속적으로 제정 및 개정되며, 그 영향이 상위법, 시행령, 시행규칙뿐 아니라 고시, 서식, 행정해석, 판례 등 다양한 준규범으로 확산된다. 즉, 법령은 '문서 집합'에 더하여 '연결된 규범 체계'이며, 개정은 단일 조문 변경에 그치지 않고 관련 규범 전반에 연쇄적 파급을 발생시키는 특성을 가진다. 그러나 문서 중심 관리와 관계형 데이터베이스 중심의 현재 방식은 이러한 연관성과 연쇄 영향을 완전하게 추적, 검증, 시각화하기에 본질적 한계를 갖는다.

다. 노동법(근로기준법 중심)의 특수성

근로기준법은 다른 법 분야에 비해 정보화 및 지능화의 필요성이 특히 절실한 영역이다. 근로기준법은 근로계약, 임금, 근로시간과 휴식, 여성과 소년, 안전과 보건, 재해보상 등 광범위한 영역을 규율하며, 각 영역별로 시행령과 시행규칙에서 세부 기준, 계산방식, 서식, 절차를 규정한다. 또한 사업장 규모별, 업종별로 적용 기준이 상이하고, 노동환경 변화에 따라 빈번한 법 개정과 판례 및 행정해석을 통한 지속적 보완이 이루어진다.

예컨대 근로시간 제도의 개편과 같은 변화는 단일 조항의 변경이 아니라 근로시간의 정의(제50조), 연장근로의 제한(제53조), 야간근로와 휴일근로(제56조), 탄력적 근로시간제(제51조, 제51조의2, 제52조), 적용 제외(제63조) 등 다수의 조문에 연쇄적으로 영향을 미친다. 나아가 시행령에서 규정하는 통상임금의 산정(시행령 제6조), 시행규칙에서 규정하는 서식과 신고절차에도 파급된다. 이러한 환경에서 근로감독관은 현장에서 "어떤 규정을 적용해야 하는지", "해당 행위가 위반인지", "예외나 특례가 존재하는지", "보고 및 조치 의무가 무엇인지"를 신속하게 판단해야 하지만, 현행 법령관리 시스템은 개정의 영향 범위를 자동 분석하거나 판단 과정을 지원하는 기능이 없기 때문에 수작업 확인과 개인 역량에 의존하는 구조적 한계를 갖는다.

또한 근로기준법은 임금, 근로시간, 휴가 등 일상적 노동관계에 직접 적용되며, 위반 시 형사처벌(제107조부터 제111조) 또는 과태료(제116조) 등 제재로 연결 될 수 있어 실시간 준수(real-time compliance)의 요구가 클 뿐만 아니라 준수 위반에 대한 판단의 중요도가 매우 높다. 2024년 기준 전체 취업자 약 2,800만 명이 근로기준법의 직접 적용 대상이다. 특히 소규모 및 영세사업장은 법률 전문가 접근성이 낮아 근로기준법의 사각지대에 놓여있으며 근로자 개인의 권리 보호 관점에서도 "이해 가능하고 적용 가능한 법 지식 제공"이 중요한 공공적 과제가 된다.

라. 인공지능 기반 법률서비스의 한계

최근 리갈테크와 인공지능 기반 법률서비스는 자연어 질의응답, 계약서 검토, 판례 분석 등 다양한 형태로 확산되고 있다. 그러나 다수의 시스템은 통계적 자연어처리와 패턴 기반 접근에 의존하여 다음과 같은 한계를 노출해 왔다. 첫째, 왜 특정 결론에 도달했는지에 대한 설명가능성이 부족하다. 둘째, 유사한 질의에 대해 상이한 답변을 제공하는 일관성 부족 문제가 있다. 셋째, 법적 추론의 핵심인 요건, 효과, 예외 구조를 안정적으로 반영하지 못하는 추론의 한계가 있다. 넷째, 공공행정에 요구되는 정확성, 재현성, 감사 가능성 측면에서 제약이 있다.

대규모 언어모델(Large Language Model, 이하 LLM) 기반 법률지원 기술도 자연어 이해와 생성에서 강점을 보이지만, 근거 기반 추론의 부족, 할루시네이션(hallucination), 조문 간 시맨틱 구조의 부재, 법령 개정의 즉시 반영 및 시점별 적용 규범 선택의 어려움 등 본질적 문제로 인해 단독으로는 공공 법률행정 분야의 요구-정확도, 일관성, 시점 정합성-를 안정적으로 충족하기 어렵다.

LLM의 법률 분야 적용에서 발생하는 문제는 선행연구에서도 반복적으로 지적되어 왔다. Dahl 등(2024)은 LLM이 법적 판단 과제에서 사실과 다른 정보를 생성하는 할루시네이션 비율이 상당하며, 특히 복잡한 다단계 추론이 요구되는 경우 오류율이 증가함을 보고하였다. Magesh 등(2024)은 LLM 기반 법률연구 도구가 생성한 판례 인용 중 상당수가 존재하지 않거나 잘못된 것임을 실증하였다. 이러한 연구들은 LLM이 법령의 시간적 효력[1)]이나 규범 상충[2)]과 같은 구조적 판단을 안정적으로 수행하기 어려움을 시사한다.

검색증강생성(Retrieval-Augmented Generation, 이하 RAG) 기법을 적용하더라도 “문서 근거를 붙인 생성“은 가능하나, 법령이 내재한 정형 구조(위계, 편제, 참조), 시간적 효력(부칙, 경과규정), 규범 상충(특별법 우선, 신법 우선 등)을 체계적으로 계산하고 검증하는 기능이 부족하면, 결국 서비스는 관련 조문을 제시하는 수준에 머물 가능성이 크다. 예를 들어, 근로기준법 제53조의 연장근로 한도가 2018년 개정으로 변경된 경우, RAG 시스템이 개정 전후의 조문을 모두 검색 결과로 반환할 때 “특정 시점에 어떤 규정이 적용되는가“를 판단하는 것은 LLM의 자연어 처리 능력만으로는 안정적으로 수행되기 어렵다.

1) 법령의 시간적 효력: 어떤 법령(및 그 조문·규범명제)이 언제부터(효력발생) 언제까지(효력소멸) 법적 구속력을 가지며, 어느 시점의 사실관계·절차·법률관계에 적용되는지(시간적 적용)를 정하는 규칙의 총합

2) 규범 상충: 같은 사안(동일/중청되는 적용요건)에 대해 동시에 적용 가능한 두 개 이상의 규범명제가 서로 양립 불가능 한 법적 효과(의무/금지/허용/권한 등)을 만들어 내는 것

마. 법령의 온톨로지 도입과 한계

1) 온톨로지 도입 필요성

법령은 자연어로 서술된 규범 텍스트이지만, 그 본질은 단순 문장 집합이 아니라 (i) 구조(위계・편제), (ii) 의미 단위(주체-행위-요건-효과-예외), (iii) 관계(위임・인용・준용・적용제외), (iv) 시간효력(시행・경과・유예・소급), (v) 상충 해결 원리(상위・특별・신법 우선)가 결합되어 작동하는 논리적・관계적 시스템이다. 따라서 공공 서비스(감독・민원・준수점검)에서 법령을 활용하려면 **"관련 조문을 찾는 것"을 넘어, 누가(주체) 어떤 상황에서(요건) 무엇을 해야/하지 말아야 하는지(양태)와 그 결과(효과), 예외・특례, 적용 시점, 우선 적용 규범을 재현 가능하고 설명 가능한 방식으로 결정**할 수 있어야 한다.

그럼에도 법령의 표현과 데이터 저장에 있어 오랫동안 관계형 데이터베이스(이하 RDB) 기반의 문서 중심 저장과 검색에 기반해 발전해 왔고, 최근에는 LLM/RAG, 벡터 검색, 지식그래프가 도입되며 자연어 표현과 처리가 향상되었으나, 각 방법 들은 현재까지도 다음과 같은 구조적 한계가 존재한다.

(표1-1) 공공서비스에서의 법령의 기대 역할

기존 역할		기대 역할
관련 조문을 찾는 것	⇒	누가(주체) 어떤 상황에서(요건) 무엇을 해야/하지 말아야 하는지(양태)와 그 결과(효과), 예외・특례, 적용 시점, 우선 적용 규범을 재현 가능하고 설명 가능한 방식으로 결정

첫째, RDB 기반 저장 방식의 한계이다. RDB는 조문・항・호・목 등 정형 구조를 테이블로 정규화하여 저장하고, 키워드/필드 검색과 이력 관리를 수행하는 데 강점을 가진다. 그러나 법령의 핵심은 테이블 스키마로 고정하기 어려운 다층 관계(참조 유형, 예외/특례의 override, 위임 범위, 상충 원리)와 판단 단위(규범명제(Normative Proposition)[3])의 재구성에 있다. RDB에서는 이러한 관계가 다수의 조인과 응용 로직으로 흩어지고, "적용 판단"은 결국 애플리케이션 코드와 사람의 해석에 의존하게 된다. 특히 개정이 잦은 노동법 영역에서는 스키마・로직의 잦은 변경과 회귀 검증 부담이 커져, 운영 단계에서 일관성과 감사가능성(결정 근거 추적)을 유지하기 어려운 한계가 있다.

둘째, 벡터 DB(검색증강: RAG) 적용의 한계이다. 벡터 검색은 자연어 질의

3) 규범명제(Normative Proposition): 법령 조문에서 도출되는 개별적인 규범적 진술로, 특정 상황(요건)에서 어떠한 권리나 의무(효과)가 발생하는 지를 명시하는 법적 판단의 최소 논리 단위를 의미.

와 유사한 조문·해설·판례를 빠르게 찾아주는 데 탁월하지만, 본질적으로 "유사 텍스트 회수"에 최적화되어 있어 (i) 규범명제의 요건-효과 구조화, (ii) 시점별 유효 규범 선택, (iii) 상충 해결(우선순위 결정), (iv) 예외·특례의 누락 방지, (v) 산식·기한·절차 요건 검증을 스스로 수행하지 못한다. 결과적으로 검색된 근거가 현행인지 연혁인지, 경과규정 적용이 필요한지, 동일 사안에 더 우선하는 특별 규정이 있는지에 대한 판정은 다시 LLM 생성 또는 사용자 해석에 의존하게 되며, 이는 법집행·민원 응답에서 요구되는 재현성과 정확성에 취약점을 남긴다.

셋째, 단순 지식그래프 기반 법령 표현의 한계이다. 법령의 조문 간 참조관계를 추출해 그래프로 연결하거나, 용어 간 연관을 노드·엣지로 표현하는 지식그래프는 법령 네트워크의 가시화와 탐색성 개선에 기여한다. 그러나 많은 단순 KG는 관계를 "연결"하는 데 머물고, 그 관계가 갖는 규범적 의미(위임/준용/적용제외 등 관계의 유형과 효과)와 판단 가능한 단위(규범명제)를 충분히 내재화하지 못한다. 또한 그래프가 "구조적 링크" 중심으로 구축되면, 동일 사안에 적용 가능한 규범들 중 무엇이 최종 적용되는지(상충 해결), 어느 시점의 법이 유효한지(시간효력), 예외가 무엇인지(override) 같은 핵심 질문에 대해 결정 절차와 근거를 산출하는 '추론/검증 계층'이 부재한 경우가 많다.

(표1-2) 법의 디지털 표현 및 저장

기술	강점	법 적용 관점 한계	온톨로지로 보완되는 지점
RDB	편제/메타데이터/이력 관리, 정형 조회	참조 의미·예외·상충 해결·시점 선택이 코드/수작업에 분산	규범 요소/관계를 "의미 단위"로 모델링, 결정 경로 추적
Vector DB/RAG	유사 텍스트 검색·최신 문서 회수	시점별 유효 규범 선택 불안정, 상충·예외· 산식 검증 불가	시점 질의(time-slice), 예외/특례, 우선순위 로직으로 "결정" 수행
지식그래프 (링크 그래프)	참조 네트워크 연결·시각화	관계의 효과/유형·규범명제·검증/추론 계층 부재 시 탐색 도구에 머묾	규범명제(요건→효과)와 규칙/제약을 결합한 규제 지식화
LLM/NLP	질의 해석·요약·설명 생성	재현성/정합성(Temporal)/상충 해결 일관성/검증 취약	온톨로지 기반 결정 결과를 근거로 설명 생성(역할 분담)

넷째, 자연어처리/LLM 중심 표현의 한계이다. LLM은 자연어 이해・요약・설명 생성에 강점이 있지만, 법령이 요구하는 구조적・규칙적 처리를 안정적으로 보장하기 어렵다. 특히 (i) 규범 요소의 엄밀한 분해・결합, (ii) 시행일・경과규정에 따른 시점 정합성, (iii) 특별법・신법 우선 등 상충 해결의 일관성, (iv) 산식・기한・절차 요건의 검증과 감사가능한 근거 제시 측면에서 취약점이 드러난다. RAG를 결합하더라도, 검색이 제공하는 것은 '근거 텍스트'이며, '적용 규범의 결정' 자체는 여전히 불안정할 수 있다.

2) 온톨로지 적용 한계

온톨로지는 위 방법들이 갖는 한계를 극복할 수 있는 방안으로 대두되어 온톨로지를 이용한 법의 표현과 처리를 위한 많은 연구와 개발이 있었다. 온톨로지는 법령을 데이터 표현(구조화)과 지식화(판단・규칙・검증)로 전환하는 핵심 기반으로 (i) 법령・조문・규범요소를 식별 가능한 객체로 만들고, (ii) 참조・위임・준용・예외 등 관계의 의미를 명시화하며, (iii) 규범을 규범명제(요건→효과(규범이 적용되기 위해 필요한 사실・상태・절차・조건이 충족될 때 발생하는 법적 결과(의무/금지/권리/허용/제재 등), 예외 포함)라는 판단 단위로 모델링하고, (iv) 시간효력과 상충 해결을 속성과 규칙(우선순위)로 내재화함으로써, 법령을 "검색 가능한 문서"가 아니라 **"적용 가능한 규제 지식"**으로 만든다. 또한 온톨로지는 **결정 경로를 그래프와 규칙 기반으로 추적 가능하게 하여**, 공공서비스에서 필수적인 재현성・설명가능성・감사가능성을 확보하는 기반이 될 수 있다.

(표1-3) 온톨로지가 법령을 적용 가능한 **규제 지식**으로 만드는 주요 절차

(i)	(ii)	(iii)	(iv)
법령・조문・규범요소를 식별 가능한 객체로 구성	참조・위임・준용・예외 등 관계의 의미를 명시화	규범을 규범명제(요건→효과)라는 판단 단위로 모델링	시간효력과 상충 해결을 속성과 규칙(우선순위)로 내재화

이런 기능적 장점에도 불구하고 그러나 실제 적용 과정에서 법온톨로지는 기대만큼의 실무적 효과를 안정적으로 구현하지 못한 측면이 있으며, 그 핵심 한계는 다음과 같이 정리될 수 있다.

(1) 규범명제 수준의 판단 단위 표현의 부족

많은 법온톨로지 적용은 조문 구조(편제)와 참조관계의 네트워크화에 성과가 있었으나, 실제 판단의 최소 단위인 규범명제(요건→효과, 예외/특례 포함)

를 일급 객체로 모델링하지 못하거나 제한적으로만 다루었다. 이 경우 시스템은 “관련 조문 탐색”은 개선하더라도, 현장에서 요구되는 “해당 사실관계에 무엇이 적용되는지”를 규범 단위로 결정하기 어렵고, 결국 전문가의 해석과 수작업에 재의존하게 된다.

(2) 시간적 효력의 연산적 처리(Temporal Logic) 미흡

법령은 부칙의 시행 시점, 경과규정, 단계적 시행, 유예, 소급 여부 등 시간 조건에 의해 적용 규범이 달라진다. 그럼에도 시간효력을 단순 메타데이터 수준(공포일/시행일 표기)에 머물게 하거나, 시점 t에서 유효한 규범을 선택(time-slice selection)하고 경과규정에 따라 구법/신법을 분기하는 연산을 온톨로지 내부에서 수행하지 못한 경우가 많았다. 결과적으로 법령이 잦게 개정되는 노동법 영역에서는 최신성・정합성・재현성을 확보하기 어렵고, 운영 단계에서 신뢰도가 저하된다.

(3) 규범 상충 해결 메커니즘(Priority) 부재

법령 체계에서는 상위법 우선, 특별법 우선, 신법 우선, 원칙-예외(적용제외) 등 우선순위 원리가 빈번히 적용되며, 이는 판단 결과를 좌우한다. 그러나 온톨로지 적용이 “규범을 연결”하는 수준에 머물러, 동일 사안에 복수 규범이 경합할 때 우선순위(Priority) 기반으로 최종 적용 규범을 결정하고, 그 결정을 설명 가능한 근거(왜 이 규범이 우선인지)로 제시하는 기능이 내재화되지 못한 경우가 많았다.

(4) 실제 서비스와 결합되지 못하고 학술적 시도로 그친 경우가 많았음

법온톨로지는 구축 비용과 유지관리 부담(개정 대응, 정합성 검증, 회귀 테스트, 운영 자동화)을 수반하며, 현장 업무 프로세스와 결합되어야 효과가 발생한다. 그러나 많은 시도는 프로토타입 수준에서 종료되거나, 운영 환경에서 요구되는 성능・갱신・감사 가능성(로그/추적성)을 확보하지 못해 실제 업무의 핵심 의사결정(위반 판단, 조치 의무 안내, 민원 답변 근거 제시)에까지 확장되지 못했다.

요컨대, 법온톨로지는 법을 데이터로 표현하고 지식화하기 위한 유력한 기반임에도, (a) 규범명제 기반 판단 단위의 부재, (b) 시간효력의 연산적 처리 미흡, (c) 상충 해결을 위한 우선순위 메커니즘 부재, 그리고 운영・서비스 결합의 실패로 인해 실무에서 체감 가능한 성과로 연결되지 못한 한계가 존재해 왔다.

본 연구는 이러한 한계를 전제로, 노동법 영역을 대상으로 (i) 규범명제를

중심으로 한 판단 단위 모델링, (ii) 시간효력·경과규정의 연산적 처리, (iii) 우선순위 기반 상충 해결, (iv) 감독·민원·준수점검 시나리오와 결합 가능한 실험 시스템 설계를 통해, 법온톨로지를 "구조화된 정보" 수준을 넘어 "적용 가능한 규제 지식 시스템"으로 확장하는 연구 필요성을 제시한다.

2. 연구 문제의 재정의

본 연구는 법령지식 시스템을 **"문서 검색"** 뿐만 아니라 **"규제판단 및 변화 관리"** 관점에서 재정의한다. 이를 위해 **법령의 이중 구조**를 다음과 같이 설정하여 모델링의 기준 축으로 하였다.

첫째, **정형적 체계(structured system) 축**이다. 근로기준법, 근로기준법 시행령, 근로기준법 시행규칙의 삼층 위계와 편, 장, 절, 조, 항, 호, 목의 체계는 법령의 **물리적 구조이자 근거 추적성과 변경 관리의 기반**이다. 본 연구는 이 정형 구조를 유지하되, **참조관계(인용, 준용, 위임, 근거)를 유형화**하여 **개정 영향도의 전파 경로를 분석 가능한 형태로 모델링**한다.

둘째, **규범명제(normative proposition) 축**이다. 실제 법 적용 판단은 조문 번호가 아니라 **"특정 요건 하에서 특정 법적 효과가 발생한다"는 규범명제 단위로 수행**된다. 예를 들어, 근로기준법 제56조 제1항은 "사용자는 연장근로에 대하여 통상임금의 100분의 50 이상을 가산하여 근로자에게 지급하여야 한다"는 규범명제를 포함한다. 이 규범명제는 주체(사용자), 대상(근로자), 요건(연장근로의 발생), 법적 효과(가산임금 지급 의무), 산식(통상임금의 50% 이상)으로 구성된다. 본 연구는 이러한 규범명제를 일급 객체(first-class object)로 정의하고, 규범명제를 구성하는 주체, 대상, 행위, 요건, 법적효과, 예외, 특례, 준용을 구조화하여 사실관계와의 매칭을 통해 판단을 지원하는 체계를 제안한다.

이 이중 구조가 실무적 효과를 가지기 위해서는 본 연구는 추가로 두 가지 로직을 온톨로지 내부에서 명시하였다.

첫째, **시간적 효력(temporal validity)의 모델링**이다. 법령의 각 조항과 규범명제는 시행일(effective date), 효력 종료일, 경과규정 조건을 속성으로 가지며, 질의 시점 t를 입력으로 "해당 시점에 적용되는 규범 선택(select applicable norm at time t)"이 가능해야 한다.

시간적 효력 모델 적용 예시
근로기준법 제53조의 **연장근로 한도**는 **2018년 개정 전후**로 내용이 다르며, **사업장 규모**에

따라 **시행 시점이 단계적으로 적용**되었다. **시간적 효력 모델은 이러한 복잡한 경과규정을 처리**할 수 있어야 한다.

둘째, **규범 상충 해결을 위한 우선순위 로직**이다. 법의 우선순위는 상위법 우선(근로기준법 > 시행령 > 시행규칙), 특별법 우선, 신법 우선 원칙과 원칙-예외 관계를 우선순위 속성 및 충돌 규칙으로 명시하여, **복수의 규범명제가 경합할 때 최종 적용 규범을 결정하고 설명할 수 있어야 한다**. 예를 들어, 근로기준법 제63조는 제4장(근로시간과 휴식)의 규정 적용을 배제하는 예외 조항으로서, 감시·단속적 근로자에 대해서는 제50조 내지 제56조의 일반 규정보다 우선하여 적용된다.

3. 연구 목적 및 기대효과

가. 연구 목적 및 범위

본 연구의 목적은 근로기준법, 근로기준법 시행령, 근로기준법 시행규칙을 대상으로, **법령의 정형적 체계와 규범명제를 결합한 법령온톨로지 모델을 구축하고, 시간적 효력 및 버전 관리 로직과 우선순위 기반 규범 상충 해결 메커니즘을 내장함으로써, LLM 및 RAG 단독 접근의 한계를 보완하는 설명가능하고 운영 가능한 법지식 시스템 프레임워크를 제시**하는 데 있다.

연구의 범위는 근로기준법 중에서도 특히 근로시간, 휴게, 휴일, 연장근로, 야간근로, 휴일근로 및 이에 따른 가산임금에 관한 규정(제4장 근로시간과 휴식, 제50조 내지 제63조)을 중심으로 한다. 이 영역은 규범명제의 구조(요건-효과-예외)가 명확하고, 시간적 효력의 변화(2018년 근로시간 단축 개정 등)가 있으며, 규범 상충(일반규정과 적용제외, 원칙과 특례)이 빈번하게 발생하여 본 연구가 제안하는 모델의 효과를 검증하기에 적합하다.

아울러 본 연구는 제안한 모델이 단순한 개념 제시에 머물지 않고 실제로 규범명제 기반 판단 기능을 수행할 수 있음을 검증하기 위해, 온톨로지, 규칙, 질의 계층을 결합한 실험 시스템(프로토타입)을 구현하였다. 본 프로토타입은 근로기준법 영역의 규범을 규범명제(요건, 효과, 예외) 단위로 구조화한 뒤, 이를 실제 사실관계와 매칭하여 **"적용 규범 선택 및 판단 결과"**를 도출하도록 설계되었다.

구체적으로 시스템은 다음의 네 가지 핵심 기능을 중심으로 구성하였다. 첫째, **시점 기반 적용 규범 선택**(Temporal Query)으로서, 질의 시점 t에 유효한 규범 버전을 선택한다. 둘째, **원칙-예외-특례 처리**(Override Handling)로서, 적

용제외 조항(제63조 등)과 특례 규정의 우선 적용을 처리한다. 셋째, 규범 상충 해결(Conflict Resolution)로서, 상위법 우선, 특별법 우선, 신법 우선 원칙에 따른 최종 적용 규범을 결정한다. 넷째, 요건 및 산식 검증(Compliance Validation)으로서, 근로시간 한도, 가산임금률 등 수치 조건의 충족 여부를 검증한다.

프로토타입의 처리 흐름은 다음과 같다. 먼저 사용자가 입력한 사실관계(예: 행위 시점, 사업장 규모 및 업종, 주간 근로시간, 연장근로시간, 야간근로 여부 등)와 질의 의도("연장근로 한도 위반 여부", "가산임금 지급 의무", "적용제외 해당 여부")를 수집한다(Input 단계). 다음으로 온톨로지 상에서 정형 구조(근로기준법-시행령-시행규칙, 조문 체계, 참조관계) 및 규범명제 그래프를 조회하여 적용 후보 규범명제를 구성한다(Process-1 단계). 이후 SWRL 규칙을 활용해 사실관계가 충족하는 요건 패턴에 따라 규범명제의 적용 후보를 분류하고, SPARQL 질의를 통해 특정 시점 t에서 유효한 버전의 규범을 선택하며, 원칙-예외 관계에 따른 덮어쓰기(override) 결과를 반영하고, 충돌 해결 원리에 기반한 우선순위 로직을 적용하여 최종 적용 규범을 결정한다(Process-2 단계). 마지막으로 근로시간 한도, 가산임금률 등 수치 조건에 대해 SPARQL 기반 검증 규칙을 통해 충족 여부를 점검하고, 최종적으로 적용되는 규범명제, 결론(의무, 금지, 허용 등 법적 효과), 적용 시점 및 버전 근거, 예외・특례 적용 근거, 검증 결과(충족/미충족)와 근거 조문 링크를 함께 출력한다(Output 단계).

이때 LLM은 "판정의 주체"가 아니라 보조적 역할로 한정하였다. 구체적으로 LLM은 규범명제 후보 추출 및 용어 정규화, 자연어 질의 해석과 사실관계 구조화 보조, 시스템이 산출한 적용 규범 및 근거와 검증 결과를 이용자 관점에서 설명 가능한 형태로 재구성하는 역할을 수행한다. 반면 시점 선택, 상충 해결, 예외 처리, 산식 검증 등 판정의 핵심 연산은 온톨로지와 규칙(SWRL), 질의(SPARQL), 검증 규칙으로 수행함으로써, LLM 및 RAG 단독 접근에서 문제로 지적되는 할루시네이션, 시점 정합성 오류, 상충 규범 처리 불안정, 재현성 부족을 구조적으로 보완하도록 설계하였다.

이러한 역할 분담의 근거는 다음과 같다. 첫째, 시간적 효력 판단은 시행일, 경과규정 조건 등 명시적 날짜 비교와 조건 평가를 요구하는데, LLM은 날짜 계산과 조건 분기에서 오류를 발생시키는 경우가 보고되어 왔다. 둘째, 규범 상충 해결은 법적용 원칙(상위법 우선, 특별법 우선)의 일관된 적용을 요구하는데, LLM은 맥락에 따라 상이한 원칙을 적용하거나 원칙 자체를 무시하

는 비일관성을 보일 수 있다. 셋째, 수치 검증(근로시간 한도, 가산임금률)은 정확한 산술 연산을 요구하는데, LLM은 산술 오류를 발생시키는 경우가 있다. 따라서 이러한 핵심 연산은 규칙 기반 시스템에서 수행하고, LLM은 자연어 처리가 강점인 영역—질의 해석, 용어 정규화, 설명 생성—에 활용하는 것이 합리적이다.

나. 연구의 기대효과

본 연구의 구체적인 기대효과는 다음과 같다.

첫째, 정형 구조와 규범명제의 이중 구조를 갖는 근로기준법 온톨로지 스키마를 제안한다. 근로기준법, 시행령, 시행규칙의 삼층 위계와 조문 편제를 표현하는 정형 구조 계층과, 요건-효과-예외로 구성되는 규범명제 계층을 통합적으로 모델링한다.

둘째, 부칙과 경과규정을 포함하는 시간적 효력 모델과 시점 기반 적용 규범 선택 방법을 제시한다. 각 규범명제에 validFrom, transitionCondition, validTo, 속성을 부여하고, 질의 시점과 사실관계의 발생 시점을 입력으로 적용 가능한 규범 버전을 선택하는 알고리즘을 제안한다.

셋째, 상위법 우선, 특별법 우선, 신법 우선 및 원칙-예외 관계를 반영한 우선순위 기반 규범 상충 해결 메커니즘을 설계한다. 규범명제 간의 override, exception, specialization 관계를 온톨로지에 명시하고, 충돌 발생 시 우선순위 규칙에 따라 최종 적용 규범을 결정하는 로직을 구현한다.

넷째, LLM은 “추출, 질의해석, 설명 생성“ 역할로 제한하고, 핵심 판단은 온톨로지 기반 검증 및 우선순위 로직으로 수행하는 결합 아키텍처를 제시한다. 이를 통해 LLM의 자연어 처리 강점과 규칙 기반 시스템의 정확성 및 일관성을 결합한다.

다섯째, 개정 영향도 분석, 시점별 규범 적용, 상충 해결, 준수 진단 등 공공행정 과업에 대한 실증 평가 프레임을 제안한다. 평가 지표로는 정확도(accuracy), 일관성(consistency), 설명가능성(explainability), 시점 정합성(temporal correctness)을 설정하고, 비교 대상으로 LLM 단독 접근, RAG 기반 접근, 본 연구의 온톨로지 기반 접근을 설정하여 비교 평가를 수행한다.

여섯째, 규범명제 기반 법령온톨로지가 실제 판단 단위로 기능하도록 SWRL(적용 후보 분류), SPARQL(시점 선택, 우선순위 및 예외 반영, 결과 생성), 검증 규칙(근로시간 한도, 가산임금률 등 수치 조건 점검)을 결합한 실험 시스템을 구현하고, 이를 통해 LLM 및 RAG 단독 접근의 한계를 보완할 수 있음을 실증적으로 평가한다.

제2절. 소결 및 시사점

1. 소결

1장은 법이 사회의 보편적 규범이자 공공 지식임에도 불구하고, 현실의 법령 정보서비스가 여전히 **"접근 가능하지만 이해·적용이 어려운 정보"** 수준에 머무르고 있다는 문제의식을 제시한다. 특히 법령은 단순 텍스트가 아니라 조문 구조, 예외·특례, 위임·참조, 상·하위 법령 관계, 시간적 효력(시행·경과·적용례) 등이 결합된 복합 체계이므로, 키워드 검색과 조문 열람만으로는 사용자의 구체적 상황에 적용되는 규범을 식별하기 어렵다는 점을 명확히 정리하고 있다.

또한 1장은 기존 법령 정보화 방식(RDB 중심 저장·검색)과 최근의 기술적 대안(RAG/벡터검색, 단순 지식그래프, LLM 기반 질의응답)의 강점과 한계를 비교하면서, 이들이 공통적으로 **규범명제(요건→효과) 단위의 판단, 시간적 효력의 연산적 처리, 규범 상충 해결, 감사가능한 근거 제시**를 안정적으로 수행하기 어렵다는 점을 지적한다. 이를 통해 본 연구가 단순 검색 고도화가 아니라, 법령을 "적용 가능한 규제 지식"으로 재구성하는 문제를 다룬다는 점을 분명히 한다.

본 연구의 핵심 기여는 연구 문제를 **문서 검색 중심이 아니라 규제판단 및 변화관리 중심으로 재정의**한 데 있다. 이를 위해 법령의 이중 구조(① 정형적 체계: 법령 위계·편제·참조관계, ② 규범명제 축: 주체·행위·요건·효과·예외)를 모델링 기준으로 설정하고, 여기에 시간적 효력(시점 기반 적용 규범 선택)과 우선순위 기반 상충 해결(상위법/특별법/신법 우선, 원칙-예외)을 온톨로지 내부 로직으로 명시해야 한다는 방향을 제시한다.

아울러 연구 범위를 근로기준법(특히 근로시간·휴게·휴일·연장근로·가산임금 관련 조문) 중심으로 설정하고, 온톨로지-규칙(SWRL)-질의(SPARQL)-검증 규칙을 결합한 프로토타입 시스템 구상을 제시함으로써, 본 연구가 이론적 제안에 머무르지 않고 실제 판단 기능(적용 규범 선택, 예외 처리, 상충 해결, 수치 검증)까지 검증하려는 실증 연구임을 분명히 했다. 또한 LLM의 역할을 '판정 주체'가 아니라 **질의해석·용어정규화·설명생성 보조**로 제한하여, 정확성과 설명가능성을 동시에 확보하는 역할 분담 원칙을 제시한 점도 중요한 정리이다.

2. 시사점

1장은 법첫째, 법령 정보화의 다음 단계는 **"검색 가능한 법령"** 이 아니라 **"적용 가능한 법지식"**으로의 전환이라는 점이 분명해졌다. 즉, 향후 법률정보 시스템은 조문 조회 중심 구조에서 벗어나, 규범명제 단위(요건-효과-예외)로 법을 표현하고 사실관계와 매칭할 수 있는 형태로 발전해야 한다.

둘째, LLM/RAG의 도입은 필요하지만, 공공 법률행정 영역에서는 단독 활용보다 **온톨로지·규칙 기반 판단 계층과의 결합**이 전제되어야 한다는 점이 확인된다. 특히 시점 정합성, 상충 해결, 산식 검증처럼 오류 허용도가 낮은 영역은 규칙 기반으로 처리하고, LLM은 질의해석·설명생성에 집중시키는 구조가 실무적으로 타당하다.

셋째, 법령 온톨로지의 실무적 성공 여부는 단순 스키마 구축이 아니라, **시간효력·우선순위·예외처리 로직**을 얼마나 연산 가능하게 내재화하느냐에 달려 있다. 제1장에서 제시한 문제 재정의는 이후 장의 모델링(3장)과 검증(4장)을 평가하는 기준이 되며, 본 연구의 차별성을 설명하는 핵심 축으로 활용될 수 있다.

넷째, 본 장에서 제시한 연구 범위 설정(근로기준법 제4장 중심)은 적절하다. 근로시간·연장근로·가산임금 영역은 요건·효과·예외 구조가 분명하고 수치 검증 요소가 많아, **규범명제 기반 모델과 추론 체계의 유효성을 검증**하기에 적합한 대표 사례로 기능한다. 향후에는 이를 다른 노동법령(고용보험법, 산업안전보건법 등)으로 확장해 일반화 가능성을 검증하는 방향이 필요하다.

다섯째, 제1장은 후속 장의 평가 프레임(정확도, 일관성, 설명가능성, 시점 정합성)을 이미 제시하고 있어, 향후 실증 결과를 단순 데모가 아닌 비교 가능한 성능평가로 연결할 수 있는 장점을 갖는다. 이는 최종 보고서에서 연구의 정책적·실무적 설득력을 높이는 중요한 기반이 된다.

제2장 기존 연구 동향 및 분석

제1절. 법령의 구조 및 특징

법령 온톨로지를 설계·구축하기 위해서는, 먼저 법령이 어떤 방식으로 구성되고(구조), 어떤 의미 요소를 규범을 표현하며(내용), 어떤 관계 속에서 작동하는지 (관계·규범성)를 체계적으로 이해할 필요가 있다. 법령은 일반 문서와 달리 조·항·호·목의 계층 구조, 부칙·별표·서식 등 부속 구성요소, 그리고 의무·금지·허용과 같은 규범적 강제력(양상성), 조건부 적용, 예외·특례·준용과 같은 수정관계가 결합되어 하나의 규칙 체계를 이룬다. 이러한 특성을 고려하지 않으면 조문을 단순 텍스트로 저장하는 수준에 머물러, “누가 무엇을 어떤 조건에서 해야 하는가”를 기계적으로 표현·추론하기 어렵다.

따라서 본 절에서는 법령의 구조 및 특징을 정리함으로써, 이후 장에서 제시하는 규범명제 기반 온톨로지 설계(개체·속성·관계 정의)와 지식 추출·추론 범위가 어떤 근거 위에서 도출되는지 명확히 하고자 한다. 이는 기존 연구들이 주로 문서 구조화, 개념 그래프, 학습 기반 질의응답 등 일부 요소에 집중해 온 한계를 보완하고, 본 연구의 방법론적 차별성을 논리적으로 연결하는 기초가 된다.

1. 법(법령)의 체계와 구조

가. 법의 위계 구조

우리 법규범은 일반적으로 헌법을 최상위 규범으로 하여, 그 아래에 국회가 제정하는 법령과 그 하위규범인 대통령령(시행령), 총리령・부령(시행규칙) 및 국제조약, 명령, 규칙, 조례, 자치규칙, 국회규칙 등 각종의 법 형식을 총칭하는 의미로 쓰인다.

대통령령은 시행령, 총리령과 부령은 시행규칙이라고도 한다. 시행령은 법률의 하위에 있으면서 실제 법률의 시행을 위한 세부적인 사항을 기술하며, 시행규칙은 시행령을 기반으로 더욱 구체적인 규칙들을 지정하고, 총리령은 총리에 의한 명령, 부령은 각 부처별 장에 의한 명령을 토대로 지정되는 규칙을 의미한다. 따라서 시행령과 시행규칙은 법률 아래에서 서로 밀접한 관계를

맺고 있다. 그 외, 각급 행정기관의 훈련, 예규, 고시 및 자치법규도 법령의 하위에 속하는 구조이다.

(표2-1) 법체계의 단계와 유형

단계	구분	세부 유형	제정/발령 주체	법적 성격·효력(요지)	예시/비고
1	헌법	헌법	국민 (국민투표)	최상위 규범(모든 법령의 근거·한계)	대한민국헌법
2	법령	법률	국회 의결 + 대통령 공포	일반적으로 국민·국가기관을 직접 구속(상위법)	근로기준법, 고용보험법 등
		시행령 (대통령령)	대통령	법률의 위임 또는 집행을 위해 제정(법률 하위)	○○법 시행령
		시행규칙 (총리령·부령)	국무총리 /각 부처 장관	시행령·법률 집행을 위한 세부사항 규정(법률·시행령 하위)	○○법 시행규칙
3	행정규칙	훈령	행정기관 (기관장)	원칙적으로 행정조직 내부 기준(대외적 구속력은 제한적)	업무처리 기준, 내부 지침
		예규	행정기관 (기관장)	원칙적으로 내부 기준·해석지침 성격(대외 효력 제한)	법령해석 지침, 처리예규
		고시	행정기관 (장관 등)	일반적으로는 행정규칙이나, 법률/명령의 위임에 따라 국민을 구속하는 “법규적 고시”가 될 수 있음	기준·요건·고시문(위임 여부가 핵심)
4	자치법규	조례	지방의회	해당 지방자치단체 구역 내에서 효력(상위법 범위 내)	○○시 조례
		규칙	지방자치 단체장	조례의 위임 또는 집행을 위한 세부 규정(조례 하위)	○○도 규칙

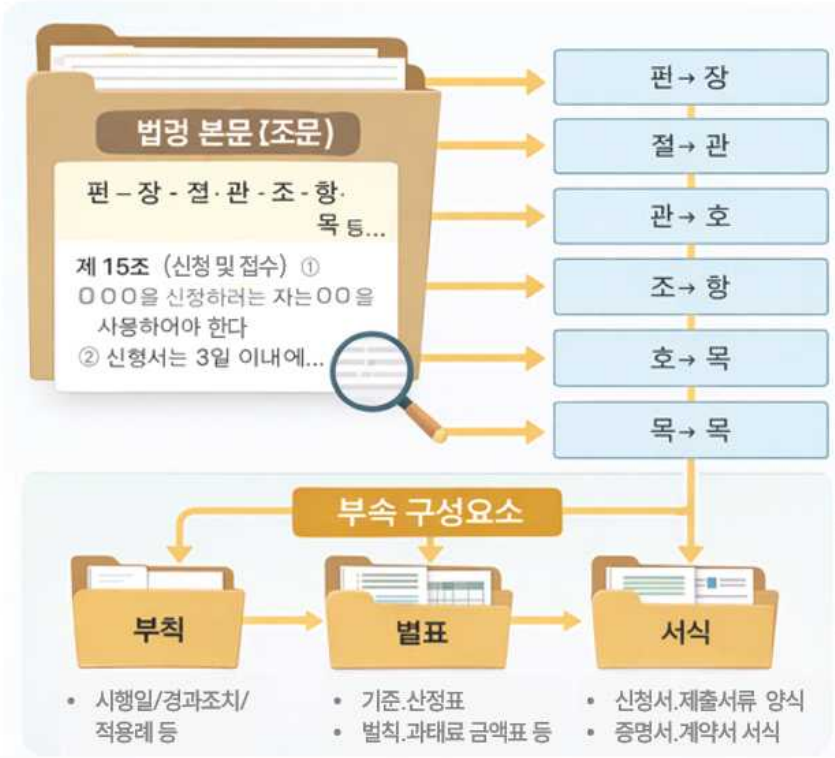
법체계의 단계적 구조는 상위규범이 기본 원칙・권리의무의 골격을 정하고, 하위 법규범이 그 범위 안에서 세부 기준과 절차를 구체화하는 구조로 운영된다. 이 위계 구조는 어떤 규정이 다른 규정을 참조하거나 위임할 때, 그 관계가 적법한지(위임 근거・범위・저촉 여부)와 어떤 규범이 우선 적용되는지(상위규범 우선)를 판단하는 전제가 된다.

나. 법조문의 구조

일반적으로 본문은 “편-장-절-관-조-항-호-목”과 같은 단위로 세분화되며, 상위 단위는 규범의 적용 범위를 개략적으로 설정하고 하위 단위로 갈수록 구체적 행위・요건・예외가 기술되는 경향이 있다. 또한 법령은 본문(본칙) 외

에 부칙, 별표, 서식 등을 포함하는데, 이는 단순한 부록이 아니라 시행과 집행을 구체화하는 실질 규범으로 기능한다. 특히 별표·서식은 기준표/산정표/제출서류 목록/신청서 양식 등 "데이터 테이블 및 입력 스키마" 로서 행정 집행과 직접 연결된다.

온톨로지화를 위해 조문 텍스트의 보존과 함께 (a) 조문 구조의 기계적 파싱, (b) 별표(테이블)·서식(필드)의 스키마화, (c) 조문↔별표/서식 참조 연결이 동시에 필요하다.



(그림2-1) 법문서의 구조와 부속 구성요소

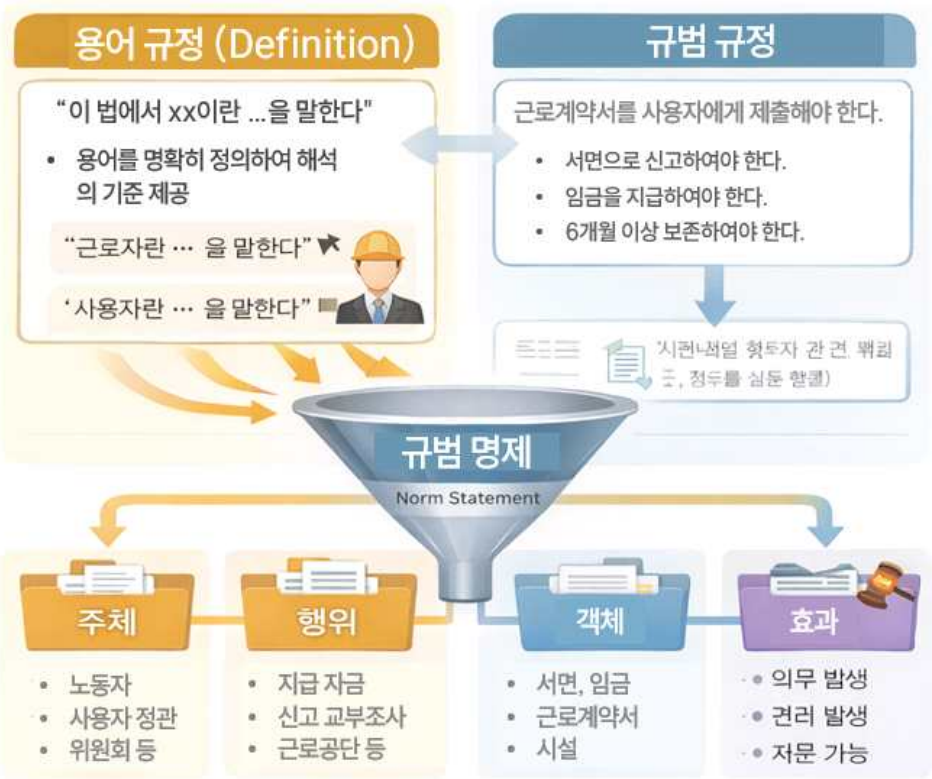
2. 법(법령)의 내용·규범·관계적 특성과 지식화

가. 법의 내용적 특징: 개념(용어)과 사실관계 구성요소의 결합

법령의 내용은 크게 정의(용어 규정)와 규범(행위 규율)로 구성된다. 정의 규정은 법령 내부의 개념을 고정하여 해석의 기준을 제공한다. 규범 규정은 특정 주체가 일정 조건에서 어떤 행위를 해야/하지 말아야/할 수 있는지를 정한다. 이때 규범은 보통 (i) 주체, (ii) 행위, (iii) 객체, (iv) 요건(조건), (v) 효과로 구성되며, 노동·고용 분야에서는 사업장 규모, 기간, 절차적 전제(신고·교부·보존 등)와 같은 실무형 조건이 빈번하게 등장한다.

온톨로지화를 위해서는 명사 추출을 넘어 "주체–행위–객체–요건–효과"의

의미역할을 분해하고 정규화해야 하며, 이 의미 단위가 본 연구의 규범명제로 연결된다.



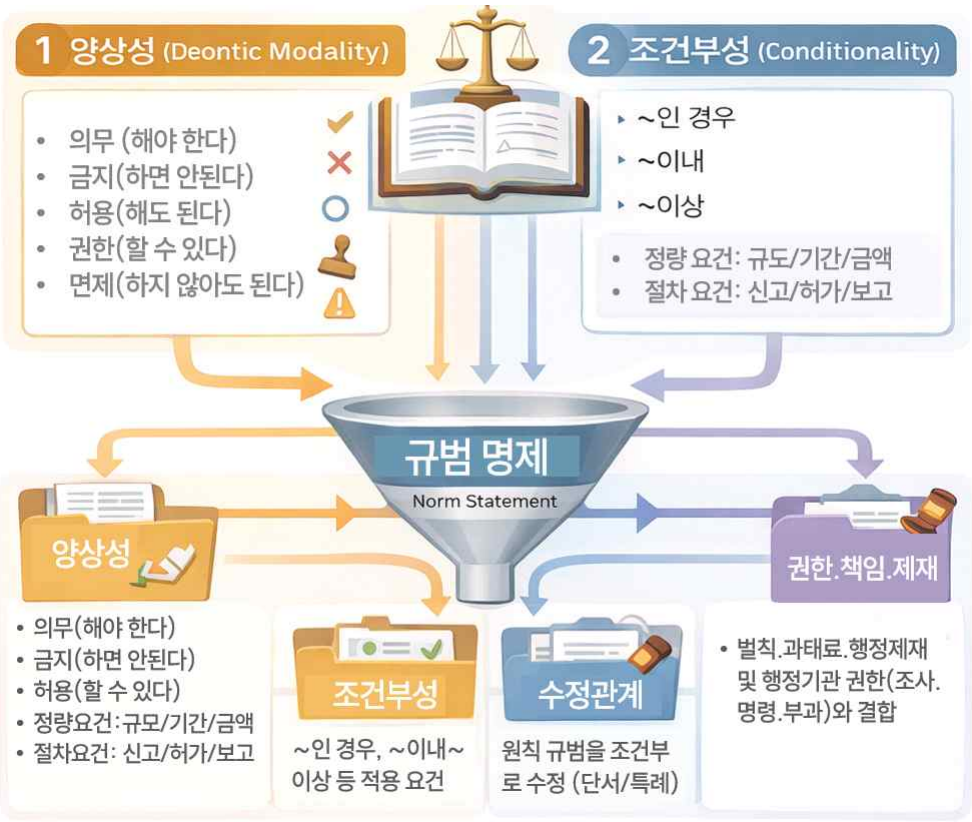
(그림2-2) 법문서의 구조와 부속 구성요소

나. 규범적 특징: 양상성, 조건부성, 수정관계 및 제재의 결합

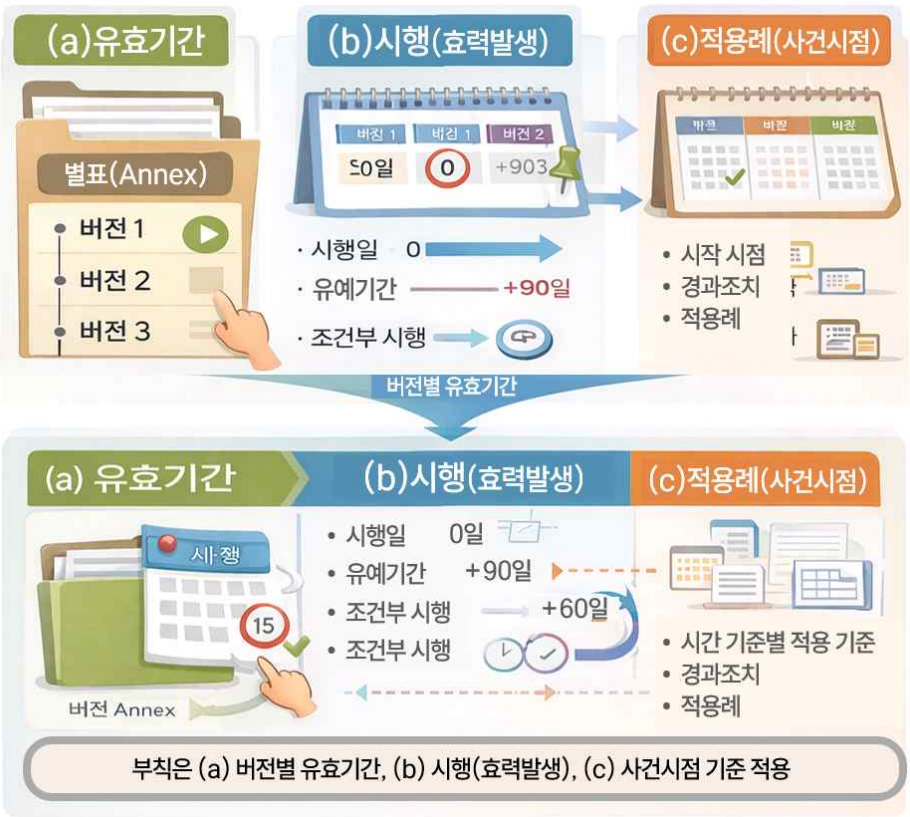
법령 문장은 단순 서술이 아니라 규범적 강제력을 가진다. 특히 다음 요소는 규범 지식화를 위해 반드시 구조화되어야 한다.

- 양상성(Deontic modality): 의무/금지/허용/권한/면제 등 규범력의 강도와 형태
- 조건부성(Conditionality): "~인 경우", "~이내", "~이상" 등 적용 요건(정량·절차 요건 포함)
- 수정관계(예외·특례·준용): 원칙 규범을 조건부로 수정하는 관계(단서/특례/준용)
- 권한·책임·제재 연결: 의무 위반 시 벌칙·과태료·행정제재 및 행정기관 권한(조사·명령·부과)과의 결합

온톨로지화를 위해서는 예외/특례/준용은 단순 태그가 아니라 "무엇을(원칙 규범) 어떤 조건에서 어떻게 수정하는지"를 나타내는 관계(modifies)로 모델링해야 한다.



(그림2-3) 법문서의 규범적 특징(AI 생성 후 보완 그림)



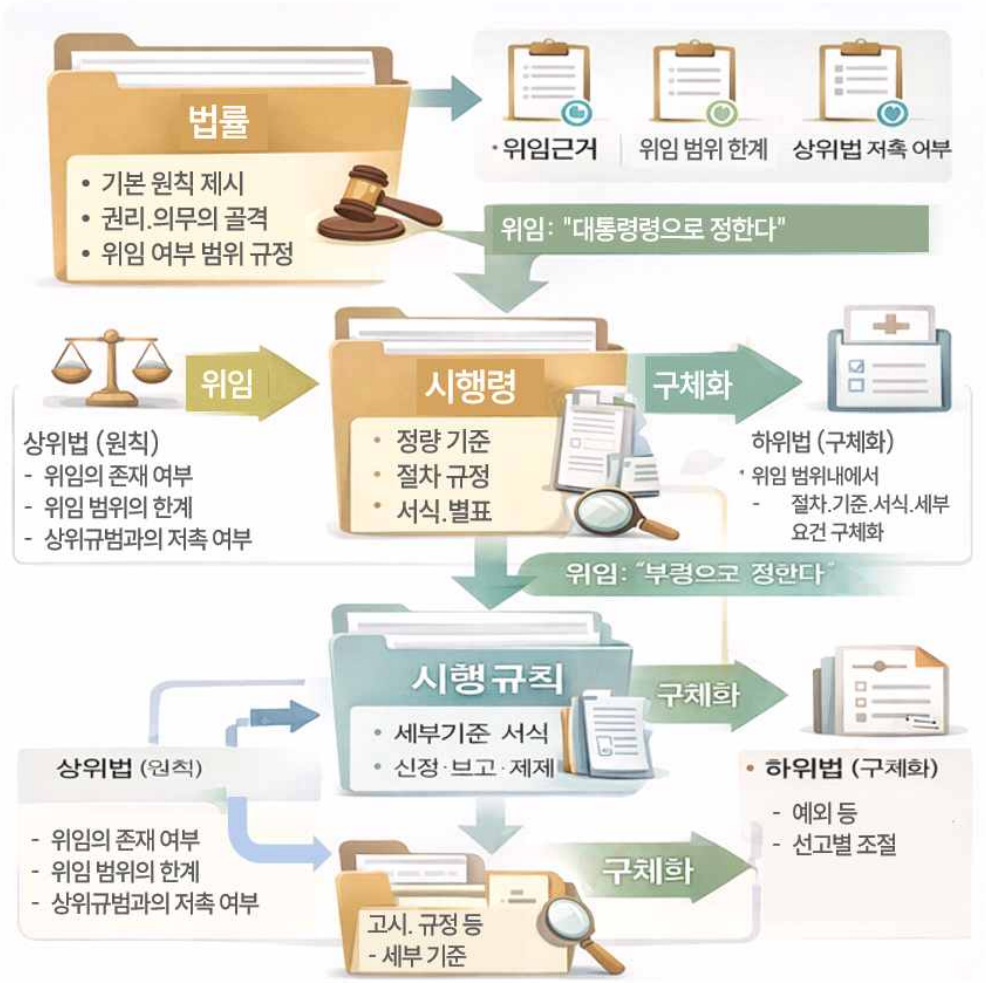
(그림2-4) 법문서의 시간적 효력과 적용(AI 생성 후 보완 그림)

다. 시간적 효력과 적용: 시행·경과·적용례에 따른 규범 운영

법령은 "언제부터 효력이 발생하는지(시행)"와 "어떤 사건에 적용되는지(적용)"가 분리되어 작동한다. 부칙은 시행일・유예・조건부 시행과 함께 경과조치・적용례를 통해 시행 전후 사건(진행 중 계약・절차・처분 등)의 처리 기준을 규정한다. 따라서 시간 정보는 단일 날짜가 아니라 (a) 버전별 유효기간, (b) 시행(효력발생), (c) 사건시점 기준 적용 규칙을 분리해 표현할 필요가 있다.

라. 관계적 특징: 법은 '단독 규범'이 아니라 '법체계 네트워크'로 작동

법령의 핵심 특성 중 하나는 개별 조문이 독립적으로 존재하지 않고, 상.하위 규범과 다수의 연관 규범 속에서 의미와 효력을 획득한다는 점이다. 법체계는 다음과 같은 관계를 통해 네트워크 구조를 형성한다.



(그림2-5) 법문서의 상.하위법 관계 (AI 생성 후 보완 그림)

1) 상・하위법 관계 (규범 위계)

법령은 일반적으로 법률-시행령-시행규칙(및 위임과 고시・규정 등 세부기준)과 같은 위계 구조를 갖는다. 상위법은 기본 원칙과 권리・의무의 골격을 제공하고, 하위법은 위임 범위 내에서 절차・기준・서식・세부요건을 구체화한다. 이 위계는 (i) 위임의 존재 여부, (ii) 위임 범위의 한계, (iii) 상위 규범과의 저촉 여부를 판단하는 기준이 된다.

온톨로지화를 위해서는 위계, 위임근거, 구체화, 저촉과 같은 관계・속성이 필요하다.

2) 위임・구체화 관계

상위법이 "대통령령/부령으로 정한다"와 같이 위임하면, 하위법은 그 위임을 근거로 정량기준(규모/기간/횟수), 절차 흐름, 제출서식, 별표 기준표 등을 설계한다. 노동분야에서 특히 자주 나타나는 패턴은 "법률은 의무를 선언하고, 시행령·시행규칙/별표·서식이 실무 데이터를 규정"하는 형태다.

온톨로지화를 위해서는 위임 문장을 단순 참조가 아니라 위임유형(기준/절차/서식/제재/산정)으로 유형화하여, 하위법 규정이 무엇을 구체화하는지 추적 가능해야 한다.

3) 연관법 관계(관련 법령 간 결합)

노동・고용 영역의 규범은 단일 법률로 완결되지 않고, 예컨대 근로기준, 고용보험, 산업안전, 직업훈련, 개인정보/기록관리 등과 사실관계가 겹치는 다법령 연계로 구성되는 경우가 많다. 이는 동일 행위(예: 서류 보존, 신고·보고, 안전조치)가 여러 법령에서 서로 다른 요건·효과로 규율될 수 있음을 의미한다.

온톨로지화를 위해서는 단순 "관련 있음"을 넘어서, 연관법을 "공유주어/공유술어/공유목적(Subject-Predicate-Object)"관계로 연결하고, 관계 유형을 "보완, 제한, 전제" 등으로 구분해야 실무 판단이 가능해진다.

4) 참조・인용・준용 관계(Intertextual links)

법령은 조문 내부에서 다른 법령 또는 다른 조문을 인용·참조·준용하는 방식으로 법체계의 정합성을 유지한다.

- 인용/참조: 근거를 제시하거나 정의·요건을 불러오는 연결
- 준용: 다른 규정을 일부 변경해 적용(실질적으로 규범의 복제+변형 적용)
- 특례/예외: 일반 규정을 특정 상황에서 수정하는 연결(상충 해결과 직결)

온톨로지화를 위해서는 참조는 단순 링크가 아니라 참조유형(인용/근거/위임/준용/특례/예외/적용배제)로 유형화되어야 하며, 특히 예외·특례·준용은 원칙규범명제로 귀속시켜야 한다.

5) 상충과 우선순위(체계정합성 유지 메커니즘)

다법령·다규범이 동시에 적용되는 상황에서는 규범 간 상충이 발생할 수 있다. 이때 법체계는 일반적으로 (i)상위규범 우선, (ii)특별규정 우선, (iii) 후법 우선과 같은 원리를 통해 정합성을 유지한다. 노동 현장에서는 "일반 규정과 특례", "구법과 신법(경과조치)", "법률 규정과 하위법령의 구체 기준" 사이의 충돌이 대표적으로 나타날 수 있다.

온톨로지화를 위해서는 상충을 1급 객체로 두고, 해결원리, 우선규범, 근거를 기록해야 설명 가능한 추론이 가능하다.

(표2-2) 법의 연관성 특징 요약

구분	관계 유형	정의/의미	연결 대상 (범위)	핵심 구성요소 (무엇을 표현?)	전형적 표현 (문장 패턴)	노동.고용 분야 예시
1	상·하위법 관계 (규범 위계)	법규범이 위계 구조를 이루며 상위 규범이 원칙·골격, 하위 규범이 세부를 규율	. 헌법–법률–시행령–시행규칙(위임 시 고시) . 조례–규칙	.위계(상위/하위), .적용 우선순위의 기본 전제	. "법률에 따라", . "이 법에서 정한 바에 따라", . (간접적) "대통령령/부령으로 정한다"	법률(근로기준법)이 의무의 골격을 정하고, 시행령·시행규칙이 적용 범위·절차·서식을 구체화
2	위임·구체화 관계	상위 규범이 하위 규범에게 세부사항 규율 권한을 부여 (위임)하고 하위 규범이 이를 구체화	. 법률→시행령/시행규칙/고시, . 시행령→시행규칙, . 조례→규칙	. 위임 근거, . 위임 범위/한계 . 구체화 대상 (기준/절차/서식 등)	. "대통령령으로 정한다" ."부령으로 정한다" . "고시로 정한다" . "별표로 정한다"	법률: "세부 기준은 대통령령으로 정한다" 시행령: "상시근로자 ○명 이상" 등 정량기준 제시
3	연관법 관계(관련 법령 결합)	동일 사실관계가 여러 법령에 의해 중첩 규율되어, 다 법령을 함께 봐야 적용 판단 가능	. 서로 다른 법률/명령 간(근로기준–고용보험–산안법–직업훈련–개인정보 등)	. 공유 주체/행위/객체 (SPO), . 관계 성격(보완/제한/전제)	"~에 따라야 한다"가 다수 법령에서 중첩 등장, 동일 객체(자료/기록/안전조치 등) 반복	"서류 보존"은 근로기준(근로관계문서) + 개인정보(처리/보관 제한) + 기록물 관리 규정과 함께 고려
4	참조·인용·준용 관계 (Intertextual links)	조문이 다른 조문/법령/별표/서식을 불러와 정합성과 재사용성을 확보(준용은 변형 적용)	. 동일 법령 내 조문 간, . 법령 간, . 조문↔별표/서식	참조유형(근거/정의/요건/절차/준용/예외/특례/적용배제), 참조 범위	. 인용/참조: "~에 따른다", "~에 의한다", "~을 준수한다" . 준용: "제○조를 준용한다"	. 준용: "제○조를 준용한다" . 예외: "다만 ○○한 경우에는 ..." 별표의 과태료 기준표 참조

구분	관계 유형	정의/의미	연결 대상 (범위)	핵심 구성요소 (무엇을 표현?)	전형적 표현 (문장 패턴)	노동.고용 분야 예시
					. 예외/특례: "다만 ...", "~에도 불구하고 ..."	
5	상충과 우선순위 (체계정합성 메커니즘)	다규범이 동시에 적용될 때 충돌을 해결하기 위한 원리(위계·특별·후법 등)	동일 사안에 적용 가능한 규범명제들 (원칙 vs 특례, 구법 vs 신법, 상위 vs 하위)	. 상충 유형(의무 vs 금지 등), . 해결 원리(우선 순위 규칙), . 승패 규범, 근거	. 충돌 해결 단서: "특례", "예외", "경과조치/적용례", "~에도 불구하고"	일반 규정의 의무 vs 특례 규정의 완화/면제, 개정 전후 규정 충돌(부칙 경과조치로 해결)

제2절. 법률의 지식 표현 기술

법률은 사회의 규범적 질서를 규정하는 규범 명제(normative proposition)의 집합체로서, 단순한 텍스트 정보(information)가 아니라 규범을 통해 사실관계를 평가하고 법적 효과를 결정하는 전문적 지식의 성격을 갖는다. 따라서 법률을 컴퓨터로 처리하기 위해서는 조문을 "조·항·호"와 같은 형식적 구조로만 분절·저장하는 수준을 넘어, 의무·권리·금지·허가 등 규범적 개념이 서로 얽혀 형성하는 요건-효과 구조, 예외·특례, 우선순위, 시간적 효력(시행·경과·소급), 상위·하위 규범 간 충돌과 같은 핵심 특성을 함께 표현할 수 있어야 한다.

결국 법률지식의 전산 처리는 법률의 근본 원리와 적용 메커니즘을 충실히 반영하는 최적의 지식 표현 방식을 전제로 하며, 이는 법률 텍스트의 구조화와 더불어 의미론적 개념화 및 규범추론 가능성을 통합적으로 확보하는 방향을 고려해야 한다.

기존의 관계한 데이터베이스(RDB)는 법령의 구조적 정보를 저장하고 검색하는 데는 효과적이지만, 법률 개념 간의 의미적 관계, 규범명제의 논리적 구조, 법적 추론의 복잡성을 처리하는 데는 한계가 있다. 이러한 한계를 극복하기 위해 온톨로지(ontology), 지식 그래프(knowledge graph), 논리 프로그래밍 (logic programming), 의무논리(deontic logic) 등 다양한 지식 표현 기술이 연구되어 왔다.

본 절에서는 법률의 지식 표현 기술을 체계적으로 조사하고 분류하여 각 기술의 특징과 적용 범위, 관련 연구 동향을 정리한다. 구체적으로는 다음의 다섯 가지 범주로 분류하여 살펴본다: (i) 법률 온톨로지, (ii) 논리 기반 표현, (iii) 법령 XML 표준, (iv) 법률 지식 그래프, (v) 뉴로 심볼릭 접근법.

1. 법률 지식표현 관련 연구 동향

가. 문서 구조·메타데이터 중심 표현 (Legal XML Standards)

법령을 ‘구조화된 문서’로 표현하는 방법으로서 법령이나 판결문을 컴퓨터가 안정적으로 파싱·교환하도록 표준 XML/메타데이터 모델로 구조화하는 방법으로 대표 표준/연구는 “Akoma Ntoso[4)]”와 “LegalDocML[5)]”가 있다.

1) Akoma Ntoso

Akoma Ntoso(AKN)는 입법, 사법, 행정 문서를 XML 기반으로 구조화하는 국제 기술 표준으로, 2004년 유엔 경제사회국(UNDESA) 프로젝트에서 시작되었다. 2018년 OASIS 표준으로 채택되었으며, 문서의 구문적(구조)적 요소와 의미적 요소를 모두 기계가 처리할 수 있도록 하여, 법령의 시점 관리(point-in-time consolidation) 등 고급 서비스를 가능하게 한다.

2) LegalDocML

LegalDocML은 법령을 장기적으로 보존·검색·시각화를 고려한 법률문서 표준화 (기술위원회)이다. 미국의 USLM(United States Legislative Markup), 브라질의 LexML, 독일의 LegalDocML.de 등 각국은 Akoma Ntoso를 기반으로 자국의 응용 프로필을 개발하여 활용하고 있으며, 영국의 국립보관소(National Archives)는 2014년 모든 법령을 AKN으로 변환하여 글로벌 개방 데이터 지수(Global Open Data Index)에서 법령 분야 1위를 달성하였다.

나. 규칙/논리 기반 표현

전통적으로 전문가 시스템 계열에서 활용하는 방법으로 규범명제를 ‘규칙’으로 모델링하고, 추론 엔진을 적용하여 예외·우선순위·의무 등을 표현하는 방식이다.

1) LegalRuleML

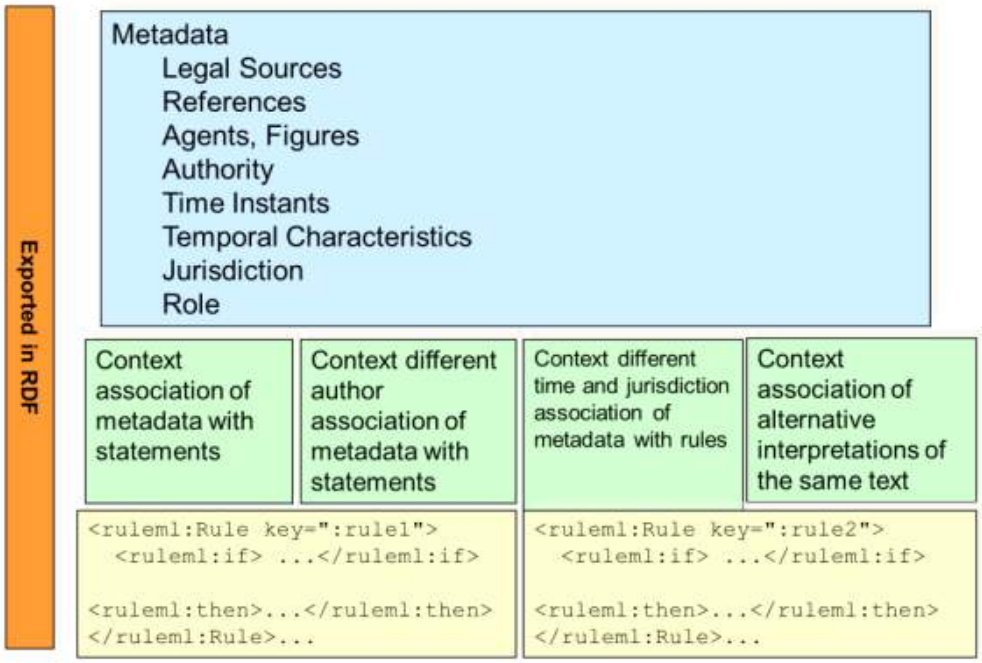
LegalRuleML은 법적 논증의 생성, 평가, 비교를 지원하는 규칙 표현 도구로, 2020년 OASIS 위원회 명세(Committee Specification)로 채택되었다. Akoma Ntoso가 법령 문서의 구조적 표현에 초점을 맞춘다면, LegalRuleML은 규범명제(권리, 의무, 위반, 허가, 금지 등)의 논리적 내용을 표현하는 데 특화되어 있다. GDPR 준수 검사[6)]와 같은 규제 컴플라이언스 분야에서의 활용이 대표

4) Akoma Ntoso Version 1.0, OASIS LegalDocumentML TC, 2018
5) OASIS LegalDocumentML, OASISOPEN,

적이다.

즉, 입법·규정·계약 같은 "법 텍스트" 안의 규범(의무, 권리, 금지 등)을 논리 규칙으로 표현하고, 동시에 그 출처·시점·관할·해석 등 메타데이터를 함께 모델링하는 것이 목적이며, 규범 논리(Deontic), 비단조/가산 논리(Defeasible), 시제 논리(Temporal)를 고려한 법적 추론을 지원하도록 설계되었다.[7)]

법령 지식 표현 관점에서 LegalRuleML 문서는 크게 세 블록으로 나눠지며, (i) Rules 블록은 규범을 논리 규칙으로 표현하는 부분으로 RuleML 구문을 확장해 의무(O), 허용(P), 금지(F), 권리, 책임 같은 법적 개념을 표시할 수 있다. (ii)Metadata 블록은 규범의 출처(어느 법, 어느 조문), 공포·시행일, 적용 관할(jurisdiction), 버전, 작성자, 해석 관점(interpretation) 등을 설명하는 메타데이터를 담는 부분으로 이를 통해 동일한 규범 표현에 대해 대안적 해석·개정 이력등을 함께 유지 가능하다. (iii) Context/parameters 블록은 시간·공간·당사자 범위 등 규범이 작동하는 맥락을 명시해, 추론 시 조건부 적용을 가능하게 한다.[8)]



(그림2-6) LegalRuleML main blocks [15)]

2) Defeasible Logic (비단조/가산 논리)

6) GDPR 준수 검사: 유럽연합(EU)의 개인정보보호 규정인 GDPR(General Data Protection Regulation)을 조직이 적절히 이행하고 있는지 다각도로 평가하고 점검하는 과정

7)Guido Governatori외, "OASIS LegalRuleML", Proceedings of the Fourteenth International Conference on Artificial Intelligence and Law - ICAIL '13, 2013

8) Monica Palmirani, "D2: A LegalRuleML specialisation for reporting obligations, as well as the legal rationale that motivates the design choices",

법령・판례처럼 “일반 규칙 + 수많은 예외” 구조를 가진 지식을 표현할 때, Defeasible Logic(가산/비단조 논리)은 기본 규칙과 예외·우선순위를 함께 기술할 수 있는 경량 논리 프레임워크이다.[9] 예외가 많은 법규범을 “반박 가능 규칙”으로 표현・추론하는 연구 흐름으로서 전통 1차논리/규칙 기반 시스템은 새로운 정보가 추가돼도 이미 도출된 결론이 그대로 유지되는 단조(monotonic) 추론을 전제하지만, 실제 법률에서는 예외・상위법・특별법・후법 때문에 결론이 뒤집히는 일이 흔하다.

비단조 논리는 “보통은 ~이다”라는 가산 규칙(defeasible rule)과, 이를 무효화할 수 있는 defeater(반증 규칙), 그리고 항상 성립하는 엄격 규칙(strict rule), 이들 사이의 우선순위(priorities)를 명시적으로 표현해, 새로운 규정이나 사실이 들어오면 기존 결론을 철회할 수 있게 설계된 논리[10][11]이다. 아래 규칙에서 Defeasible Logic은 두 규칙을 모두 표현하고, ‘집합금지 명령’ 사실이 추가되면, “영업 가능” 결론을 가산적으로 철회하고 “영업 불가”를 채택하도록 만들 수 있다.

일반규칙: “원칙적으로, 허가받은 업소는 영업할 수 있다.”
예외규칙: “하지만, 감염병 예방법에 따른 집합금지 명령이 있으면 영업할 수 없다.”

비단조 논리는 아래와 같은 점에 있어 법령 지식 표현에서 유리하다.

- **예외·충돌·우선순위 표현**: 법령·계약·규정은 “일반 규칙 + 여러 예외 + 상위·특별법 우선” 구조를 가지므로, 규칙 간 우선순위와 예외를 논리 내에 직접 표현 가능
- **규범적(deontic) 연산자와 결합**: 최근 작업들은 의무(O), 허용(P), 금지(F) 같은 규범 연산자를 Defeasible Logic에 통합한 “defeasible deontic logic”을 사용해, 법적 의무·권리·예외를 동시에 표현·추론
- **부분적 정보·모순 처리:** 불완전하거나 모순된 사실(증거가 불충분, 규정이 충돌 등)이 있어도, 가장 강한 규칙·예외를 기준으로 “현재로서는 정당화 가능한 결론”을 계산할 수 있어, 법적 추론(특히 판례·계약 해석)에 적합

9) Benjamin Johnston 외, “Induction of defeasible logic theories in the legal domain”, ICAIL '03: Proceedings of the 9th international conference on Artificial intelligence and law, 2003.
10) Benjamin Johnston외, “Induction of defeasible logic theories in the legal domain”, ICAIL '03: Proceedings of the 9th international conference on Artificial intelligence and law, ACM Digital Library, 2003
11) Grigoris Antoniou 외, “DR-Prolog: A System for Defeasible Reasoning with Rules and Ontologies on the Semantic Web”, IEEE Transactions on Knowledge and Data Engineering, 2007

법령 지식 표현 관점에서 자주 쓰이는 구문은 다음과 같으며, 아래 네 가지를 함께 모델링하여 실제 법 텍스트의 "원칙-예외-특례-경합" 구조를 비교적 자연스럽게 규칙 집합으로 구성할 수 있다[12].

- **Strict rule(엄격 규칙):** 항상 성립하는 규칙
  - 예: "형사처벌은 법률에 근거해야 한다" (무예외 원칙 수준에서 모델링)
- **Defeasible rule (가산 규칙):** 보통은 성립하지만, 더 강한 규칙이나 예외가 있으면 깨지는 규칙
  - 예: "근로자는 1주에 1회 이상 유급주휴일을 갖는다." (단, 특정 예외 규정이 있으면 패배)
- **Defeater(반증 규칙):** 결론을 직접 내리지는 않지만, 다른 규칙의 적용을 막는 정보
  - 예: "재해가 근로자의 중대한 과실에 의한 경우, 통상 산업재해로 보지 않을 수 있다." → '산재로 본다'는 가산 규칙을 깨는 역할
- **Priority (우선순위):** 결론을 규칙들 사이의 우열 관계
  - 예: "특별법이 일반법에 우선", "상위 법령이 하위 법령에 우선", "후법이 전법에 우선"

법령 지식 표현의 활용에서 많은 연구에서 공통적으로 강조하는 포인트는 아래와 같다.

- **규정 자동 해석·분류**: 조항들을 기본적으로 가산 규칙으로 추출하고, "단, ~인 경우에는 그러하지 아니하다" 등 예외 문장을 별도의 defeater 또는 우선순위가 높은 규칙으로 모델링[13]
- **판례·분쟁 해결 지원**: 판례에서 도출된 법리("원칙 + 예외")를 Defeasible Logic 또는 Defeasible Logic Programming(DeLP)으로 표현하여, 양측 주장의 충돌을 "경합하는 논증들의 비교"로 계산하고, 어느 쪽 규칙이 우선하는지 설명 가능한 형태로 제시[14]
- **규정 충돌 탐지와 정합성 검토:** 상충하는 규칙들의 집합에 우선순위를 부여해, 어떤 상황에서 어느 규칙이 이기는지 자동 검증하고, 법령 개정·신설 시 불필요한 충돌이나 "구멍"을 탐지[15]

12) Qing Liu,외 "Towards an efficient rule-based framework for legal reasoning", Knowledge-Based Systems, Vol.224, 2021
13) How Khang Lim외, "Automating Defeasible Reasoning in Law", Artificial Intelligence, 2022
14) C. BEIERLE 외, "Using Defeasible Logic Programming for Argumentation-Based Decision Support in Private Law", 2022
15) Cleyton Mario 외, "LEGIS: A Proposal to Handle Legal Normative Exceptions and Leverage Inference Proofs Readability", Journal of Applied Logics, 2019

(표2-3) 법령 지식 표현 기준의 장점 및 한계점[16)17)]

장점	한계점
- 규칙 기반이라 이해·설명(설명 가능성)이 쉽고, 법률 문장 구조와도 비교적 잘 대응 - 예외·특례·우선순위 모델링이 자연스러워 실제 법령 구조를 그대로 반영하기 좋음 - 계산적으로 비교적 경량이라, 대규모 규칙 집합에도 실용적	- 의미론이 비교적 단순해, 복잡한 양상(상호 의존적 예외, 해석론적 논쟁, 가치형형평성 판단 등)을 모두 담기엔 부족할 수 있음 - 어떤 것을 strict rule로, 어떤 것을 defeasible rule· defeater로 볼지, 우선순위를 어떻게 줄지에 대해 법이론적·실무적 합의가 필요 - 시제·행위·책임귀속, 절차 규범까지 포함한 정교한 모델을 만들려면, Deontic logic, Tense logic, Action logic 등 다른 형식들과의 결합이 필요

다. 온톨로지/지식그래프 기반 (시맨틱웹: 법률 개념을 '의미 구조'로) 표현

법률 온톨로지는 법 영역의 개념적 구성 요소를 명시적으로 개념화하고 체계화하는 기술이다. Gruber(1993)의 정의에 따르면, 온톨로지는 '개념화에 대한 명시적인 명세(an explicit specification of a conceptualization)'로, 법률 온톨로지는 법적 개념, 관계, 규칙을 컴퓨터가 처리할 수 있는 형태로 구조화함으로써 의미론적 검색, 지식베이스 구축, 자동추론 등의 기반을 제공한다.

법률에 대한 온톨로지 기반 표현은 법률 텍스트를 단순 저장·검색하는 수준을 넘어, 법률에서 사용되는 개념(예: '근로자', '사용자', '해고')과 그 관계(예: '해고'는 '근로계약' 관계를 전제, '정당한 이유'는 '요건'에 해당)를 형식적으로 정의하여 기계가 의미 단위로 처리할 수 있도록 만드는 접근이다.

온톨로지는 일반적으로 시맨틱웹 표준(예: RDF/OWL) 위에서 개념·관계를 모델링하고, 질의(SPARQL) 및 제약 검증(SHACL)과 결합되어 데이터의 정합성과 의미 기반 탐색·연계를 지원한다.[18)] 온톨로지의 지식 그래프는 지식 그래프는 법적 개체(entity)와 그 관계(relation)를 트리플(주어-술어-목적어) 구조로 표현하여, 다원적이고 이질적인 법률 지식을 인간의 인지에 가까운 형태로 변환하는 기술이다. 법령 조항, 사법 해석, 판례, 법적 개념 등을 노드와 엣지로 연결하여 복잡한 법적 관계를 시각적으로 탐색하고 추론할 수 있게 한다.

1) 법률 온톨로지의 핵심 특징

가) 의미 기반 상호 운용성(Interoperability)

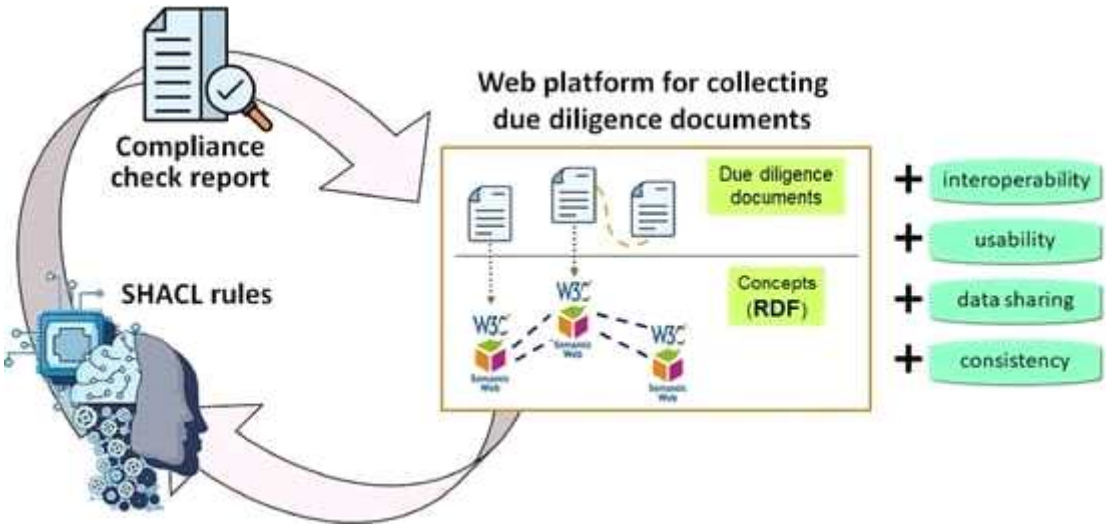
16) Grigoris Antoniou 외, "DR-Prolog: A System for Defeasible Reasoning with Rules and Ontologies on the Semantic Web", IEEE Transactions on Knowledge and Data Engineering, 2007
17) Rigoni, Adam W, "Legal Rules, Legal Reasoning, and Nonmonotonic Logic", 2014
18) Rinke Hoekstra 외, "The LKIF Core Ontology of Basic Legal Concepts", IST-2004-027665, 2004

서로 다른 시스템/기관에서 만든 데이터라도 동일(또는 매핑 가능한) 개념체계를 쓰면, 통합 검색·연계·교환이 용이해진다(유럽권에서 LKIF·ELI 등 "교환/재사용" 목적이 강조됨)[19]

나) 정합성 검증과 제약(Constraints/Validation)

SHACL[20] 등으로 "데이터가 법률 모델의 제약을 위반하는지" 자동 점검(예: 의무/금지 충돌, 필수 속성 누락)하는 연구가 활발하다.[21]

(그림2-4)는 SHACL의 개념도로서 자동 규정 준수 검사에 대한 프로세스를 나타낸다. SHACL의 자동규정준수 검사는 다양한 도메인에서 법적 규정을 집행하기 위한 핵심 도구로 부상했으며, 최근에는 시맨틱 웹 기술인 RDF(Resource Description Framework) 데이터에 대해 SHACL-SPARQL 규칙 실행에 대해 연구가 진행되고 있고, OWL이 제공하는 표현력으로는 특정 규범의 의미를 충분히 표현할 수 없는 경우가 있기 때문이다.



(그림2-7) SHACL 개념 [15]

(그림2-7)는 SHACL 기반 자동규준 검사의 전체 워크 플로우를 보여주는 것으로

(1) Due diligence documents (실사 문서) — 규정 준수를 확인해야 할 대상 문서들이 웹 플랫폼에 수집

(2) Concepts (RDF) — 이 문서들의 핵심 개념과 데이터가 W3C 시맨틱웹 표준(RDF)으로

19) ELI Ontology, http://publications.europa.eu/resource/dataset/eli, POE
20) Shapes Constraint Language, https://www.w3.org/TR/shacl-af/#SPARQLRule
21) Joseph Anim 외, "A SHACL-Based Approach for Enhancing Automated Compliance Checking with RDF Data",

구조화됨 (3) SHACL rules — RDF로 변환된 데이터에 SHACL(Shapes Constraint Language) 규칙을 적용하여 법적 규정 준수 여부를 자동으로 검증 (4) Compliance check report — 검증 결과가 준수 검사 보고서로 생성 이 과정이 순환적으로 반복되며, 새로운 문서가 들어오면 다시 RDF로 변환 → SHACL 검증 → 보고서 생성의 사이클을 돌게 됨

다) 추론 가능성(Reasoninig)

분류/상속(개념 추론)뿐 아니라, 규범 (의무·금지)을 다루려면 규칙/논리(예: SWRL, defeasible logic, LegalRuleML 등)와의 결합이 자주 필요하다.[22]

라) 시간・개정(Temporal/Diachronic) 처리를 구조로 담기 쉬움

법령은 시행·개정·폐지에 의해 의미가 달라지므로 "시간축을 포함한 그래프/온톨로지"로 모델링하려는 시도가 늘고 있다.[23] 기존의 표준적인 평문 검색은 법의 계층적·통시적·인과적 구조에 대해 한계가 있어 시간 착오적이고 신뢰할 수 없는 답변을 생성할 때가 많기 때문에 이를 보완할 수 있는 방법으로 온톨로지를 활용

2) 법률 온톨로지의 유형

가) 구조적 관점의 온톨로지

법령·판례 같은 법률 문서를 정확히 구조화·식별·연결하여 검색/참조/버전관리를 안정화할 수 있는 것으로 조문 단위 식별자, 개정 이력, 공포/시행, 인용/참조 관계, 관보·원문 위치 등 메타데이터 및 문서 구조의 표준화가 중심인 온톨로지이다.

- **Akoma Ntoso:** 입법/사법 문서를 기계가 읽는 구조와 메타데이터 모델로 표준화[24]
- **CEN MetaLex:** 법원(법령) 및 참조를 XML로 표현하는 표준화 흐름[25]
- **ELI(European Legislation Identifier):** 법령 메타데이터를 RDF로 표현·연계하는 온톨로지/프로파일(예: ELI-EP)[26]

22) LegalRuleML Core Specification Version 1.0, OASIS Standard, 2021
23) Hudson de Martim, "An Ontology-Driven Graph RAG for Legal Norms: A Hierarchical, Temporal, and Deterministic Approach",
24) Akoma Ntoso Version 1.0, 2018
25) CEN Metalex, EU Semantic Interoperability Catalogue, 2019
26) ELI Ontology, Publications Office of the European Union, 2024

(표2-4) 법령 지식 표현 기준의 장점 및 한계점

장점	한계점
- 조문 단위 주소/참조(인용·준용·위임 등)와 버전(개정이력) 관리가 강함 - 법령 지식그래프(legislation KG) 구축의 기반(ETL/그래프 분석/검색 최적화)에 유리	- 조문을 "잘 쪼개고 연결"하는 데 강하지만, 요건→효과 같은 규범명제 추론을 온톨로지만으로 완결하기는 어려워 규칙/논증 계층이 필요

나) 규범명제 관점의 온톨로지

법률의 핵심인 규범명제(요건→효과)와 의무·금지·허용·권리·권한 같은 규범적 개념을 명시적으로 모델링하고, (가능하면) 기계 추론까지 연결하는 역할의 온톨로지 유형이다. 규범명제는 "어떤 요건이 충족되면 어떤 법적 효과(의무/금지/허용/권리/제재)가 발생한다"는 형태이므로, 온톨로지만으로 완결하기 보다 보통 아래 두 층을 분리하여 활용한다.

(1) 어휘·개념 층(온톨로지): "행위주체, 행위, 객체, 상태, 기간, 예외, 절차' 같은 개념과 관계를 정의 (의미 일관성/재사용)

(2) 규범 층(규칙/논리/규범표현 표준): 요건과 효과, 예외·특례, 우선순위, 시간적 효력, 반박가능성 (비단조) 등을 표현하고 추론

(3) 대표 사례

- **LKIF Core:** 법률 기본 개념(행위자, 규범, 행위, 시간 등)과 교환/재사용을 위한 코어 온톨로지 제시[27]
- **LegalRuleML:** 규범 규칙(법규범/지침/정책)을 교환 가능한 규칙 마크업으로 표현(온톨로지 계층 위 "법률 규칙/추론 계층"을 지향[28]

(표2-5) 규범명제 관점 온톨로지의 장점 및 한계점

장점	한계점
- 규범적 의미(의무/금지/권리/효력)"를 데이터 구조에 포함하므로, 설명가능성(왜 의무가 성립하는가)과 준수점검(위반 탐지)으로 확장하기 용이	- 예외·특례·우선순위·상충 규범 등은 전통 논리만으로 다루기 어려워 비단조/가산 논리, 규칙엔진, 논증(Argumentation)과 결합 설계가 필요 - 모델링 비용(정의 합의, 범위 설정, 유지보수)이 크고, 법 개정 시 버전/시간 관리까지 포함하면 복잡도가 급증

27) Rinke Hoekstra외, The LKIF Core Ontology of Basic Legal Concepts, 2007.
28) LegalRuleML Core Specification Version, https://docs.oasis-open.org/legalruleml/legalruleml-core-spec/v1.0/os/legalruleml-core-spec-v1.0-os.html (Authoritative), 2021

3) 법률 온톨로지의 최근 연구 동향

가) 표준 기반 법령 지식 그래프

Akoma Ntoso 같은 구조 표준을 기반으로 법령을 지식그래프로 구축하고, 그래프 분석/질의/LLM 연계를 시도하는 연구가 증가하고 있다.[29)]

나) 온톨로지・그래프 기반 RAG(GraphRAG/Ontology-grounded RAG)

평면 텍스트 검색의 한계(계층구조·개정·참조관계 무시)를 극복하기 위해, 법률의 구조/시간/참조를 그래프·온톨로지로 모델링한 뒤 RAG에 투입하는 접근이 제안되었다.

온톨로지를 개념 층, 지식그래프를 사실/문서 연결 층으로 두고 법률 질의응답의 정확도・설명가능성을 높이려는 연구가 지속되고 있다.(예: 2026년 Legal-Onto 모델처럼 온톨로지와 KG를 결합해 지식기반 검색을 구현)[30)]

LLM을 활용해 법률 지식그래프를 자동 구축하거나(추출/정규화/삼중화), 온톨로지로 구조를 고정한 뒤 RAG・그래프 검색을 결합하는 연구가 증가하고 있다.[31)]

다) 규범 준수(Compliance) 자동점검

OWL과 SHALE를 결합하여 규범 준수를 수행할 수 있는 법령 온톨로지를 개발하는 것으로 규정 문장을 지식 그래프로 옮기고 SHACL 규칙을 실행하여 준수/위반을 판정하는 연구가 활발(법규정의 “검증 가능성”을 목표)하게 진행되고 있다.[32)]

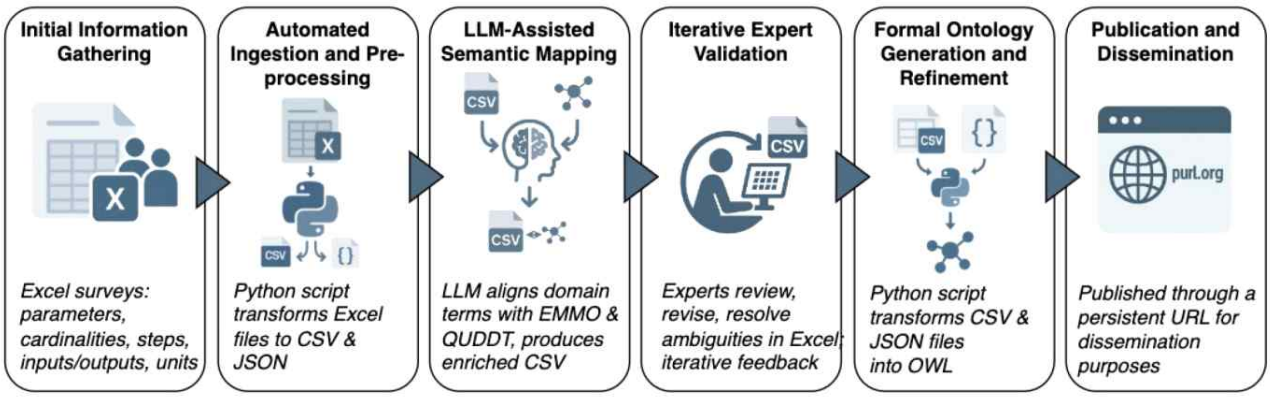
라) 온톨로지 구축과 정렬의 LLM 활용

온톨로지 설계 (요구사항→개념 후보→제약) 과정에 LLM을 보조 도구로 활용(온톨로지 협업 설계, 자동 온톨로지 구조 생성, 기존 온톨로지 정렬)하는 연구 증가[33)][34)]

---

29) Andrea Colombo외, “LegisSearch: navigating legislation with graphs and large language models”, Artificial Intelligence and Law, 2025
30) Vuong T. Pham외, “Ontology-Based Knowledge Graph Approach for Legal Queries”, https://doi.org/10.15388/ 25-INFOR617, 2026
31) Jun Li 외, “Construction of Legal Knowledge Graph Based on Knowledge-Enhanced Large Language Models”, https://doi.org/10.3390/info15110666, 2024
32) Joseph Anim 외, “A SHACL-Based Approach for Enhancing Automated Compliance Checking with RDF Data”,
33) Janis Kampars1외, “LLM-supported collaborative ontology design for data and knowledge management platforms” ORIGINAL RESEARCH article, 2025
34) Janis Kampars

(그림2-8)는 Janis Kampars의 연구 “HyWay 프로젝트”에서 제안한 HyWay 방법론의 전체 워크플로(6단계)를 요약한 도식으로, 엑셀 설문에서 시작해 OWL 온톨로지와 퍼시스턴트 URL 출판까지 이어지는 파이프라인을 통해 온톨로지 구축 과정을 제시하였다.



(그림2-8) SHACL 개념 [15]

4) 이중 온톨로지와 LLM 통합 아키텍처 방향

가) 이중 온톨로지 구성 근거

Janis Kampars의 연구를 기반으로 본 연구는 3장에서 (1) 법령구조온톨로지, (2) 규범연제 온톨로지라는 두 영역으로 나눈 온톨로지 체계에 LLM을 결합하여 LLM의 강력한 자연어 처리 기술에 온톨로지의 정형화된 정보와 판단을 제공하는 구조를 구성한다.

(표2-6) 이중 온톨로지 설계의 근거

구분	법령구조 온톨로지	규범명제 온톨로지
모델링 대상	법령의 물리적·계층적 구조	법령 조문이 표현하는 의무·권리·조건 등 규범적 의미
엔티티 예시	법률, 편, 장, 조, 항, 호, 부칙 등	의무(Obligation), 금지(Prohibition), 허용(Permission), 조건(Condition), 제재(Sanction)
관계 예시	hasPart, hasVersion, amendedBy, enactedOn	imposes, appliesTo, triggeredBy, exemptedIf
참조 모델	SAT-Graph RAG의 LRMoo 기반 Work/Expression 모델	SHACL 논문의 규정 준수 규칙 체계 + deontic logic 전통
시간 모델링	CTV/CLV 버전 체인, Action 노드	TVC 패턴 (값-시간-맥락), 시행일·유효기간
주된 용도	“2020년 1월 1일 시점의 제15조 텍스트는?"	"개인정보 처리 시 정보주체 동의를 받아야 하는가?"

이런 분리 구조는 HyWay 논문의 HMIO가 실험(experiment) 모듈과 시뮬레이션(simulation) 모듈을 별도 .ttl 파일로 분리한 모듈식 아키텍처와 같은 설계 원리를 따라, 구조와 규범이 독립적으로 진화할 수 있고, 각각 다른 전문가 그룹이 검증할 수 있는 구조가 될 수 있다.

나) LLM 통합 방안

HyWay는 LLM을 "제한된 역할의 co-pilot"으로 활용하는 전략이 효과적임을 실증했으며, 이를 법률 온톨로지에 적용하면:

(1) 구조 온톨로지 구축 시 LLM 역할

단계	법령구조 온톨로지	규범명제 온톨로지
1. 초기 정보 수집	법제처 법령정보센터 API에서 법령 XML 파싱 → 편·장·조·항 구조 추출	법령 XML은 이미 구조화됨
2. 자동 전처리	- XML → JSON/CSV 변환 - 법령 메타데이터 정규화	결정적(deterministic) 파싱
3. 시맨틱 매핑	법령 구조 요소를 LRMoo·Akoma Ntoso 등 표준 온톨로지에 매핑	- LLM co-pilot: "조"를 LRMoo:Component에, - "개정"을 Action에 매핑하는 후보 제안
4. 전문가 검증	법률 전문가가 매핑 정확성 검토	-
5. OWL 생성	OWL 생성 스크립트 자동 생성	구조가 정형적 이므로 규칙 기반 생성

구조 온톨로지의 경우 법령 XML이 이미 반구조화 데이터이므로, HyWay처럼 Excel 설문이 아닌 자동 파싱이 주된 지식 획득 경로가 될 수 있기 때문에 LLM의 역할은 최소화될 수 있다.

(2) 규범명제 온톨로지 구축 시 LLM 역할 - 핵심 영역

단계	법률 도메인 적용	LLM 역할
1. 초기 정보 수집	법 조문 자연어 텍스트 수집	-
2. 자동 전처리	조문 텍스트를 문장 단위로 분절	-
3. 시맨틱 매핑	① 자연어 조문에서 규범명제 (의무/금지/조건/제재) 분류 ② 규범명제 구성요소 추출	LLM 핵심 역할
4. 전문가 검증	법전문가가 deontic 분류와 논리 구조 검토	-
5. OWL 생성	검증된 규범명제를 SHACL-SPARQL 규칙으로 변환	LLM 보조 - 규칙 초안 생성, 전문가 수정

(3) LLM과 전문가의 역할 구분

LLM 작업	전문가 검증 작업
· 조문 텍스트에서 deontic 유형(의무/금지/허용) 분류 · 규범의 구성요소 추출: 수범자(addressee), 행위(action), 조건(condition), 제재(sanction) · 유사 조문 간 참조 관계 후보 제안 · SHACL-SPARQL 규칙 초안(template) 생성	· deontic 분류의 정확성 (법적 미묘한 차이 - "할 수 있다" vs "하여야 한다" vs "하도록 노력하여야 한다") · 예외 조항과 패배 가능성(defeasibility) 관계 · SHACL 규칙의 논리적 정합성 · 법률 흠결(lacuna) 식별

마. 뉴로심볼릭 AI 접근법 (Neurosymbolic AI)

1) LLM과 기호적 추론의 통합

최근 법률 지식 표현 연구의 가장 주목할만한 흐름은 대규모 언어모델(LLM)과 기호적 추론(symbolic reasoning)을 결합하는 뉴로심볼릭 AI 접근법이다.

LLM은 자연어 이해에 강점을 가지지만 환각(hallucination) 문제가 있고, 기호적 시스템은 정확한 추론이 가능하지만 자연어 처리에 한계가 있다. 이 두가지를 통합함으로써 각각의 한계를 보완할 수 있다.

Stanford CodeX의 Kant[35] 등 (2024)은 NeuraIPS 2024 워크샵에서 LLM과 논리 프로그래밍을 결합한 보험 계약 분석 연구를 발표하였다. 그들은 바닐라 LLM, 비유도록 접근(unquided), 유도적 접근(guided)의 세 가지 방법론을 비교하였으며, LLM과 Logic의 유도적 접근이 가장 가장 우수한 성과를 보였다. 특히 OpenAI 계열의 추론 특화 LLM이 기존 LLM 대비 큰 성능 도약을 보여주었다.

2) 통합 프레임워크

L4M 프레임워크는 대립적 LLM 에이전트(adversarial LLM agent)와 SMT 솔버 기반 증명을 결합하여 자연어의 해석적 유연성과 기호적 검증의 엄밀성을 통합한 접근법이다. NS-LCR(Neural-Symbolic enhanced Legal Case Retrieval)는 사건 수준과 법령 수준의 논리 규칙을 학습하여 유사 판넬 검색에서 설명 가능한 결과를 제공하는 프레임워크이다.

이러한 뉴로심볼릭 접근법은 법률 지식 표현의 미래 방향을 제시하고 있으며, 특히 법률 서비스의 정확성, 반복성, 투명성을 동시에 충족시킬 수 있는 가능성을 보여주고 있다.

35) https://law.stanford.edu/codex-the-stanford-center-for-legal-informatics/projects/codex-insurance-initiative/

제3절. 법률정보(지식)시스템

앞 절에서 법의 구조·내용·규범·관계적 특성을 정리한 이유는, 본 연구가 궁극적으로 지향하는 대상이 '법 그 자체'의 설명이 아니라 법을 기반으로 작동하는 법률지식시스템(Legal Knowledge System)의 구현 가능성과 한계를 분석하기 위함이다. 즉, 법이 어떤 형태로 구성되고(조문 구조·부속 요소), 어떤 방식으로 의미가 생성되며(정의·규범명제), 어떤 관계망 속에서 효력이 결정되는지(위계·위임·참조·상충·시간효력)를 먼저 규명하지 않으면, 이후 논의할 법률지식시스템의 기능 요구사항과 기술적 난점을 타당하게 도출할 수 없다. 이러한 선행 정리는 "법을 어떻게 모델링해야 하는가"에 대한 기준점을 제공하고, 시스템 설계 관점에서 무엇이 핵심 데이터이며 어떤 관계가 추론의 전제가 되는지를 명확히 한다.

이러한 배경에서 본 절에서는 법률정보(지식)시스템에 대한 현황과 분석을 수행하였다. 법률정보(지식)시스템은 단순한 법령 검색을 넘어, 사용자의 상황에 맞는 규범을 탐색하고, 조문 간 관계를 따라가며, 적용 요건과 예외를 반영해 결과를 제시하는 등 '이해·적용·설명' 중심의 서비스를 목표로 한다. 그러나 앞서 정리한 법의 특성 – 복잡한 조문 구조와 부속 요소(별표·서식), 규범의 조건부 적용과 양상성, 예외·특례·준용에 따른 규범 수정, 상·하위법 및 다법령 연계, 빈번한 개정과 시간적 효력 – 은 기존의 키워드 검색이나 단순 문서 기반 데이터베이스만으로는 충분히 처리하기 어렵다. 다시 말해, 법률지식시스템이 제공해야 할 서비스 수준이 높아질수록, 법의 '텍스트'만으로는 사용자의 질의에 대해 일관되고 설명 가능한 답을 생성하기 힘들어지며, 이는 곧 기술적·구조적 한계로 이어진다.

따라서 법률정보(지식)시스템 조사는 단순한 관련 분야 소개가 아니라, 법의 특성에서 출발한 필연적 연구 절차로서 위치한다. 본 연구는 법률지식시스템의 구현 목표와 기존 접근(검색 중심, 규칙 기반, 통계·딥러닝 기반 등)을 검토함으로써, 왜 현행 기술만으로는 '온전한' 법률지식 서비스(정확성·적용가능성·설명가능성)를 안정적으로 제공하기 어려운지 논증하고자 한다. 그리고 이러한 한계를 극복하기 위한 대안으로, 법의 구조와 관계를 명시적으로 표현할 수 있는 시맨틱 기술인 온톨로지 기반 지식화와, 자연어 처리·추론·요약 등 비정형 처리를 담당하는 AI 기술의 결합 필요성을 후속 절에서 구체적으로 제시한다.

1. 법률지식시스템 개요

가. 법률지식시스템 정의

법률지식시스템(Legal Knowledge System)은 법률 영역의 전문 지식을 체계적으로 표현, 저장, 추론하여 법률 관련 문제 해결을 지원하는 컴퓨터 기반 시스템이다. Valente & Breuker(1999)는 ON-LINE 아키텍처에서 법률지식시스템을 "법률 정보 제공(Legal Information Serving)과 법률 분석(Legal Analysis)을 결합한 법률 워크벤치" 로 정의하였다.

법률지식시스템은 일반적인 지식 기반 시스템(Knowledge-Based System)의 특수한 형태로, 다음과 같은 법률 도메인의 고유한 특성을 반영한다.

- 규범성(Normality): 법률은 당위(ought to be)를 규정하며, 이는 이는 사실(is)가 구별됨
- 해석 다양성(Interpretive Plurality): 동일 법조문에 대한 다양한 해석 가능성
- 시간적 변화(Temporal Dynamics): 법령의 재・개정에 따른 지속적 변화
- 계층적 구조(Hierarchical Structure): 헌법-법률-시행령-시행규칙의 위계
- 복합 관계적 구조(Relational Structure): 법조문 내 타법, 동법, 하법, 상위법 등의 관계 참조

원론적인 정의에 더하여 최근에는 법률지식시스템은 법률 전문가(변호사, 법무사, 노무사, 감독관 등)나 일반 사용자가 법률 정보를 검색, 분석, 해석, 적용하는 과정을 자동화하고 지원하기 위해 설계된 컴퓨터 기반 시스템을 총칭하며, 법률, 판례, 행정규칙, 고시 등 복잡한 규범지식을 체계적으로 수집, 구조화, 저장하고 이를 기반으로 질의응답, 규제 적용 판단, 법률 해석 지원, 행정 업무 자동화 등에 활용하는 기술적・지식적 체계를 의미한다.

또한 디지털 정부, 규제 자동화(RegTech[36]), 리걸테크(LegTech)[37]와 같은 법률 AI 발전과 함께 법률 지식 시스템의 중요성은 크게 확대되고 있으며, 단순한 법령 조회 시스템을 넘어 규범의 의미적 구조를 반영한 지식 기반 추론 시스템으로 진화하고 있다.

36) RegTech는 규제(Regulation)와 기술(Technology)의 합성어로 IT 기술을 활용하여 기업의 규제 준수(Compliance) 및 보고 의무를 보다 효율적이고 효과적으로 관리하도록 돕는 혁신적인 기술 분야를 의미

37) 리걸테크(LegalTech)는 법률(Legal)과 기술(Technology)의 합성어로, 정보통신기술(ICT)을 활용하여 법률 서비스의 제공 방식, 법률 전문가의 업무 효율성, 그리고 일반 대중의 법률 접근성을 혁신하는 기술 및 산업 분야를 의미

나. 법률지식시스템의 발전 역사

법률지식시스템의 발전은 크게 세 단계로 구분할 수 있다.

(표2-7) 법률지식시스템의 발전 단계

세대	시기	주요 특징	대표 시스템
1세대	1970-1990	규칙 기반 전문가 시스템, 상징적 AI	TAXMAN, HYPO, CATO
2세대	1990-2010	온톨로지 기반 지식 표현, 웹 기술 통합	ON-LINE, LKIF, MetaLex
3세대	2010~현재	기계학습/딥러닝 통합, LLM 기반 RAG	Legal-Bert, Lawbot, GPT-4 Legal

다. 관련 이론 및 선행 연구

법률지식시스템의 이론적 기초는 크게 세 가지 학문 영역에서 발전해 왔다.

1) 법률 온톨로지(Legal Ontology)

Breuker et al.(2006)은 FOLaw(Functional Ontology of Law)에서 법률지식을 다섯 가지 범주로 구분하였다: 규범적 지식(Normative), 세계 지식(World), 책임 지식(Responsibility), 반응적 지식(Reactive), 창조적 지식(Creative). 이러한 분류는 법률 추론에서 지식 유형 간 의존관계를 설명하는 인식론적 프레임워크를 제공한다.

LKIF(Legal Knowledge Interchange Format) Core 온톨로지는 유럽의 ESTRELLA 프로젝트의 산출물로, 법률 도메인 내에서 정책, 법령, 판례뿐만 아니라 그에 따르는 정당성 근거(Justificatory arguments)[38]를 표현하고 상호 교환하기 위한 표준이 되는 것을 목표로 설계되었다. LKIF는 개념 표현을 위한 웹 온톨로지 언어인 OWL(Web Ontology Language)을 기반으로 구축되었으며, 재사용 가능한 기초 법률 개념 온톨로지를 포함한다.

LKIF는 법률지식을 재사용하고 교환하기 위한 표준 형식을 제공한다. LKIF는 상위 온톨로지(장소, 부분관계, 시간), 핵심 온톨로지(행위, 규범, 역할), 법률 도메인 온톨로지의 계층 구조로 구성되며, 이를 위한 핵심 표현 언어로 OWL-DL(OWL-Desription Language)과 SWRL(Semantic Web Rule Language)의 결합하여 제공된다.

38) 정당성 근거(Justificatory arguments): 단순한 법령 텍스트 저장이 아니라, '왜 이런 결론이 나왔는가'에 대한 논증 구조까지 표준화 하는것

(표2-8) LKIF Core 온톨로지 개요

순번	항목	내용
1	명칭	LKIF Core (Legal Knowledge Interchange Format Core)
2	개발	ESTRELLA 프로젝트 (EU IST-2004-027665)
3	버전	1.1
4	라이선스	GNU LGPL 2.1
5	네임스페이스	http://www.estrellaproject.org/lkif-core/
6	표현언어	OWL-DL + SWRL
7	목적	법률 지식 교환을 위한 표준 어휘 제공

LKIF는 4계층 15개 모듈로 구성된 온톨로지 라이브러리로서 본 연구도 지속적으로 법률의 현행화와 표준화를 위해 LKIF Core 온톨로지 모듈의 표현을 참고하였다.

(표2-9) LKIF Core 온톨로지 모듈 구조

순번	계층	모듈	설명
1	추상계층 (Abstract Level)	top.owl	최상위 개념 (Mental, Physical, Abstract)
2		places.owl	상대적 장소 이론 (Donelly, 2005)
3		mereology.owl	부분-전제 관계
4		time.owl	시간 온톨로지 (Allen, 1984)
5		spacetime.owl	시공간 통합 모듈
6	기본 계층 (Basic Level)	process.owl	변화와 프로세스
7		role.owl	역할 유형론
8		action.owl	행위 어휘
9		expression.owl	명제와 명제적 태도
10	법률 계층 (Legal Level)	legal-action.owl	법적 행위
11		legal-role.owl	법적 역할
12		norm.owl	규범, 의무, 금지, 허용
13	프레임워크 (Frameworks)	modification.owl	법령 개정/변경
14		lkif-rules.owl	LKIF 규칙 어휘

법률지식시스템에서 법률적 가치를 달성하기 위한 핵심 계층이 법률 계층이며 법적 행위, 법적 역할, 규범/의무/금지/허용 등을 표현하고 데이터를 저장할 수 있는 모델이 구성되어 있다. (표2-5)는 법률 계층의 Norm 모듈과 Action을 표현한 일부이다.

(표2-10) LKIF의 Norm 모듈 클래스 구성

순번	상위 클래스	하위 클래스 (2 레벨)	하위 클래스 (3 레벨)	하위 클래스 (4 레벨)
1	Qualification (from expression.owl)	Normatively_Qualified	Allowed (허용된 것)	Obliged(의무인 것) - subClassOf Allowed
			Disallowed (금지된 것)	
2	Norm (규범)	Permission (허용) - allows → Allowed		
		Obligation (의무) - allows → Obliged, disallows → Disallowed		
		Prohibition (금지) - disallows → Disallowed [Obligation ≡ Prohibition - 동치 관계]		
3	Right (권리)	Potestative_Right (형성권)		
		Permissive_Right (허용적 권리)		
		Immunity (면제권)		
4	Power (권한)	Liability (책임)		
		Disability (무능력)		

(표2-10)는 Manchester Syntex 표현법에 따라 규범의 클래스를 기술한 것이며, (표2-11)은 규범의 의무논리에 대한 LKIF 표현과 OWL-DL 패턴의 대응표이다.

(표2-11) 규범 클래스의 의무논리 대응표

순번	의무 논리	LKIF 표현	OWL-DL 패턴
1	O(p) - 의무	Obligation allows Obliged	Obligation ⊑ ∃allows.Obliged
2	F(p) - 금지	Prohibition disallows Disallowed	Prohibition ⊑ ∃disallows.Disallowed
3	P(p) - 허용	Permission allows Allowed	Permission ⊑ ∃allows.Allowed
4	O(p) ↔ F(¬p)	Obligation ≡ Prohibition	Obligation ≡ Prohibition
5	O(p) → P(p)	Obliged ⊑ Allowed	Obliged rdfs:subClassOf Allowed

2) 의무 논리(Deontic Logic)

게오르크 헨리크 폰 릭트(Georg Henrik von Wright)의 의무논리는 법률지식시스템에서 규범적 명제의 형식적 표현 기반을 제공한다. 즉, ‘규범명제’를 기계가 이해할 수 있는 논리식으로 변환하는 가장 고전적이면서도 강력한 틀을 제공한다.

폰 릭트의 체계에서는 행위(Action)을 대상으로 하는 양상 연산자(Deontic Operators)를 통해 규범의 강도를 표현한다. 의무(Obligation), 금지(Prohibition), 허용(Permission)의 양상 연산자를 통해 법규범의 논리적 관계를 표현한다. 이는 법률지식시스템의 추론 엔진 설계에 직접적 영향을 미쳤다.

von Wright는 비형식적 언어 사용을 분석하면서, 전통적인 진리함수적 논리만으로는 규범·행위·시간·진리 개념의 의미를 다 포착할 수 없다고 보고 새로운 논리 체계들을 제시했다. 특히 규범문(“해야 한다”), 행위술어, 시간 연산자 등이 지니는 논리 형식이 곧 그 의미 역할(어떻게 사용되는가)을 드러낸다는 입장을 취한다.[39]

(표2-12) 의미 논리 핵심 연산자

순번	구분	설명	비고
1	O (Obligatory)	의무(~해야 한다)	
2	P (Permitted)	허용 (~할 수 있다 / 해도 된다)	
3	F 또는 Ph (Forbidden)	금지 (~해서는 안 된다)	
4	I (Indifferent/Optional)	자유/선택 (해도 되고 안 해도 된다)	

기본 논리적 상관관계에서

- Op ≡ ¬P¬p (의무는 '하지 않는 것'이 허용되지 않음과 같다)
- Fp ≡ ¬Pp ≡ O¬p (금지는 '하는 것'이 허용되지 않음, 즉 '하지 않을 의무'와 같다)

(표2-13) 의미 논리 연산자를 이용한 노동법 규범명제의 형식적 표현 예시

구분	자연어 (규범명제)	형식 표현
의무	사용자는 근로자에게 퇴직금을 지급해야 한다	O(Pay_Severance)
금지	사용자는 근로자를 폭행하지 못한다	F(Violence) 또는 O(¬ Violence)
허용	당사자 합의 시 1주 12시간을 한도로 근로시간을 연장할 수 있다.	Agreement→P(Extend₩_Work)
자유/선택	근로자는 휴게시간을 자유롭게 이용할 수 있다	P(Use_Break) ∧ P(¬ Use₩_Break)

기존 RDB는 이러한 양상적 차이를 단순히 텍스트로 저장하지만, 폭 릭트의 논리를 온톨로지에 반영하면 다음과 같은 지능형 서비스가 가능해진다.

(표2-14) 규범 요소의 의무논리 기반 모델링의 시스템적 의의

규범 요소	의무논리 기반 모델링	시스템적 의의
양상성	O, P, F 연산자 매핑	'재량 행위'와 '기속 행위'의 기계적 구분 가능
조건부성	Condition → Operator(Action)	특정 상황(요건) 발생 시에만 규범력 활성화
수정 관계	¬P(A)(원칙이나 특례에 의해 $P(A)$로 전환	예외 조항에 의한 규범 강도 변경 추적

39) Von Wright's truth-logic and around

3) 사례 기반 추론 (Case-Based Reasoning, CBR)

법률 분야의 사례기반 추론은 단순히 유사한 문서를 찾는 검색 기술을 넘어, 과거의 판결 사례(Precedents)를 바탕으로 새로운 사건(Current Case)에 대한 법적 논리를 구성하는 결론을 도출하는 인공지능 추론 방법론이다.

Ashley(1990)의 HYPO 시스템부터 시작된 이 흐름은 법률 AI가 '단순한 법조문 암기'에서 '판례를 활용한 변론과 추론'의 영역으로 진입하게 된 이정표가 되었다. 법률 CBR은 판사를 설득하기 위해 "과거의 유사한 사건에서 이러한 결론이 났으므로, 이번 사건도 동일하게 취급되어야 한다"는 유추 해석(Analogical Reasoning)에 기반한다. 이를 위해 판례를 정형화된 지식으로 구조화하는 작업이 선행되어야 한다.

(표2-15) 대표적인 법률 사례기반 추론 예시

구분	설명	주요 특징
HYPO 시스템 (Ashley, 1990)	영업비밀(Trade Secret) 보호법 분야를 대상으로 개발되었으며, 판례를 분석하기 위해 '차원(Dimensions)'이라는 개념을 도입	· 차원(Dimensions): 사건의 승패에 영향을 미치는 법적 쟁점을 수치나 척도로 표현한 것입니다 (예: 영업비밀을 아는 직원의 수, 기밀 유지를 위한 조치의 강도) · 논리 구성: 현재 사건과 가장 유사한 '가장 가까운 이웃'을 찾아내고 유리한 판례와 불리한 판례를 비교하며 변론을 생성
CATO 시스템 (Aleven, 1997)	HYPO를 계승한 CATO는 교육적 목적을 결합하여, 차원보다 단순화된 '요소(Factors)' 개념을 정립	· 요소(Factors): 해당 사례에서 원고나 피고에게 유리하게 작용하는 사실관계의 유무를 이진법(Boolean)으로 표현 · 요소 위계(Factor Hierarchy): 하위의 구체적 사실관계가 어떻게 상위의 법적 쟁점으로 연결되는지를 구조화하여, 판례 간의 유사성을 더 논리적으로 파악하게 함
IBP 시스템 (Issue-Based Prediction)	CBR에 규칙 기반 모델(Rule-based Model)을 결합한 하이브리드 형태	· 논리 구조: 법령의 정형적 체계(조문 구조)를 '이슈 트리'로 만들고, 각 이슈의 판단이 모호한 지점에 CBR을 적용하여 판결 결과를 예측

법률 지식 시스템에서 CBR은 보통 다음의 4단계를 거치며 작동한다.

(표2-16) CBR의 추론 프로세스 (4RE Cycle)

순번	프로세스	설명
1	Retrieve (검색)	현재 사건의 사실관계와 가장 유사한 요소를 가진 과거 판례를 지식 베이스에서 추출
2	Reuse (재사용)	과거 판례에서 적용된 법적 논리와 결론을 현재 사건의 변론 구성에 차용
3	Revise (수정)	현재 사건과 과거 판례 간의 차이점을 분석하여 논리를 보완
4	Retain (보존)	새로운 판결이 나면 이를 다시 지식 베이스에 저장하여 향후 추론의 근거로 삼음

CBR과 RAG와의 차별점은 RAG 기술이 텍스트의 '유사성'에 의존한다면, CBR은 법적 '논리'와 '쟁점'의 구조에 집중한다.

- **구조적 표현력의 한계 극복**: 현재의 텍스트 중심 검색은 조문 간의 유기적 연결이나 체계적 맥락이 단절되는 한계가 있음
- **규범적 판단 지원**: 단순 검색은 법이 사회적으로 수행하는 다양한 기능을 제공하지 못하지만, CBR은 과거의 구체적 판단 사례를 연결함으로써 실질적인 '규제 판단'을 지원함
- **할루시네이션 방지**: LLM 단독으로는 법적 추론의 근거가 부족하고 일관성이 없으나 , CBR은 명확한 '판례'라는 지식 기반(Knowledge-based)을 통해 답변의 신뢰성을 확보할 수 있음

2. 법률지식시스템 핵심 구성요소

법률지식시스템은 일반적인 전문가 시스템의 구조를 기반으로 하면서, 법률 도메인의 특수성을 반영한 구성요소를 포함한다.

(표2-17) 법률지식시스템 핵심 구성 요소

구성요소	핵심 기능	법률 도메인의 특수성
지식베이스	규칙과 사실의 저장소	법령, 판례, 학설의 구조화
추론 엔진	지식 적용 및 결론 도출	의무논리 기반 규범 추론
법률 온톨로지	개념과 관계의 형식적 정의	법적 개념의 계층적 분류
사용자 인터페이스	입출력 및 상호작용	법률 문서 형식 지원
지식 획득 모듈	새로운 지식의 추가/갱신	법령 개정 자동 반영
설명 기능	추론 과정의 투명성 제공	법적 근거 제시

가. 지식 베이스

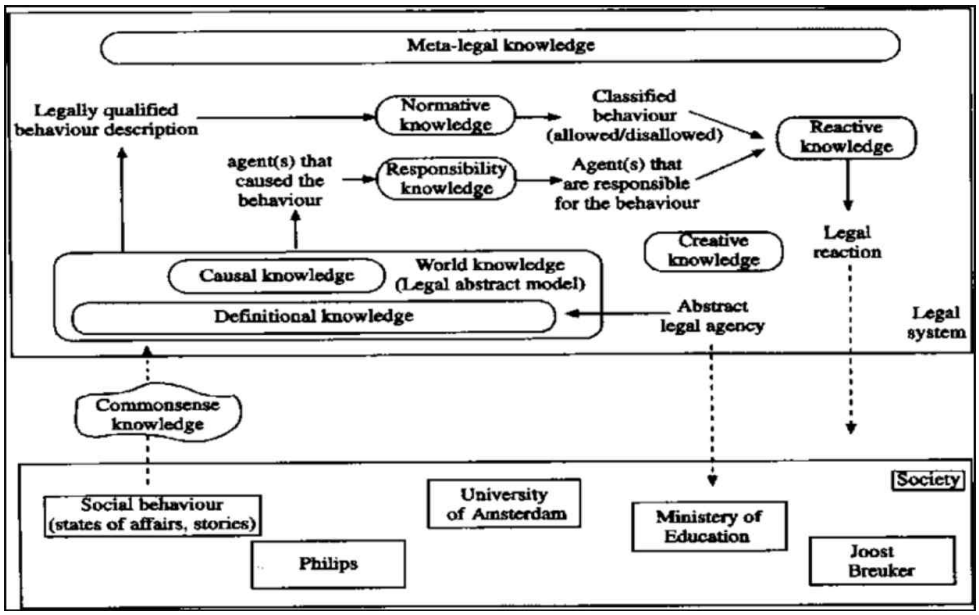
지식 베이스는 법률지식시스템의 핵심 저장소로, 법률 도메인의 전문 지식을 구조화된 형태로 보관한다. 법률지식 베이스는 일반적인 전문가 시스템의 지식베이스와 달리, **법률 고유의 지식 유형을 체계적으로 수용**해야 한다.

1) 법률지식의 유형

Valente & Breuker(1999)의 FOLaw[40] 프레임워크에 따르면, 법률지식은 (그림2-9)와 같이 6개의 지식과 행위 및 사회로 구성이 된다.

40) JOOST BREUKER외, "Legal Ontologies in Knowledge Engineering and Information Management", Artificial Intelligence and Law, 2004

- Meta-legal knowledge: 메타 법률 지식(법률 시스템 전체를 관장하는 상위 지식)
- 규범적 지식(Normative Knowledge): 법적 의무, 금지, 허용을 규정하는 규칙
- 세계 지식(World Knowledge): 법이 규율하는 대상 영역에 대한 상식적 지식
  - Causal knowledge: 인과적 지식
  - Definitional knowledge: 정의적 지식
- 책임 지식(Responsibility Knowledge): 법적 행위에 대한 책임 귀속 규칙
- 반응적 지식(Reactive Knowledge): 법 위반에 대한 제재와 구제 수단
- 상식적 지식(Commonsense Knowledge): 상식적 지식
- 창조적 지식(Creative Knowledge): 법적 요력을 발생시키는 행위 규칙
- 행위 기술(Legally qualified behaviour description): 법적으로 평가된 행위 기술
- 사회(Society): 실제 행위와 주체들이 존재하는 영역



(그림2-9) Legal Ontologies의 프레임워크

이 프레임워크는 현실 세계의 사회적 행위가 어떻게 법률적으로 해석되고 처리되는지를 보여주며, 법률지식시스템의 목적을 위해 고려될 주요 요소가 될 수 있다.

- 사회영역: 실제 기업, 기관, 개인 간의 행위가 발생
- 상식과 정의: 사회적 행위는 '상식적 지식'을 거쳐 법률 시스템 내의 '정의적 지식'과 '인과적 지식'을 통해 추상화 됨
- 판단 프로세스: 행위가 법적으로 기술되면, 그것이 허용되는지(규범적 지식)와 누

구에게 책임이 있는지(책임 지식)을 따짐

- 법적 대응: 최종적으로 '대응적 지식'을 통해 사회로 다시 법적 처벌이나 보상 같은 '법적 대응'이 됨

2) 지식 표현 형식

법률지식베이스에서 지식을 표현하는 주요 형식은 다음과 같다.

(표2-18) 법률지식베이스에서 지식 표현 형식

구성요소	핵심 기능	법률 도메인의 특수성
생산규칙 (Production Rule)	IF <조건> THEN <결론/행위> 조건-행위 쌍의 집합	법률 요건-효과 표현
프레임(Frame)	슬롯-값 구조의 템플릿 상속 관계 지원	법적 개념의 속성 정의
시맨틱 네트워크	노드와 엣지로 구성된 그래프 개념 간 관계 표현	법률 개념 관계 시각화
OWL / RDF	웹 온톨로지 언어 논리적 추론 지원	LIF Core, MetaLex

3) T-Box와 A-box 분리

법최신 법률지식시스템에서는 Description Logic의 T-Box(Termonological Box)와 A-Box(Assertional Box) 분리 원칙을 적용한다. T-Box는 법률 개념의 일반적 정의와 관계를 포함하고, A-Box는 구체적 사실과 인스턴스를 포함한다. 이러한 분리는 지식의 재사용성을 높이고 새로운 사례에 대한 추론을 효율화 한다.

나. 추론 엔진 (Inference Engine)

추론 엔진은 지식 베이스에 저장된 규칙과 사실을 활용하여 새로운 결론을 도출하는 핵심 모듈이다. 법률지식시스템의 추론 엔진은 특수한 요구사항을 반영해야 한다.

1) 추론 전략

추론 엔진은 크게 두 가지 추론 전략을 사용한다.

2) 법률 추론의 특수성

법률 추론은 일반적인 논리적 추론과 다음 측면에서 구별된다.

- 비단조 추론(Non-monotonic Reasoning): 새로운 정보 추가 시 기존 결론이 철회될 수 있음
- 결손 추론(Defeasible Reasoning): 예외 규정에 의해 일반 규칙이 무효될 수 있음
- 유추 추론(Analogical Reasoning): 유사 사례에 기반한 법적 판단

- 목적론적 추론(Teleological Reasoning): 법의 목적에 기초한 해석

3) 규범 추론을 위한 의무논리

법률지식시스템의 추론 엔진은 von Wright의 의무논리를 기반으로 규범적 명제를 처리한다. 핵심 연산자는 (표2-19)와 같다.

(표2-19) 규범 추론을 위한 의무논리 핵심 연산자

연산자	의미	형식	예시
O(p)	의무 (Obligation)	p를 해야 한다.	신고 의무
F(p)	금지 (Prohibition)	p를 해서는 안 된다	뇌물 수수 금지
P(p)	허용 (Permission)	p를 해도 된다.	계약 해지권

이들 연산자 간에는 (표2-20)의 논리적 관계가 성립하며, 이 관계별로 근로기준법에서 의무논리 기반 규범을 추론할 수 있다.

(표2-20) 의무논리 기반 근로기준법 규범 추론 사례

연산자	의미	원칙
O(p)↔F(¬p)	의무는 반대 행위의 금치와 동치	어떤 행위를 할 의무가 있다는 것은, 그 행위를 하지 않는 것이 금지됨을 의미한다.
F(p) ↔ O(¬p)	금지는 반대 행위의 의무와 동치	어떤 행위가 금지된다는 것은, 그 행위를 하지 않을 의무가 있음을 의미한다
P(p) ↔ ¬F(p)	허용은 금지의 부정	어떤 행위가 허용된다는 것은, 그 행위가 금지되어 있지 않음을 의미한다
O(p) → P(p)	의무는 허용을 함의	어떤 행위를 해야 할 의무가 있다면, 그 행위를 하는 것은 당연히 허용된다

다. 법률 온톨로지 (Legal Ontology)

법률 온톨로지는 법률 도메인의 개념과 관계를 형식적으로 정의한 명세(specification)이다. 온톨로지는 지식 베이스의 구조적 골격을 제공하며, 시스템 간 지식 교환과 추론의 기반이 된다.

1) 법률 온톨로지의 계층

법률 온톨로지는 추상화 수준에 따라 (표2-21)과 같이 계층화 된다.

(표2-21) 법률 온톨로지 계층

계층	설명	예시
상위 온톨로지 (Upper Ontology)	도메인 독립적 기본 개념 (존재, 시간, 공간, 부분관계 등)	DOLCE, SUMO, BFO
핵심 온톨로지 (Core Ontology)	법률 도메인 공통 개념 (규범, 행위, 역할, 문서 등)	LKIF Core, LRI-Core

계층	설명	예시
도메인 온톨로지 (Domain Ontology)	특정 법률 분야 개념 (형법, 민법, 노동법 등)	노동법 온톨로지, 세법 온톨로지 등
응용 온톨로지 (Application Ontology)	특정 시스템/서비스용 개념 (판결문 분석, 계약서 검토 등)	판례 분석 온톨로지

2) 온톨로지와 지식 그래프의 통합

최근 법률지식시스템에서는 온톨로지를 지식그래프(Knowledge Graph)의 스키마 계층으로 활용하는 통합 접근법이 주목받고 있다. Legal-Onto 모델(Pham et al., 2024)은 온톨로지를 개념 계층(Conceptual Layer)으로, 지식그래프를 구현 계층(Implementation Layer)으로 결합하여 의미 기반 법률정보 검색의 효율성을 높였다.

3) 온톨로지와 LLM의 통합

최근 법률지식시스템에서는 대규모언어모델(LLM)의 자연어 이해·생성 능력을 활용하여 법률정보 검색, 질의응답, 문서 요약, 규정 해석 지원 등의 기능을 고도화하려는 시도가 확대되고 있다. 그러나 LLM은 학습 데이터 기반의 확률적 생성 모델이라는 특성상, 법률 분야에서 요구되는 정확성, 일관성, 근거 제시, 최신성을 안정적으로 보장하기 어렵고, 특히 법률 개념의 경계, 규범 구조(요건-효과), 예외·특례, 상하위 법령 관계와 같은 구조적 지식을 명시적으로 유지하는 데 한계가 있다.

이러한 한계를 보완하기 위해, 최근에는 온톨로지를 LLM의 의미 제약 및 지식 정렬(semantic grounding) 계층으로 결합하는 통합 접근이 주목받고 있다. 즉, 온톨로지는 법률 개념, 속성, 관계, 규범 구조를 명시적으로 정의하는 개념·의미 계층(Conceptual/Semantic Layer) 으로 기능하고, LLM은 이를 바탕으로 질의 해석, 자연어 생성, 사례 설명, 정보 추출을 수행하는 언어 처리 계층(Language Interaction Layer) 으로 기능한다. 이와 같은 구조는 LLM의 유연한 언어처리 능력과 온톨로지의 정합성·추적 가능성을 결합하는 뉴로심볼릭(neuro-symbolic) 법률지식처리의 실현 방식으로 볼 수 있다.

라. 사용자 인터페이스 (User Interface)

사용자 인터페이스는 사용자와 법률지식시스템 간의 상호작용을 담당하는 모듈이다. 법률 도메인의 특수성을 고려하여, 다양한 사용자 유형(법률 전문

가, 일반인, 기업 및 공공기관)에 맞는 인터페이스 설계가 필요하다.

1) 인터페이스 유형

- 질의응답 인터페이스: 자연어 질문에 대한 법률정보 제공 (예: Lawbot)
- 폼 기반 인터페이스: 구조화된 입력을 통한 법률 상담 (예: 법률구조공단 상담 또는 행정의 규범 분석)
- 문서 분석 인터페이스: 법률 문서 업로드 및 분석 결과 제공
- 대화형 인터페이스: 챗봇 기반의 단계별 법률 안내
- 시각화 인터페이스: 법령 체계도, 판례 관계 그래프 등

2) 접근성 고려사항

법률지식시스템의 사용자 인터페이스는 다음 사항을 고려해야 한다.

- 법률 용어의 쉬운 해설 제공 (예: 찾기쉬원 생활법령정보)
- 다국어 지원(외국인 접근성)
- 모바일 환경 최적화 및 시각 장애인을 위한 스크린리더 호환성 등

마. 지식 획득 모델(Knowledge Acquisition Module)

지식 획득 모듈은 새로운 법률지식을 시스템에 추가하거나 기존 지식을 갱신하는 기능을 담당한다. 법률 도메인에서는 법령 개정, 새로운 판례, 학설 변화 등을 지속적으로 반영해야 하므로 이 모듈의 중요성이 크다.

1) 지식 획득 방법

- 전문가 인터뷰: 법률 전문가로부터 암묵적 지식 추출
- 문서 분석: 법령, 판례, 학술문헌에서 지식 자동 추출
- 기계학습: 대량의 법률 데이터에서 패턴 학습
- API 연동: 국가법령정보센터 등 외부 시스템과 실시간 연계

2) 법령 개정 반영 자동화

법률지식시스템은 법령의 빈번한 개정을 신속히 반영해야 한다. 국가법령정보센터 Open API를 활용한 자동 업데이터, 신구조문 대조표 기반 변경사항 추적, 관련 규칙의 연쇄 수정 등의 기능이 필요하다.

바. 설명 기능(Explanation Facility)

설명 기능은 시스템의 추론 과정과 결론의 근거를 사용자에게 투명하게 제시하는 모듈이다. 법률 분야에서는 결론의 법적 근거 제시가 필수적이므로, 설명 기능의 중요성이 특히 높다.

1) 설명의 유형

- How 설명: 어떤 규칙이 적용되어 결론에 도달했는지 추론 체인 제시
- Why 설명: 특정 정보가 왜 필요한지 추론 목표 설명
- Why-Not 설명: 특정 결론에 도달하지 못한 이유 설명
- What-If 설명: 가정 변경 시 결과 변화 시뮬레이션

2) 법적 근거 제시

법률지식시스템의 설명 기능은 단순하 추론 과정 설명을 넘어, 법적 근거(법령 조문, 판례, 학설)를 명시적으로 인용해야 한다. 이는 시스템 결과의 신뢰성을 높이고, 사용자가 독립적으로 결론을 검증할 수 있게 한다.

3. 구성요소 간 상호작용 및 통합 아키텍처

가. 구성요소 간 의존관계

법률지식시스템의 6가지 핵심 구성요소는 독립적으로 기능하지 않고, 복잡한 상호의존관계를 형성한다. FOLaw[41]의 인식론적 프레임워크에 따르면, 지식 유형 간에도 의존 관계가 존재한다.

Breuker & Hoekstra[42]는 FOLaw가 “법이 무엇인가”를 추상적으로 정의하는 개념 온톨로지라기보다, “법률가가 어떻게 알고 추론하는가”를 구조화하는 인식론적 프레임워크에 가깝다고 평가한다.

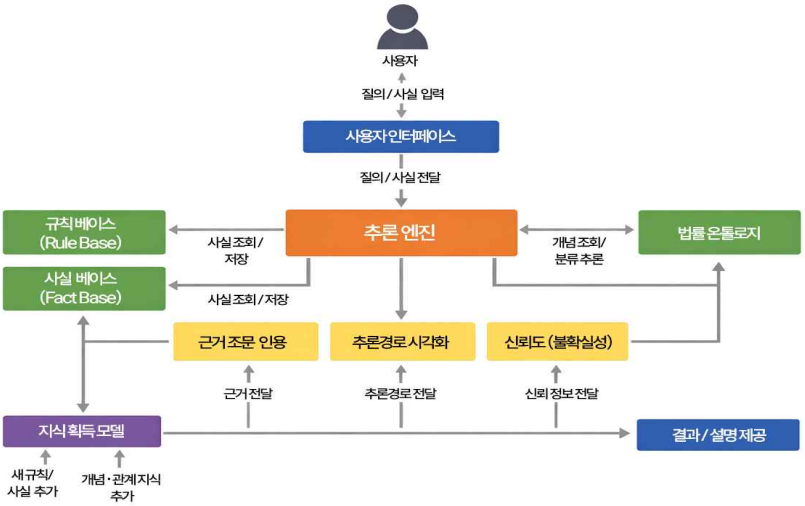
1) 구성요소 간 데이터 흐름

법률지식시스템에서 6가지 구성요소는 독립적으로 작동하기보다, 질의 처리 – 지식 참조 – 추론 – 설명 – 지식 갱신이 순환하는 상호의존적 데이터 흐름을 형성한다. 먼저 사용자 인터페이스는 사용자의 질의와 사실 입력을 수집·정규화하여 추론 엔진으로 전달함으로써 시스템의 추론 과정을 개시한다. 추론 엔진은 질의 해석과 결론 도출을 위해 지식 베이스와 양방향으로 연동되며, 규

41) Valente(1995): 법적 추론에 관여하는 지식 유형들 사이의 기능적 의존성(functional dependencies)을 체계적으로 명시

42) Joost Breuker 외, “Epistemology and ontology in core ontologies: FOLaw and LRI-Core, two core ontologies for law”, Phycologia, 2004

칙을 조회하는 동시에 추론 과정에서 도출된 새로운 사실을 저장·갱신한다. 또한 추론 엔진은 법률 온톨로지와의 상호작용을 통해 개념 정의를 참조하고, 법률 개념의 분류·관계 추론을 수행함으로써 의미론적 일관성과 개념 기반 추론의 정합성을 확보한다.



(그림2-10) 법령지식시스템 구성요소 간 데이터 흐름

추론 결과의 신뢰성과 투명성을 제고하기 위해, 추론 엔진은 설명 기능으로 추론 체인(적용 규칙, 참조 개념, 도출 사실의 연쇄)을 전달하고, 설명 기능은 이를 바탕으로 사용자가 이해 가능한 설명 텍스트를 생성하여 사용자 인터페이스에 제공한다. 한편 시스템의 지속적 확장과 최신성 유지를 위해 지식 획득 모듈은 외부 자료, 전문가 입력, 학습 결과 등으로부터 새로운 지식을 추출·정제하고, 이를 지식 베이스 및 온톨로지에 추가함으로써 규칙·사실·개념 지식의 품질과 커버리지를 강화한다. 결과적으로 본 데이터 흐름은 **"질의 입력 → 추론 및 지식 참조/갱신 → 설명 생성 → 사용자 제공 → 지식 확장"** 의 구조를 가지며, 각 구성요소 간 연계가 시스템 성능(정확성·설명가능성·적응성)을 좌우하는 핵심 메커니즘으로 작동한다.

2) 지식유형 간 의존 관계

Breuker & Hoekstra(2004)[43]는 법률지식시스템에서 지식이 단일 층위로 존재하는 것이 아니라 지식이 단일 층위로 존재하는 것이 아니라, 지식 유형 간 위계적·인과적 의존구조를 이룬다고 설명한다. 즉, 법적 판단과 추론은 규범(의무·금지·권한 등) 자체만으로 완결되지 않으며, 규범이 적용되는 대상 세계에 대한 이해, 규범 위반의 평가, 그에 따른 책임 확정, 그리고 최종적인 법

43) Joost Breuker and Rinke Hoekstr, "DIRECT: Ontology-based Discovery of Responsibility and Causality in Legal Case Description", Legal Knowledge and Information Systems, 2004

적 반응(제재·구제 등)으로 단계화된 지식 연쇄를 필요로 한다.

첫째, 규범적 지식(normative knowledge)은 세계 지식(world knowledge)에 의존한다. 규범은 현실 세계의 행위자, 행위, 객체, 상황 및 사실관계를 전제로 효력을 갖기 때문에, 법이 규율하는 대상에 대한 개념적·사실적 이해 없이는 규범의 의미 해석과 적용 조건(요건) 판단이 불가능하다. 다시 말해, 세계 지식은 규범적 지식의 적용 범위와 의미론적 경계를 규정하는 기초 토대이다.

둘째, 책임 지식(responsibility knowledge)은 규범적 지식에 의존한다. 책임의 성립은 일반적으로 '무엇이 요구되었는가(규범의 내용)'와 '그 규범이 위반되었는가(위반 판단)'를 전제로 하므로, 선행 단계에서 규범의 존재, 적용 가능성, 위반 여부가 확정되어야 책임 귀속이 논리적으로 가능해진다. 따라서 책임 지식은 규범적 지식에 의해 제공되는 의무·금지·권한 등 규범적 기준을 입력으로 하여, 위반의 법적 평가와 귀속 판단을 구성한다.

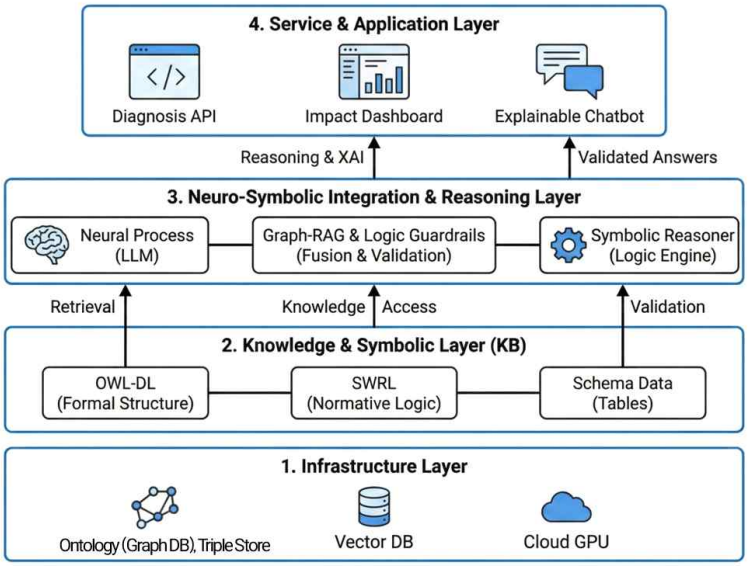
셋째, 반응적 지식(reactive knowledge)은 책임 지식에 의존한다. 법적 반응(제재, 행정처분, 손해배상, 권리구제 등)은 단순한 사실 발생이 아니라 책임의 확정 또는 책임 요건 충족을 전제로 발생하는 후속 효과이기 때문이다. 즉, 반응적 지식은 책임 성립의 결과를 조건으로 하여 어떤 법적 조치가 선택·적용되는지, 그 강도·형태·절차가 어떻게 구성되는지를 산출하는 단계적 지식으로 이해된다.

넷째, 창조적 지식(creative knowledge)은 규범적 지식에 의존한다. 창조적 지식은 계약·설립·등기·허가 등과 같이, 특정 행위나 절차의 충족을 통해 새로운 법적 상태(권리·의무·지위)의 발생 또는 변경을 다루는데, 이때 '효력 발생 조건(요건)'과 '효력의 내용(효과)'은 규범에 의해 정의된다. 따라서 창조적 지식은 규범적 지식이 제공하는 구성요건과 효과 규정을 기초로, 법적 효력의 생성·변동을 설명하고 판단한다.

요컨대 Breuker & Hoekstra(2004)의 관점에서 법률지식은 세계 지식 → 규범적 지식 → (책임 지식 → 반응적 지식) 및 규범적 지식 → 창조적 지식으로 이어지는 의존 그래프를 형성하며, 이러한 구조는 법률 추론이 '사실 인식–규범 적용–위반 평가–책임 확정–제재/구제'라는 일련의 논리적 절차를 통해 체계적으로 수행됨을 시사한다.

나. 통합 시스템 아키텍처

현대 법률지식시스템의 통합 아키텍처는 전통적인 전문가 시스템 구조를 기반으로 하면서, 웹 기술과 AI 기술을 통합한 형태로 발전하고 있으며 통합 계층도의 주요 구성 요소와 역할은 (표2-23)로 요약된다.



(그림2-11) 통합 시스템 구성도

1) 지식 구축 및 심볼릭 계층(Knowledge & Symbolic Layer)

- 정형적 체계 그래프: 법-시행령-시행규칙의 위계와 조-항-호-목의 구조, 참조 관계를 온톨로지DB 또는 그래프 DB(Triple Store)에 저장하여 개정 시 연쇄 영향(Propagation)을 추적
- 규범명제 지식 베이스: 주체-행위-요건-효과-예외를 SWRL 및 의무논리 형식으로 모델링하여 기계적 추론이 가능한 형태로 관리
- 별표/서식 스키마: 부속 문서를 데이터 테이블화 하여 수치 계산과 절차 검증을 자동화

2) 신경망 처리 및 검색 계층 (Neural & Retrieval Layer)

방대한 법률 텍스트를 처리하고 사용자 질의를 이해하는 '유연한 지능' 영역이다.

- 법률 특화 LLM: 법률 도메인에 특화된 언어 모델을 통해 복잡한 법률 문맥 파악
- 임베딩 및 벡터 DB: 조문 텍스트와 판례를 벡터화하여 시맨틱 검색(Semantic Search) 기능을 제공
- RAG (Retrieval-Augmented Generation): 답변 생성 시 관련 법조문을 실시간으로 검색하여 LLM에 전달함으로써 정확도를 향상

3) 뉴로-심볼릭 계층 (Neuro-Symbolic Integration Layer)[44] 결합

이 계층은 신경망의 유연한 자연어 처리 능력과 기호 추론의 엄밀한 논리 검증 능력을 결합하여 '설명 가능하고 신뢰할 수 있는 법률 판단'을 생성하는 역할을 수행하는 핵심 엔진으로 구성된 계층이다.

- Graph-RAG (Knowledge-Augumented RAG): 단순 텍스트 검색을 넘어, 사용자의 질문과 연관된 온톨로지 내의 상위법, 참조 조문, 규범명제를 패키지 형태로 추출하여 LLM에 전달
- 논리 가드레일 (Logic Guardrails): LLM이 생성한 답변이 온톨로지에 정의된 규범적 논리(예: O(p)↔¬P(¬p))나 SWRL 규칙과 충돌하는지 실시간으로 검증. 만약 LLM이 "연소자는 1주 45시간 근로가 가능합니다"라고 환각을 일으키면, 심볼릭 엔진이 이를 차단하고 정정
- 시맨틱 매핑(Semantic Mapping): 사용자의 일상적인 질문 (예: "알바생이 일주일 넘게 일했는데..")을 온톨로지상의 개념으로 변환하여 정밀한 쿼리를 수행하게 함
- 추론 경로 가시화(XAI): 심볼릭 추론 엔진이 내린 결론의 경로를 LLM이 자연어로 변환하여 사용자에게 제공

뉴로-심볼릭 계층 내에서 두 기술은 다음과 같이 보완적으로 작동한다.

(표2-22) Neural vs Symbol vs Neuro-Symbol 통합 비교

구분	Neural (신경망-LLM)	Symbol(기호-온톨로지)	Neuro-Symbol
강점	자연어 이해, 유연한 소통	엄밀한 논리, 근거 기반 추론	정확한 논리에 기반한 자연어 답변
역할	인터페이스 및 문맥 파악	규제 판단의 '준거' 제공	할루시네이션이 억제된 법률 지원
비유	말을 잘하는 '변호사'	법전을 통달한 '판사'	실무에 능통하면서도 법리에 완벽한 전문가

(표2-23) 통합 시스템 계층도 주요 구성 요소와 역할

계층	주요 구성 요소	역할
인프라	온톨로지DB, Vector DB, 클라우드 GPU	정형/비정형 데이터의 통합 저장
지식 계층	OWL-DL (정형 구조) + SWRL (규범 명제)	법의 논리적 구조와 시맨틱 체계 구현
추론 계층	논리 추론기(Symbolic) + LLM (Neural)	정확도와 일관성을 동시에 확보한 규제 판단
서비스 계층	대화형 규제 진단, 개정 영향도 대시보드	실시간 준수 및 현장 의사결정 지원

44) Chen Li외, 'SOLAR: Scalable Optimization of Large-scale Architecture for Reasoning', arXiv, 2025

나. 법률정보(지식)시스템 현황

우리나라의 주요 법률지식시스템은 법령, 판례, 행정규칙, 해석례 등을 제공하는 공공 및 민간 데이터베이스를 중심으로 운영되며, 가장 대표적인 시스템은 법제처의 ‘국가법령정보센터’이며 주요 법률지식시스템 및 서비스는 다음과 같다.

1) 국가 운영 공공 법률정보시스템

국내 공공 법률정보서비스는 대체로 (i) 현행 법령의 권위 있는 제공·연계, (ii) 판례 중심의 해석·적용 정보 제공, (iii) 국민 생활 중심의 이해 지원(해설·분류), (iv) 자치법규·해외법령 등 범위 확장의 축으로 분화되어 운영된다. 이들 시스템은 공통적으로 방대한 법률 텍스트를 축적·제공하되, 서비스 목적에 따라 데이터 조직 방식과 검색·탐색 기술을 상이하게 채택한다.

첫째, 국가법령정보센터는 법령·행정규칙·자치법규 연계, 법령 체계도, 신구법 및 비교, 위임법령 조회 등 “현행 규범의 통합 제공과 구조화 조회”를 핵심 기능으로 한다. 특히 OpenAPI를 통해 법령 목록·본문·조항호목, 신구법/3단 비교, 법령 체계도, 위임법령 조회 등 다양한 구조화 데이터를 외부 시스템에서 활용 가능하도록 제공함으로써, 법률정보를 단순 열람을 넘어 서비스·시스템 연계 자원으로 확장하고 있다.

둘째, 지능형 법령검색(Lawbot)은 국민이 정확한 법령명이나 법률 용어를 알지 못하더라도 일상 용어·문장 기반 질문으로 법조문을 탐색할 수 있도록 설계된 AI 기반 검색 서비스다. 공개된 자료에 따르면 질문 의도 분석과 더불어, 법령용어-일상용어 관계 및 법령 간 관계를 포함하는 지식베이스를 구축하고, 지식그래프를 활용한 검색 및 관련 법령 추천·시각화 기능을 고도화하는 방향으로 발전해 왔다. 이는 전통적 키워드 검색의 진입 장벽을 낮추고 “연관 규범 탐색”을 강화한다는 점에서 의미가 있으나, 검색 결과가 곧바로 ‘적용 규범의 확정’으로 이어지기 위해서는 규범의 요건·예외·시간효력까지 정밀하게 설명하는 체계가 추가로 필요하다.

셋째, 대법원 종합법률정보 계열 서비스는 판례를 중심으로 법령·규칙·예규 등과의 통합 탐색을 지원하여, 규범 텍스트만으로는 포착하기 어려운 “해석과 적용의 층위”를 제공한다. 즉 법령이 선언하는 규범이 실제 사건에서 어떻게 해석되고 적용되는지를 확인할 수 있는 축을 공식적으로 제공한다는 점에서 법률지식시스템 연구의 필수 기반이 된다. 다만 판례는 사실관계가 다양하고 서술 구조가 길어, 기계적 처리와 서비스화(요건 매핑, 설명 가능성 확

보)를 위해 별도의 구조화·정규화가 요구된다.

넷째, 찾기 쉬운 생활법령정보는 법령을 정부 조직 중심이 아니라 국민의 생활 주제 중심으로 재분류하고, 어려운 법령 내용을 알기 쉽게 해설하는 방식으로 "이해 지원"에 초점을 둔다. 이는 대국민 서비스 관점에서 접근성과 학습성을 크게 높이지만, 서비스가 '쉬운 설명' 수준을 넘어 개인의 구체적 상황에 대한 적용 판단(요건 충족 여부, 예외·특례의 적용, 경과규정 반영)까지 수행하려면, 원문 규범의 구조적 표현과 추론 가능한 지식모형이 뒷받침되어야 한다.

다섯째, 자치법규정보시스템(ELIS)은 전국 지자체의 조례·규칙을 통합 제공하고, 최근 제·개정, 입법예고, 비교검색 등을 지원함으로써 중앙법령 중심 시스템에서 놓치기 쉬운 "현장 집행 규범"을 포괄한다. 그러나 자치법규는 지자체별 표현·형식 편차가 커서 전국 단위 지식서비스로 통합하려면 표준화와 관계 정규화 비용이 상대적으로 크다.

여섯째, 세계법제정보센터는 주요 국가의 법령 원문·요약·번역과 입법동향, 연구보고서를 제공하고 법령체계도를 함께 제공하는 등 해외법령 정보를 수집·가공하여 비교법·정책연구 및 해외진출 지원에 활용되도록 구성된다. 다만 국가·언어·분야별 커버리지와 현행화 시차, 번역의 범위·품질 같은 구조적 제약이 존재할 수 있어, 서비스 적용 시 범위·신뢰성 관리가 중요하다.

마지막으로 입법부 영역에서는 국회 및 국회도서관 포털을 중심으로 법률자료와 입법 관련 정보가 제공되며(예: 국회도서관 의회법률정보포털), 이는 "현행 규범(집행 중심)"과는 다른 관점에서 입법 과정 및 참고자료 기반 분석에 유리하다.

(표2-24) 국가 운영 공공 법률정보시스템 현황

순번	1	시스템명	국가법령정보센터 (www.law.go.kr)		운영기관	법제처
주요 기능			적용 기술	장점		단점
법령(법률·대통령령·부령), 행정규칙(훈령·예규·고시), 자치법규(조례·규칙), 판례·해석례, 별표·서식 등 통합 제공			- 대규모 법령·조약·판례 DB - 정형·반정형 법령메타데이터, - 고급 키워드 검색·필터, - 관계형 DB - 전문검색엔진 - API·공동활용 인터페이스	- 국가 차원의 가장 방대한 법령·조약·행정규칙 DB, - 최신성·신뢰도가 매우 높음, - 구조화된 메타데이터로 정밀 검색 가능, - 다른 공공·민간 시스템이 표준 레퍼런스로 활용		- 법령 구조·용어에 익숙하지 않으면 검색 난이도 높음 - 기본 검색은 키워드 중심이라 질문 의도 이해에는 한계
순번	2	시스템명	지능형 법령검색 (Lawbot)		운영기관	법제처
주요 기능			적용 기술	장점		단점

일상 질문(자연어)로 법조문·서식 탐색	- 자연어 질의 처리(NLU), - 의미 기반 검색 - 지식그래프(법령·용어·일상어 관계 약 70만~75만건), - Q&A 데이터(2만건대) 기반 머신러닝·생성형 AI	- 일상 언어·불완전한 용어로 관련 조문 검색 가능, - 질문 의도 파악을 통한 의미 검색 - 시각화된 지식그래프 제공	- 학습 데이터·지식베이스 범위 밖 질의에 취약, - AI 추천 결과의 법적 정확성 검증이 필요, - 복잡한 쟁점·판례까지는 아직 심층 법률해석 기능 부재

순번	3	시스템명	국회법률정보시스템(likms.assembly.go.kr/law	운영기관	국회사무처
주요 기능		적용 기술	장점		단점
국회에서 운영하며, 제·개정 법률, 의안, 법제 현안 및 참고자료를 전문적으로 제공		- 입법 과정 중심 메타데이터(의안, 심사보고서, 회의록 연계), - 고급 검색·필터, 버전 관리(제·개정 이력), - 관계형 DB와 의안/법률 연계 검색	입법 과정(의안 중심) 분석·추적에 강점(법제처 '현행 법령 DB'와 상보적)		접근/공개범위, 제공형태가 "현행 집행규범"보다 입법 자료 중심일 수 있어 서비스 목적에 따라 병행 필요

순번	4	시스템명	대법원 종합법률정보(glaw.scourt.go.kr)	운영기관	대법원
주요 기능		적용 기술	장점		단점
판례, 대한민국 법령, 대법원 규칙·예규·내규, 법률문헌을 "정보검색시스템" 형태로 제공		- 방대한 판례·선고문 DB, - 판례 메타데이터(사건종류, 법원, 선고일 등) - 고급 키워드·조건 검색	- 판례 검색에 특화 - 선고문 원문과 법령·판례 인용관계 확인 가능 - 실무에서 활용되는 신뢰도 높은 공식 판례 데이터 제공		- 자연어 질의·의도 기반 탐색은 상대적으로 제한 - 비전문가가 키워드·사건 유형을 선정하기 어려울 수 있음 - 실무자 중심 인터페이스

순번	5	시스템명	찾기 쉬운 생활법령정보(easylaw.go.kr)	운영기관	법제처
주요 기능		적용 기술	장점		단점
국민 생활 주제 기반 콘텐츠(250여 개) 제공, 연령·직업 등 조건 기반 "맞춤형 생활법령"		- 생활주제 분류체계(콘텐츠 큐레이션) - 사용자 프로필(연령/직업/상황) 기반 개인화 제공	"쉬운 설명"과 맥락형 안내에 강점(대국민 서비스에 유리)		원문 법령·판례 수준의 세밀한 탐색에는 한계, 주제화되지 않은 세부 규정까지 포괄하지는 못함

순번	6	시스템명	자치법규정보시스템(ELIS)	운영기관	행안부
주요 기능		적용 기술	장점		단점
전국 자치법규(조례·규칙 등), 입법예고, 현황 등의 비교검색 제공		- 지자체별 자치법규 메타데이터/전문검색 - 최근 제·개정 추적 - 비교검색 기능	중앙법령 중심 DB에서 놓치기 쉬운 현장 집행 규범(조례/규칙) 확보에 강점		지자체별 문서 품질·표현·구조 편차가 커서 "전국 단위 서비스"는 정규화 비용이 큼

순번	7	시스템명	세계법제정보센터 (world.moleg.go.kr)	운영기관	법제처
주요 기능		적용 기술	장점		단점
해외 주요 법령·연구보고서·입법동향 수집·가공 제공		- 다국어·다국가 법령 DB - 국가·분야별 분류 - 메타데이터 기반 검색 - 해외 입법 동향·연구보고서 콘텐츠 관리	- 여러 국가 법령·입법 동향을 한 번에 조회 가능 - 비교법·입법정책 연구에 적합 - 영문 등 다국어 자료 제공.		국가별 업데이트 주기·포함 범위가 상이, 원문·현행성 확인에 추가 검증 필요

요약하면, 현행 공공 시스템들은 데이터 축적·권위 있는 제공, 검색 편의, 생활형 해설, 판례 제공, 자치·해외 영역 확장 등에서 강점을 보인다. 반면 공통적으로 규범의 의미 단위(주체-행위-객체-요건-효과)로의 분해, 예외·특례·준용의 '수정관계' 모델링, 위임·참조 관계의 유형화, 시간적 효력(시행·경과·적용례) 반영, 다법령 적용 시 상충 해결의 설명 가능성까지를 일관되게 처리하는 '지식추론형' 구현은 전무하다.

따라서 법률지식시스템이 "검색/열람"을 넘어 "적용·설명·추론" 수준의 서비스를 목표로 할 경우, 텍스트 중심 인덱싱만으로는 한계가 발생하며, 온톨로지(관계·정규화·규범명제 구조)와 AI(자연어 이해·의미역할 추출·요약·질의응답)의 결합이 요구된다.

2) 민간 및 기타 법률정보시스템

민간 영역의 법률정보 시스템은 공공 법률정보서비스(현행 법령의 공적 제공)와 달리, 실무·시장 수요(로펌/기업 법무, 변호사-의뢰인 매칭, 고도화된 검색, 계약·문서 자동화, 데이터 분석)에 초점을 맞추며 서비스가 분화되어 있다. 본 연구에서 검토한 민간 법률정보 시스템은 대체로 (i) 상용 법률정보 DB, (ii) 변호사·법률서비스 플랫폼, (iii) 지능형 법률검색, (iv) 법률정보·리포트 기반 서비스의 네 유형으로 구분할 수 있다.

첫째, 상용 법률정보 DB는 판례·법령·실무자료를 통합한 데이터베이스를 기반으로 전문 검색과 해설, 실무서 연계를 제공하며, 로펌·기업 법무 업무에서 활용되는 검색(서치) 시스템으로 자리한다. 이 유형은 리걸테크 이전부터 존재해 온 전통적 상용 법률DB 서비스로서, 데이터의 축적과 전문적 검색·해설 제공에 강점을 가진다.

둘째, 변호사·법률서비스 플랫폼은 변호사 검색·매칭을 핵심으로 사건분야·지역·경력 등의 필터링, 온라인 상담 및 사건관리 등 서비스 중개 기능을 제공하고, 최근에는 AI 기반 상담 및 문서 자동화 등으로 고도화를 추진하는 경향이 나타난다. 즉, '법률 플랫폼'의 네트워크 효과에 AI-SaaS 성격의 문서/업무 자동화를 결합하여 확장하는 모델로 정리된다.

셋째, 지능형 법률검색은 판례·법령 등 법률데이터의 풀텍스트 인덱싱(full-text indexing)[45]을 기반으로, 다단계 동시단어 검색, 검색 내 검색, 관련 법령 다이어그램 제공 등 고급 탐색 기능을 강조한다. 이는 공공 검색이 제공하는 범용 검색을 넘어, 실무자 관점에서 검색 정밀도·탐색성(발견 가능성)을

45) 풀텍스트 인덱싱: 방대한 텍스트 데이터 속에서 특정 단어나 문구를 빠른 속도로 찾아내기 위해 데이터의 '모든 단어'를 미리 정리해두는 기술

높이는 방향으로 설계된 유형이다.

넷째, 법률정보・리포트 기반 서비스는 단일 제품형 서비스라기보다, 다양한 리걸테크・컨설팅・데이터 기업들이 판례・입법 동향 분석, 리걸 데이터 분석 리포트 제공, 계약서 자동작성・검토 등 부가 기능을 결합하여 제공하는 형태로 나타난다. 이 유형은 특정 브랜드의 통합 플랫폼보다는, 영역별 특화 서비스가 분화・조합되는 경향이 강하다.

(표2-25) 민간 운영 법률정보시스템 현황

유형	대표 서비스	적용 기술	강점	한계
상용 법률정보 DB	로앤비 등	판례·법령·실무자료 DB, 전문 검색, 해설/실무서 연계, 로펌·기업 법무용 서치	전문성·검색/해설 품질, 실무 워크플로 친화	'적용 판단'까지는 제한 (텍스트/검색 중심), 서비스 이용 비용/접근성 이슈 가능
변호사·법률서비스 플랫폼	로톡 등	변호사 검색·매칭, 사건/지역/경력 필터, 온라인 상담·사건관리, AI 상담·문서 자동화	이용자 관점 UX, "상담–문서–사건관리"로 서비스 연계	규범 적용의 설명가능성/책임소재, 상담 자동화의 정확성·근거 제시 문제
지능형 법률검색	리걸서치 등	풀텍스트 인덱싱, 다단계 검색, 검색 내 검색, 관련 법령 다이어그램	탐색성 강화(발견 가능성), 실무형 검색 정밀도	법령의 예외·특례·준용/ 시간 효력까지 반영한 규범 추론은 별도 모델 필요
법률정보·리포트 기반	다양한 기업	판례·입법 동향 분석, 리걸데이터 분석/리포트, 계약서 자동작성·검토 결합	"분석·문서 자동화"로 부가가치 창출, 기업 법무 수요 대응	데이터/규범 구조가 표준화되지 않으면 재현성·추적성 확보 어려움

민간 법률정보 시스템의 공통적 특징은 다음 세 가지로 요약된다.

첫째, 단순 검색에서 업무로의 확장으로서, 민간 서비스는 단순 열람・검색을 넘어 상담, 사건관리, 계약・문서 작성 및 검토 등 업무 프로세스 단위로 기능을 확장하고 있다. 즉, 법률정보를 '지식' 이 아니라 '업무 수행 도구' 로 전환하는 방향성이 강하다.

둘째, 현 단계에서 민간 서비스의 AI는 대체로 상담 응대, 문서 자동작성 및 검토, 요약/분류 등 비정형 텍스트 처리 자동화에 우선 적용되는 경향이 크다. 반면, 예외・특례・준용, 위임범위, 상충 해결, 시간적 효력까지 반영하는 정밀 규범 추론은 구현 난이도가 높아 보조적 수준에 머물 가능성이 높다.

셋째, 법률정보는 '조문 간 참조 및 준용' 과 '시행・경과・적용례' 처럼 관계와 시간축이 본질적으로 결합되어 있다. 이 구조를 충분히 반영하지 못하면, 서비스는 결국 고급 검색이나 요약을 제공하더라도 사용자의 상황에 대한 적용 가능성 판단과 설명가능한 근거 제시에서 한계를 갖게 된다.

종합하면, 민간 법률정보 시스템은 공공서비스 대비 사용자 경험과 실무 워크플로 측면에서 빠르게 진화하고 있으나, 법이 갖는 관계적·시간적·규범적 복잡성을 일관되게 반영하기에는 텍스트 중심의 검색이나 자동화만으로는 한계가 존재한다. 따라서 법률지식시스템이 "검색/열람"을 넘어 "적용·설명·추론" 수준의 서비스를 제공하려면, **규범명제 단위의 온톨로지(관계·버전·근거의 명시화)**와 **AI(의미역할 추출·질의응답·요약)의 결합**이 요구된다.

제4절. 소결 및 시사점

1. 소결

본 장에서는 법령 온톨로지 기반 지식화 및 추론 체계의 필요성을 세 가지 층위에서 검토하였다.

첫째, 법령 자체의 특성 측면에서는 법이 단순 텍스트가 아니라 문서 구조(편·장·절·조·항·호), 규범 구조(주체·행위·객체·요건·효과), 시간적 효력(시행·경과·적용례), 관계 구조(위계·위임·참조·준용·예외·특례)가 중첩된 복합 지식이라는 점을 확인하였다. 이에 따라 법령 지식화는 조문 저장을 넘어, 규범 의미와 관계를 기계가 해석 가능한 형태로 구조화하는 방향으로 설계되어야 함을 도출하였다.

둘째, 법률 지식 표현 기술 측면에서는 XML 기반 법령 표준, 규칙 기반 표현, 법률 온톨로지, 지식그래프, 뉴로심볼릭 접근이 각각 상이한 강점을 가진다는 점을 검토하였다. 즉, 구조 보존과 버전 관리에는 법령 XML이, 개념·관계의 의미 표현과 정합성 관리에는 온톨로지가, 요건-효과 및 예외 처리에는 규칙/논리 기반 표현이, 자연어 처리 및 확장성에는 LLM 기반 기술이 유효하다. 따라서 단일 기술로 문제를 해결하기보다, 구조-의미-규범-추론을 분리하고 결합하는 계층형 설계가 필요하다는 결론에 이르렀다.

셋째, 법률지식시스템 관점에서는 T-Box/A-Box 분리, 추론 엔진, 설명 가능성, 사용자 인터페이스, 개정 반영 자동화, 외부 데이터 연계(Open API) 등 실제 서비스 구현에 필요한 구성요소를 정리하였다. 또한 공공·민간 법률정보 시스템의 현황 분석을 통해, 현재 시스템들은 권위 있는 데이터 제공과 검색·조회 기능에는 강점을 가지나, 사용자의 구체적 사실관계를 입력받아 규범 적용 여부를 구조적으로 판정하고 근거를 추적·설명하는 기능은 제한적

이라는 점을 확인하였다.

종합하면, 2장은 본 연구가 제안하는 법령 온톨로지 기반 접근이 단순한 데이터 구조 개선이 아니라, 법령의 구조적 복잡성·규범성·시간성·관계성을 통합적으로 다루기 위한 필수적 기반임을 이론적으로 정리한 장이다. 또한 이후 장에서 제시하는 규범명제 모델링, 전방/후방추론 엔진, 개정 영향 탐지 체계의 설계가 왜 필요한지에 대한 선행 논리를 제공한다.

2. 시사점

2장의 검토 결과는 향후 법령 지식화 및 AI 기반 법률서비스 설계에 다음과 같은 시사점을 제공한다.

첫째, 법령 데이터 구축은 조문 원문 수집 중심에서 벗어나, 규범명제 중심의 의미 단위(주체-행위-요건-효과-예외)로 재구성되어야 한다. 이는 단순 검색을 넘어 사실관계 기반 판단, 위반 탐지, 영향 분석으로 확장되는 핵심 전제이다.

둘째, 법률 지식 표현은 단일 저장 방식으로 해결하기 어렵기 때문에, 문서 구조(XML/메타데이터)-의미 구조(온톨로지)-규범 규칙(논리/규칙)-사례 사실(A-Box)를 분리·연계하는 아키텍처가 필요하다. 특히 노동법과 같이 예외·특례·위임·준용이 빈번한 영역에서는 이 계층 분리가 필수적이다.

셋째, 법령 개정이 빈번한 실무 환경을 고려할 때, 시스템은 정적 지식베이스가 아니라 개정 반영 자동화와 버전/시점 관리가 가능한 동적 지식체계로 설계되어야 한다. 즉, “현재 조문 조회”를 넘어 “특정 시점 기준 적용 법령”과 “개정 영향 대상 탐지”까지 지원해야 실무 활용성이 확보된다.

넷째, 실제 적용 단계에서는 추론 결과의 정확도만큼이나 설명 가능성(explainability) 이 중요하다. 법률 분야는 결론보다 근거(적용 조문, 요건 충족 여부, 예외 적용 여부)를 제시해야 신뢰를 얻을 수 있으므로, 추론 과정의 단계별 추적과 근거 제시 기능이 시스템의 핵심 요구사항이 된다.

다섯째, 공공·민간 시스템 분석 결과를 고려할 때, 본 연구의 차별성은 “검색 고도화”가 아니라 규범 적용 판단의 구조화에 있다. 따라서 후속 연구 및 실증에서는 검색 정확도뿐 아니라, 사실관계 입력의 정규화 수준, 규범명제 매핑 정확도, 추론 경로 설명 가능성, 개정 영향 탐지의 정밀도·재현율 등 실질적 성능지표를 함께 설정할 필요가 있다.

제3장 노동법 온톨로지

제1절. 노동 관련 법률 지식 구조화

1. 노동 관련 법률 및 구조

가. 노동법 온톨로지 구축을 위한 대상 조사

노동법 온톨로지를 구축하여 법령지식시스템에 활용하기 위해서는 기본적으로 노동 관계 법령의 범위를 파악하는 것이 필요하다. (표3-1)은 공인노무사법 시행령 별표1에 기재된 노동관계 법령의 범위로서 기본 법령과 확장 법령으로 구성된다.

1) 개별적 근로 관계법

개별 근로자와 사용자 간의 근로조건을 규율하는 법률

가) 핵심 기본법

번호	법률	약칭	주요 내용
1	근로기준법	근기법	근로조건의 최저기준 (임금, 근로시간, 해고제한, 휴가 등)
2	최저임금법	-	근로자 최저임금 보장
3	산업안전보건법	산안법	산업재해 예방, 안전·보건 기준
4	산업재해보상보험법	산재보험법	업무상 재해에 대한 보상

나) 고용 관련법

번호	법률	약칭	주요 내용
5	고용보험법	-	실업급여, 고용안정사업, 직업능력개발
6	고용정책기본법	-	고용정책의 기본 방향
7	직업안정법	-	직업소개, 직업지도, 고용정보 제공
8	고용상 연령차별금지 및 고령자고용촉진에 관한 법률	고령자고용법	연령차별 금지, 고령자 고용촉진
9	장애인고용촉진 및 직업재활법	장애인고용법	장애인 의무고용, 직업재활

번호	법률	약칭	주요 내용
10	외국인근로자의 고용 등에 관한 법률	외국인고용법	외국인 근로자 고용절차·보호
11	청년고용촉진 특별법		청년 고용촉진 정책

다) 비정규직・특수형태 근로자 보호법

번호	법률	약칭	주요 내용
12	기간제 및 단시간근로자 보호 등에 관한 법률	기간제법	기간제·단시간 근로자 차별금지, 사용기간 제한
13	파견근로자 보호 등에 관한 법률	파견법	파견근로자 보호, 파견업종 제한
14	가사근로자의 고용개선 등에 관한 법률	가사근로자법	가사서비스 종사자 보호

라) 평등・모성보호 관련법

번호	법률	약칭	주요 내용
15	남녀고용평등과 일·가정 양립 지원에 관한 법률	남녀고용평등법	성차별 금지, 육아휴직, 일·가정 양립
16	고용상 성별등 차별금지 및 구제에 관한 법률	-	고용상 성차별 구제절차

마) 임금・복지 관련법

번호	법률	약칭	주요 내용
17	임금채권보장법		체불임금 대지급금 제도
18	근로복지기본법	-	근로자 복지정책 기본 사항
19	근로자퇴직급여보장법	퇴직급여법	퇴직금.퇴직연금 제도
20	근로자의 날 제정에 관한 법률		5월1일 근로자의 날 지정

바) 직업훈련・능력개발법

번호	법률	약칭	주요 내용
21	근로자직업능력개발법	-	직업훈련, 능력개발 지원
22	숙련기술장려법	-	숙련기술인 우대

사) 특수분야 근로자법

번호	법률	약칭	주요 내용
23	선원법		선원의 근로조건 (근기법 특별법)
24	진폐의 예방과 진폐근로자의 보호 등에 관한 법률	진폐법	진폐 예방·보상
25	건설근로자의 고용개선 등에 관한 법률	건설근로자법	건설일용직 보호
26	항공사업법 (항공기승무원 관련 규정)		승무원 근로조건
27	전공의의 수련환경 개선 및 지위 향상을 위한 법률	전공의법	전공의 근로조건 보호

2) 집단적 노사관계법

노동조합과 사용자(단체) 간의 관계를 규율하는 법률

가) 핵심 기본법

번호	법률	약칭	주요 내용
28	노동조합 및 노동관계조정법	노조법	노동3권(단결권, 단체교섭권, 단체행동권) 보장, 노동쟁의 조정, 부당노동행위
29	노동위원회법	-	노동위원회 설치·운영

나) 노사협력법

번호	법률	약칭	주요 내용
30	근로자참여 및 협력증진에 관한 법률	근참법	노사협의회 설치.운영
31	노사관계발전 지원에 관한 법률	-	노사관계 발전 지원
32	경제사회노동위원회법	-	경사노위 설치.운영

다) 공무원・교원 노동관계법

번호	법률	약칭	주요 내용
33	공무원의 노동조합 설립 및 운영 등에 관한 법률	공무원노조법	공무원 노동조합
34	교원의 노동조합 설립 및 운영 등에 관한 법률	교원노조법	교원 노동조합

3) 사회보험관련법

번호	법률	약칭	주요 내용
35	국민연금법	-	노후소득 보장
36	국민건강보험법	-	의료보장
37	고용보험 및 산업재해보상 보험의 보험료징수 등에 관한 법률	보험료징수법	4대 보험료 징수

4) 공무원 관련법(노동관계 규정)

번호	법률	약칭	주요 내용
38	공무원연금법	-	제4장(급여), 제8장(심사의 청구)
39	공무원 재해보상법	-	제2장(급여), 제5장(심사의 청구)
40	사립학교교직원 연금법	-	급여 및 심사청구 관련 규정
41	군인연금법	-	급여 및 심사청구 관련 규정
42	별정우체국법	-	급여 및 심사청구 관련 규정

5) 안전・재해 관련법

번호	법률	약칭	주요 내용
43	중대재해 처벌 등에 관한 법률	중대재해법	사업주·경영책임자 처벌 (제2장 중대산업재해)
44	석면안전관리법	-	석면 관련 안전관리
45	화학물질관리법	-	유해화학물질 관리

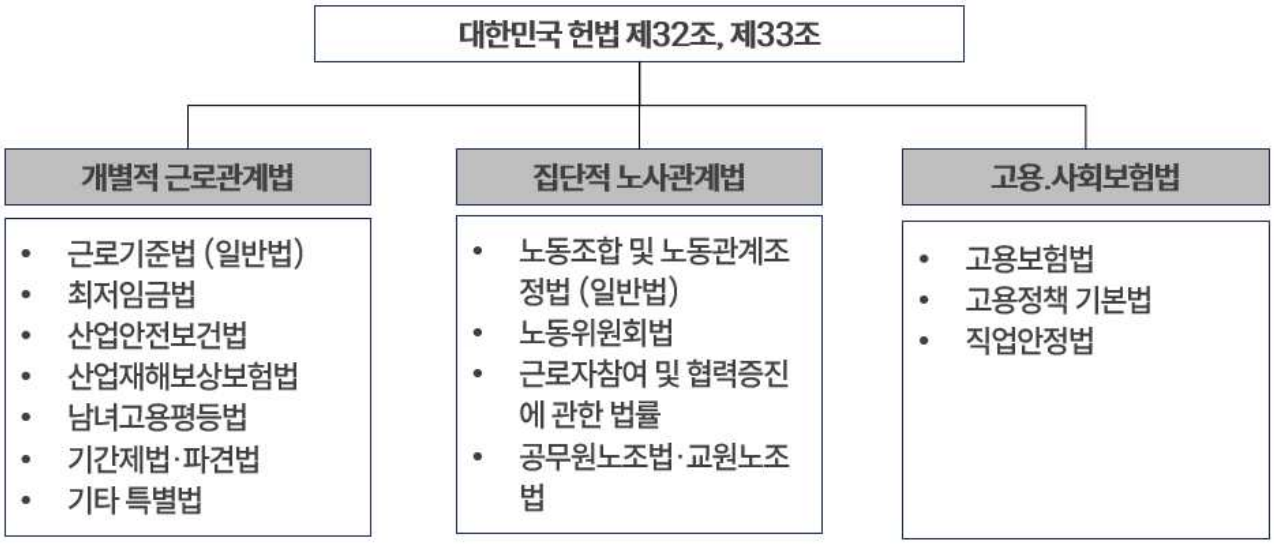
6) 기타 노동관련법

번호	법률	약칭	주요 내용
46	국가인권위원회법	-	노동관계 차별 조사 (제30조 관련)
47	채용절차의 공정화에 관한 법률	-	채용절차 공정성 확보
48	사회적기업 육성법	-	사회적기업 근로자 보호
49	협동조합 기본법		협동조합 근로자 관련
50	대중문화예술산업발전법		연예인 보호
51	영화 및 비디오물의 진흥에 관한 법률		영화근로자 보호

나. 노동법 관계 법률 체계도

노동법 들은 헌법 제32조와 제33조를 기준으로 (그림3-1)과 같이 법령 체계를 갖고 있다. 개별적 근로관계법, 집단적 노사관계법, 고용.사회보험법으로 분류되고 있으며, 각 분야별로 관련된 법률들로 분리될 수 있다.

본 연구는 개별적 근로관계에 해당되는 "근로기준법"을 중심으로 법률 지식 구조화를 처리한 후 온톨로지를 설계・구축하였다.



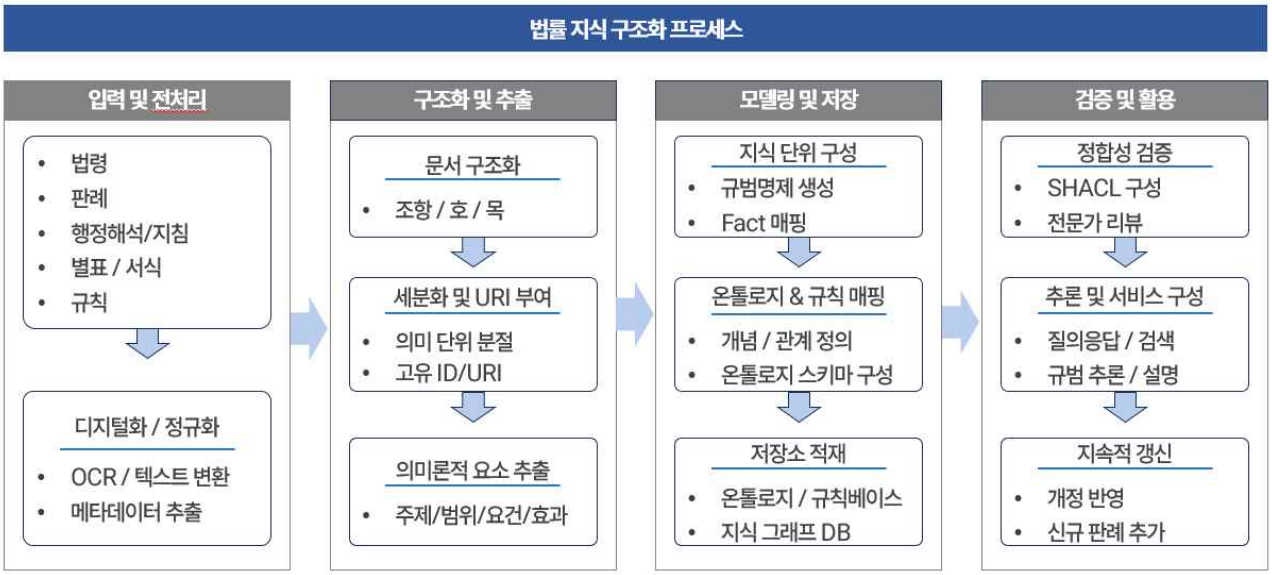
(그림3-1) 노동법 관계 법률 체계

다. 노동 관련 법률 지식 구조화

1) 법률 지식 구조화 개요 및 프로세스

법률 지식 구조화(Legal Knowledge Structuring)란 사람의 언어(Natural Language)로 기술된 법령, 판례, 행정해석 등 비정형 텍스트 데이터를 컴퓨터가 해석하고 추론할 수 있는 **기계 가독형(Machine-readable) 데이터 모델로 변환하는 일련의 공학적 과정**을 의미한다. 즉, 법령・행정규칙・판례・지침・서식 등 법적 규범 요소를 일관된 의미 모델로 조직화하여, 개념・관계・조건・제약을 시스템적으로 표현하는 과정을 의미한다.

이는 단순히 법전의 내용을 디지털 텍스트로 옮기는 전산화(Digitization)를 넘어, 법률 문장 내에 기술된 **주체(Subject), 행위(Action), 요건(Condition), 효과(Effect)** 등의 의미론적 요소를 분해하고, 이들 간의 논리적 관계를 체계적으로 정의하여 '컴퓨팅 가능한 지식(Computable Knowledge)'으로 재구성하는 것으로 단순 텍스트 저장(예: PDF, DB row)이 아니라 **지식 단위(Knowledge Unit)**로서 법의 논리 구조를 재구성하는 행위라 할 수 있다.



(그림3-2) 법률 지식 구조화 프로세스

2) 법률 지식 구조화 프로세스 상세 설계

AI가 정밀한 추론을 수행하기 위해서는 통문장 형태의 조문을 식별 가능한 최소 단위(Granularity)로 쪼개고, 각 단위에 고유한 주소(URI)를 부여해야 한다. 온톨로지 구축의 관점에서 법령의 내부 구조의 분류는 크게 ① 구조적 위계(Structure), ② 문장 성분(Sentence Part), ③ 의미론적 요소(Semantic Element)로 구성하였다.

(표3-1) 온톨로지 구축을 위한 3대 기본 구조

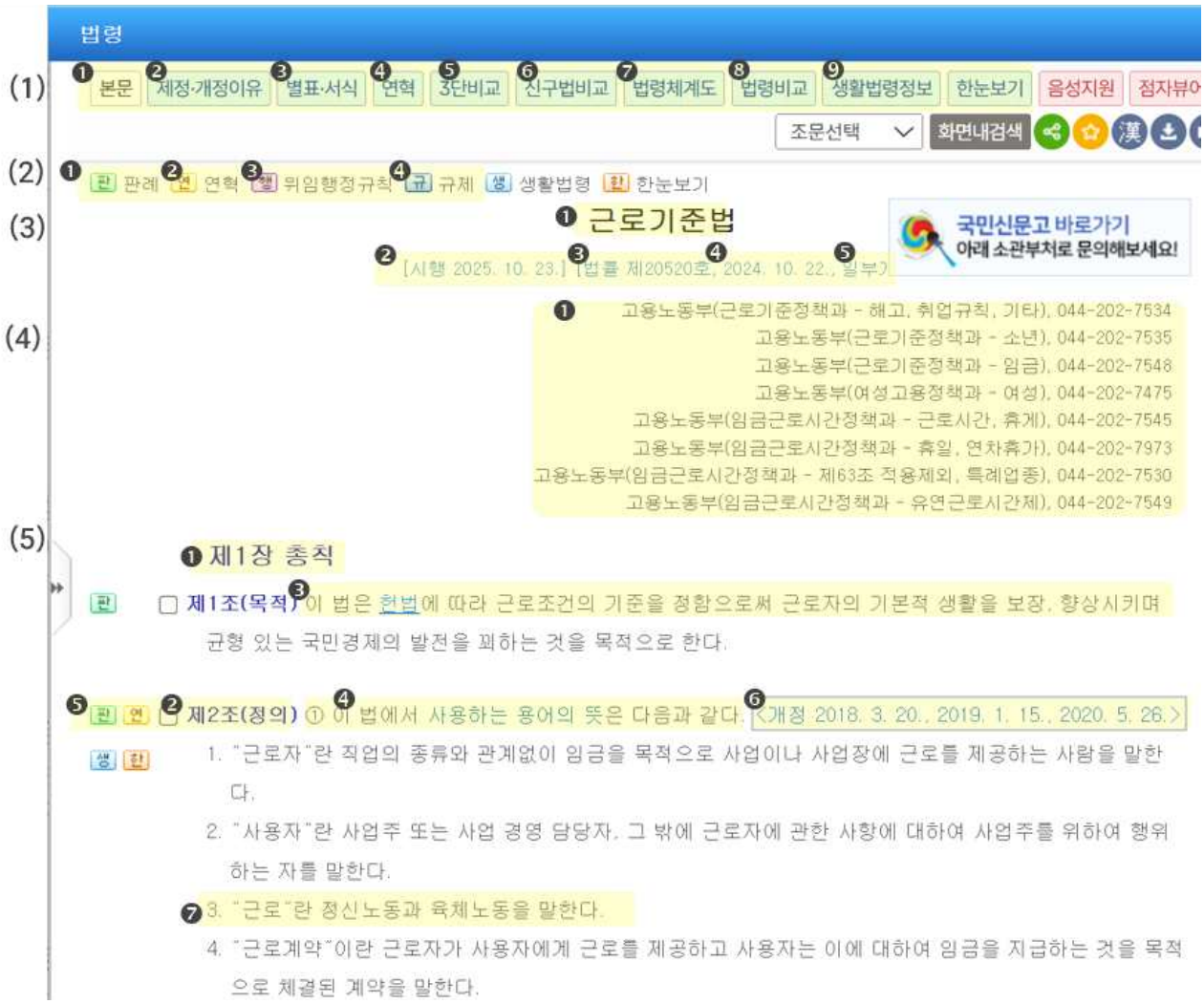
구조	설명	예시
구조적 위계(Structure)	법령의 물리적 계층	편→장→절→조→항·호·목
문장 성분 (Sentence Part)	조문 내 문장 구성	본문, 단서("다만..."), 후단, 괄호
의미론적 요소 (Semantic)	법적 의미단위	주체, 객체, 행위, 요건, 효과, 제재

3) 법령의 법문 구조 메타데이터 체계화

본 연구는 근로기준법을 기준으로 법문 구조 메타데이터를 7개 계층으로 체계화했고 우선 구조화 대상에 따라 법령 체계에 따른 구조적 위계 분류(Structural Hierarchy와 논리적 구성요소 분해(Logical Components)로 구성하였다

(그림3-3)은 국가법령정보센터에서 제공하는 법 검색 결과로서 이를 바탕으

로 (표3-3)의 기본 메타데이터를 도출하였다.



(그림3-3) 법문 구조 예시 (근로기준법)

(표3-3) 법문 구조 메타데이터 체계 요약

구분	구분	메타항목	하위 요소	온톨로지 연결
(1)	법령 메타 정보	❶ 본문	법령 본칙 조문(편/장/절/관/조/항/호/목) 열람 진입점	LegalInstrument ↔ StructuralUnit
		❷ 제정.개정 이유	법령 제정/개정의 취지·배경(입법 이유)	LegislativeHistory
		❸ 별표·서식	별표(테이블)·서식(양식) 목록/원문	AnnexTable, LegalForm
		❹ 연혁	법령 전체의 개정 연혁(버전 히스토리)	InstrumentVersion
		❺ 3단비교	신·구 조문 비교(3단 비교 뷰)	VersionDiff
		❻ 신구법비교	신법/구법 대조(변경점 비교)	VersionDiff

구분	구분	메타항목	하위 요소	온톨로지 연결
		❼ 법령체계도	법령의 체계(구조/관계) 시각화	HierarchyLink, ReferenceLink
		❽ 법령 비교	타 법령과 비교 (유사 / 관련 비교)	InterLawLink
		❾ 생활법령정보	생활 주제 기반 안내(대국민 해설 연계)	PlainExplanation(연계 리소스)
		❿ 한눈보기	요약/핵심정보(요건·절차 등) 형태의 뷰	SummaryView
(2)	연관 콘텐츠	❶ 판례	해당 법령과 연관된 판례로의 연결	CaseLawLink
		❷ 연혁	법령의 개정 이력 바로가기 (중복)	InstrumentVersion
		❸ 위임행정규칙	법령이 위임/근거로 연결되는 행정규칙 (훈령.예규.고시 등)	Delegation, AdministrativeRule
		❹ 규제	규제 정보(규제 관련 메타/연계 페이지)	RegulationMetadata
		❺ 생활법령	생활법령정보 연계 (주제형 설명)	PlainExplanation
		❻ 한눈보기	요약 보기 (중복 제공)	SummaryView
(3)	법령 기본 메타	❶ 법령명	법령의 공식 명칭(예: 근로기준법	LegalInstrument.title
		❷ 시행일	효력 발생일(예: [시행 2025.10.23.])	effectiveFrom
		❸ 공포번호/법령번호	법률 제○○○○호 등 식별 번호	promulgationNo
		❹ 공포일(또는 개정일 표기 포함)	예: “2024.10.22.	promulgatedOn
		❺ 재.개정 유형	일부개정/전부개정/제정/폐지 등	revisionType
(4)	소관부처	❶ 소관기관	법령의 공식 명칭(예: 근로기준법	LegalInstrument.title
		❷ 담당부서	효력 발생일(예: [시행 2025.10.23.])	effectiveFrom
		❸ 연락처	법률 제○○○○호 등 식별 번호	promulgationNo
		❹ 행정분류	예: “2024.10.22.	promulgatedOn
(5)	본문 구조	❶ 장(상위구조)	예: “제1장 총칙”	Chapter
		❷ 조 번호	예: “제1조”, “제2조”	Article.number
		❸ 조명(제목)	예: “(목적)”, “(정의)”	Article.title
		❹ 조문 내용(본문 텍스트)	조의 규정 문장(원문)	Article.text
		❺ (동일 패턴의 다른 조 표시)	화면에서 다음 조가 같은 구조로 반복됨을 보여주는 포인트	(5)-2~(5)-4 반복
		❻ 조문 단위 개정이력	예: “[개정 2018.3.20., …]”	ProvisionVersion / amendedOn
		❼ 호(각호) 및 호내용	예: 제2조의 1.,2.,3.… 항목(정의 각호)	Item + Item.text
		❽ 항/항내용	“① … ② …” 형태	
		❾ 목/목내용	“가. … 나. …” 형태	
		❿ 단서	“다만,… ” (예외/수정 관계)	
		⓫ 별표	“별표 ○” (조문 -> 별표 연결)	

구분	구분	메타항목	하위 요소	온톨로지 연결
		⓬ 서식	“별지 제○호 서식” (조문 -> 서식연결)	

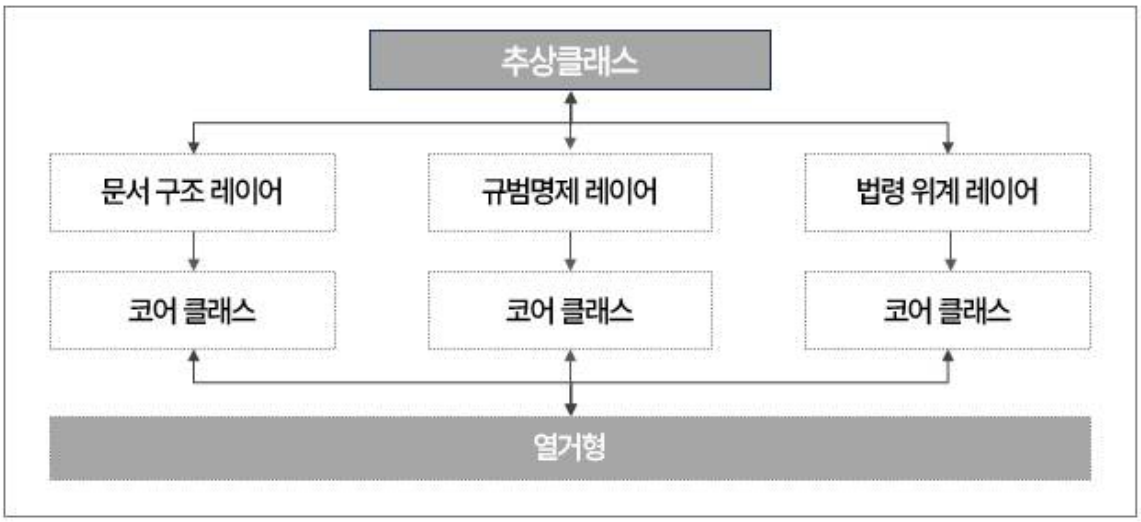
2. 법령 온톨로지 명세서 개요

가. 설계 목적과 범위

본 절은 온톨로지 설계를 위해 우리나라 법령정보시스템의 법령 서비스 요소를 기반으로 기준을 도출하였으며, 법령의 물리적 문서구조, 규범적 내용, 법체계상의 위계를 통합적으로 표현하는 것을 목적으로 한다. 이를 통해 법령 간 관계 탐색, 규범 상충 검증, 위임 근거 추적, 상위법 우선 원칙의 구조적 적용 등 설명 가능한 검색과 추론을 지원을 목적으로 한다.

나. 법령 온톨로지 전체 구조

제안한 온톨로지는 3개 레이어, 8개 추상 클래스와 48개의 핵심 클래스로 구성된다.



(그림3-4) 법령 온톨로지 구성 레이어

레이어	클래스수	역할	예시
문서 구조 레이어	24	- 법령 문서의 물리적 구조(편/장/절/관/ /조/항/호/목), - 부속요소(부칙/별표/서식), - 버전 이력, - 소관부처, - 서비스 기능 등을 표현	- LegalInstrument - StructuralUnit - Appendix

레이어	클래스수	역할	예시
규범명제 레이어	13	- 조문에서 추출한 규범명제 (주체-행위-객체-요건-효과 + 양상성) - 제재 - 수정관계(예외/특례/준용) - 상충 및 우선순위 해결	- NormStatement, - Condition, - LegalEffect
법령 위계 레이어	3	- 헌법→법률→대통령령→총리령/부령→조례→규칙의 위계 구조, - 법령 간 상하위 관계, - 헌법적 근거 관계를 표현	- NormativeLevel, - NormativeHierarchy - Link
추상 클래스	8	OWL domain 교집합 문제 방지를 위한 공통 상위 클래스 (IdentifiedEntity, TitledEntity, TextBearingEntity 등)	- IdentifiedEntity, - TitledEntity, - LinkEntity

다. 핵심 설계 원칙

첫째, 추상 클래스 기반 상속 설계를 적용하였다. OWL 의미론에서 title 속성에 rdfs:domain을 복수 클래스에 선언하면 합집합이 아닌 교집합으로 해석되는 문제가 발생한다. 이를 방지하기 위해 TitledEntity, TextBearingEntity 등 8개 추상 클래스를 도입하여 공통 속성의 domain을 통일하였다. 3개 레이어 모두 동일한 추상 클래스를 재사용하여 일관성을 유지한다.

둘째, 관계의 1급 객체화(Reification)를 적용하였다. 참조, 위임, 연관법, 예외/특례, 법령 위계 등의 관계를 단순한 Object Property가 아닌 별도의 클래스(LinkEntity 계열)로 모델링하여, 관계의 유형・근거문구・범위 등 부가정보를 표현할 수 있도록 하였다.

셋째, 법령 위계의 구조적 모델링을 도입하였다. 헌법부터 규칙까지의 위계를 NormativeLevel 클래스의 hierarchyRank로 순서화하고, 법령 간 상하위 관계를 NormativeHierarchyLink로 표현하였다. 이를 통해 상위법 우선 원칙(lex superior)을 hierarchyRank 비교로 자동 판단할 수 있으며, 위임의 방향성 검증과 합헌성 심사의 구조적 기반을 제공한다.

라. 레이어별 구성 요약

1) 문서구조 레이어

LegalInstrument를 중심으로 법령 문서의 물리적 구조를 모델링한다. StructuralUnit의 하위 클래스로 Part(편), Chapter(장), Section(절), Subsection(관), Article(조), Paragraph(항), Item(호), Subitem(목)을 정의하여 실제 법령의

계층 구조를 완전하게 반영하였다. 부속요소는 Appendix를 상위 클래스로 하여 Addendum(부칙), AnnexTable(별표), LegalForm(서식)으로 분류하였으며, 버전 관리는 InstrumentVersion(법령 단위)과 ProvisionVersion(조문 단위)으로 분리하였다.

2) 규범명제 레이어

NormStatement를 중심으로 조문에서 추출한 규범명제를 모델링한다. 각 규범명제는 주체(LegalRole), 행위(LegalAction), 객체(LegalObject), 요건(Condition), 효과(LegalEffect), 양상성(DeonticModality)으로 분해되며, 제재(Sanction)와 행정기관 권한행위(AuthorityAction)를 결합한다. 예외·특례·준용은 ModificationLink로, 규범 상충은 NormConflictCase와 ResolutionDecision으로 표현하여 설명 가능한 추론을 지원한다.

3) 법령위계 레이어

NormativeLevel을 중심으로 헌법(rank=1)부터 규칙(rank=6)까지의 위계 단계를 모델링한다. 각 LegalInstrument는 hasNormativeLevel을 통해 정확히 하나의 위계 단계에 연결된다. 구체적인 법령 간 상하위 관계는 NormativeHierarchyLink로 표현하며, 헌법 조항과 법률 간의 헌법적 근거 관계는 ConstitutionalBasis로 명시한다. 이 레이어는 기존 DelegationLink(조문 단위 위임)와 연동되어 위임기반 위계(hierarchyType = DELEGATION_BASIS)를 구분하며, ResolutionPrinciple의 상위법 우선 원칙이 hierarchyRank 비교로 자동 판단 가능하도록 연결된다.

마. 온톨로지 구성 레이어별 설계 현황

(표3-4) 온톨로지 구성 레이어별 구성 현황*

구분	클래스	Datatype Property	Object Property	열거형
추상 클래스	8	5	1	-
문서 구조 레이어	24	21	30_	7
규범명제 레이어	13	6	20+	5
법령위계 레이어	3	7	10	3
합계	48	39	61+	15

*실제 구축 단계에서는 증감될 수 있음

바. 주요 제약 및 일관성 규칙

본 온톨로지는 11개의 제약 규칙을 정의하여 일관성과 설명가능성을 보장한다. 주요 규칙은 다음과 같다.

① **근거 추적성**: 모든 NormStatement는 최소 1개의 근거 조문(expressedInUnit)을 가져야 한다.

② **양상성 필수:** 모든 NormStatement는 규범력 유형(hasModality)을 명시해야 한다.

③ **위계 필수**: 모든 LegalInstrument는 정확히 1개의 NormativeLevel을 가져야 한다.

④ **위계 방향성 검증**: NormativeHierarchyLink의 상위법령 hierarchyRank는 하위법령보다 작아야 한다.

⑤ **위임 정합성**: DelegationLink의 위임 방향은 상위법→하위법이어야 한다.

⑥ **공통 속성 중복 금지**: title, text, identifier는 추상 클래스에서만 정의하고 개별 클래스에서 재정의하지 않는다.

3. 법령 온톨로지 명세서

가. 상위(추상) 클래스 정의

3개 레이어 모두에 일관되게 적용되는 8개 추상클래스 목록이다.

순번	추상 클래스	목적/필요성	대표 공통 속성
1	IdentifiedEntity	시스템 내 모든 개체에 식별자 부여	identifier: xsd:string
2	TitledEntity	제목/표제 title의 공통 도메인 확보 (OWL domain 교집합 문제 방지)	title: rdf:langString
3	TextBearingEntity	본문 텍스트 text의 공통 도메인 확보	text: rdf:langString
4	OrderedEntity	화면/문서 내 정렬을 위한 순서값	orderIndex: xsd:integer
5	NumberedEntity	번호 표기 (제1장/제2조/① 등)	number: xsd:string
6	VersionedEntity	버전(연혁/개정)을 갖는 대상	hasVersion (Object Property)
7	ReferencableEntity	참조 링크의 타깃 공통 범주 (조문/부속/법령)	–
8	LinkEntity	관계(링크)를 1급 객체로 표현	anchorText, scope

나. 문서구조 레이어

1) 핵심 클래스

구분	클래스	정의(역할)	상위클래스(상속)	주요 Datatype 속성
(3)(5)	LegalInstrument	법령 문서 단위 (법률/대통령령/부령/조례)	IdentifiedEntity, TitledEntity, VersionedEntity, ReferencableEntity	identifier, title, instrumentType, promulgationNo, promulgatedOn, effectiveFrom, revisionType, language, sourceSystem
(5)	StructuralUnit (추상)	법문 구조 단위 상위 클래스 (편/장/절/관/조/항/호/목)	IdentifiedEntity, TitledEntity, TextBearingEntity, OrderedEntity, VersionedEntity, ReferencableEntity	identifier, title, text, orderIndex, unitType
(5)	Part	편 (예: 제1편 총론)	StructuralUnit, NumberedEntity	number
(5)	Chapter	장 (예: 제1장 총칙)	StructuralUnit, NumberedEntity	number
(5)	Section	절 (예: 제1절)	StructuralUnit, NumberedEntity	number
(5)	Subsection	관 (예: 제1관)	StructuralUnit, NumberedEntity	number
(5)	Article	조 (예: 제2조(정의))	StructuralUnit, NumberedEntity	number(조번호)
(5)	Paragraph	항 (①②③…)	StructuralUnit, NumberedEntity	number
(5)	Item	호 (1. 2. …)	StructuralUnit, NumberedEntity	number
(5)	Subitem	목 (가. 나. …)	StructuralUnit, NumberedEntity	number
(3)	InstrumentVersion	법령(문서) 버전/연혁 단위	IdentifiedEntity	identifier, validFrom, validTo, promulgatedOn, revisionType, noteText
(5)	ProvisionVersion	조문 단위 버전/개정 이력	IdentifiedEntity	identifier, validFrom, validTo, amendedOn, amendmentNoteText
(1)(5)	Appendix (추상)	부속 구성요소 상위 (부칙/별표/서식)	IdentifiedEntity, TitledEntity, TextBearingEntity, ReferencableEntity	identifier, title, text, appendixType
(1)(5)	Addendum	부칙 (시행/경과/적용례)	Appendix	effectiveRuleText
(1)(5)	AnnexTable	별표 (기준표/산정표 등)	Appendix	tableType, schemaJson

구분	클래스	정의(역할)	상위클래스(상속)	주요 Datatype 속성
(1)(5)	LegalForm	서식 (별지 제○호)	Appendix	formNo, fieldsJson
(1)(2)	ServiceFeature	화면 탭/바로가기 (본문/연혁/비교 등)	IdentifiedEntity, TitledEntity	identifier, title, featureType, description
(4)	CompetentAuthority	소관기관/담당부서	IdentifiedEntity, TitledEntity	identifier, title, orgName, deptName, contactPhone, adminClassCode
(5)(8)	ReferenceLink	인용/참조/준용/예외/특례 텍스트 내 링크	IdentifiedEntity, LinkEntity	identifier, referenceType, anchorText, scope
(5)(8)	DelegationLink	위임 관계 링크	IdentifiedEntity, LinkEntity	identifier, delegationType, delegationScopeText
(2)(8)	InterLawLink	연관법(관련 법령 간 결합)	IdentifiedEntity, LinkEntity	identifier, relationType, sharedElement
(2)	CaseLawLink	판례 연계	IdentifiedEntity, TitledEntity	identifier, title, court, date, caseNo
(1)	SummaryView	한눈보기/요약 리소스	IdentifiedEntity, TextBearingEntity, TitledEntity	identifier, title, text, generatedBy
(1)(2)	PlainExplanation	생활법령/쉬운 설명 리소스	IdentifiedEntity, TextBearingEntity, TitledEntity	identifier, title, text, topic

2) Datatype Property

공통 속성(identifier, title, text, orderIndex, number)은 추상 클래스에서 정의되므로 클래스 전용 속성만 기술하였다.

순번	속성	Domain	Range	의미
1	instrumentType	LegalInstrument	xsd:string (enum)	법령 유형
2	promulgationNo	LegalInstrument	xsd:string	공포번호
3	promulgatedOn	LegalInstrument	xsd:date	공포일
4	effectiveFrom	LegalInstrument	xsd:date	시행일
5	revisionType	LegalInstrument	xsd:string (enum)	재·개정 유형

순번	속성	Domain	Range	의미
6	language	LegalInstrument	xsd:language	언어 (ko)
7	sourceSystem	LegalInstrument	xsd:string	원문 시스템
8	unitType	StructuralUnit	xsd:string (enum)	구조 단위 유형
9	amendedOn	ProvisionVersion	xsd:date	개정일
10	amendmentNoteText	ProvisionVersion	rdf:langString	개정 주석
11	validFrom / validTo	ProvisionVersion	xsd:date	버전 유효 기간
12	orgName	CompetentAuthority	xsd:string	기관명
13	deptName	CompetentAuthority	xsd:string	부서명
14	contactPhone	CompetentAuthority	xsd:string	연락처
15	adminClassCode	CompetentAuthority	xsd:string	행정분류 코드
16	appendixType	Appendix	xsd:string (enum)	부속 유형
17	featureType	ServiceFeature	xsd:string (enum)	기능 유형
18	effectiveRuleText	Addendum	rdf:langString	시행/경과 조치 텍스트
19	tableType	AnnexTable	xsd:string	테이블 유형
20	formNo	LegalForm	xsd:string	서식 번호

3) Object Property

공통 속성(identifier, title, text, orderIndex, number)은 추상 클래스에서 정의되므로 클래스 전용

가) 포함/소속 관계

순번	Object Property	Domain	Range	Card.	Inverse	의미
1	hasStructuralUnit	LegalInstrument	StructuralUnit	1..n	partOfInstrument	법령 → 조문 구조
2	partOfInstrument	StructuralUnit	LegalInstrument	1	hasStructuralUnit	구조 단위의 소속 법령
3	hasAppendix	LegalInstrument	Appendix	0..n	appendixOfInstrument	법령 → 부칙/별표/서식
4	hasAuthority	LegalInstrument	CompetentAuthority	0..n	authorityOfInstrument	소관기관 연결
5	hasServiceFeature	LegalInstrument	ServiceFeature	0..n	featureOfInstrument	화면 기능 탭

나) 구조 계층 관계

순번	Object Property	Domain	Range	Card.	Inverse	의미
1	hasChildUnit	StructuralUnit	StructuralUnit	0..n	hasParentUnit	상위→하위 (장→조→항…)
2	hasParentUnit	StructuralUnit	StructuralUnit	0..1	hasChildUnit	하위→상위

다) 버전/연혁 관계

순번	Object Property	Domain	Range	Card.	Inverse	의미
1	hasInstrumentVersion	LegalInstrument	InstrumentVersion	0..n	versionOfInstrument	법령 문서 버전
2	hasProvisionVersion	StructuralUnit	ProvisionVersion	0..n	versionOfUnit	조문 단위 버전
3	derivedFrom	ProvisionVersion	ProvisionVersion	0..1	–	이전 버전 계승

라) 참조/위임/연관법/판례/요약 관계

순번	Object Property	Domain	Range	Card.	Inverse	의미
1	hasReferenceLink	StructuralUnit	ReferenceLink	0..n	–	조문의 참조 링크
2	referenceSource	ReferenceLink	StructuralUnit	1	–	링크 출발 조문
3	referenceTarget	ReferenceLink	ReferencableEntity	1	–	링크 대상
4	hasDelegationLink	StructuralUnit	DelegationLink	0..n	–	위임 링크 보유
5	delegationSource	DelegationLink	StructuralUnit	1	–	위임 근거 조문
6	delegationTarget	DelegationLink	LegalInstrument ⊎ StructuralUnit	1..n	–	위임 대상
7	hasInterLawLink	LegalInstrument	InterLawLink	0..n	–	연관법 링크
8	relatedCase	LegalInstrument ⊎ StructuralUnit	CaseLawLink	0..n	caseRelatedTo	판례 연결
9	hasSummary	LegalInstrument	SummaryView	0..n	summaryOfInstrument	한눈보기/요약
10	hasPlainExplanation	LegalInstrument	PlainExplanation	0..n	explainsInstrumentOrUnit	생활법령/쉬운 설명
11	hasAddendumChild	Addendum	StructuralUnit	0..n	–	부칙 내부 조/항 구조

다. 규범명제 레이어

1) 핵심 클래스

클래스	상위 클래스	정의(역할)	예시
NormStatement	IdentifiedEntity, TextBearingEntity, VersionedEntity	조문에서 추출한 규범명제 1개 단위 (주체-행위-객체-요건-효과 + 양상성)	"사용자는 근로자에게 임금을 지급하여야 한다."
LegalRole	IdentifiedEntity, TitledEntity	규범의 주체/상대방 역할	사용자, 근로자, 고용노동부 장관
LegalAction	IdentifiedEntity, TitledEntity	규범의 행위	지급, 교부, 보존, 신고
LegalObject	IdentifiedEntity, TitledEntity	규범의 객체	임금, 근로계약서, 서류

클래스	상위 클래스	정의(역할)	예시
Condition	IdentifiedEntity, TextBearingEntity	요건 (정량/기간/절차/사유 등)	"상시 5명 이상", "30일 이내"
LegalEffect	IdentifiedEntity, TitledEntity, TextBearingEntity	효과 (의무발생/권리발생/무효/제재)	"의무 발생", "과태료 부과 가능"
DeonticModality	IdentifiedEntity, TitledEntity	양상성 코드 (의무/금지/허용/권한/면제)	OBLIGATION, PROHIBITION, PERMISSION, POWER, EXEMPTION
Sanction	IdentifiedEntity, TitledEntity, TextBearingEntity	제재 (벌칙/과태료/행정제재/무효)	"과태료", "벌금", "시정명령"
AuthorityAction	IdentifiedEntity, TitledEntity, TextBearingEntity	행정기관 권한행위 (조사/명령/부과)	조사한다, 명령한다
ModificationLink	IdentifiedEntity, LinkEntity	예외·특례·준용 수정관계 링크	"다만 …인 경우 제외"
NormConflictCase	IdentifiedEntity, TextBearingEntity	규범 상충 케이스	일반규정 vs 특례규정
ResolutionPrinciple	IdentifiedEntity, TitledEntity	상충 해결 원리	상위규범우선, 특별규정우선, 후법우선
ResolutionDecision	IdentifiedEntity, TextBearingEntity	상충 해결 결과	"특별규정 우선으로 A 적용"

2) Object Property

순번	Object Property	Domain	Range	Card.	Inverse	의미
1	expressedInUnit	NormStatement	StructuralUnit	1..n	hasNormStatement	규범명제의 근거 조문
2	hasSubjectRole	NormStatement	LegalRole	1..n	–	주체 역할
3	hasAction	NormStatement	LegalAction	1..n	–	행위
4	hasObject	NormStatement	LegalObject	0..n	–	객체
5	hasCondition	NormStatement	Condition	0..n	–	요건 연결
6	hasEffect	NormStatement	LegalEffect	1..n	–	효과 연결
7	hasModality	NormStatement	DeonticModality	1	–	규범력 유형
8	hasSanction	NormStatement	Sanction	0..n	–	위반 시 제재
9	authorityMayDo	NormStatement	AuthorityAction	0..n	–	기관 권한행위
10	enforcedByAuthority	NormStatement	CompetentAuthority	0..n	–	집행/소관 기관
11	hasModificationLink	NormStatement	ModificationLink	0..n	–	예외/특례/준용 링크
12	modifiesTarget	ModificationLink	NormStatement	1	–	수정 대상(원칙 규범)
13	conflictingNorm	NormConflictCase	NormStatement	2..n	–	상충 당사 규범

순번	Object Property	Domain	Range	Card.	Inverse	의미
14	resolvesConflict	ResolutionDecision	NormConflictCase	1	–	해결 대상 상충
15	resolutionPrinciple	ResolutionDecision	ResolutionPrinciple	1..n	–	적용 해결 원리
16	preferredNorm	ResolutionDecision	NormStatement	1	–	우선 적용 규범

라. 법령위계 레이어

1) 우리나라 법령 위계 체계

위계	코드	한글명	영문명	제정기관	설명
1	CONSTITUTION	헌법	Constitution	국회(헌법개정)	최상위 규범
2	ACT	법률	Act / Statute	국회	형식적 법률
3	PRESIDENTIAL_DECREE	대통령령	Presidential Decree	대통령	시행령
4	PM / MIN_ORD	총리령/부령	PM/Ministerial Ord.	총리/장관	시행규칙
5	MUNICIPAL_ORD	조례	Municipal Ordinance	지방의회	지방의회 제정
6	MUNICIPAL_RULE	규칙	Municipal Rule	자치단체장	자치단체장 발령
(별도)	ADMIN_RULE	행정규칙	Admin. Rule	행정기관	훈령/예규/고시/공고

2) 핵심 클래스 (3개)

클래스	상위 클래스	정의(역할)	예시
NormativeLevel	IdentifiedEntity, TitledEntity, OrderedEntity	법령 위계 단계를 나타내는 1급 객체. hierarchyRank로 순서화.	CONSTITUTION(1), ACT(2), PRESIDENTIAL_DECREE(3), …
NormativeHierarchyLink	IdentifiedEntity, LinkEntity	법령 간 위계 관계를 1급 링크로 표현. 상위법→하위법 방향성 명시.	ACT → PRESIDENTIAL_DECREE (위임근거 포함)
ConstitutionalBasis	IdentifiedEntity, LinkEntity	헌법 조항과 법률 간의 헌법적 근거 관계.	헌법 제32조 → 근로기준법

3) Object Property

순번	Object Property	Domain	Range	Card.	Inverse	의미
1	hasNormativeLevel	LegalInstrument	NormativeLevel	1	instrumentsAtLevel	법령이 속한 위계 단계
2	instrumentsAtLevel	NormativeLevel	LegalInstrument	0..n	hasNormativeLevel	해당 위계에 속한 법령들
	superiorLevel	NormativeLevel	NormativeLevel	0..1	subordinateLevel	상위 위계 단계
3	subordinateLevel	NormativeLevel	NormativeLevel	0..n	superiorLevel	하위 위계 단

순번	Object Property	Domain	Range	Card.	Inverse	의미
						계(들)
	hasHierarchyLink	LegalInstrument	NormativeHierarchyLink	0..n	–	위계 링크 보유
4	hierarchySource	NormativeHierarchyLink	LegalInstrument	1	–	상위법령
	hierarchyTarget	NormativeHierarchyLink	LegalInstrument	1	–	하위법령
5	hasConstitutionalBasis	LegalInstrument	ConstitutionalBasis	0..n	–	헌법적 근거 보유
	basisSource	ConstitutionalBasis	StructuralUnit	1	–	헌법 조항 (근거)
6	basisTarget	ConstitutionalBasis	LegalInstrument	1	–	해당 법률

4) 기존 모델과의 통합

① DelegationLink 연동: 조문 단위 위임관계(DelegationLink)와 법령 단위 위계관계(NormativeHierarchyLink)를 연동하여 hierarchyType = DELEGATION_BASIS로 구분한다.

② 규범 상충 해결 연동: ResolutionPrinciple의 SUPERIOR_NORM_PRIORITY가 NormativeLevel.hierarchyRank를 기반으로 자동 판단 가능하도록 연결된다.

마. 열거형(Controlled Vocabulary) 정의

15개의 열거형 클래스를 정의하였다. 열거형은 owl:oneOf 또는 SKOS ConecptScheme로 구현할 수 있다.

1) InstrumentType - 법령 유형

- 적용속성: LegalInstrument.instrumentType. NormativeLevel과 연동된다.

열거값	한글명	의미/설명
CONSTITUTION	헌법	대한민국헌법. 최상위 규범으로서 모든 법령의 합헌성 기준이 된다. (hierarchyRank = 1)
ACT	법률	국회 의결로 제정되는 형식적 법률. 예: 근로기준법, 민법, 형법. (hierarchyRank = 2)
PRESIDENTIAL_DECREE	대통령령	법률의 위임에 의해 대통령이 발하는 명령. 일반적으로 '시행령'으로 불린다. (hierarchyRank = 3)
PRIME_MINISTER_DECREE	총리령	국무총리가 소관 사무에 관하여 발하는 명령. (hierarchyRank = 4)
MINISTERIAL_ORDINANCE	부령	법률 또는 대통령령의 위임에 의해 각부 장관이 발하는 명령. 일반적으로 '시행규칙'으로 불린다. (hierarchyRank = 4)

열거값	한글명	의미/설명
MUNICIPAL_ORDINANCE	조례	지방의회가 법령의 범위 안에서 의결로 제정하는 자치법규. (hierarchyRank = 5)
MUNICIPAL_RULE	규칙	지방자치단체장이 법령 또는 조례의 범위 안에서 발하는 규칙. (hierarchyRank = 6)
ADMINISTRATIVE_RULE	행정규칙	법령 외 행정기관 내부규범. 훈령·예규·고시·공고 등을 포함하며, 공식적 법령 위계에는 포함되지 않는다.

2) RevisionType - 재.개정 유형

- 적용속성: LegalInstrument.revisionType, InstrumentVersion.revisionType.과 연동된다.

열거값	한글명	의미/설명
ENACTMENT	제정	법령을 최초로 만들어 공포하는 것. 새로운 법령의 탄생.
PARTIAL_AMENDMENT	일부개정	기존 법령의 일부 조항만 수정하는 것. 가장 빈번한 개정 형태.
FULL_AMENDMENT	전부개정	법령 전체를 새로 작성하여 교체하는 것. 법령번호는 유지되지만 내용이 전면 변경된다.
REPEAL	폐지	법령의 효력을 소멸시키는 것. 해당 법령은 더 이상 적용되지 않는다.

3) UnitType - 구조 단위 유형

- 적용속성: StructuralUnit.unitType. 법령 문서의 물리적 계층 구조를 나타낸다.

열거값	한글명	의미/설명
PART	편	법령의 최상위 분류 단위. 예: '제1편 총론'. 민법·형법 등 대규모 법률에서 사용
CHAPTER	장	편의 하위 또는 법령의 주요 분류 단위. 예: '제1장 총칙'
SECTION	절	장의 하위 분류 단위. 예: '제1절 통칙'
SUBSECTION	관	절의 하위 분류 단위. 예: '제1관'. 민법 등 복잡한 법률에서 사용
ARTICLE	조	법령의 기본 규정 단위. 예: '제1조(목적)', '제2조(정의)'. 모든 법령의 핵심 구성 요소
PARAGRAPH	항	조의 하위 구분. 원문 ①② 형태로 표기된다. 예: '제2조제1항'
ITEM	호	항 또는 조의 하위 나열 항목. 1. 2. 3. 형태로 표기된다. 예: 정의 각호
SUBITEM	목	호의 하위 나열 항목. 가. 나. 다. 형태로 표기된다

4) AppendixType - 부속 유형

- 적용속성: Appendix.appendixType

열거값	한글명	의미/설명
ADDENDUM	부칙	법령의 시행일, 경과조치, 적용례 등을 규정하는 부속 규정. 예: '부칙 제1조(시행일)'.
ANNEX_TABLE	별표	법령 본문에서 참조하는 기준표·산정표 등의 테이블 형태 부속물. 예: '별표 1'.
FORM	서식	법령이 정한 공식 서식/양식. 예: '별지 제1호 서식'.

5) FeatureType - 서비스 기능 유형

- 적용속성: ServiceFeature.featureType. 법령정보시스템 화면의 탭/기능을 분류한다

열거값	한글명	의미/설명
BODY	본문	법령 본칙 조문 열람 기능.
HISTORY	연혁	법령 전체의 개정 연혁(버전 히스토리) 조회 기능.
THREE_WAY_DIFF	3단비교	신·구 조문을 3단으로 비교하는 뷰 기능.
OLD_NEW_DIFF	신구법비교	신법/구법을 대조하여 변경점을 비교하는 기능.
HIERARCHY	법령체계도	법령의 체계(구조/관계)를 시각화하는 기능.
COMPARISON	법령비교	타 법령과 비교(유사/관련 비교)하는 기능.
LIFE_LAW	생활법령정보	생활 주제 기반으로 법령을 쉬운 언어로 안내하는 연계 기능.
SUMMARY	한눈보기	법령의 요약/핵심정보(요건·절차 등)를 압축하여 보여주는 뷰 기능.

6) ReferenceType - 참조 유형

- 적용속성: ReferenceLink.referenceType. 조문 간 텍스트 내 참조 관계의 의미적 유형을 구분한다

열거값	한글명	의미/설명
CITATION	인용	특정 조항을 명시적으로 언급하는 관계. 예: '제00조에 따른', '제00조에서 정하는'.
REFERENCE	참조	다른 조항의 내용을 참고하도록 지시하는 관계. 예: '제00조를 참조한다'.
MUTATIS_MUTANDIS	준용	다른 조항의 규정을 필요한 수정을 가하여 적용하는 관계. 예: '제00조를 준용한다'. 법학적으로 중요한 관계로, 단순 참조와 구별된다.

열거값	한글명	의미/설명
EXCEPTION	예외	원칙 규정의 적용을 배제하는 관계. 예: '다만, …인 경우에는 그러하지 아니하다'.
SPECIAL_PROVISION	특례	일반 규정에 대한 특별한 예외를 규정하는 관계. 예: '제00조에도 불구하고'.
APPENDIX_REF	별표/서식 참조	조문에서 별표 또는 서식을 참조하는 관계. 예: '별표 1에 따른다'.

7) ReferenceType - 참조 유형

- 적용속성: ReferenceLink.referenceType. 조문 간 텍스트 내 참조 관계의 의미적 유형을 구분한다

열거값	한글명	의미/설명
CITATION	인용	특정 조항을 명시적으로 언급하는 관계. 예: '제00조에 따른', '제00조에서 정하는'.
REFERENCE	참조	다른 조항의 내용을 참고하도록 지시하는 관계. 예: '제00조를 참조한다'.
MUTATIS_MUTANDIS	준용	다른 조항의 규정을 필요한 수정을 가하여 적용하는 관계. 예: '제00조를 준용한다'. 법학적으로 중요한 관계로, 단순 참조와 구별된다.
EXCEPTION	예외	원칙 규정의 적용을 배제하는 관계. 예: '다만, …인 경우에는 그러하지 아니하다'.
SPECIAL_PROVISION	특례	일반 규정에 대한 특별한 예외를 규정하는 관계. 예: '제00조에도 불구하고'.
APPENDIX_REF	별표/서식 참조	조문에서 별표 또는 서식을 참조하는 관계. 예: '별표 1에 따른다'.

8) DelegationType — 위임 유형

- 적용속성: DelegationLink.delegationType. 조문에서 '…으로 정한다' 형태의 위임 대상을 구분한다.

열거값	한글명	의미/설명
TO_PRESIDENTIAL_DECREE	대통령령 위임	'대통령령으로 정한다' 형태의 위임. 법률→시행령.
TO_ORDINANCE	부령 위임	'부령으로 정한다' 형태의 위임. 법률/시행령→시행규칙.
TO_RULE	규칙 위임	'규칙으로 정한다' 형태의 위임. 조례→규칙.
TO_PROVISION	특정 조항 위임	특정 조항으로 세부 사항을 위임하는 관계. 예: '제00조에서 정하는 바에 따른다'.

9) InterLawRelationType — 연관법 관계 유형

- 적용속성: InterLawLink.relationType.

열거값	한글명	의미/설명
RELATED	관련	내용적으로 관련되는 법령 간의 일반적 연결 관계.
SUPERIOR	상위법	해당 법령의 상위 근거가 되는 법령과의 관계.
SUBORDINATE	하위법	해당 법령에 근거하여 제정된 하위 법령과의 관계.
IMPLEMENTING	시행법	상위법의 시행을 위해 구체적 사항을 규정하는 법령과의 관계. 예: 법률→시행령.
SUPPLEMENTARY	보충법	본법을 보충·보완하는 법령과의 관계.

10) DeonticModalityCode — 양상성 코드

- 적용속성: InterLawLink.relationType.

열거값	한글명	의미/설명
RELATED	관련	내용적으로 관련되는 법령 간의 일반적 연결 관계.
SUPERIOR	상위법	해당 법령의 상위 근거가 되는 법령과의 관계.
SUBORDINATE	하위법	해당 법령에 근거하여 제정된 하위 법령과의 관계.
IMPLEMENTING	시행법	상위법의 시행을 위해 구체적 사항을 규정하는 법령과의 관계. 예: 법률→시행령.
SUPPLEMENTARY	보충법	본법을 보충·보완하는 법령과의 관계.

11) ConditionType — 요건 유형

- 적용속성: Condition.conditionType

열거값	한글명	의미/설명
QUANTITATIVE	정량적 요건	수치로 판단할 수 있는 요건. 예: '상시 5명 이상', '5인 미만'.
TEMPORAL	기간적 요건	시간/기간과 관련된 요건. 예: '30일 이내', '3개월 이내'.
PROCEDURAL	절차적 요건	특정 절차나 방식을 요구하는 요건. 예: '서면으로', '신청을 거쳐'.
CAUSAL	사유적 요건	특정 사유나 원인의 발생을 전제로 하는 요건. 예: '정당한 사유가 있는 경우'.
SUBJECTIVE	주관적 요건	판단 주체의 재량이나 판단을 요구하는 요건. 예: '필요하다고 인정하는 경우'.

12) EffectType — 효과 유형

- 적용속성: LegalEffect.effectType.

열거값	한글명	의미/설명
OBLIGATION_ARISE	의무 발생	특정 주체에게 의무가 발생하는 효과. 예: '임금을 지급할 의무가 있다'.
RIGHT_ARISE	권리 발생	특정 주체에게 권리가 발생하는 효과. 예: '휴가를 청구할 수 있다'.
NULLITY	무효	법률행위나 계약 등이 무효가 되는 효과. 예: '무효로 한다'.
SANCTION_ARISE	제재 발생	벌칙·과태료 등 제재가 발생하는 효과. 예: '과태료를 부과한다'.
PERMISSION_ARISE	허용 발생	특정 행위가 허용되는 효과. 예: '…할 수 있다'.

13) ModificationType — 수정 유형

- 적용속성: ModificationLink.modificationType.

열거값	한글명	의미/설명
EXCEPTION	예외	원칙 규정의 적용을 배제하는 수정 관계. 예: '다만 …인 경우에는 적용하지 아니한다'.
SPECIAL_PROVISION	특례	일반 규정에 대한 특별한 규정을 두는 수정 관계. 예: '제00조에도 불구하고'.
MUTATIS_MUTANDIS	준용	다른 규정을 필요한 수정을 가하여 적용하는 수정 관계. 예: '제00조를 준용한다'.
EXCLUSION	적용배제	특정 대상이나 상황에 대해 원칙 규정의 적용을 전면 배제하는 관계.

14) SanctionType — 제재 유형

- 적용속성: Sanction.sanctionType.

열거값	한글명	의미/설명
CRIMINAL_PENALTY	벌칙	형사처벌(징역/벌금 등). 법률의 벌칙 조항에 규정된다.
FINE	과태료	행정질서 위반에 대한 금전적 제재. 벌칙보다 경미한 제재 수단.
ADMINISTRATIVE_ORDER	행정명령	행정기관이 발하는 시정명령·중지명령 등의 행정적 제재.
ADMINISTRATIVE_PENALTY	행정처분	허가 취소·영업정지·등록 말소 등의 행정적 불이익 처분.
NULLITY	무효	법률행위의 효력을 부정하는 제재. 예: '무효로 한다'.

15) HierarchyType — 위계 관계 유형

- 적용속성: NormativeHierarchyLink.hierarchyType.

열거값	한글명	의미/설명
SUPERIOR_INFERIOR	상위-하위	법체계상 상위법령과 하위법령 간의 일반적 위계 관계. 예: ACT → PRESIDENTIAL_DECREE.
DELEGATION_BASIS	위임근거	상위법의 특정 조항이 하위법 제정을 위임한 근거에 기반한 위계 관계. DelegationLink와 연동된다.
IMPLEMENTATION	시행 관계	상위법의 시행을 위해 구체적 사항을 규정하는 관계. 예: 법률→시행령→시행규칙 체인.
CONSTITUTIONAL_BASIS	헌법적 근거	헌법 조항이 특정 법률의 헌법적 근거가 되는 관계. ConstitutionalBasis 클래스와 연동된다.

16) IssuingAuthorityType — 제정/발령 기관 유형

- 적용속성: NormativeLevel.issuingAuthority. 각 위계 단계에서 법령을 제정・발령하는 기관의 유형이다.

열거값	한글명	의미/설명
NATIONAL_ASSEMBLY	국회	법률(ACT)을 의결·제정하는 입법기관. 헌법개정안도 국회가 의결한다.
PRESIDENT	대통령	대통령령(시행령)을 발하는 행정부 수장. 긴급명령·긴급재정경제처분도 포함.
PRIME_MINISTER	국무총리	총리령을 발하는 기관. 국무총리 소관 사무에 관한 명령.
MINISTER	각부 장관	부령(시행규칙)을 발하는 기관. 해당 부처 소관 법률의 시행을 위한 세부사항 규정.
LOCAL_COUNCIL	지방의회	조례를 의결·제정하는 지방자치단체의 의결기관.
LOCAL_HEAD	자치단체장	규칙을 발하는 지방자치단체의 집행기관(시장·군수·구청장·도지사 등).
ADMIN_AGENCY	행정기관	행정규칙(훈령·예규·고시·공고 등)을 발하는 중앙행정기관 및 그 소속기관.

바. 제약 및 일관성 규칙 정의

11개의 제약 규칙을 정의하여 온톨로지의 일관성과 설명 가능성을 보장한다.

ID	규칙명	설명	OWL 표현
C-01	근거 추적성	모든 NormStatement는 최소 1개의 expressedInUnit을 보유	owl:minCardinality 1
C-02	양상성 필수	모든 NormStatement는 hasModality를 명시	owl:exactCardinality 1
C-03	수정관계 방향성	ModificationLink는 modifiesTarget으로	owl:exactCardinality 1

ID	규칙명	설명	OWL 표현
		원칙 규범를 지정	
C-04	상충 설명가능성	ResolutionDecision은 resolutionPrinciple을 필수 포함	owl:minCardinality 1
C-05	공통 속성 중복 금지	title/text/identifier는 추상 클래스에서만 정의	설계 규칙
C-06	구조 단위 소속	모든 StructuralUnit는 정확히 1개의 LegalInstrument에 소속	owl:exactCardinality 1
C-07	상충 최소 규범 수	NormConflictCase는 최소 2개의 conflictingNorm 보유	owl:minCardinality 2
C-08	버전 분리 원칙	InstrumentVersion과 ProvisionVersion은 분리 관리	설계 규칙
C-09	위계 필수	모든 LegalInstrument는 정확히 1개의 hasNormativeLevel 보유	owl:exactCardinality 1
C-10	위계 방향성	NormativeHierarchyLink의 source.rank < target.rank	SWRL / SHACL
C-11	위임 정합성	DelegationLink의 위임 방향은 상위법→하위법	SWRL / SHACL

제2절. 법령 온톨로지 모델 및 데이터화

1. 법령 온톨로지 모델

가. 법령 온톨로지 전체 구조 특징

1) 추상 클래스 기반의 공통 메타 통일

- IdentifiedEntity / TitledEntity / TextBearingEntity / VersionedEntity / ReferencableEntity / LinkEntity 등 공통 상위 클래스를 두어, 식별자・제목・원문・버전・참조 가능성・링크 구조를 전 레이어에 일관되게 적용한다.

- 결과적으로 문서구조(조문)・규범명제・관계 링크가 모두 같은 규격으로 관리되어 확장・유지보수・정합성 검증을 용이하게 하였다.

2) 법령위계 레이어로 규범의 ‘법체계적 위치’ 표현

- 헌법-법률-시행령-시행규칙 등 상・하위 규범의 위계를 모델링하고, 위임・근거・저촉 판단의 기반이 되는 계층(랭크/레벨) 정보를 구조화한다.

- 이는 단순 링크가 아니라, 뒤의 “상충 해결/우선순위” 판단에서 상위규범 우선 원리를 적용할 수 있는 기준축을 제공한다.

3) 문서구조 레이어: ‘조문 단위’의 정밀한 구조화와 버전 관리

- LegalInstrument-StructuralUnit(편/장/절/조/항/호/목)로 본문을 계층 구조로 표현하고, ProvisionVersion 등으로 개정・시행에 따른 버전/유효기간을 분리한다.

- 별표・서식 등 부속 구성요소를 AppendixTable / AnnexTable 등으로 포함하여, 법령이 실제 행정 집행에서 요구하는 테이블/입력 스키마(기준표・산정표・서식 필드)까지 포괄한다.

4) 규범명제 레이어: ‘주체-행위-객체-요건-효과’로 의미 단위 표준화

- 조문 텍스트를 단순 저장하지 않고, NormStatement 중심으로 (주체) LegalRole - (행위) LegalAction - (객체) LegalObject - (요건) Condition - (효과) LegalEffect로 분해한다.

- 동시에 DeonticModality(의무/금지/허용/권한/면제), Sanction/AuthorityAction, ModificationLink(예외・특례・준용)을 포함하여, 법령의 규범적 성격(강제력・

조건부성・수정관계・제재 결합)을 추론 가능한 구조로 표현하였다.

5) 레이어 간 연결의 핵심: 조문-규범명제 '추적성(Traceability)'

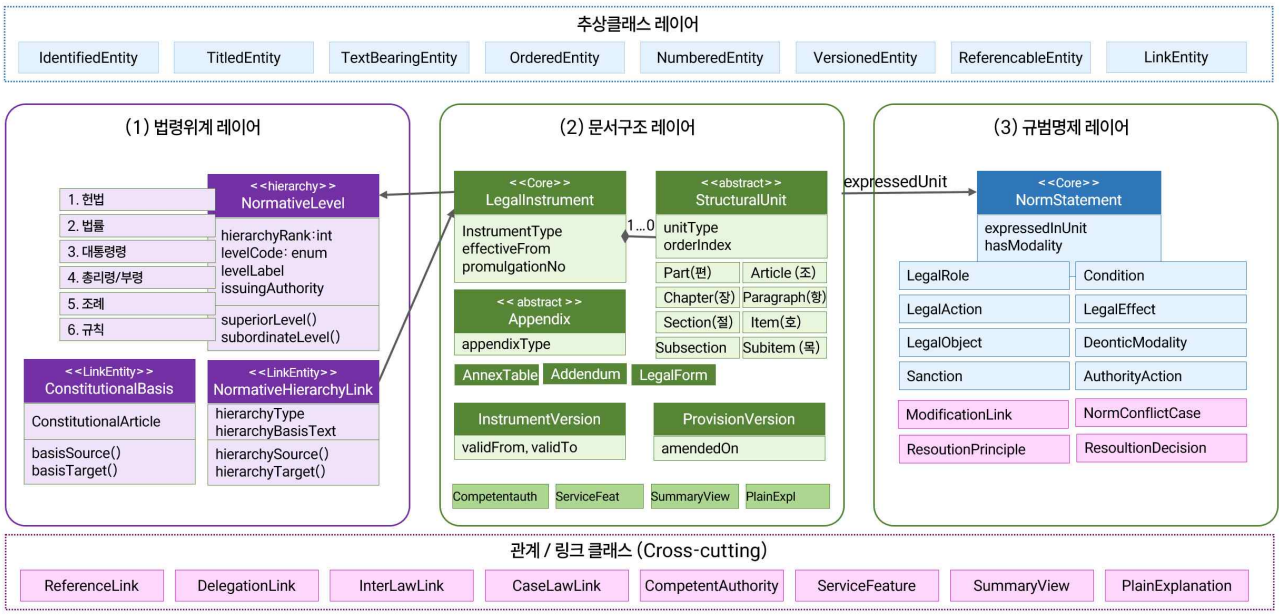
- 규범명제가 expressedInUnit 등으로 근거 조문(StructuralUnit)에 연결되어, "이 규범 판단이 어느 조문에서 나왔는가"를 역추적할 수 있다.

- 이는 법률지식시스템에서 설명가능한 검색/답변(근거 제시)을 가능하게 하는 핵심 설계 포인트다.

6) Cross-cutting 관계/링크 클래스로 법체계 네트워크를 별도 관리

- ReferenceLink / DelegationLink / InterLawLink / CaseLawLink 등 링크 클래스를 공통 레이어로 분리해, 참조・인용・위임・연관법・판례연계를 동일한 링크 스키마로 관리한다.

- 또한 NormConflictCase + ResolutionPrinciple + ResolutionDecision을 통해 상충을 1급 객체화하여, 상위규범 우선/특별규정 우선/후법 우선 등 우선순위 원리에 따른 해결 결과를 기록할 수 있다.



(그림3-5) 법령온톨로지 전체 구조

나. 법령 온톨로지 법령 위계 구조 설계 특징

본 연구에서 구성한 모델의 핵심 특징은 "위계(레벨)-법령(문서)-위계링크(근거/유형)"를 분리해서 자동판단 + 근거추적 + 다대다 연결이 가능하고 설명가능하도록 만들었다.

이 구조는 (a) 위계 레벨의 정규화(NormativeLevel) + (b) 근거를 담는 링크 엔티티(NormativeHierarchyLink/DelegationLink/ConstitutionalBasis)를 조합해, 노동법처럼 위임·연결이 많은 영역에서도 추적가능/설명가능/자동화 가능한 위계 모델을 만든 게 가장 큰 특징이다.

1) 위계 단계를 별도 객체로 정규화

- 헌법→법률(ACT)→대통령령→부령/총리령→조례→규칙처럼 법령 단계 자체를 클래스로 두고, superiorLevel / subordinateLevel로 단계 간 체인(연속 구조)을 구성하였다.

- 각 단계는 hierarchyRank, levelCode(InstrumentType)로 순서·유형이 명시되어 자동 판정의 기준이 된다.

2) 개별 법령(LegalInstrument)은 '정확히 1개의 단계' 에 귀속

- LegalInstrument — hasNormativeLevel (1) 구조로, "이 법령이 법체계에서 어느 위치인가" 를 데이터로 확정할 수 있다(분류의 일관성 확보)

3) 법령 간 관계를 '링크 엔티티(NormativeHierarchyLink)' 로 분리

- 법령↔법령 관계를 단순 엣지가 아니라 NormativeHierarchyLink라는 1급 객체로 모델링하여,

  - hierarchyType(SUPERIOR/INFERIOR/DELEGATION_BASIS/IMPLEMENTATION …)
  - hierarchyBasisText, anchorText, scope

같은 근거·범위 메타데이터를 함께 저장할 수 있으며, 결과적으로 "연결됨" 이 아니라 "왜/어떤 범위로 연결되는지" 까지 설명가능(Traceable)하다.

4) '위임근거' 는 조문(StructuralUnit)에서 발생하도록 분리(DelegationLink)

- DelegationLink가 source=조문(StructuralUnit) → target=하위 법령(LegalInstrument)로 연결된다.

  - 즉 "대통령령/부령으로 정한다" 같은 위임 문장이 있는 조문이 근거가 되고,
  - 이를 바탕으로 NormativeHierarchyLink의 DELEGATION_BASIS를 생성/검증할 수 있는 구조임.

5) '헌법적 근거(ConstitutionalBasis)를 별도 링크로 관리

- ConstitutionalBasis: 헌법 조문(StructuralUnit) → 법률(ACT)의 근거 관계를 명시하여 법률의 정당화/근거를 온톨로지에 포함시켜 체계적 근거 추적이 가

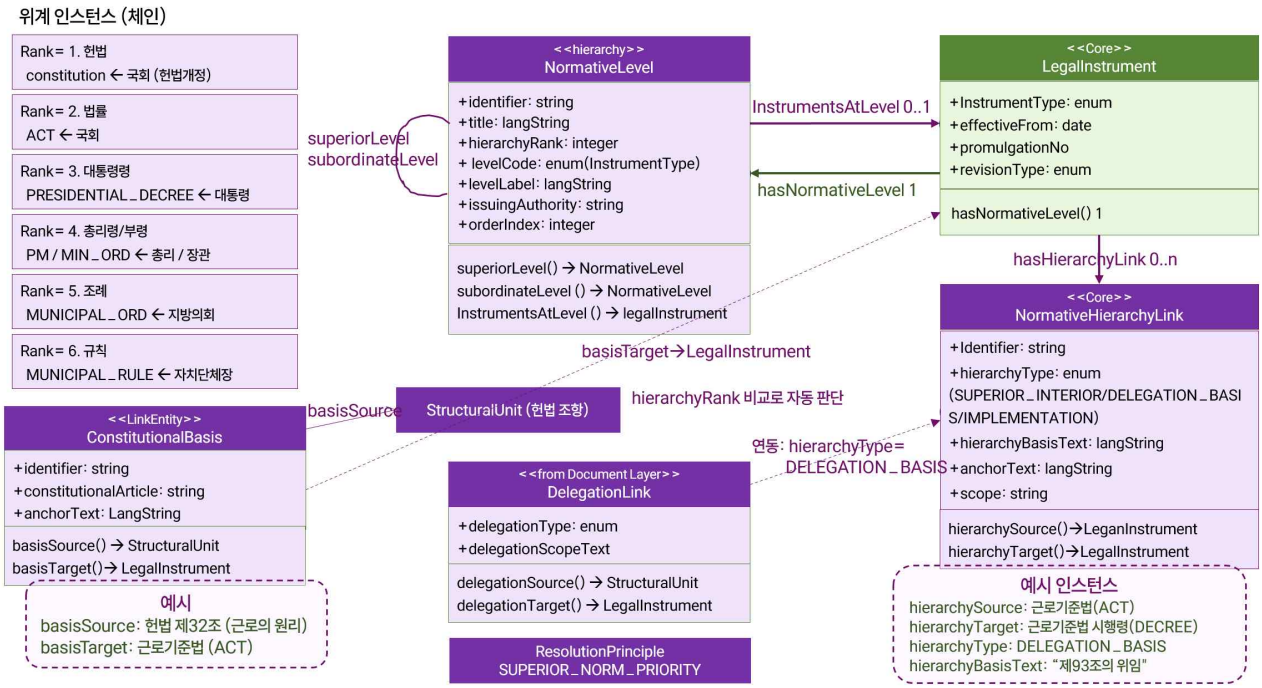
능하다.

6) ‘자동화·검증에 유리한 구조(‘rank 비교’ 기반)

- hierarchyRank가 있으므로 기본적인 상위/하위 판정은 규칙으로 자동 생성 가능하고, 위임은 DelegationLink로 근거가 있는 경우에만 DELEGATION_BASIS로 강화할 수 있도록 하여, 정합성 검사(위임 없는 하위규정, 저촉 가능성 등)에 활용 가능하다.

7) 상충 해결 원리와 연결될 확장 포인트를 제공

- 위계(레벨/랭크)가 명확하므로, 이후 상충 모델(상위규범 우선 등)과 결합 시 우선순위 판단의 객관적 기준축으로 바로 사용할 수 있다.



(그림3-6) 법령온톨로지 법령위계 설계 구조

다. 법령 온톨로지 문서 구조 설계 특징

본 연구에서 구성한 모델의 핵심 특징은 “법령 문서를 기계적으로 파싱·버전관리·부속(별표/서식)까지 구조화” 할 수 있게 만든 점으로 검색·추적·개정관리까지 바로 가능하게 만든 설계가 핵심이다.

이 설계는 (1) 조문 트리 구조의 정밀한 표현, (2) 별표·서식의 스키마화, (3) 법령/조문 버전의 분리 관리, (4) 위계·담당기관·서비스 기능까지 결합

가능한 확장성을 동시에 갖는 "실무형 문서구조 레이어"가 특징이다.

1) 법령 문서의 최상위 단위: LegalInstrument 중심 "컨테이너" 모델

- LegalInstrument가 법령 1건(법률/대통령령/부령 등)을 대표하고, 본문 구조(hasStructuralUnit), 부속(hasAppendix), 소관기관(hasAuthority), 서비스 요소(hasServiceFeature), 버전(hasVersion)을 한 곳에서 묶어 관리한다.

- 또한 hasNormativeLevel로 법령위계 레이어(NormativeLevel)와 연결되어, "이 문서가 법체계에서 어느 단계인지"를 문서구조 레이어에서 바로 참조 가능하다.

- hasAuthority, hasServiceFeature처럼 운영/서비스 확장 포인트를 루트에 둬서 시스템 기능(담당기관, 제공 기능)을 붙이기 쉽도록 하였다.

2) 본문 구조는 StructuralUnit(추상)로 정규화 + 타입 확장

- StructuralUnit을 추상 클래스로 두고 Part/Chapter/Section/Subsection/Article/Paragraph/Item/Subitem로 확장하였다.

- 모든 단위가 공통으로 unitType, orderIndex를 가져 표시 순서와 구조 파싱 결과를 안정적으로 유지할 수 있다.

- hasChildUnit(0..n), hasParentUnit(0..1)로 자기참조 계층 구조를 형성해, "편-장-절-조…" 트리 탐색을 자연스럽게 하였다.

- partOfInstrument(1)로 "이 구조단위가 어느 법령에 속하는지"가 명확하도록 구성하여 추적성을 확보하였다.

- 이를 공통 관계인 Part/Chapter/Section/Subsection(상위 구조)와 Article / Paragraph/Item/Subitem (조문 세부)로 명확히 분류해, 자동 파싱 결과를 안정적으로 담을 수 있게 설계하였다.

3) "문서 버전"과 "조문 버전"을 분리(시간적 효력/개정 반영)

- InstrumentVersion(법령 전체 버전): validFrom/validTo, revisionType, noteText

- ProvisionVersion(조문 단위 버전): validFrom/validTo, amendedOn, amendmentNoteText, derivedFrom

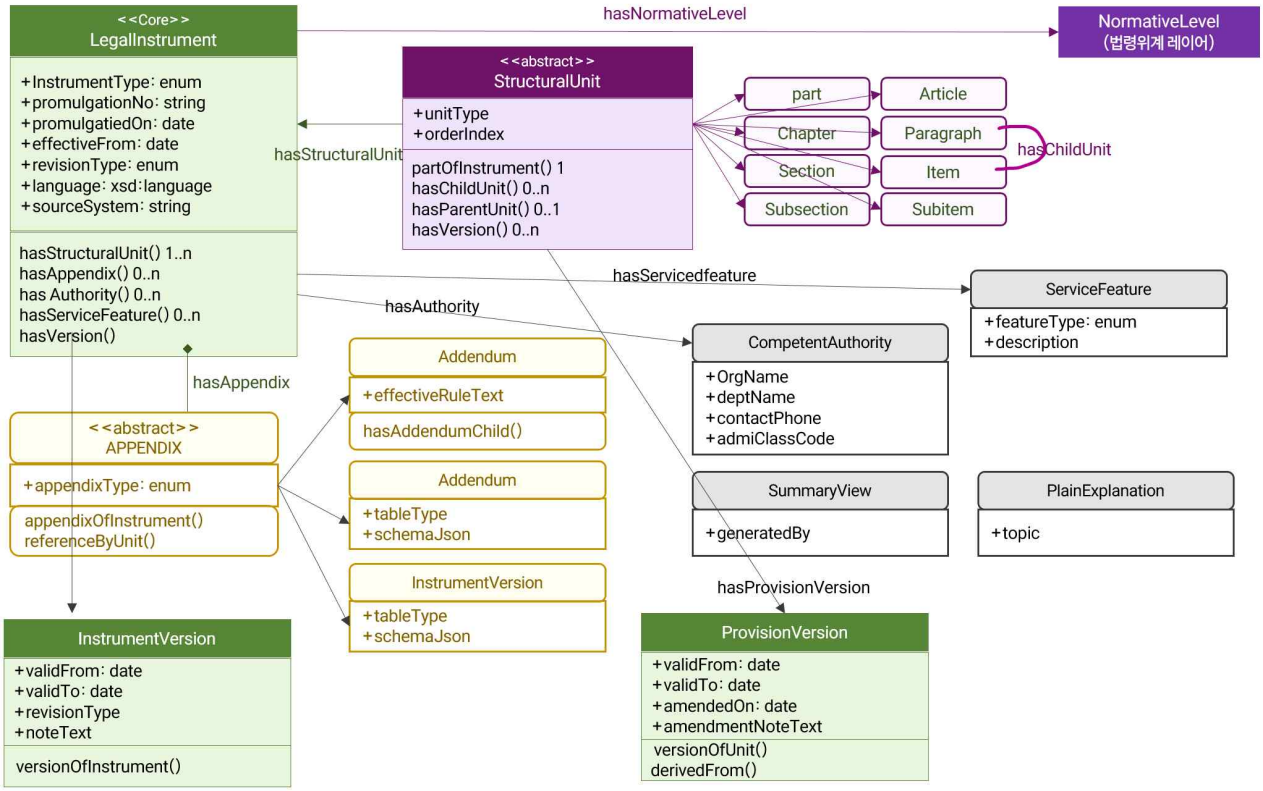
- 즉, 법령 전체의 개정 이력과 특정 조문의 개정/유효기간을 분리해 표현하므로 "법은 언제 시행되고, 어느 시점에는 어떤 조문 문구가 유효했는지"

를 정밀하게 관리하도록 구성하였다.

4) 부속 구성요소(부칙/별표/서식)를 '부록' 이 아니라 구조적 객체

- APPENDIX(abstract) + appendixType(enum)로 부속 유형을 구분하고, Addendum 쪽에 effectiveRuleText(부칙/경과/적용례 등 규정 텍스트) tableType + schemaJson(별표 기준표/산정표/서식 필드 등 테이블·입력 스키마)를 담도록 설계하였다.

- 특히 schemaJson을 두어 별표·서식이 실제 집행에서 "데이터 테이블/입력양식" 역할을 하므로, 이를 데이터 구조로 직접 연결할 수 있도록 하였다.



(그림3-7) 법령온톨로지의 문서 구조 레이어

5) 법령위계 레이어(NormativeLevel)와의 연결을 내장

- LegalInstrument — hasNormativeLevel → NormativeLevel로, 문서구조 레이어가 법령 위계(법률/시행령/시행규칙…)와 즉시 결합 가능하기 때문에 이후 위임/저촉/우선순위 같은 규칙 기반 추론의 기반이 될 수 있다.

6) "서비스용 표현" 을 별도 클래스로 분리해 확장 가능

- SummaryView, PlainExplanation 같은 클래스를 두어, 원문 구조와 분리된 요약/쉬운 설명을 붙일 수 있게 하였다(생성주체 generatedBy 등 확장 가능)

라. 법령 온톨로지 규범명제 설계 특징

본 연구는 법령 문서(조문) 구조”를 그대로 두면서, 그 위에 규범을 실행·추론 가능한 단위로 구조화한 규범명제 중심 법령구조 레이어라는 게 핵심 특징이다. 이 레이어는 조문 텍스트를 ‘규범명제 단위’로 분해하면서도 근거 조문을 잃지 않고, 동시에 양상성·조건·효과·제재·예외·상충해결까지 한 번에 표현해서 정확도(규범 적용) + 설명가능성(근거 제시) + 확장성(위계/연관법 결합)을 같이 확보한 구조이다.

1) 조문(StructuralUnit)-규범명제(NormStatement) ‘추적성’

- StructuralUnit — expressedInUnit(1..n) → NormStatement로 연결되어, 어떤 규범이 어느 조문에서 나왔는지를 항상 역추적할 수 있게 설계하였다.

2) 법의 내용적 특징을 슬롯으로 정규화(주체-행위-객체)

- NormStatement가 의미역할 슬롯을 갖고 LegalRole / LegalAction / LegalObject로 분해하였다(예: 사용자(주체)-지급(행위)-임금(객체))

- 조문 텍스트를 “명사 추출”로 끝내지 않고 사실관계 구성요소로 표준화해 검색/추론에 유리하도록 구성하였다.

3) 규범적 특징을 별도 노드로 분리(양상성·조건·효과)

- hasModality → DeonticModality로 의무/금지/허용/권한/면제를 명시(규범력 표현)하였다.

- hasCondition → Condition으로 “~인 경우/기한/정량/절차” 같은 조건부성을 구조화하였다.

- hasEffect → LegalEffect로 규범의 법적 효과(의무 발생, 권리 발생, 제재 가능 등)를 분리하였다.

4) 제재·집행(권한)까지 한 그래프에서 연결

- hasSanction → Sanction, authorityMayDo → AuthorityAction, enforcedBy-Authority → CompetentAuthority를 구성해 "지켜야 함"에서 끝나지 않고 "위반 시 무엇이 발생하고(제재), 누가 집행하는지(권한/기관)까지 연결"하였기 때문에 → 행정 집행형 서비스에 적합하도록 구성하였다.

5) 예외·특례·준용을 ‘태그’가 아니라 수정관계로 모델링

- hasModificationLink → ModificationLink가

- modifiesTarget(원칙 NormStatement)
- modificationCondition(언제/어떤 조건에서)
- modificationType(예외/특례/준용 등), anchorText, scope

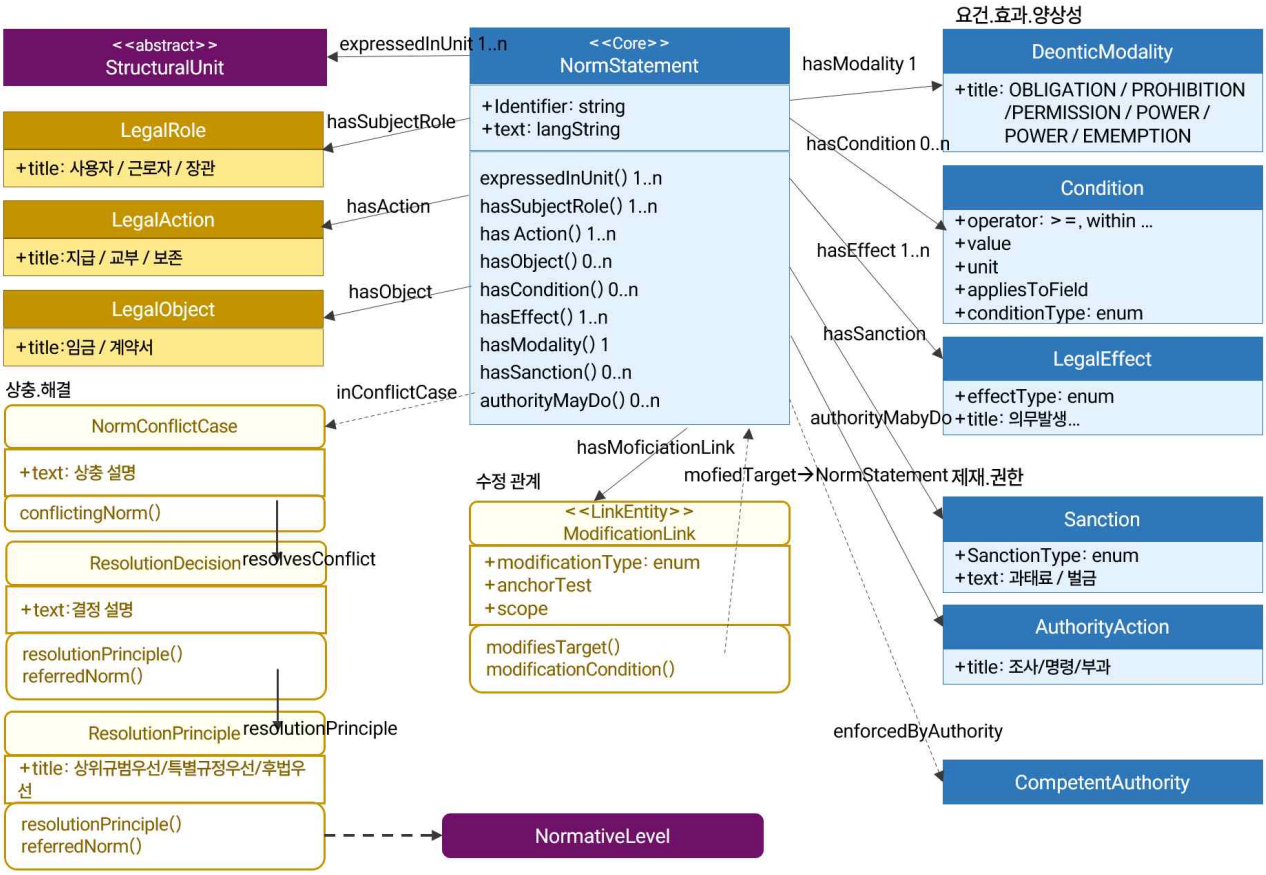
- 즉 "원칙을 어떤 조건에서 어떻게 바꾸는지"가 남아 규정 적용의 정확도를 올리도록 설계하였다.

6) 상충과 우선순위를 1급 객체로 처리(설명 가능한 충돌 해결)

- NormConflictCase에 상충 규범들을 묶고, ResolutionDecision이 Resolution-Principle (상위우선/특별우선/후법우선)을 근거로 우선 적용 규범을 기록하였다. → 단순 결과가 아니라 왜 그 규범이 우선인지를 데이터로 설명 가능.

7) 위계 레벨(NormativeLevel)과의 연결 포인트 확보

- 하단 NormativeLevel이 붙어 있어, 규범명제/조문/법령 위계 레이어와 결합 시 상위규범 우선 같은 원리를 기계적으로 적용·검증할 수 있는 확장 구조를 가진다.



(그림3-8) 법령온톨로지의 규범명제 레이어

마. 법령 온톨로지 관계/링크 설계 특징

이 설계의 핵심은 LegalInstrument(법령 문서)와 StructuralUnit(조문 구조 단위)를 "내용의 뼈대(Core)"로 두고, 법체계에서 중요한 연결(참조・위임・연관・위계・헌법근거・판례연계)을 전부 LinkEntity(링크 엔티티)로 분리・타입화(reify) 했다는 점으로 그 결과, "법령은 네트워크로 작동한다"는 특성을 추적 가능(Traceable)・설명 가능(Explainable)하게 모델링되었다.

특히 모델링 관점에서 관계/링크 구조를 통해 아래와 같은 특징을 보완하였다.

- **정합성(Consistency)**: 각 관계는 도메인/레인지를 명확하도록 설계 (예: 위임은 반드시 source=조문)
- **확장성(Extensibility):** 링크 타입을 추가로 기능 확장하였음(예: "적용배제", "부분준용" 등)
- **추적성(Traceability):** 모든 관계가 근거 조문/문구(anchorText)로 역추적 가능
- **설명가능성(Explainability)**: "왜 이 답이 나왔는지"를 링크 메타로 제시 가능
- **영향분석(Impact Analysis)**: 특정 법령 개정 시 참조/위임/연관 링크를 따라 영향 범위 계산 가능

1) Core(문서/조문)와 Link(관계)를 분리한 구조

- 가운데 LegalInstrument ↔ StructuralUnit이 "법령 자체"와 "조문 구조"를 담고, 주변의 ReferenceLink / DelegationLink / InterLawLink / CaseLawLink / NormativeHierarchyLink / ConstitutionalBasis가 관계를 담당하도록 구성하였다.

- 이렇게 분리하여 아래와 같은 기능을 모델링 하였다.
  - 법령/조문은 안정적으로 유지(정규화)
  - 관계는 필요에 따라 확장(새 링크 타입 추가) 가능
  - 관계 메타데이터(근거문구/범위/유형)를 풍부하게 저장 가능

2) 관계를 1급 객체로 두어 "근거・범위・유형"을 저장(설명가능성)

- 각 링크 엔티티는 단순 "선"이 아니라 속성을 가진다.

(1) ReferenceLink — 참조/인용/준용을 타입화

- referenceType에 CITATION / REFERENCE / MUTATIS_MUTANDIS(준용) / RECEPTION / SPECIAL(특례)처럼 법적 의미를 구분
- anchorText, scope로 어떤 문구를 근거로 어디까지 적용되는지 표현

→ **"왜 그 조문을 가져왔는가 / 어느 범위로 가져왔는가"를 기록 가능**

(2) DelegationLink — 위임을 "조문 근거"로 고정

· referenceSource() → StructuralUnit로, 위임의 근거는 반드시 조문 텍스트가 되도록 구성

· delegationType(TO_PRES_DECREE/TO_ORDINANCE/TO_RULE)로 위임 대상 하위 규범 유형을 명시

· delegationScopeText로 위임 범위를 텍스트로 보존

→ **"대통령령으로 정한다"** 같은 문장을 근거로 하위법령 생성/검증이 가능

(3) InterLawLink — 법령 간 연관을 연결 이유까지

· relationType, sharedElement를 둬서 "관련 있음" 수준을 넘어서

· 어떤 요소(행위/객체/요건/목적 등)를 공유해서 연결되는지 표현 가능

→ **노동·고용처럼 다법령 결합이 많은 영역**에 적합

3) 위계(법령↔법령)와 위임(조문→하위법령)을 분리하고 "연동" 가능

- NormativeHierarchyLink — 법령 간 위계/구체화 관계

· hierarchySource/Target : LegalInstrument → LegalInstrument

· hierarchyType : SUPERIOR/INFERIOR/DELEGATION_BASIS/IMPLEMENTATION

· hierarchyBasisText로 근거 텍스트를 보관

- DelegationLink → NormativeHierarchyLink 연동(그림의 점선 의미)

· 그림에 "연동: hierarchyType = DELEGATION_BASIS"가 표시된 것처럼, 조문에서 발견된 위임(DelegationLink)을 근거로 법령 간 위계 링크(NormativeHierarchy-Link)를 생성/검증하도록 설계

→ "문장 근거 기반(조문)" + "법체계 관계(법령↔법령)"가 끊기지 않음

4) 헌법 근거와 판례 연계를 같은 프레임으로 포괄

- ConstitutionalBasis — 헌법 조문 → 법률 근거

· basicSource() → StructuralUnit, basisTarget() → LegalInstrument

· constitutionalArticle로 조문 식별

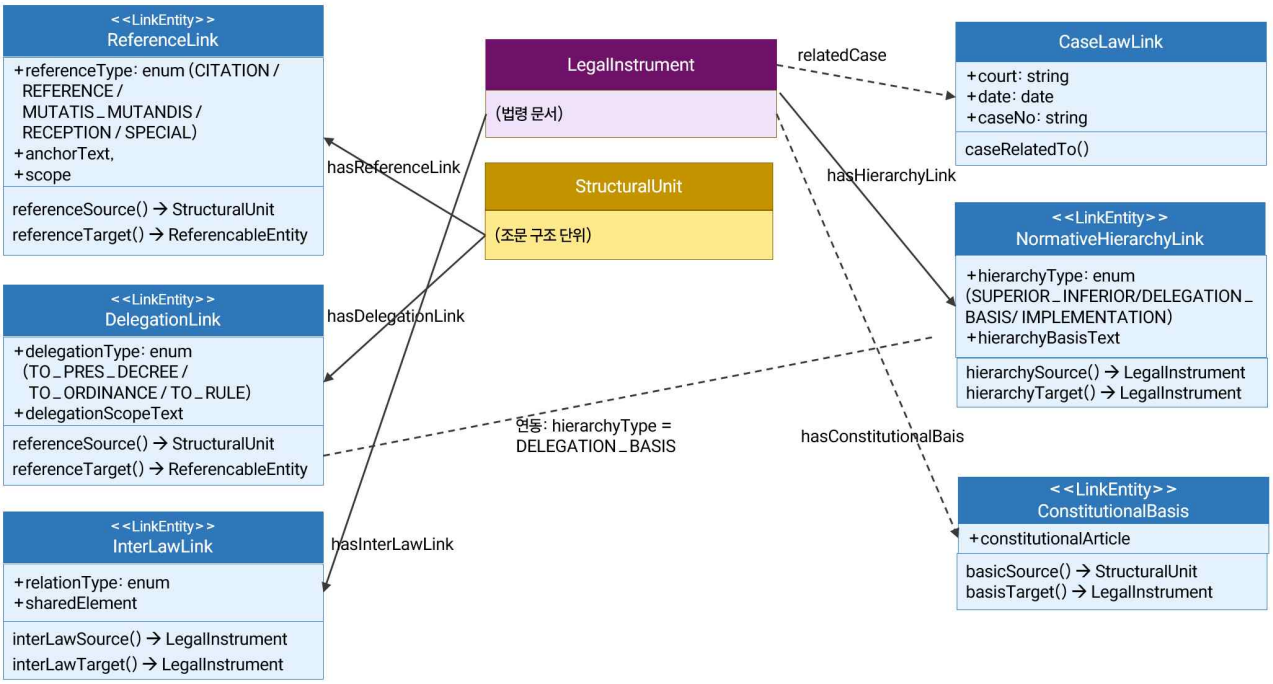
→ 법률의 "헌법적 근거"를 데이터로 보존해 체계적 정당화/근거 추적 가능

- CaseLawLink — 법령과 판례 연결

· court/date/caseNo 메타를 포함

· caseRelatedTo()로 연결

→ 법령 중심 지식그래프가 판례(해석/적용)까지 확장될 수 있음



(그림3-9) 법령온톨로지의 관계/링크 레이어

바. 법령 온톨로지 추상클래스 설계 특징

법령 온톨로지 설계의 추상 클래스(공통 상위 클래스) 설계는 한마디로 "법령을 구성하는 모든 객체에 공통 메타(식별・표제・원문・버전・참조・링크)를 '기능별 믹스인(mixin) 상속'으로 표준화한 구조로 아래와 같은 주요 특징을 갖는다.

특히 모델링 관점에서 추상화 기법을 통해 아래와 같은 특징을 보완하였다.

- **일관성**: title/text/version/reference/link 메타가 전 도메인에 동일 규칙으로 적용
- **확장성**: 새 클래스 추가 시 필요한 기능만 상속하면 되어 스키마 확장 부담 감소
- **추적성/설명가능성**: LinkEntity(앵커·범위) + ReferenceableEntity(참조대상 통일)로 "왜/어디를 근거로"가 남음
- **시간/개정 대응성:** VersionedEntity 축이 있어 버전·효력 기반 질의로 자연 확장
- **운영 편의성**: 공통 속성의 도메인/레인지 정의가 상위 클래스에서 정리되므로 검증(Shape/Constraint) 설계가 쉬움

1) 설계 철학: "기능별 공통 메타의 분리 + 재사용"

- Identified / Titled / TextBearing / Versioned / Referenceable / LinkEntity처럼 역할(기능) 단위로 추상 클래스를 쪼개두고, LegalInstrument, StructuralUnit, Appendix, NormStatement 등 도메인 클래스가 필요한 기능만

선택적으로 상속하였기 때문에 결과적으로 같은 속성(예: title, text, identifier)을 각 클래스에 반복 정의하지 않아 중복(DRY) 제거, 일관성을 확보하였다.

2) 핵심 추상 클래스들의 의미(표준화 대상 구성)

(1) IdentifiedEntity / NumberedEntity

- IdentifiedEntity: 모든 리소스에 identifier를 강제해 IRI/키 관리와 통합검색·연계의 기준을 제공
- NumberedEntity: "제○조/제○항/제○호" 같은 번호 체계가 중요한 개체에 별도 축을 제공(정렬·표시·검색에 유리)

(2) TitledEntity / TextBearingEntity

- TitledEntity: "표제/조명/별표명"처럼 라벨이 필요한 객체의 공통 속성(title)을 통일
- TextBearingEntity: 조문 원문, 부칙 문장, 규범명제 문장 등 텍스트 본문을 갖는 객체를 한 축으로 묶어 원문 보존/인용을 일관되게 처리

(3) VersionedEntity

- hasVersion → Version로 분리해 시간적 효력(유효기간/개정 이력)을 구조적으로 붙일 수 있게 함
- 법령/조문/규범명제가 모두 "버전"을 가질 수 있어, '언제' 기준 질의(시행·경과·적용례)로 확장하기 좋은 기반임

(4) ReferenceableEntity

- "참조 대상이 될 수 있는 것"을 별도 축으로 두어, 참조의 타깃을 LegalInstrument만이 아니라 StructuralUnit, Appendix 등으로 넓히는 범용 참조 모델을 구성

(5) LinkEntity

- 링크(참조/위임/준용/특례/연관)를 단순 엣지가 아니라 1급 객체로 모델링하기 위한 기반
- anchorText, scope 같은 속성이 공통으로 들어가서 "어떤 문구를 근거로(앵커)" "어느 범위까지(스코프)"를 링크마다 동일한 방식으로 기록
  → 설명가능성(왜 연결됐는가)을 강화

3) 레이어별 적용이 자연스러운 구조

(1) 문서구조 레이어

- LegalInstrument: Identified + Titled + Versioned + Referenceable (법령 전체 단위)

· StructuralUnit: Identified + Titled + TextBearing + Ordered/Numbered + Versioned + Referenceable (조문 구조 단위)

· Appendix: Identified + Titled + TextBearing + Referenceable (별표/서식)

(2) 규범명제 레이어

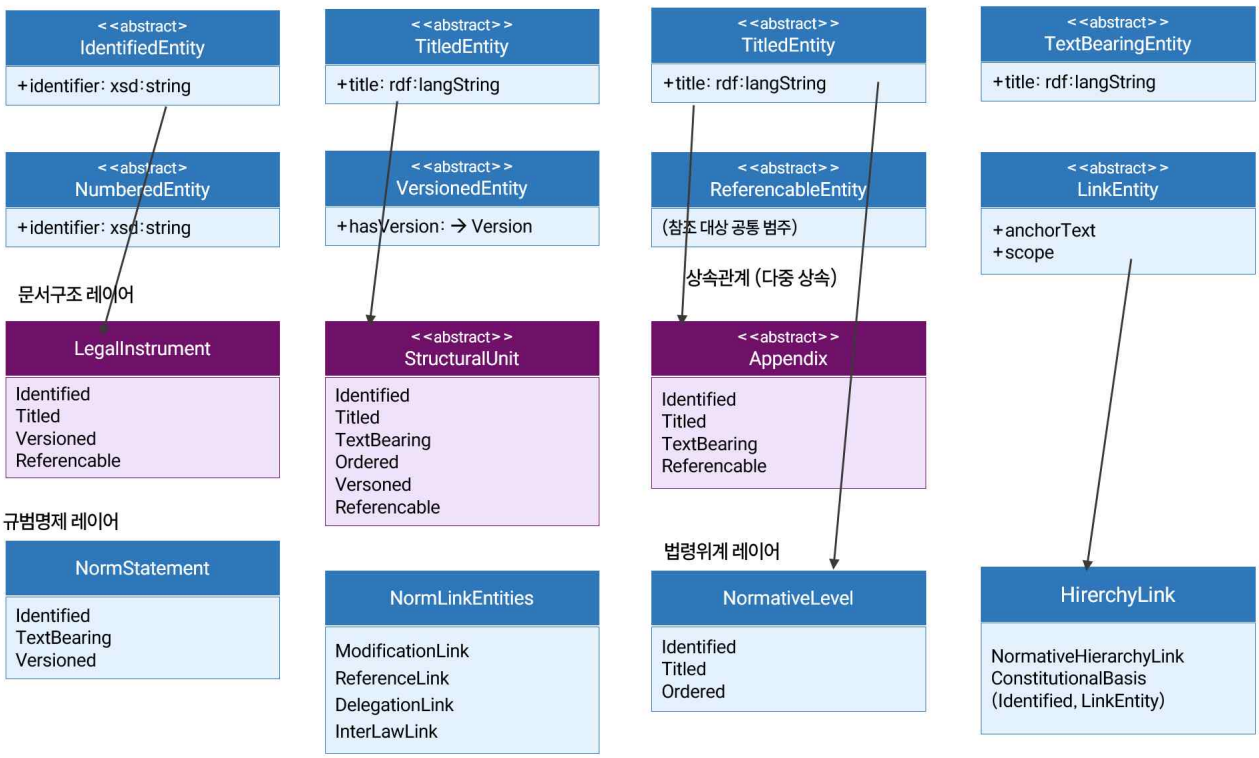
· NormStatement: Identified + TextBearing + Versioned (규범 단위는 원문/버전이 핵심)

· NormLinkEntities: Modification/Reference/Delegation/InterLawLink 등은 LinkEntity 계열로 정리

(3) 법령위계 레이어 레이어

· NormativeLevel: Identified + Titled + Ordered (위계 단계는 정렬/표시가 중요)

· HierarchyLink: NormativeHierarchyLink, ConstitutionalBasis 등(Identified + LinkEntity)로 묶어 위계·근거 링크를 일관화



(그림3-10) 법령온톨로지의 추상클래스

2. 법령 온톨로지 모델 특징

가. 법령 온톨로지 설계 원칙

본 연구는 법령 온톨로지가 갖추어야 할 핵심 요구를 일관성(consistency),

확장성(extensibility), 추적성(traceability), 설명가능성(explainability)으로 설정하고, 이를 스키마 수준에서 보장하는 것을 설계 원칙으로 삼았다.

특히 법령은 (i) 문서 구조가 정형화되어 있으면서도, (ii) 개정・시행・경과 등 시간적 효력이 강하게 작동하고, (iii) 참조・준용・특례・위임 등 관계를 통해 네트워크로 의미를 획득한다. 따라서 개별 클래스에 속성을 중복 정의하기보다, 법령 전반에 공통적으로 요구되는 메타데이터를 상위 추상 클래스로 분리하여 재사용하고, 관계 또한 단순 링크가 아닌 1급 객체로 구조화함으로써 규범 지식의 정합성과 운용 가능성을 동시에 확보하였다.

나. 법령 온톨로지 구현전략

첫째, 공통 메타데이터를 기능 단위로 분해한 추상 클래스(예: IdentifiedEntity, TitledEntity, TextBearingEntity, VersionedEntity, ReferenceableEntity, LinkEntity)를 정의하고, 도메인 클래스가 필요 기능을 선택적으로 상속하도록 구성하였다. 이를 통해 식별자・표제・텍스트・버전・참조 가능성 등의 속성 정의를 상위 계층에서 통제하며, 클래스 간 속성 의미를 통일하였다.

둘째, 법령 문서(legal instrument), 조문 구조(structural unit), 부속 구성요소(appendix), 규범명제(norm statement) 등 핵심 클래스를 레이어별로 배치하되, 동일한 추상 메타를 공유하도록 설계하여 레이어 간 단절을 방지하도록 하였다.

셋째, 참조・위임・준용・연관・위계・판례 연계 등 법체계 연결은 LinkEntity 기반의 링크 클래스로 객체화하고, anchorText・scope 등 근거 및 적용 범위를 공통 필드로 유지한다. 마지막으로 시간적 효력은 VersionedEntity 및 Version 객체를 통해 모델링하여, 법령/조문/규범명제 단위로 개정 이력과 유효기간을 분리 관리할 수 있도록 하였다.

다. 법령 온톨로지 모델 설계의 기대효과

본 설계는 (1) 속성 중복을 최소화하고 상위 계층에서 메타데이터 정책을 통제함으로써 스키마 일관성과 유지보수성을 향상시킨다. (2) 법령・조문・부속・규범명제가 동일한 공통 메타를 공유하므로, 데이터 확장(법령 추가, 부속요소 확장, 규범 레이어 확장) 시 구조 변경 비용이 감소하여 확장성이 강화된다. (3) 관계를 링크 엔티티로 객체화함으로써 “무엇을/어떤 근거로/어느 범위에서” 연결되는지 기록이 가능해져, 법률지식시스템에서 요구되는 근거 제시 및 설명가능한 추론을 지원한다. (4) 버전과 유효기간의 분리 표현은 시

행·경과·적용례 등 시간 기반 질의를 가능하게 하여, 시점에 따라 달라지는 법 적용을 정밀하게 처리하는 시간적 정확성을 제고한다.

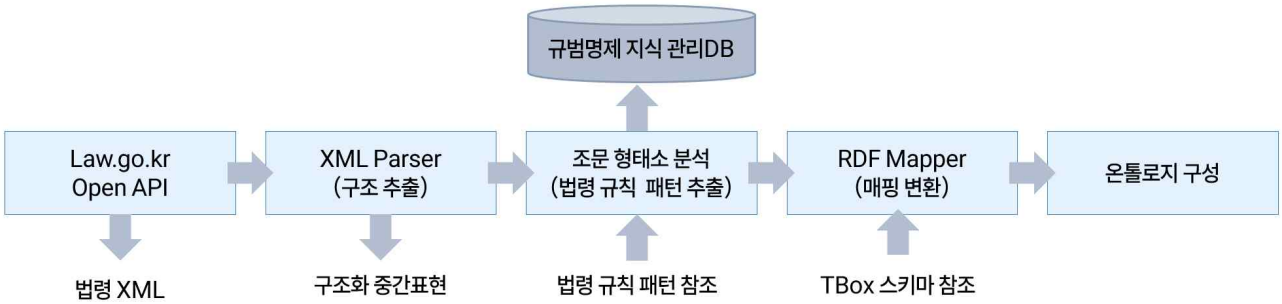
종합하면, 본 연구의 추상 클래스 기반 설계는 법령의 구조적·시간적·관계적 특성을 온톨로지 차원에서 통합적으로 수용함으로써, 법률지식시스템 구현을 위한 안정적 기반을 제공할 수 있다.

제3절. 노동법(근로기준법 중심) 온톨로지 구축

1. (노동)법령 온톨로지 구축 프로세스

가. 파싱 파이프라인 구성

파싱 파이프라인은 law.go.kr API의 XML 구조를 온톨로지 매핑 규칙에 따른 변환 로직을 이용하여 온톨로지 인스턴스를 자동 생성하는 과정이다. 이를 위해 법령 온톨로지의 ABox(인스턴스)를 자동 생성하기 위한 데이터 파이프라인을 설계하였고, 이는 국가법령정보센터(law.go.kr)가 제공하는 Open API의 XML 응답을 파싱하여 RDF 트리플(Turtle 형식)로 변환하는 5단계 프로세스로 구성하였다.



(그림3-11) 법령온톨로지 구축을 위한 파싱 파이프라인

나. 데이터 소스: law.go.kr Open API

1) 사용 API

API	엔드포인트	반환 데이터	매핑 대상 레이어
법령 본문 조회	LawService.do ?OC=...&target=law&MST=...	법령 메타 + 조문 전체 XML	문서구조 레이어
법령 연혁 조회	LawService.do ?target=hist	개정 연혁 목록	InstrumentVersion
법령 체계도 조회	LawService.do ?target=prec	상·하위법, 위임근거	법령위계 레이어
법령 검색	LawSearchService.do	법령 목록/메타	LegalInstrument 기본

2) XML 응답 구조

law.go.kr API의 법령 본문 XML은 다음과 같은 계층 구조를 갖는다.

<법령>

```
<기본정보>
  <법령명>근로기준법</법령명>
  <법령ID>005309</법령ID>
  <공포번호>제5309호</공포번호>
  <공포일자>19970313</공포일자>
  <시행일자>19970313</시행일자>
  <법령구분>법률</법령구분>
  <소관부처>고용노동부</소관부처>
</기본정보>
<조문>
  <조문단위>
    <조문키>001</조문키>
    <조문번호>제1조</조문번호>
    <조문제목>목적</조문제목>
    <조문내용>이 법은 헌법에 따라 ...</조문내용>
    <항>
      <항번호>①</항번호>
      <항내용>이 법에서 사용하는 ...</항내용>
      <호>
        <호번호>1</호번호>
        <호내용>"근로자"란 ...</호내용>
      </호>
    </항>
  </조문단위>
</조문>
<부칙> ... </부칙>
</법령>
```

다. XML의 온톨로지 매핑 규칙

1) 문서구조 레이어 매핑

XML 요소/속성	온톨로지 클래스	온톨로지 속성	변환 규칙
<법령>	LegalInstrument	—	루트 요소 → 1개 인스턴스 생성
<법령명>	—	:title	텍스트 → rdf:langString @ko
<법령ID>	—	:identifier	'LSA_KR_' + 텍스트

XML 요소/속성	온톨로지 클래스	온톨로지 속성	변환 규칙
<공포번호>	—	:promulgationNo	텍스트 그대로
<공포일자>	—	:promulgatedOn	YYYYMMDD → xsd:date 변환
<시행일자>	—	:effectiveFrom	YYYYMMDD → xsd:date 변환
<법령구분>	—	:instrumentType	'법률'→ACT, '대통령령'→ PRESIDENTIAL_DECREE 등 매핑
<법령구분>	—	:hasNormativeLevel	instrumentType에 따라 해당 NormativeLevel IRI 연결
<소관부처>	CompetentAuthority	:orgName	기관명으로 CompetentAuthority 인스턴스 생성
<편>	Part	:number, :title	편번호/편제목 추출, hasParentUnit=LegalInstrument
<장>	Chapter	:number, :title	장번호/장제목 추출
<절>	Section	:number, :title	절번호/절제목 추출
<관>	Subsection	:number, :title	관번호/관제목 추출
<조문단위>	Article	:number, :title, :text	조번호/조제목/조내용 추출
<항>	Paragraph	:number, :text	항번호/항내용 추출, hasParentUnit=Article
<호>	Item	:number, :text	호번호/호내용 추출, hasParentUnit=Paragraph
<목>	Subitem	:number, :text	목번호/목내용 추출, hasParentUnit=Item
<부칙>	Addendum	:effectiveRuleText	부칙 내용 → Appendix 인스턴스
<별표>	AnnexTable	:tableType	별표 메타 추출
<서식>	LegalForm	:formNo	서식번호/서식명 추출

2) 법령위계 레이어 매핑

소스데이터	온톨로지 클래스	변환 규칙
<법령구분> + 고정 NormativeLevel 인스턴스	hasNormativeLevel	instrumentType을 NormativeLevel IRI로 매핑 (ACT→LV_ACT, PRESIDENTIAL_DECREE→ LV_PRESIDENTIAL_DECREE 등)
법령체계도 API: 상위법 정보	NormativeHierarchyLink	hierarchyType=SUPERIOR_INFERIOR, hierarchySource=상위법, hierarchyTarget=현재법
조문 내 '대통령령으로 정하는' 패턴	DelegationLink + NormativeHierarchyLink	delegationType=TO_PRESIDENTIAL_DECREE, hierarchyType=DELEGATION_BASIS로 이중 생성
제1조 '헌법에 따라' 패턴	ConstitutionalBasis	basisSource=헌법 관련 조항, basisTarget=현재 법령

3) 규범명제 레이어 매핑

규범명제 레이어의 인스턴스 생성은 조문 텍스트의 의미 분석이 필요하므

로, 완전 자동화가 어렵기 때문에 다음의 반자동 접근으로 설계하였다.

단계	처리 방식	설계
1. 양상성 추출	규칙 기반 자동	패턴 매칭: '~하여야 한다'→OBLIGATION '~하지 못한다'→PROHIBITION '~할 수 있다'→PERMISSION '~을 금지한다'→PROHIBITION
2. 주체/객체 추출	NER + 규칙 기반	법적 주체 사전(사용자, 근로자, 장관 등) 기반 매칭 + 조사 패턴('은/는' 앞 명사구 → 주체)
3. 요건 추출	규칙 기반 + LLM 보조	수치 패턴('N명 이상', 'N일 이내')→QUANTITATIVE 시간 패턴('이내', '까지')→TEMPORAL '경우에는', '때에는'→조건절 범위 추출
4. 벌칙 연결	규칙 기반 자동	벌칙 조항의 '제N조를 위반한 자' 패턴 → ReferenceLink(CITATION) + Sanction 연결

라. 법령 조문 패턴 분석 및 온톨로지 매핑 규칙

법령 조문에는 위계, 위임, 참조, 제재 등의 관계를 나타내는 정형화된 언어 패턴이 존재한다. 이러한 패턴을 유형별로 분류하고, 각 패턴이 온톨로지의 어떤 클래스와 속성으로 매핑되는지를 정의하였다.

1) 양상성 추출 패턴

법령 조문의 문말(文末) 표현에서 규범의 의무・금지・허용・권한・면제를 자동으로 판별한다.

양상성 코드	한글명	문말 패턴 (정규표현식)	예시 조문
OBLIGATION	의무	· ~하여야 한다 · ~할 의무가 있다 · ~하여야 한다[.)] · ~하도록 하여야 한다	사용자는 근로자에게 임금을 지급하여야 한다 (제43조)
PROHIBITION	금지	· ~하지 못한다 · ~하여서는 아니 된다 · ~아니 된다 · ~을(를) 금지한다 · ~할 수 없다 (맥락상 금지)	사용자는 … 강요하지 못한다 (제7조)
PERMISSION	허용	· ~할 수 있다 · ~을(를) 허용한다 · ~을(를) 인정한다	근로자가 … 청구하면 거부하지 못한다 (제10조)
POWER	권한	· ~을(를) 명할 수 있다 · ~할 권한이 있다 · ~을(를) 명하여야 한다 · ~처분할 수 있다	고용노동부장관은 … 명할 수 있다 (제104조)
EXEMPTION	면제	· ~적용하지 아니한다 · ~을(를) 면제한다 · ~에서 제외한다 · ~적용을 배제한다	… 적용하지 아니한다 (제11조 단서)

하나의 조문에 복수 패턴이 존재할 경우에는 **PROHIBITION > OBLIGATION > EXEMPTION > POWER > PERMISSION 순서로 판별**한다. 단서('다만, …')가 포함된 경우 본문과 단서를 분리하여 각각 별개의 NormStatement로 생성한다.

2) 위임 추출 패턴

상위법이 하위법에 세부사항을 위임하는 표현 패턴이다. DelegationLink와 NormativeHierarchyLink(DELEGATION_BASIS)를 동시에 생성한다

위임 패턴	DelegationType	생성 인스턴스	예시 조문
대통령령으로 정한다 대통령령으로 정하는 바에 따라	TO_PRESIDENTIAL_DECREE	DelegationLink NormativeHierarchyLink (DELEGATION_BASIS)	이 법의 일부 규정을 적용할 수 있다. 대통령령으로 정하는 바에 따라 (제11조③)
총리령으로 정한다 총리령으로 정하는 바에 따라	TO_ORDINANCE (총리령)	DelegationLink + HierarchyLink	-
부령으로 정한다 고용노동부령으로 정한다 ○○부령으로 정하는	TO_ORDINANCE (부령)	DelegationLink + HierarchyLink	서식은 고용노동부령으로 정한다
규칙으로 정한다 규칙으로 정하는 바에 따라	TO_RULE	DelegationLink + HierarchyLink	조례 조문에서 규칙 위임
제○○조에서 정하는 바에 따른다 제○○조에 따라 정한다	TO_PROVISION	DelegationLink (동일 법령 내)	제93조에서 정하는 바에 따라

위임 패턴이 감지되면 (1) delegationSource는 해당 조문(Article/Paragraph), (2) delegationTarget은 위임 유형에 따른 하위 법령(LegalInstrument), (3) delegationScopeText는 패턴 주변 문맥을 포함한다. 동시에 Normative-HierarchyLink(hierarchyType=DELEGATION_BASIS)를 생성하여 법령위계 레이어와 연동한다.

3) 참조・인용(ReferenceLink) 추출 패턴

조문 간 텍스트 내 참조 관계를 식별한다.

참조 패턴	ReferenceType	의미	예시
제○○조에 따른 제○○조에 따라 제○○조에서 정하는	CITATION	특정 조항을 명시적으로 언급	이 법은 헌법에 따라 (제1조)

참조 패턴	ReferenceType	의미	예시
제○○조를 참조한다 제○○조를 참고한다	REFERENCE	다른 조항의 내용을 참고하도록 지시	-
제○○조를 준용한다 제○○조의 규정을 준용	MUTATIS_MUTANDIS	다른 조항을 필요한 수정을 가하여 적용	제50조의 규정을 준용한다
다만, …인 경우에는 그러하지 아니하다	EXCEPTION	원칙 규정의 적용을 배제	다만, 동거 친족만을 사용하는 … 적용하지 아니한다 (제11조①)
제○○조에도 불구하고 제○○조에 관계없이	SPECIAL_PROVISION	일반 규정에 대한 특별한 예외	제55조에도 불구하고
별표 ○에 따른다 별표 ○과 같다 별지 제○호 서식	APPENDIX_REF	별표 또는 서식 참조	별표 1과 같다 (시행령 제7조)

조문 번호 추출 정규 표현식은 다음과 같이 구성하였다.

제(₩d+)조(?:의(₩d+))?(?:제(₩d+)항)?(?:제(₩d+)호)?

이 패턴으로 '제11조제3항', '제7조', '제2조제1항제1호' 등의 참조 대상을 구조적으로 파싱하여 referenceTarget의 StructuralUnit 인스턴스와 자동 연결한다.

4) 벌칙·제재(Sanction) 연결 패턴

벌칙 조항(제12장)에서 위반 대상 조문과 제재 내용을 추출한다

벌칙 패턴	ReferenceType	의미	예시
제○○조를 위반한 자는 N년 이하의 징역 또는 N만원 이하의 벌금에 처한다	CRIMINAL_PENALTY	· referenceTarget: 위반 조문 · Sanction.text: 형량 ReferenceLink(CITATION)	제7조를 위반한 자는 5년 이하의 징역 또는 5천만원 이하의 벌금에 처한다 (제107조)
N만원 이하의 과태료를 부과한다	FINE	· Sanction.text: 과태료액 대상 조문 연결	500만원 이하의 과태료를 부과한다
○○을(를) 취소할 수 있다 ○○을(를) 정지할 수 있다 폐쇄를 명할 수 있다	ADMINISTRATIVE_ PENALTY	· AuthorityAction 생성 · Sanction.text: 처분 내용	허가를 취소하거나 영업정지를 명할 수 있다
시정을 명할 수 있다 시정명령을 할 수 있다	ADMINISTRATIVE_ ORDER	· AuthorityAction 생성 · Sanction.text: 명령 내용	시정을 명할 수 있다
무효로 한다. 효력이 없다	NULLITY	· LegalEffect(NULLITY) 생성	이 기준에 미치지 못하는 근로조건은 무효로 한다 (제15조)

형량 추출 정규 표현식은 다음과 같이 구성하였다.

(₩d+)년₩s*이하의₩s*징역|([₩d,]+)만?원₩s*이하의₩s*(벌금|과태료)

5) 요건(Condition) 추출 패턴

규범명제의 적용 조건을 유형별로 식별하였다.

요건 패턴	ConditionType	추출 규칙	예시
N명 이상 / N명 미만 N인 이상 / N세 이상 N시간 이상 / N일 이상	QUANTITATIVE	· operator: >=, <=, >, < · value: 수치 · unit: 명, 인, 세, 시간	상시 5명 이상의 근로자를 사용하는 (제11조)
N일 이내 / N개월 이내 N년 이내 / ○○까지 ○○ 이전에 / ○○ 이후	TEMPORAL	· operator: <=, >= · value: 수치 · unit: 일, 월, 년	30일 이내에 통보하여야 한다
서면으로 / 신청을 거쳐 신고한 후 / ○○의 승인을 받아 ○○에게 통보하여	PROCEDURAL	· procedure: 절차명 · actor: 행위자	서면으로 명시하여야 한다 (제17조)
정당한 사유가 있는 경우 불가피한 사유로 ○○의 사유가 발생한 때	CAUSAL	· cause: 사유 텍스트	정당한 사유 없이 해고하지 못한다 (제23조)
필요하다고 인정하는 경우 상당하다고 인정되는 때 ○○이 적절하다고 판단하는	SUBJECTIVE	· judge: 판단 주체 · criterion: 판단 기준	고용노동부장관이 필요하다고 인정하는 경우

6) 수정 관계(ModificationLink) 추출 패턴

원칙 규정에 대한 예외・특례・준용・적용배제 관계를 식별한다.

수정 패턴	ModificationType	의미	추출 규칙
다만, ...인 경우에는 ...하지 아니한다 ...그러하지 아니하다	EXCEPTION	원칙 규정의 적용 배제	'다만' 키워드 탐지 앞 문장→ 원칙 NormStatement 뒷 문장→ 예외 NormStatement ModificationLink로 연결
제○○조에도 불구하고 제○○조에 관계없이 제○○조의 규정에도 불구하고	SPECIAL_PROVISION	일반 규정에 대한 특별 규정	'불구하고' 키워드 탐지 참조 조문 → modifiesTarget 현재 조문→ 특례 NormStatement
제○○조를 준용한다 제○○조의 규정을 준용	MUTATIS_MUTANDIS	다른 규정을 수정 적용	'준용' 키워드 탐지 참조 조문 → modifiesTarget ReferenceLink(MUTATIS_MUTANDIS) 동시 생성

수정 패턴	ModificationType	의미	추출 규칙
○○에 대하여는 적용하지 아니한다 적용을 배제한다 적용에서 제외한다	EXCLUSION	특정 대상에 대한 전면 배제	'적용하지 아니한다' 패턴 대상 범위 추출 Condition 연동

7) 헌법적 근거(ConstitutionalBasis) 추출 패턴

법률의 제1조(목적) 또는 전문에서 헌법과의 연결을 식별한다.

패턴	생성 인스턴스	추출 규칙	추출 규칙
이 법은 헌법에 따라 헌법 제○○조에 따라 헌법의 이념에 기초하여 헌법 정신에 따라	ConstitutionalBasis	basisSource: 헌법 조문 (명시된 경우) 또는 관련 헌법 조항 수동 매핑 - basisTarget: 현재 법령 - constitutionalArticle: 조항 텍스트	이 법은 헌법에 따라 근로조건의 기준을 정함으로써... (근로기준법 제1조)

대부분의 법률은 제1조에서 '헌법에 따라'만 명시하고 구체적 헌법 조항 번호를 적시하지 않는다. 따라서 헌법 조문과의 정확한 매핑은 법학적 판단이 수반되며, 자동 추출 후 전문가 검수가 필수적인 것으로 확인되었다.

8) 법령 간 관계(InterLawLink) 추출 패턴

패턴	relationType	의미	예시
「○○법」에 따른 「○○법」에서 정하는 「○○법」 제○○조	RELATED	내용적 관련 법령 연결	「산업안전보건법」에 따른 (제76조)
이 법에 따른 ... 시행령 이 법 시행에 필요한 사항	IMPLEMENTING	시행을 위한 하위법	이 법 시행에 필요한 사항은 대통령령으로 정한다
「○○법」에 우선하여 적용 「○○법」에 대한 특별법	SUPERIOR	상위/특별법 관계	이 법은 ... 에 우선하여 적용한다

법령명 추출 정규 표현식은 다음과 같다.

「([^」]+)」

꺾쇠괄호(「」) 안의 법령명을 추출하여 LegalInstrument 인스턴스와 매칭한다.

9) 패턴 종합 및 자동화 수준

추출 대상	패턴 수	자동화 수준	비고
양상성 (DeonticModality)	5 유형 12+ 패턴	완전 자동	문말 패턴 매칭, 정확도 높음
위임 (DelegationLink)	5 유형 8+ 패턴	완전 자동	'~으로 정한다' 패턴, 명확한 표현
참조·인용 (ReferenceLink)	6 유형 10+ 패턴	완전 자동	조문 번호 정규표현식 매칭
벌칙·제재 (Sanction)	5 유형 8+ 패턴	완전 자동	'위반한 자' 패턴 + 형량 추출
요건 (Condition)	5 유형 15+ 패턴	반자동	QUANTITATIVE/TEMPORAL은 자동 SUBJECTIVE/CAUSAL은 검수 필요
수정 관계 (ModificationLink)	4 유형 10+ 패턴	반자동	키워드 탐지 자동, 대상 식별은 검수
헌법적 근거 (ConstitutionalBasis)	4 패턴	수동+규칙	'헌법에 따라' 탐지 자동 구체 조항 매핑은 수동
법령 간 관계 (InterLawLink)	3 유형 6+ 패턴	반자동	법령명(「」) 추출 자동 관계유형 판별은 검수

총 8개 추출 대상, 70+ 패턴을 정의하였다. 이 중 양상성・위임・참조・벌칙의 4개 영역은 법령 텍스트의 정형성이 높아 규칙 기반으로 완전 자동화가 가능하며, 요건・수정관계・법령간관계의 3개 영역은 키워드 탐지는 자동으로 할 수 있으나 의미적 판별에 전문가 검수가 필요한 반자동 방식이다. 헌법적 근거는 패턴 탐지 후 수동 매핑이 필요하다.

마. 법령 조문 패턴에 따른 파싱 의사 코드

노동법(근로기준법)을 기준으로 앞 절에서 연구한 패턴에 따른 파싱 코드를를 구성하였고, 아래 표는 근로기준법 조문을 패턴에 따라 파싱 후 값을 추출한 결과이다.

1) 근로기준법 내 조문별 패턴 예시

조문	양상성	위임	참조	제재	요건	수정	헌법	관련법
제1조	-	-	-	-	-	-	헌법에 따라	-

조문	양상성	위임	참조	제재	요건	수정	헌법	관련법
제3조	금지	-	-	-	-	-	-	-
제6조	금지	-	-	-	-	-	-	-
제7조	금지	-	-	-	-	-	-	-
제10조	금지+허용	-	-	-	-	-	-	-
제11조	단서+허용+적용	대통령령	-	-	>5명	허용	-	-
제107조	-	-	제7조	5년징역 5천만원벌금	-	-	-	-
제114조	-	-	제6조	500만원벌금	-	-	-	-
시행령 제7조	-	-	제11조	-	<4명	-	-	-
시행령 제76조	-	-	-	-	-	-		산업안전보건법

2) 패턴별 핵심 파싱 로직 (의사코드)

순번	구분	의사 코드

1	문서구조 파서	`def parse_legal_instrument(xml_root):`
		`  meta = xml_root.find('기본정보')`
		`  inst = create_instance('LegalInstrument', {`
		`  'identifier': f'LSA_KR_{meta.find("법령ID").text}',`
		`  'title': meta.find('법령명').text,`
		`  'instrumentType': map_type(meta.find('법령구분').text),`
		`  'promulgatedOn': parse_date(meta.find('공포일자').text),`
		`  'effectiveFrom': parse_date(meta.find('시행일자').text),`
		`  'hasNormativeLevel': map_level(instrument_type),`
		`  })`
		
		`  for article_xml in xml_root.findall('.//조문단위'):`
		`  art = create_instance('Article', {`
		`  'number': article_xml.find('조문번호').text,`
		`  'title': article_xml.find('조문제목').text or '',`
		`  'text': article_xml.find('조문내용').text,`
		`  'partOfInstrument': inst,`
		`  'hasParentUnit': find_parent_chapter(article_xml),`
		`  })`
		`  parse_paragraphs(art, article_xml)  # 항→호→목 재귀`

2	양상성 추출	`def extract(self, text: str) -> List[ModalityResult]:`
		`  results = []`
		`  for code, patterns in self.PATTERNS.items():`
		`  for pattern, confidence in patterns:`
		`  m = re.search(pattern, text)`
		`  if m:`
		`  results.append(ModalityResult(`
		`  code=code, matched_pattern=pattern,`
		`  matched_text=m.group(0), confidence=confidence,`
		`  ))`
		`  break`
		`  results.sort(key=lambda r: self.PRIORITY.get(r.code, 0), reverse=True)`
		`  return results`

순번	구분	의사 코드

		`def extract_primary(self, text: str) -> Optional[ModalityResult]:`
		`results = self.extract(text)`
		`return results[0] if results else None`
		
		`def split_proviso(self, text: str) -> Tuple[str, Optional[str]]:`
		`"""본문과 단서('다만, ...')를 분리한다."""`
		`m = re.search(r'다만[,\s]', text)`
		`if m:`
		`return text[:m.start()].strip(), text[m.start():].strip()`
		`return text, None`

3	위임 추출	`def extract(self, text: str) -> List[DelegationResult]:`
		`results = []`
		`for del_type, patterns in self.PATTERNS.items():`
		`for pattern in patterns:`
		`m = re.search(pattern, text)`
		`if m:`
		`results.append(DelegationResult(`
		`delegation_type=del_type, matched_pattern=pattern,`
		`matched_text=m.group(0),`
		`scope_text=get_context(text, m.start(), 80),`
		`))`
		`break`

4	참조 인용	`def extract(self, text: str) -> List[ReferenceResult]:`
		`results = []`
		`seen = set()`
		`for pattern, ref_type in self.PATTERNS:`
		`for m in re.finditer(pattern, text):`
		`for ref in parse_article_ref(m.group(0)):`
		`key = (ref.article_no, ref.paragraph_no, ref_type)`
		`if key not in seen:`
		`seen.add(key)`
		`results.append(ReferenceResult(`
		`reference_type=ref_type, target_ref=ref,`
		`matched_pattern=pattern, anchor_text=m.group(0),`
		`))`

5	벌칙.제재 추출	`def extract(self, text: str) -> List[SanctionResult]:`
		`results = []`
		`violated = self._extract_violated(text)`
		`imprisonment = self._extract_imprisonment(text)`
		`fine_info = self._extract_fine(text)`
		
		`# 형사벌칙`
		`if violated and (imprisonment is not None or fine_info):`
		`s_type = SanctionType.FINE if (fine_info and fine_info[1] == "과태료" and imprisonment is None) else SanctionType.CRIMINAL_PENALTY`
		`parts = []`
		`if imprisonment is not None:`
		`parts.append(f"{imprisonment}년 이하의 징역")`
		`if fine_info:`
		`parts.append(f"{fine_info[0]}원 이하의 {fine_info[1]}")`
		`results.append(SanctionResult(`
		`sanction_type=s_type, violated_articles=violated,`
		`penalty_text=" 또는 ".join(parts),`
		`imprisonment_years=imprisonment,`
		`fine_amount=fine_info[0] if fine_info else None,`
		`))`
		`# 행정처분`
		`for pat, s_type in self.ADMIN_PATTERNS:`
		`m = re.search(pat, text)`
		`if m:`

순번	구분	의사 코드

		`results.append(SanctionResult(`
		`sanction_type=s_type, violated_articles=violated,`
		`penalty_text=m.group(0),`
		`))`
		
		`# 무효`
		`for pat in self.NULLITY_PATTERNS:`
		`m = re.search(pat, text)`
		`if m:`
		`results.append(SanctionResult(`
		`sanction_type=SanctionType.NULLITY, violated_articles=[],`
		`penalty_text=m.group(0),`
		`))`
		`return results`
		
		`def _extract_violated(self, text: str) -> List[ArticleRef]:`
		`return [ArticleRef(`
		`article_no=int(m.group(1)),`
		`article_sub=int(m.group(2)) if m.group(2) else None,`
		`paragraph_no=int(m.group(3)) if m.group(3) else None,`
		`item_no=int(m.group(4)) if m.group(4) else None,`
		`raw_text=m.group(0),`
		`) for m in self.VIOLATION_PATTERN.finditer(text)]`
		
		`def _extract_imprisonment(self, text: str) -> Optional[int]:`
		`m = self.IMPRISONMENT_PATTERN.search(text)`
		`return int(m.group(1)) if m else None`
		
		`def _extract_fine(self, text: str) -> Optional[Tuple[str, str]]:`
		`m = self.FINE_PATTERN.search(text)`
		`if m:`
		`amount = parse_korean_amount(m.group(1))`
		`return (amount, m.group(2))`
		`return None`

6	요건 추출	`def extract(self, text: str) -> List[ConditionResult]:`
		`results = []`
		`for cond_type, patterns in self.PATTERNS.items():`
		`for pat_tuple in patterns:`
		`pattern, unit = pat_tuple[0], pat_tuple[1]`
		`for m in re.finditer(pattern, text):`
		`if cond_type in (ConditionType.QUANTITATIVE,`
		`ConditionType.TEMPORAL):`
		`val = m.group(1) if m.lastindex >= 1 else None`
		`cw = m.group(2) if m.lastindex >= 2 else None`
		`results.append(ConditionResult(`
		`condition_type=cond_type, matched_text=m.group(0),`
		`operator=self.OP_MAP.get(cw), value=val, unit=unit,`
		`))`
		`else:`
		`results.append(ConditionResult(`
		`condition_type=cond_type, matched_text=m.group(0),`
		`))`
		`break`
		`return results`

7	수정 관계	`def extract(self, text: str) -> List[ModificationResult]:`
		`results = []`
		`seen_types = set()`
		`for pattern, mod_type in self.PATTERNS:`
		`if mod_type in seen_types:`
		`continue`

순번	구분	의사 코드

		`m = re.search(pattern, text)`
		`if m:`
		`seen_types.add(mod_type)`
		`target_refs = parse_article_ref(m.group(0))`
		`results.append(ModificationResult(`
		`modification_type=mod_type, matched_pattern=pattern,`
		`matched_text=m.group(0),`
		`target_ref=target_refs[0] if target_refs else None,`
		`))`
		`return results`

8	헌법적 근거	`def extract(self, text: str) -> List[ConstitutionalBasisResult]:`
		`results = []`
		`for pattern in self.PATTERNS:`
		`m = re.search(pattern, text)`
		`if m:`
		`const_ref = None`
		`try:`
		`if m.group(1):`
		`const_ref = ArticleRef(`
		`article_no=int(m.group(1)),`
		`paragraph_no=int(m.group(2)) if m.lastindex >= 2 and m.group(2) else None,`
		`raw_text=m.group(0),`
		`)`
		`except (IndexError, TypeError):`
		`pass`
		`results.append(ConstitutionalBasisResult(`
		`matched_pattern=pattern, matched_text=m.group(0),`
		`constitutional_article=const_ref,`
		`))`
		`break  # 첫 매칭만`

9	법령 간 관계	`def extract(self, text: str) -> List[InterLawResult]:`
		`results = []`
		`for law_name in self.LAW_NAME_PATTERN.findall(text):`
		`rel_type = InterLawRelationType.RELATED`
		`matched_pat = ""`
		`for pattern, r_type in self.RELATION_PATTERNS:`
		`if re.search(pattern, text):`
		`rel_type = r_type`
		`matched_pat = pattern`
		`break`
		`idx = text.find(f'「{law_name}」')`
		`ctx = text[idx:idx+100] if idx >= 0 else ""`
		`art_refs = parse_article_ref(ctx)`
		`results.append(InterLawResult(`
		`relation_type=rel_type, target_law_name=law_name,`
		`matched_pattern=matched_pat, matched_text=f'「{law_name}」',`
		`target_article=art_refs[0] if art_refs else None,`
		`))`
		`return results`

바. 근로기준법 전제 조문의 지식 구조 관리

법령 온톨로지는 8대 패턴을 관리하여 지속적으로 근로기준법의 추가·삭제·변경이 발생하더라도 법령 지식을 구성할 수 있도록 설계하였으며, (표 3-5)는 근로기준법의 패턴별 통계 결과이다.

(표3-5) 근로기준법 조문의 지식구조 통계

순번	패턴	해당 조문수	비율	비고
1	양상성	101개	87.8%	의무 51개, 금지 30개, 허용 18개, 권한 7개, 면제 3개
2	요건	43개	37.4%	
3	위임	21개	18.3%	
4	관련법	16개	13.9%	
5	참조	14개	12.2%	
6	제재	11개	9.6%	
7	수정	7개	6.1%	
8	헌법	1개	0.9%	

1) (노동)근로기준법 조문별 지식 구조 예시

다음은 법령온톨로지의 원활한 관리를 위해 법령 조별로 구성되는 지식 구조 코드의 일부이다.

순번	조문 지식 관리 구조	설명
1	# ========== # 제2장 근로계약 (제12조~제22조) # ========== ArticleKnowledge("Art12", "제12조", "삭제", "제2장 근로계약", 2, note="삭제 조문"), ArticleKnowledge("Art13", "제13조", "적용 제외 승인 등", "제2장 근로계약", 2, modality="권한", delegation="대통령령", inter_law="「직업안정법」"), ArticleKnowledge("Art14", "제14조", "법령 요지 등의 게시", "제2장 근로계약", 2, modality="의무", note="사업장 게시 의무"), ArticleKnowledge("Art15", "제15조", "이 법을 위반한 근로계약", "제2장 근로계약", 2, modality="금지", modification="배제", sanction="무효", note="기준 미달 조건 무효"), ArticleKnowledge("Art16", "제16조", "계약기간", "제2장 근로계약", 2, modality="금지", has_proviso=True, proviso_modality="면제", inter_law="「기간제법」"),	제12조 삭제 조문 (2007년 전부개정 시 삭제) 제13조고용노동부장관이 특정 사업장에 대해 이 법의 적용 제외를 승인할 수 있는 권한 규정. 세부 기준은 대통령령 위임, 「직업안정법」 연계 제14조 – 사용자는 이 법과 취업규칙의 요지를 사업장에 게시·비치할 의무 제15조 – 이 법 기준에 미달하는 근로계약 부분은 무효, 무효 부분은 이 법 기준으로 자동 대체 제16조 – 근로계약 기간은 원칙적으로 1년 초과 금지. 단, 「기간제법」에 따른 경우는 예외 허용
2	# ========== # 제12장 벌칙 (제107조~제116조) # ========== ArticleKnowledge("Art107", "제107조", "벌칙", "제12장 벌칙", 12, reference="제7조", sanction="5년↓징역/5천만원↓벌금"), ArticleKnowledge("Art108", "제108조", "벌칙", "제12	제107조 – 제7조(강제 근로 금지)를 위반한 자에 대한 벌칙. 근로기준법 중 가장 무거운 형벌로, 5년 이하의 징역 또는 5천만원 이하의 벌금 제108조 – 제23조②에 따른 부당해고 구제명령을 이행하지 않은 자에 대한 벌칙. 5년 이하의 징역 또는 3천만원 이하의 벌금

순번	조문 지식 관리 구조	설명
	장 벌칙", 12, reference="제23조②", sanction="5년↓징역/3천만원↓벌금", note="부당해고 구제명령 불이행"), ArticleKnowledge("Art109", "제109조", "벌칙", "제12장 벌칙", 12, reference="제36·43·43의2·44·44의2·44의3·45·53⑦", sanction="3년↓징역/3천만원↓벌금", note="임금체불, 금품미청산 등"), ~~~~~~~~~~~	제109조 – 임금 관련 핵심 위반에 대한 벌칙. 금품 미청산(제36조), 임금 미지급(제43조), 도급 임금 연대책임(제44조·44조의2·44조의3), 비상시 지급(제45조), 연장근로 한도 위반(제53조⑦) 등을 위반한 자는 3년 이하의 징역 또는 3천만원 이하의 벌금

1) (노동)근로기준법 조문별 지식 구조

다음은 본 연구에서 근로기준법 전체(116조)에 대해 지식 구조를 구성한 테이블의 일부로서, 각 조를 10가지 패턴에 해당되는지 여부를 분석하여 해당되는 내용을 구성한 패턴 지식 구조로서, 법령 온톨로지에 해당 패턴들을 이용하여 데이터화하였다.

조문	제목	장	양상성	위임	참조	제재	요건	수정	헌법	관련법	단서여부	단서 양상성	비고
제1조	목적	제1장 총칙	-	-	-	-	-	-	헌법에 따라	-	N	-	목적 조항, 헌법 제32조③
제2조	정의	제1장 총칙	-	-	-	-	-	-	-	-	N	-	9개 법률용어 정의 (근로자, 사용자, 근로, 임금 등)
제3조	근로조건의 기준	제1장 총칙	금지	-	-	-	-	-	-	-	N	-	최저기준 선언, '낮출 수 없
제4조	근로조건의 결정	제1장 총칙	의무	-	-	-	-	-	-	-	N	-	근로자와 사용자 대등결정
제5조	근로조건의 준수	제1장 총칙	의무	-	-	-	-	-	-	-	N	-	성실 이행 의무
제6조	균등한 처우	제1장 총칙	금지	-	-	-	주관적(성별·국적·신앙·신분)	-	-	-	N	-	차별 금지
제7조	강제 근로의 금지	제1장 총칙	금지	-	-	-	사유적(폭행·협박·감금)	-	-	-	N	-	
제8조	폭행의 금지	제1장 총칙	금지	-	-	-	-	-	-	-	N	-	
제9조	중간착취의 배제	제1장 총칙	금지	-	-	-	-	-	-	-	Y	면제	법률에 따라 허용된 자 제외
제10조	공민권 행사의 보장	제1장 총칙	금지+허용	-	-	-	-	-	-	-	Y	허용	본문: 거부 금지, 단서: 시간 변경 가능
제11조	적용 범위	제1장 총칙	단서+허용+적용	대통령령	-	-	정량적(≥5명)	예외+배제	-	-	Y	면제	본문: 5인↑ 적용, 단서: 동거친족 배제, 후문: 4인↓ 대
제12조	삭제	제2장 근로계약	-	-	-	-	-	-	-	-	N	-	삭제 조문
제13조	적용 제외 승인 등	제2장 근로계약	권한	대통령령	-	-	-	-	-	「직업안정법」	N	-	
제14조	법령 요지 등의 게시	제2장 근로계약	의무	-	-	-	-	-	-	-	N	-	사업장 게시 의무
제15조	이 법을 위반한 근로계약	제2장 근로계약	금지	-	-	무효	-	배제	-	-	N	-	기준 미달 조건 무효
제16조	계약기간	제2장 근로계약	금지	-	-	-	-	-	-	「기간제법」	Y	면제	
제17조	근로조건의 명시	제2장 근로계약	의무	대통령령	-	-	절차적(서면명시)	-	-	-	N	-	
~~													
제107조	벌칙	제12장 벌칙	-	-	제7조	5년↓징역/5천만원↓벌금	-	-	-	-	N	-	
제108조	벌칙	제12장 벌칙	-	-	제23조②	5년↓징역/3천만원↓벌금	-	-	-	-	N	-	부당해고 구제명령 불이행
제109조	벌칙	제12장 벌칙	-	-	제36·43·43의2·44·44의2·44의3·45·53⑦	3년↓징역/3천만원↓벌금	-	-	-	-	N	-	임금체불, 금품미청산 등

2) (노동)근로기준법 시행령 조문별 지식 구조

다음은 본 연구에서 근로기준법 시행령 전체 조문(제1조~제60조)를 분석하

여 구성한 패턴 매핑표이며, 시행령은 근로기준법의 위임을 받아 세부사항을 규정하므로, 모법참조(reference)가 핵심 패턴이 된다.

특히 시행령은 모법의 위임을 받아 구체적 수치(기간.금액.인원수)와 절차(서면.신고.신청)를 규정하는 역할이므로, 요건 커버리지가 높고 참조율이 80%에 달한다.

(표3-6) 근로기준법 시행령 조문의 지식구조 통계

순번	패턴	시행령	모법	해석
1	양상성	의무 33개 (71%)	의무 51개 금지 30개	시행령은 의무 편중 (세부 이행 규정)
2	참조	60개 (80.0%)	12.2%	시행령은 거의 모든 조문이 모법 조문을 인용
3	요건	42개 (56.0%)	37.4%	시행령이 구체적 수치.기간.절차 규정
4	제재	0개	9.6%	벌칙은 모법에만 존재
5	헌법	0개	0.9%	헌법 근거는 모법에만 존재
6	수정 관계	15개 (20.0%)	6.1%	시행령에 예외.단서 규정 다수
7	관련법	18개 (24.0%)	13.9%	시행령이 타 법령 연계 더 많음

다음은 본 연구에서 근로기준법 시행령 전체 조문(제1조~제60조)를 분석하여 구성한 패턴 매핑표이며, 시행령은 근로기준법의 위임을 받아 세부사항을 규정하였다.

조문	제목	양상성	위임	참조(법 조문)	제재	요건	수정관계	헌법	관련법	비고
제1조	목적	-	-	법 전체	-	-	-	-	「근로기준법」	시행령 목적 조항
제2조	평균임금의 계산에서 제외되는 기간과 임금	의무	고용노동부장관	법§2①⑥	-	기간적(수습3개월·휴업·출산·요양 등 8가지)	예외	-	「남녀고용평등법」·「노조법」·「병역법」	단서: 고용노동부장관이 정하는 것
제3조	일용근로자의 평균임금	권한	고용노동부장관	-	-	-	-	-	-	고용노동부장관이 직업별 정함
제4조	특별한 경우의 평균임금	권한	고용노동부장관	법§2①⑥·영§2·§3	-	-	-	-	-	산정 불가시 장관이 정함
제5조	평균임금의 조정	의무	-	법§79·§80·§82~§84	-	정량적(5%↑ 변동시)	단서	-	「근로자퇴직급여보장법」	보상금 산정 평균임금 조정
제6조	통상임금	-	-	-	-	-	-	-	-	정의 규정: 7가지 산정방법
제7조	적용범위	-	-	법§11②	-	정량적(4명↓)	-	-	-	별표 1 참조
제7조의2	상시 사용하는 근로자 수의 산정 방법	의무	-	법§11③·§93	-	정량적(5명/10명 기준)+기간적(1개월)	예외(불구하고)	-	「파견근로자보호법」·「기간제법」	4개항 상세 산정 규칙
제8조	명시하여야 할 근로조건	-	-	법§17①⑤·§93	-	-	-	-	-	3가지 근로조건 명시 항목
제8조의2	근로자의 요구에 따른 서면 교부	-	-	법§17②·§51·§51의2·§52·§53·§55·§57·§58·§59·§62	-	-	-	-	「노조법」	서면 합의·취업규칙·단체협약·법령 변경
제9조	단시간근로자의 근로조건 기준 등	-	-	법§18②	-	-	-	-	-	별표 2 참조

2. (노동)법령 온톨로지 구축

가. 법령 온톨로지 전체 개요

법령 온톨로지는 Turtle(.ttl)/N3(.n3) 형식으로 작성하였으며, TopBraid[46]에서 직접 구조를 구성하고 검증하였다.

1) 구현 범위

구성요소	수량	설명
클래스 (owl:Class)	48개	추상 8 + 문서구조 24 + 규범명제 13 + 법령위계 3
Datatype Property	39개	identifier, title, text, instrumentType 등
Object Property	61개 이상	hasStructuralUnit, expressedInUnit, hasNormativeLevel 등
OWL Restriction	7개	owl:cardinality, owl:minCardinality 제약
NormativeLevel 고정 인스턴스	7개	헌법(1)~규칙(6) + 행정규칙, superiorLevel 체인
rdfs:label (다국어)	@en + @ko	모든 클래스에 영문·한글 라벨 병기

나) TopBraid로 구현된 법령 온톨로지 구성

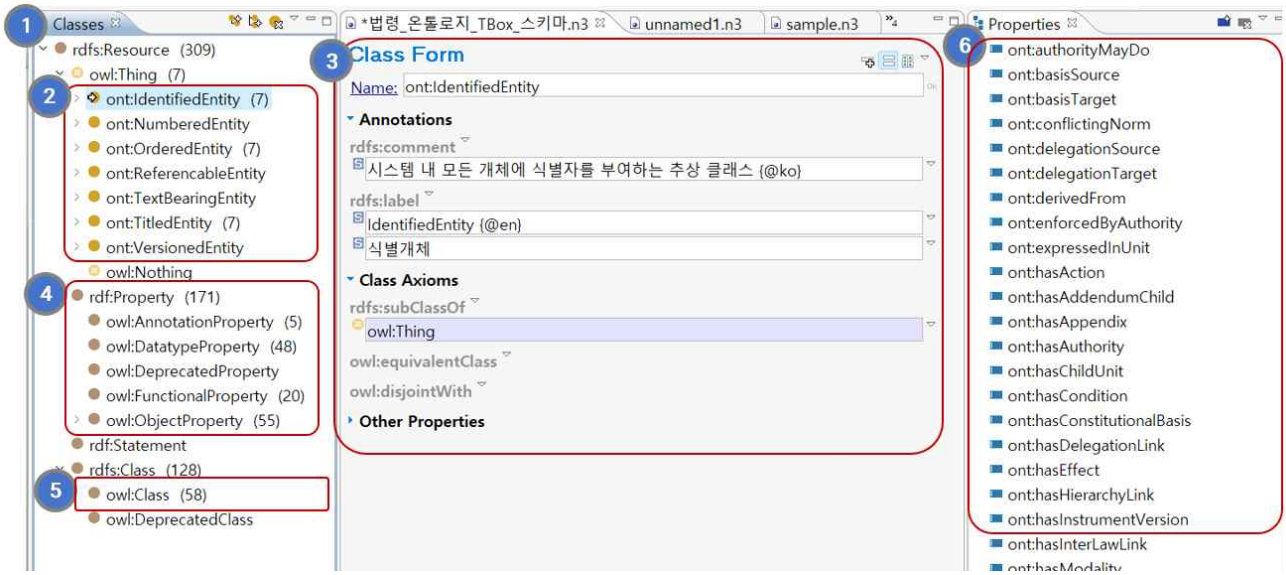
법령 온톨로지는 Turtle(.ttl)/N3(.n3) 형식으로 작성하였으며, TopBraid[47]를 이용하여 직접 온톨로지 스키마 구조를 구성하였다. (그림*)은 TopBraid로 출력한 온톨로지에 대한 설명이다.

탭	설명
Classes 탭	온톨로지의 클래스 계층 구조를 트리 형태로 보여주는 패널. rdfs:Resource → owl:Thing 아래 전체 클래스가 계층적으로 표시됨 (현재 309개 리소스, 128개 rdfs:Class)
ont:IdentifiedEntity 서브클래스들	owl:Thing의 직접 하위 클래스로 정의된 7개의 상위 클래스 그룹. NumberedEntity, OrderedEntity, ReferencableEntity, TextBearingEntity, TitledEntity, VersionedEntity 등이 있으며, 각 클래스 옆 숫자(7)는 하위 클래스 수를 나타냄

46) TopBraid: 미국의 TopQuadrant(탑쿼드란트) 사가 개발한 시맨틱 웹(Semantic Web) 및 지식 그래프(Knowledge Graph) 관련 소프트웨어 제품으로 W3C 표준(RDF, OWL, SPARQL, SHACL)을 기반으로 데이터를 모델링, 관리, 연결 및 거버넌스하기 위한 도구들을 제공

47) TopBraid: 미국의 TopQuadrant(탑쿼드란트) 사가 개발한 시맨틱 웹(Semantic Web) 및 지식 그래프(Knowledge Graph) 관련 소프트웨어 제품으로 W3C 표준(RDF, OWL, SPARQL, SHACL)을 기반으로 데이터를 모델링, 관리, 연결 및 거버넌스하기 위한 도구들을 제공

탭	설명
Class Form	선택된 클래스(ont:IdentifiedEntity)의 상세 정보 편집 화면. rdfs:comment(한국어 설명), rdfs:label(영문/한국어 라벨), rdfs:subClassOf(owl:Thing), owl:equivalentClass, owl:disjointWith 등의 어노테이션과 클래스 공리를 설정할 수 있음
rdf:Property (171)	온톨로지에 정의된 전체 프로퍼티 171개의 계층. AnnotationProperty 5개, DatatypeProperty 48개, FunctionalProperty 20개, ObjectProperty 55개로 구성됨
owl:Class (58)	rdfs:Class 128개 중 OWL 클래스로 정의된 것이 58개임을 보여줌. TBox의 핵심 클래스들이 여기에 해당
Properties 패널	온톨로지에 정의된 ObjectProperty 목록. ont:authorityMayDo, ont:basisSource, ont:conflictingNorm, ont:delegationSource/Target, ont:hasAction, ont:hasCondition, ont:hasConstitutionalBasis, ont:hasDelegationLink, ont:hasModality, ont:hasNormStatement 등 법령 온톨로지 특화 프로퍼티들로 구성



(그림3-11) TopBraid를 이용한 법령 온톨로지 실행 화면

나. 법령 온톨로지 TBox 구성

1) TBox 설계 원칙

(1) 다중 상속 구조

· LegalInstrument는 IdentifiedEntity, TitledEntity, VersionedEntity, Referencable-Entity를 다중 상속하여, 동일 속성(title, identifier 등)의 rdfs:domain 충돌 없이 여러 클래스에서 재사용할 수 있도록 함

(2) NormativeLevel 사전 정의

· 법령 위계 단계(헌법→법률→대통령령→총리령/부령→조례→규칙)는 변하지 않는 제도적 구조이므로, TBox에 고정 인스턴스로 미리 정의하고 superiorLevel/subordinateLevel 체인을 구성

(3) OWL Restriction 적용

· NormStatement의 hasModality(정확히 1개), expressedInUnit(최소 1개), LegalInstrument의 hasNormativeLevel(정확히 1개) 등 핵심 제약을 owl:cardinality / owl:minCardinality로 명시하여 Reasoner가 일관성을 검증할 수 있게 하함

2) TBox 코드 예시 (N3 형식)

(1) 클래스 정의 예시

```
ont:LegalInstrument a owl:Class ;
    rdfs:subClassOf ont:IdentifiedEntity ,
        ont:TitledEntity , ont:VersionedEntity ,
        ont:ReferencableEntity ;
    rdfs:label "LegalInstrument"@en ;
    rdfs:label "법령문서"@ko .
```

(2) OWL Restrict 예시

```
ont:NormStatement rdfs:subClassOf [
    a owl:Restriction ;
    owl:onProperty ont:hasModality ;
    owl:cardinality "1"^^xsd:nonNegativeInteger] .
```

(3) OWL Restrict 예시

```
ont:LV_ACT a ont:NormativeLevel ;
    ont:hierarchyRank 2 ;
    ont:levelCode "ACT" ;
    ont:issuingAuthority "NATIONAL_ASSEMBLY" ;
    ont:superiorLevel ont:LV_CONSTITUTION .
```

다. ABox 구축 (인스턴스 구현)

ABox는 TBox 스키마에 실제 법령 데이터를 인스턴스로 투입한 것이다. 본 연구에서는 근로기준법 제1장 총칙과 관련 벌칙 조항을 중심으로, 헌법·시행령을 포함한 3개 법령에 대해 인스턴스를 구성하였다.

1) 대상 법령 및 선별 기준

법령	InstrumentType	rank	선별 이유
대한민국헌법	CONSTITUTION	1	헌법적 근거(ConstitutionalBasis) 검증

법령	InstrumentType	rank	선별 이유
근로기준법	ACT	2	총칙(목적·정의·금지규범) + 벌칙 → 규범명제 검증
근로기준법 시행령	PRESIDENTIAL_DECREE	3	위임관계(DelegationLink) + 법령위계 검증

조문 선정은 3개 레이어를 모두 반영될 수 있게 다음 기준으로 선정하였다.

(1) 문서구조: Chapter→Article→Paragraph→Item 계층 구조를 포함하는 제2조(정의)를 중심으로 선정

(2) 규범명제: 금지(제3조・제6조・제7조), 허용(제10조), 의무(제11조) 등 다양한 양상성을 갖는 조문 선정

(3) 법령위계: 제1조의 헌법 인용, 제11조의 대통령령 위임, 벌칙 조문의 타 조문 참조를 포함하도록 선정

2) 인스턴스 구성 현황

온톨로지 클래스	인스턴스 수	주요 인스턴스
LegalInstrument	3	- CONST (헌법) - LSA (근로기준법) - LSA_ED (시행령)
StructuralUnit	16	- Chapter 2개 (제1장 총칙, 제12장 벌칙) - Article 12개 (헌법 2 + 근로기준법 9 + 시행령 1) - Paragraph 1개, Item 2개
NormativeLevel	3 (TBox 재사용)	- LV_CONSTITUTION (rank=1) - LV_ACT (rank=2) - LV_PRESIDENTIAL_DECREE (rank=3)
NormativeHierarchyLink	3	- HL_01: 헌법→근로기준법 - (CONSTITUTIONAL_BASIS) - HL_02: 근로기준법→시행령 (DELEGATION_BASIS) - HL_03: 근로기준법→시행령 (SUPERIOR_INFERIOR)
ConstitutionalBasis	2	- CB_01: 헌법 제32조→근로기준법 - CB_02: 헌법 제33조→근로기준법
NormStatement	5	- NS_Art3 (금지: 근로조건 저하) - NS_Art6 (금지: 차별적 처우) - NS_Art7 (금지: 강제 근로) - NS_Art10 (허용: 공민권 행사) - NS_Art11 (의무: 적용 범위)
DeonticModality	3	- PROHIBITION, OBLIGATION, PERMISSION
Condition	4	- SUBJECTIVE (제6조 차별사유) - CAUSAL (제7조 폭행·협박) - PROCEDURAL (제10조 청구)

온톨로지 클래스	인스턴스 수	주요 인스턴스
		- QUANTITATIVE (제11조 ≥5명)
Sanction	2	- SN_Art107: 5년↓징역/5천만원↓벌금 (←제7조) - SN_Art114: 500만원↓벌금 (←제6조)
ReferenceLink	3	- RL_01: 제1조→헌법 (CITATION) - RL_02: 제107조→제7조 (CITATION) - RL_03: 제114조→제6조 (CITATION)
DelegationLink	1	- DL_01: 제11조→시행령 (TO_PRESIDENTIAL_DECREE)
LegalRole	3	- 사용자, 근로자, 근로관계 당사자
LegalAction	4	- 근로조건 저하, 차별적 처우, 강제 근로, 공민권 거부
LegalObject	1	- 근로조건
LegalEffect	5	- NULLITY, SANCTION_ARISE(2), PERMISSION_ARISE, OBLIGATION_ARISE

3) ABox 코드 예시 (N3 형식)

(1) LegalInstrument + hasNormativeLevel 연결

```
inst:LSA a ont:LegalInstrument ;
    ont:identifier "LSA_KR_005309" ;
    ont:title "근로기준법"@ko ;
    ont:instrumentType "ACT" ;
    ont:effectiveFrom "1997-03-13"^^xsd:date ;
    ont:hasNormativeLevel ont:LV_ACT .
```

(2) 규범명제 (주체-행위-객체-요건-효과-양상성-제재 완전 분해

```
inst:NS_Art7 a ont:NormStatement ;
    ont:text "사용자는 폭행·협박·감금 등으로
        근로자의 자유의사에 어긋나는
        근로를 강요하지 못한다."@ko ;
    ont:expressedInUnit inst:LSA_Art7 ;
    ont:hasModality inst:Mod_PROHIBITION ;
    ont:hasSubjectRole inst:Role_Employer ;
    ont:hasAction inst:Act_ForcedLabor ;
    ont:hasCondition inst:Cond_Art7_Violence ;
    ont:hasEffect inst:Eff_SanctionArise107 ;
    ont:hasSanction inst:SN_Art107 .
```

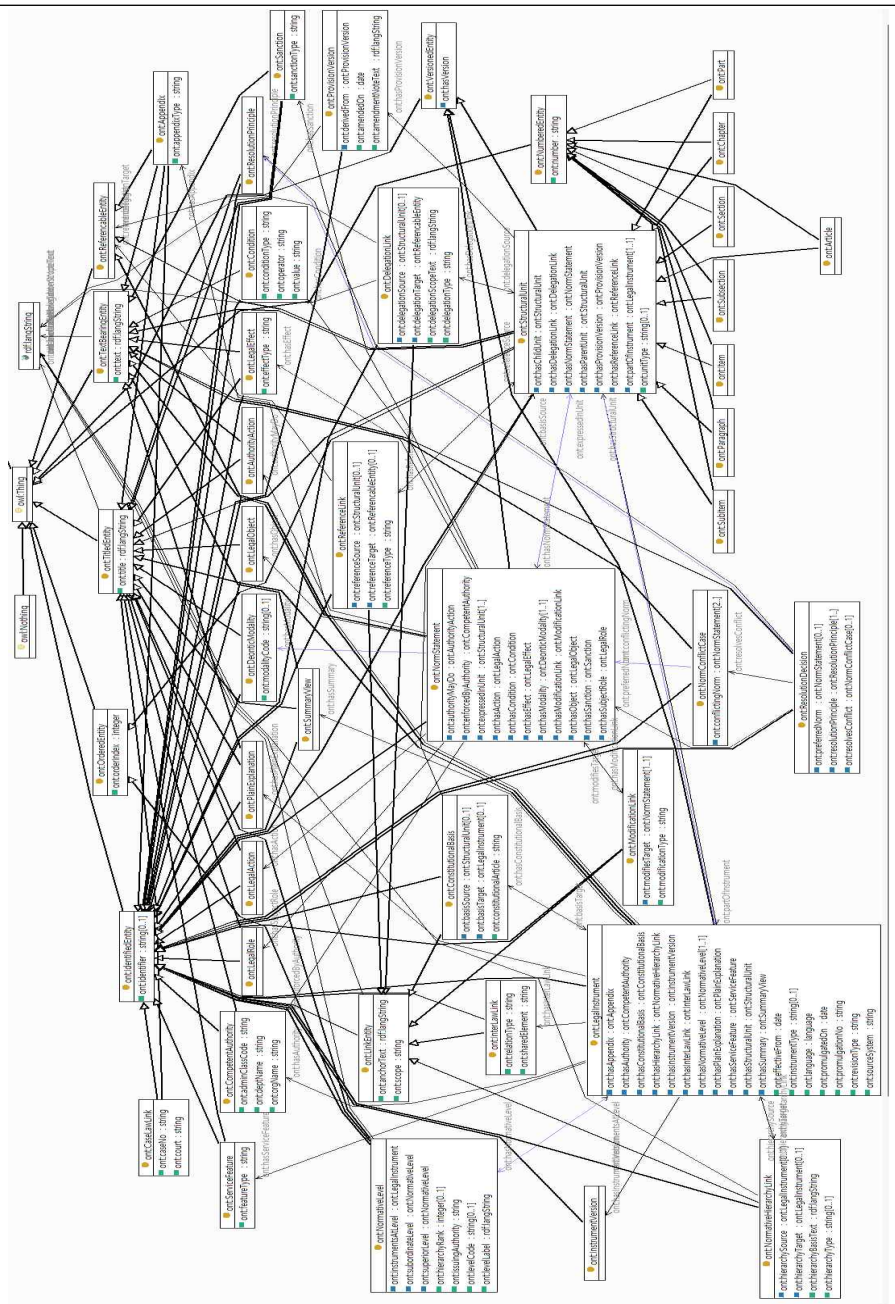
(3) 법령위계 링크 (위임근거)

```
inst:HL_02 a ont:NormativeHierarchyLink ;
    ont:hierarchySource inst:LSA ;
    ont:hierarchyTarget inst:LSA_ED ;
    ont:hierarchyType "DELEGATION_BASIS" ;
    ont:hierarchyBasisText
        "제11조제3항 '대통령령으로 정하는 바에 따라'"@ko .
```

3. (노동)법령 온톨로지

가. 온톨로지 스키마 UML

구축된 법령온톨로지에 대한 TBox인 온톨로지 스키마를 TopBraid를 이용해 출력한 전체 UML(Unified Modeling Language)은 (그림3-12)와 같다.



(그림3-12) 법령온톨로지 전체 UML

전체 구조는 "문서/조문(근거)-규범명제(의미)-관계링크(네트워크)-버전(시간)" 4축이 기준이 되며, 중요한 것은 클래스 들 간의 주요 연결에 대한 것

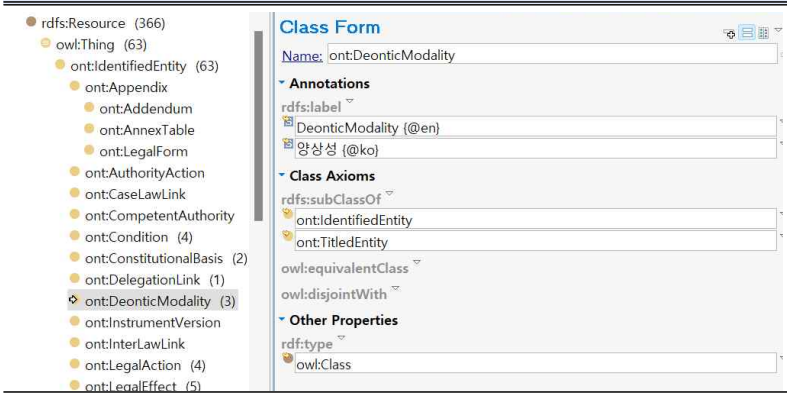
으로서, 조문(StructuralUnit)을 근거로 규범명제(NormStatement)를 생성하고, 법체계의 핵심 연결(참조・위임・연관・위계・헌법・판례)을 링크 엔티티로 객체화하며, 버전(시간)과 상충해결(우선순위)까지 포함해 법령을 "문서 + 의미 + 네트워크 + 시간" 관점에서 통합 표현하도록 설계된 것이 특징이다.

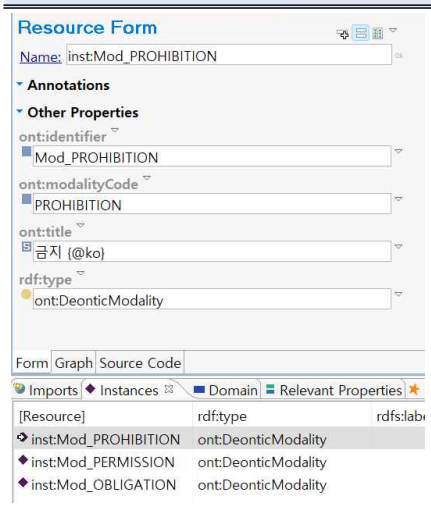
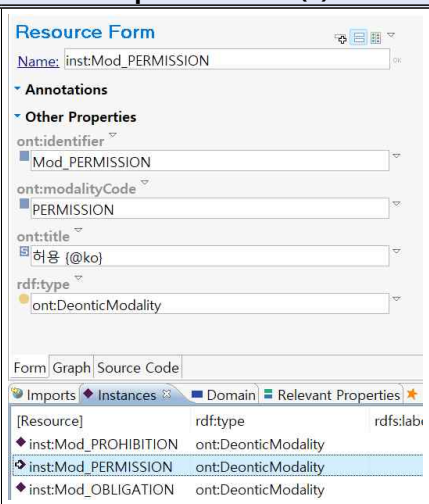
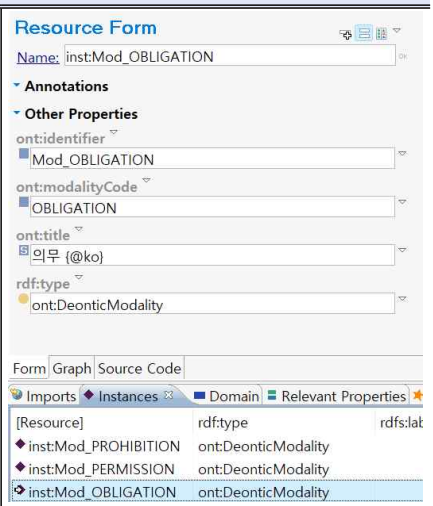
나. 법령 온톨로지 ABox

1) 법령 온톨로지 인스턴스 구성

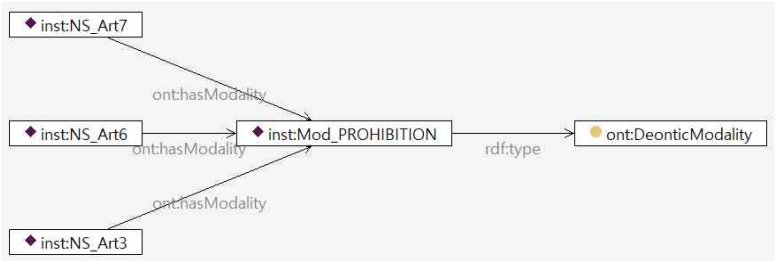
근로기준법에서 법령온톨로지 스키마에 따라 실제 데이터들을 추출하였고, 이를 클래스별 인스턴스(instance)로 구축하였다. 인스턴스는 RDB에서 레코드와 유사한 역할을 하는 것으로 온톨로지 스키마에 따라 생성된 데이터들의 구조라 할 수 있다.

다음은 실제 TopBraid 화면에서 확인할 수 있는 클래스 및 인스턴스에 대한 정보이다.

TopBraid 화면 (1)	화면 설명
	ont:DeonticModality 클래스 - rdfs:label은 영어로 DeonticModality, 한글로 '양상성' - ont:DeonticModality 클래스는 ont:IdentifiedEntity와 ont:TitledEntity의 하위 클래스 - ont:DeonticModality의 rdf:type은 owl:Class

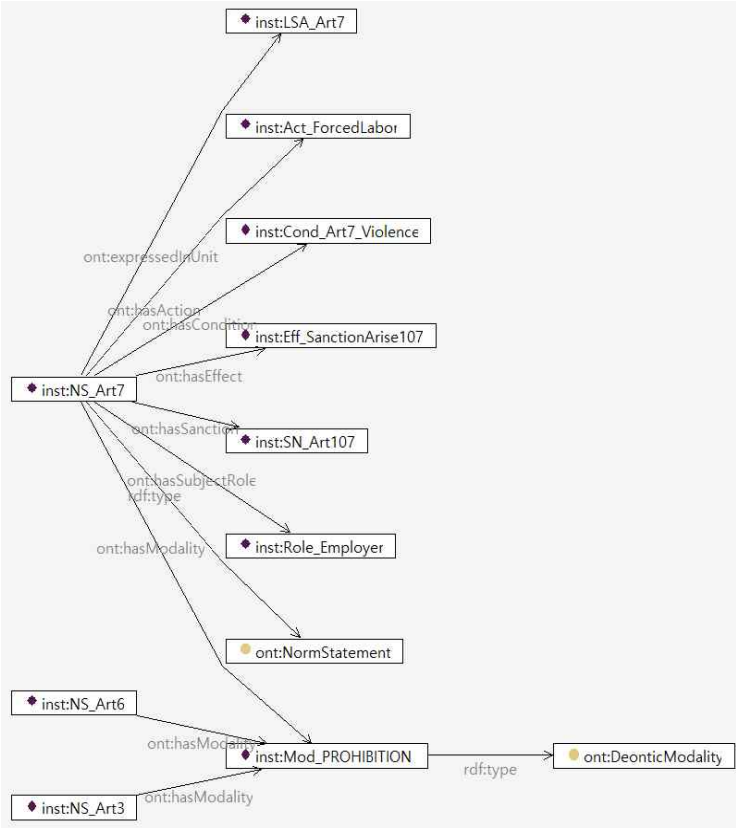
TopBraid 화면 (2)		
		
양상성 중 '금지' 인스턴스	양상성 중 '허용' 인스턴스	양상성 중 '의무' 인스턴스

TopBraid 화면 (3)



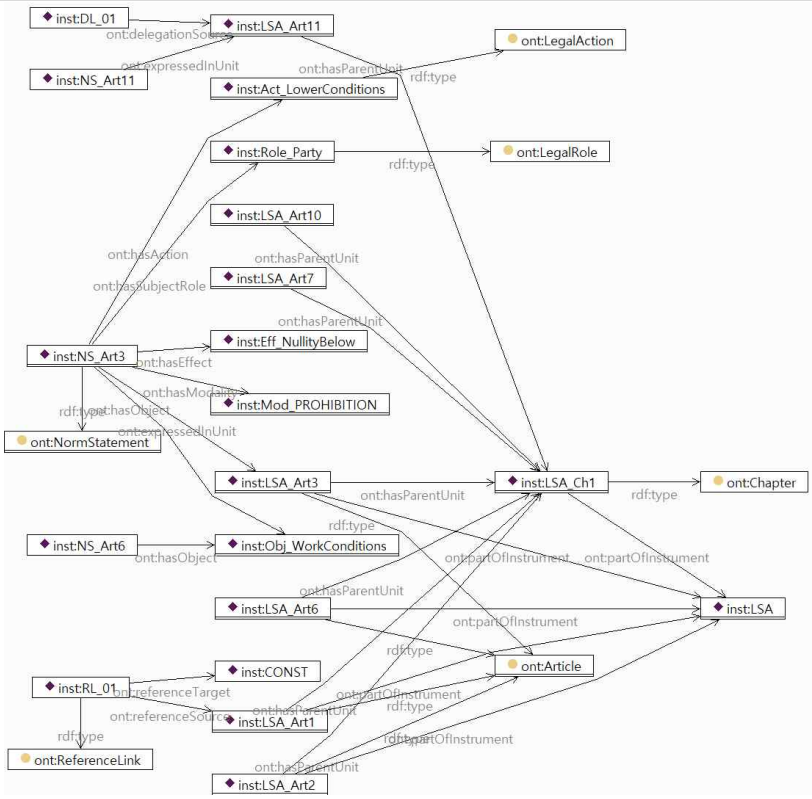
1) ont:DeonticModality type인 인스턴스는 inst:Mod_PROHIBITION
2) 조항 3개 (3조, 6조, 7조)
   - inst:NS_Art7, inst:NS_Art6, inst:NS_Art3은 inst:Mod_PROHIBITION과 ont:hasModality 관계이다.

TopBraid 화면 (4) 해석



1) 중심 노드: inst:NS_Art7
   rdf:type → ont:NormStatement
   → inst:NS_Art7는 규범명제(규범 문장 단위) 로 타입이 선언
2) 이 규범명제는 LSA_Art7(= 법 조문/단위) 에서 "표현된" 규범(근거 텍스트 링크) (ont:expressedInUnit → inst:LSA_Art7)
3) 규범의 구성 요소
   · ont:hasSubjectRole→ inst:Role_Employer
   → 규범의 주체 역할은 "Employer(사용자/사업주)"
   · ont:hasAction → inst:Act_ForcedLabor
   → 규범이 규율하는 행위는 "ForcedLabor(강제근로 시키는 행위)"
   · ont:hasCondition → inst:Cond_Art7_Violence
   → 적용 요건/조건(예: 폭행·협박 등)이 Cond_Art7_Violence로 분리돼 연결
4) 모달리티 (의무/금지/허용 등)
   · inst:Mod_PROHIBITION rdf:type → ont:DeonticModality
   → 이 규범의 모달리티는 PROHIBITION(금지) 이고, 그것이 DeonticModality(규범 모달리티) 타입
5) 효과와 제재 연결
   · ont:hasEffect → inst:Eff_SanctionArise107
   → 위반 시 법적 효과로 "제재가 발생(Art107 기반)" 같은 효과 노드로 연결
   · ont:hasSanction → inst:SN_Art107
→ 실제 제재(벌칙) 조문/규범(Art107) 쪽으로 링크를 걸어 "이 규범 위반 → 어떤 벌칙 규정이 적용되는지"를 추적

TopBraid 화면 (5)



- 법문서 구조(장.조) + 규범명제(주체.행위.객체.효과.모달리티)+ 조문간 연결(위임/참조)를 보여주는 그래프

1) 이 조문은 어디에 속하냐를 해결

inst:LSA : 법령(instrument) 자체(문서 전체)

inst:LSA_Ch1 rdf:type ont:Chapter

ont:partOfInstrument → inst:LSA

→ "Chapter 1은 LSA 문서의 일부"

2) 조문 안의 규범을 구조화

rdf:type → ont:NormStatement

→ NS_Art3는 "규범명제"로 선언

NS_Art3는 규범의 구성요소를 쪼개서 연결

ont:hasSubjectRole → inst:Role_Party

inst:Role_Party rdf:type ont:LegalRole

→ 주체 역할(당사자 등)

ont:hasAction → inst:Act_LowerConditions

inst:Act_LowerConditions rdf:type ont:LegalAction

→ 행위(근로조건을 낮추는 행위)

3) 조문 간 연결: "위임/참조"

(1) 위임 범위

inst:DL_01 ont:delegationScope → inst:LSA_Art11

inst:NS_Art11 ont:expressedInUnit → inst:LSA_Art11

→ DL_01 같은 노드는 "위임"을 사건/관계로 보고, 그 위임의 범위가 Art11에 걸린다를 명시

(2) 참조(인용) 링크

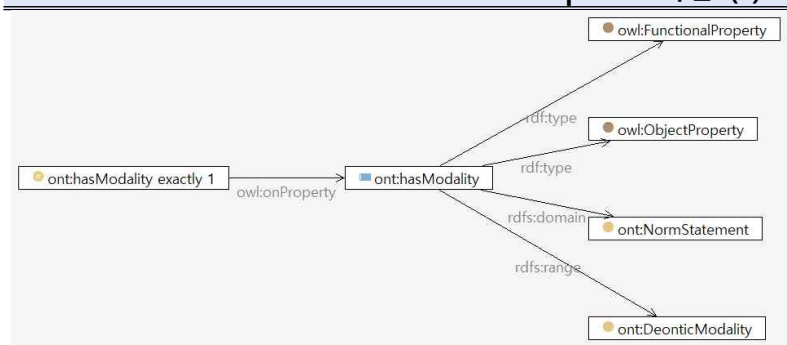
inst:RL_01 rdf:type ont:ReferenceLink

ont:referenceScope → inst:LSA_Art1

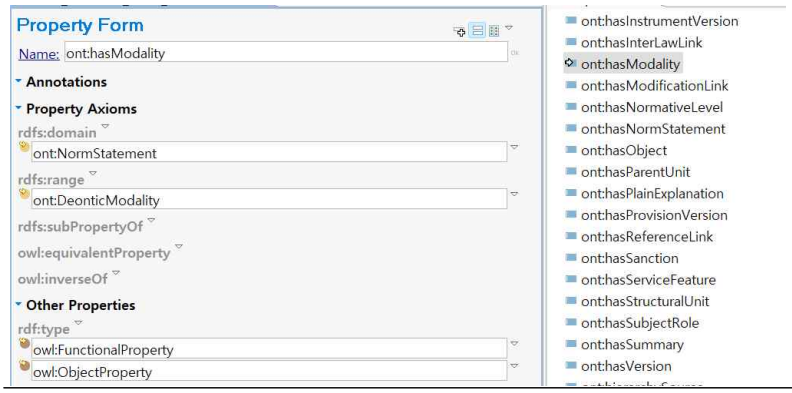
ont:referenceTarget → inst:CONST

→ "제1조가 헌법을 참조한다"를 ReferenceLink라는 중간 노드로 표현

TopBraid 화면 (6)



- ont:hasModality 속성은 "규범문(NormStatement)에 의무·허용·금지 같은 모달리티(DeonticModality)를 1개만 붙인다"로 모델링 됨
- 왼쪽의 ont:hasModality exactly 1 제약은 정확히 1개(= 최소 1개 + 최대 1개) 값을 가져야 한다는 뜻

TopBraid 화면 (7)	
	1) ont:hasModality는 - rdfs:domain은 ont:NormStatement - rdfs:range는 ont:DeonticModality 2) ont:hasModality 속성 type은 - owl:FunctionalProperty 이며 - owl:ObjectProperty이다.

2) 온톨로지 모델의 추론 테스트 개요

법령온톨로지 추론 테스트는 논리적으로 따라 나오는 사실(함의, entailment)을 계산하고 동시에 논리 모순이 있는지를 검사하여 설계된 모델의 기본적인 성능을 입증하는 단계이다.

- 모델이 의도한 대로 자동 분류/유도(inferred classification) 되는지
- 제약 (Disjointness, Functional, Domain/Range, Cardinality 등) 때문에 모순/불만족(unsatisfiable) 이 생기지 않는지

를 확인하는 테스트로서 추론 테스트를 통과하는 것은 '추론에 문제가 없다'는 의미이며 보통 아래와 같은 테스트 종류가 있다.

가) Consistency 테스트

논리적으로 모순이 없다는 의미이며, 현재 선언한 제약/정의들이 서로 충돌하지 않고, 모델이 "성립 가능한 세계"를 갖는다는 뜻이다.

나) Classification 테스트

정의 기반 자동 분류가 가능하다는 의미이며, 클래스/속성 정의가 충분히 구조화되어 있어, 추론기가 상속과 타입(유형)을 계산할 수 있다는 뜻이다.

다) Verification 테스트

제약이 '검증 장치'로 작동하는 것으로 Functional/도메인·레인지/Disjointness/카디널리티 등이 실제로 데이터(개체/트리플)를 걸러내는 역할을 수행한다는 의미이다.

라) 품질 테스트

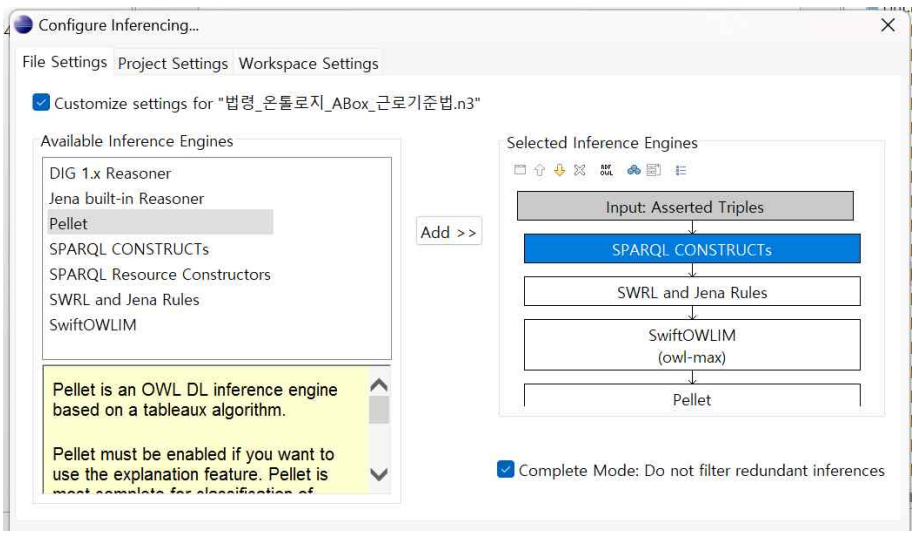
새로운 조문이나 규범 명제 등 인스턴스를 추가해도, 논리 오류를 추론이

빨리 잡아낼 가능성이 높아짐으로 운영/확장 안정성이 올라간다.

3) 법령 온톨로지 모델 추론을 위한 추론 엔진 구성

법령에 최적화된 서비스를 제공하기 위해 본 연구는 추론 엔진을 다음과 같이 구성하였으며, 성능 최적화를 위해 추론엔진의 운용 순서와 방법을 선정하는 것이 중요하기 때문이다.

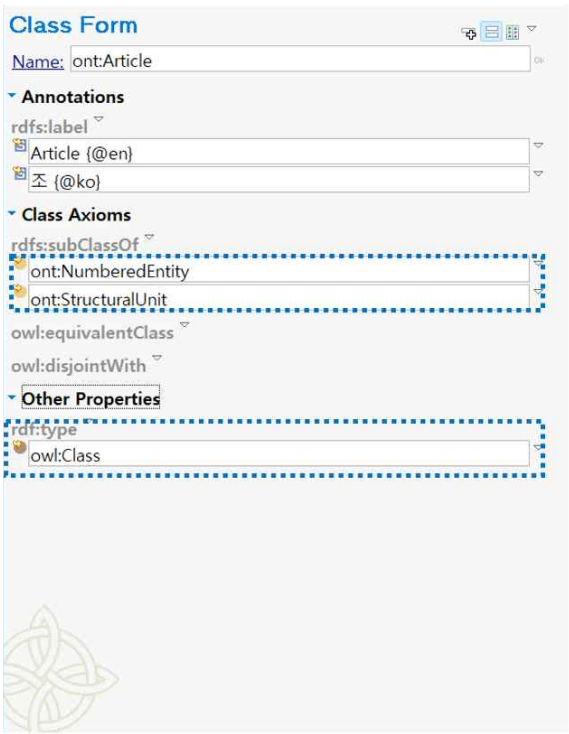
순서	추론 엔진	설 명
1	SPARQL CONSTRUCTs	- 패턴 기반으로 명시 트리플 생성하며, 가장 "통제 가능"하고 디버깅 쉬움 예: 조문 간 참조 링크 노드 생성, 규범명제 템플릿 채우기, 특정 패턴이면 관계 속성 자동 연결 등 지원
2	SWRL + Jena Rules	- SPARQL로 표현하기 번거로운 도메인 규칙(If–Then)을 추가로 반영할 수 있음 - 규칙이 많아지면 과추론/폭발 가능하기 때문에 적절한 범위를 구성하 는 것이 필요 예: (금지 + 특정 요건) ⇒ (제재발생) 같은 추론
3	SwiftOWLIM (owl-max)	- 앞 단계에서 만들어진 트리플까지 포함해서 대용량 물질화(materialization) 주로, 타입 전파(도메인/레인지) 및 서브클래스/서브프로퍼티 전파, 동일성(sameAs) 확장 등에 활용 - 운영/조회 성능을 위해 이 단계 결과를 "Inferred graph"로 쓰는 경우가 많음
4	Pellet	- 최종 Consistency checking이나 필요 시 Classification에 활용 - Functional에서 다른 모달리티 2개가 붙는 경우, 카디널니티 위반이나 Disjoint 클래스 동시 귀속(정의 자체가 모순인 클래스)



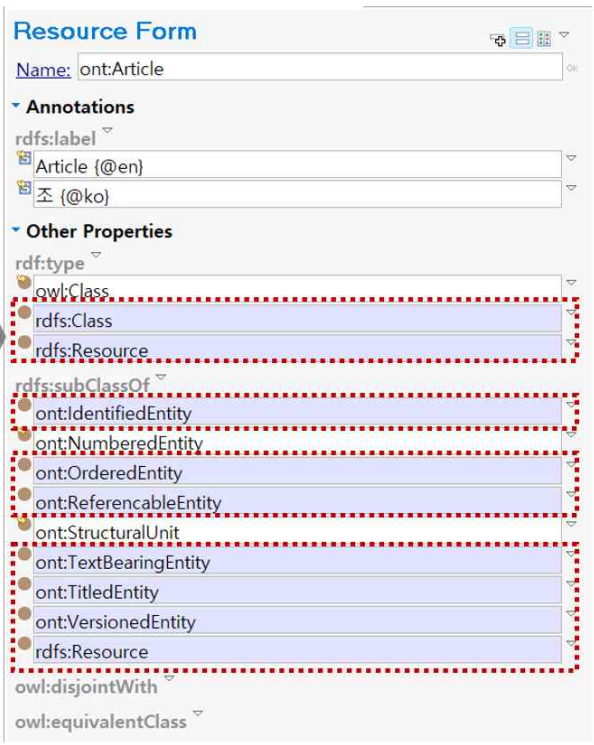
(그림3-13) 온톨로지 추론 엔진 구성 (TopBraid 환경)

온톨로지 모델 추론 검증 결과

법령 온톨로지 추론 전 클래스 현황



법령 온톨로지 추론 후 클래스 변화



(그림3-14) 온톨로지 추론 성능 검증과 검증 결과

(그림3-14)에서 좌측은 TopBraid에서 클래스로 선언한 ont:Article(‘조’)가 추론 이전에는 ont:NumberedEntity와 ont:StructuralUnit의 하위클래스로 선언되어 있는데, 추론을 실행한 결과 추론 엔진이 다양한 지식을 연계하여 ont:Article의 정보가 강화되었음을 보여준다.

추론 실행 전	추론 실행 후	설명
(1) rdf:type → owlClass (2) rdfs:subClassOf → ont:NumberedEntity → ont:StructuralUnit	(1) rdf:type → owlClass → rdfs:Class → rdfs:Resource (2) rdfs:subClassOf → ont:IdentifiedEntity → ont:OrderedEntity → ont:ReferencableEntity → ont:TextBearingEntity → ont:TitledEntity → ont:VersionedEntity → rdfs:Resource → ont:NumberedEntity → ont:StructuralUnit	추론 실행 전에는 명시된 정보들만 있으나 추론 실행 후에는 명시된 정보와 추론(inferred)된 트리플이 같이 보여진다. 추론은 이같이 모델에 기술된 공리나 관계등을 이용해 실제 상속(전이적)을 하게되며 이를 통해 on:Article가 단순히 조문이 아니라 식별될 수 있고, 순서가 있고, 참조 가능하고, 텍스트를 담고, 제목을 가질 수 있고, 버전 관리가 될 수 있된 SPARQL 문에서 “Article”을 지정하지 않아도 referencableEntity 같은 상위 개념을 질의하면 Article도 자동 포함

4) 법령 온톨로지 인스턴스 구성 확인

본 절은 법령 온톨로지 TBox 스키마와 ABox에 의해 실제 법조문에서 추출되어 인스턴스로 구성된 현황을 살펴보고 본 연구에서 설계한 온톨로지 모델과 구축 방법론이 타당함을 검토하였다.

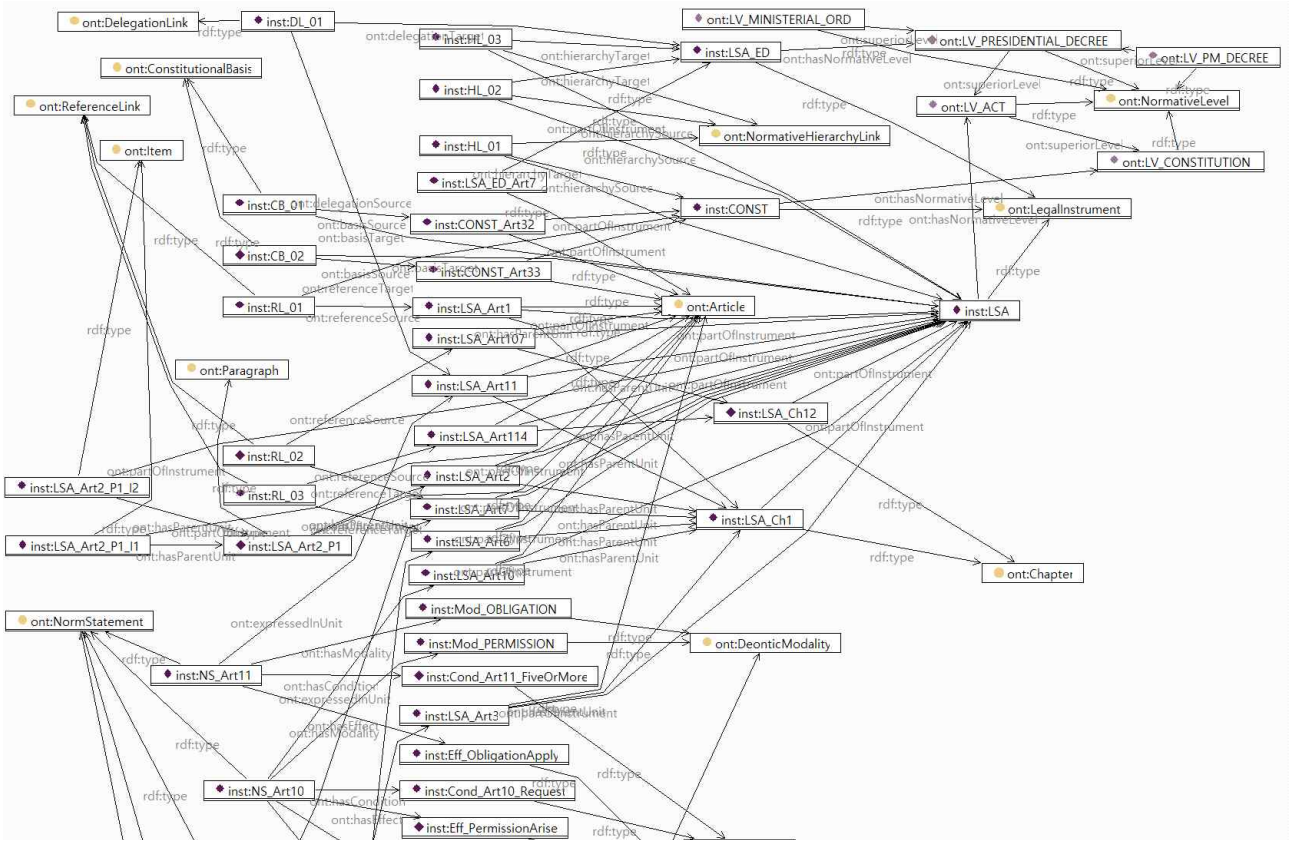
(그림3-15) 근로기준법 조문의 온톨로지 인스턴스화 결과

(그림3-15) 상단의 원문 텍스트에서 조문 및 하위 항.호 단위를 분해한 뒤, TopBraid 환경에서 각 단위를 개별 인스턴스로 구성한 모습이다. 생성된 인스턴스는 해당 구조 단위의 유형에 맞는 클래스 (예: Article, Paragraph, Item 등) 타입이 부여되며, 상.하위 단위 간 관계(조→항→호)가 부모-자식 구조로 연결된다.

또한 제안한 추출 절차를 통해 조문 표제와 본문 내용이 각각 ont:title 및 ont:title 속성에 정확히 매핑되어 입력된다. 이와 같은 결과는 비정형 법령 텍스트가 구조화된 지식 표현으로 변환될 수 있음을 보여주며, 이후 참조관계 분석, 규범명제 추출, 질의 기반 탐색과 같은 후속 처리의 기반을 제공한다.

(그림3-16)은 법령 온톨로지에서 생성된 인스턴스(개별 법령・조문・항・호・규범요소・링크) 들이 서로 어떤 관계로 연결되는지를 한 화면에 보여주는 인스턴스 그래프(부분) 이다.

이 그래프는 법령 텍스트를 “문서 구조(조문 체계) + 규범 의미(요건-효과 등) + 법령 간 연결(참조/위계)”로 동시에 모델링했음을 보여주며, 향후 정밀 검색, 연관 법령 추적, 영향 분석, 추론 기반 질의를 가능하게 하는 기반 구조임을 시각적으로 제시한다.



(그림3-16) 근로기준법 조문의 온톨로지 인스턴스 그래프(일부)

제4절. (노동)법령 온톨로지 구축 통계

1. 법령 온톨로지 스키마 (TBox) 통계

가. 법령 온톨로지 스키마 총괄

본 절에서는 (노동)법령 온톨로지 구축 결과로 시스템에 생성된 스키마와 인스턴스를 종합적으로 요약한다. 특히 인스턴스는 실제 시스템 운영 과정에서 추가·수정·삭제가 지속적으로 이루어질 수 있으므로, 생성 데이터는 고정값이 아닌 동적 데이터로 관리된다.

순서	구 분	개수	비 고
1	owl:Class	48개	8개 카테고리로 분류
2	owl:DatatypeProperty	61개	신규 13개 포함
3	owl:ObjectProperty	58개	신규 9개 포함
4	프로퍼티 합계	119개	
5	owl:FunctionalProperty	37개	DTP/OP 중복 표기
6	owl:Restriction	7개	
7	NormativeLevel 인스턴스	7개	TBox 내 고정 인스턴스

나. 법령 온톨로지 스키마(TBox) 분류

1) 상위추상 클래스

상위추상 클래스는 앞절에서 설계한 바와 같이 7개가 구축되었다.

순서	클래스명	상위클래스	비 고
1	IdentifiedEntity	owl:Thing	식별자 부여 추상 클래스
2	TitledEntity	owl:Thing	제목 보유 추상 클래스
3	TextBearingEntity	owl:Thing	본문 텍스트 보유
4	OrderedEntity	owl:Thing	정렬 인덱스 보유
5	NumberedEntity	owl:Thing	번호 보유
6	ReferencableEntity	owl:Thing	참조 가능 대상
7	VersionedEntity	owl:Thing	버전 관리 대상

2) 법령문서 클래스

법령문서 클래스는 앞 절의 설계 내용에 따라 4개로 구축되었으며, 이들의 상위 클래스는 IdentifiedEntity이다.

순서	클래스명	상위클래스	비 고
1	LegalInstrument	IdentifiedEntity	법령 (헌법, 법률, 시행령 등)
2	InstrumentVersion	IdentifiedEntity	법령 개정 버전
3	ProvisionVersion	IdentifiedEntity	조문 개정 버전
4	CompetentAuthority	IdentifiedEntity	소관 부처/기관

3) 문서구조 클래스

문서구조 클래스는 총 13개가 생성되었으며, 상위 클래스는 설계 목적에 따라 3개 범주로 구성・연계된다.

순서	클래스명	상위클래스	비 고
1	StructuralUnit	IdentifiedEntity	문서구조 추상 클래스
2	Part	StructuralUnit	편
3	Chapter	StructuralUnit	장
4	Section	StructuralUnit	절
5	Subsection	StructuralUnit	관
6	Article	StructuralUnit	조
7	Paragraph	StructuralUnit	항
8	Item	StructuralUnit	호
9	Subitem	StructuralUnit	목
10	Appendix	IdentifiedEntity	부속 문서 추상 클래스
11	Addendum	Appendix	부칙
12	AnnexTable	Appendix	별표
13	LegalForm	Appendix	서식

4) 규범명제 클래스

규범명제 클래스는 8개가 생성되었고, 상위 클래스는 목적에 따라 3개로 연계된다.

순서	클래스명	상위클래스	비 고
1	NormStatement	IdentifiedEntity	규범명제 (주체-행위-객체-양상-효과)
2	DeonticModality	IdentifiedEntity	의무양상 (OBLIGATION/PROHIBITION 등)
3	LegalRole	IdentifiedEntity	법적 역할 (사용자/근로자 등)
4	LegalAction	IdentifiedEntity	법적 행위 (차별/강제근로 등)
5	LegalObject	IdentifiedEntity	법적 객체 (근로조건 등)
6	Condition	IdentifiedEntity	적용 조건 (수량/시간/절차적 등)
7	PlainExplanation	IdentifiedEntity	평이한 설명문
8	AuthorityAction	IdentifiedEntity	권한행사/행정행위

5) 링크 / 참조 클래스

링크/참조 클래스는 7개가 생성되었고, 상위 클래스는 IdentifiedEntity로 연계된다.

순서	클래스명	상위클래스	비 고
1	LinkEntity	IdentifiedEntity	링크 추상 클래스
2	ReferenceLink	IdentifiedEntity	조문 간 참조 (CITATION 등)
3	DelegationLink	IdentifiedEntity	위임 관계
4	InterLawLink	IdentifiedEntity	법률 간 관련
5	ModificationLink	IdentifiedEntity	준용/예외/특례
6	CaseLawLink	IdentifiedEntity	판례 연결
7	ConstitutionalBasis	IdentifiedEntity	헌법적 근거

6) 법적효과/제재 클래스

법적효과/제재 클래스는 2개가 생성되었고, 상위 클래스는 IdentifiedEntity로 연계된다.

순서	클래스명	상위클래스	비 고
1	LegalEffect	IdentifiedEntity	법적 효과 (NULLITY/SANCTION_ARISE 등)
2	Sanction	IdentifiedEntity	벌칙/제재 (CRIMINAL_PENALTY 등)

7) 규범위계/충돌 클래스

규범위계/충돌 클래스는 5개가 생성되었고, 상위 클래스는 IdentifiedEntity로 연계된다.

순서	클래스명	상위클래스	비 고
1	NormativeLevel	IdentifiedEntity	법령 위계 등급 (헌법~규칙)
2	NormativeHierarchyLink	IdentifiedEntity	위계 관계 링크
3	NormConflictCase	IdentifiedEntity	규범 충돌 사례
4	ResolutionDecision	IdentifiedEntity	충돌 해결 결정
5	ResolutionPrinciple	IdentifiedEntity	해결 원칙 (상위법/신법/특별법)

8) 서비스/뷰 클래스

서비스/뷰 클래스는 2개가 생성되었고, 상위 클래스는 IdentifiedEntity로 연계된다.

순서	클래스명	상위클래스	비 고
1	ServiceFeature	IdentifiedEntity	서비스 기능 유형
2	SummaryView	IdentifiedEntity	요약 뷰

다. 법령 온톨로지 스키마(TBox) 프로퍼티

1) DatatypeProperty

DatatypeProperty는 61개로 구성되었으며, 클래스에 대한 상세화와 역할을 지원하기 위한 기본 정보로서 역할하게 된다. .

순서	FP	전방 추론	프로퍼티명	한글	DOMAIN	RANGE	설명
1			adminClassCode		CompetentAuthority	xsd:string	
2	●	★	agreementType	합의유형	Condition	xsd:string	서면합의 유형 (WRITTEN_REPRESENTATIVE 등)
3			amendedOn		ProvisionVersion	xsd:date	
4			amendmentNoteText		ProvisionVersion		
5			anchorText		LinkEntity		
6			appendixType		Appendix	xsd:string	
7			caseNo		CaseLawLink	xsd:string	
8			conditionType		Condition	xsd:string	

순서	FP	전방 추론	프로퍼티명	한글	DOMAIN	RANGE	설명
9			constitutionalArticle		ConstitutionalBasis	xsd:string	
10			court		CaseLawLink	xsd:string	
11			delegationScopeText		DelegationLink		
12			delegationType		DelegationLink	xsd:string	
13			deptName		CompetentAuthority	xsd:string	
14			effectType		LegalEffect	xsd:string	
15			effectiveFrom		LegalInstrument	xsd:date	
16			effectiveRuleText		Addendum		
17	●	★	effectiveYear	적용연도	Condition	xsd:integer	최저임금 등 기준이 적용되는 연도
18			featureType		ServiceFeature	xsd:string	
19			formNo		LegalForm	xsd:string	
20	●	★	hasWrittenContract	서면계약여부	LegalRole	xsd:boolean	서면 근로계약서 작성 여부 (17조)
21			hierarchyBasisText		NormativeHierarchyLink		
22	●		hierarchyRank		NormativeLevel	xsd:integer	위계 순서 (1=헌법, 2=법률…)
23	●		hierarchyType		NormativeHierarchyLink	xsd:string	
24	●		identifier		IdentifiedEntity	xsd:string	
25	●	★	inspectionTarget	감독대상여부	LegalRole	xsd:boolean	특별근로감독 대상 여부 (추론 결과)
26	●		instrumentType		LegalInstrument	xsd:string	열거형 InstrumentType
27			issuingAuthority		NormativeLevel	xsd:string	
28			language		LegalInstrument	xsd:language	
29	●		levelCode		NormativeLevel	xsd:string	
30			levelLabel		NormativeLevel		
31	●	★	lsaApplicable	법적용여부	LegalRole	xsd:boolean	근로기준법 적용 여부 (추론 결과)
32	●	★	minimumHourlyWage	최저시급	Condition	xsd:integer	최저임금 시간급 (원 단위)
33	●		modalityCode		DeonticModality	xsd:string	열거형 DeonticModalityCode
34			modificationType		ModificationLink	xsd:string	
35	●	★	monthlySalary	월급여	LegalRole	xsd:integer	근로자의 월 급여액 (원 단위)
36	●	★	monthlyScheduledHours	월소정근로시간	LegalRole	xsd:integer	시행령§6 통상임금 산정 기준
37			number		NumberedEntity	xsd:string	
38			operator		Condition	xsd:string	
39			orderIndex		OrderedEntity	xsd:integer	
40			orgName		CompetentAuthority	xsd:string	
41	●	★	overtimePremiumRate	연장가산율	LegalRole	xsd:integer	지급 임금 비율(%), 150=통상+50%가산
42			promulgatedOn		LegalInstrument	xsd:date	
43			promulgationNo		LegalInstrument	xsd:string	
44			referenceType		ReferenceLink	xsd:string	
45	●	★	regularWorkerCount	상시근로자수	LegalRole	xsd:integer	상시 근로자 수 (법§11 적용 기준)
46			relationType		InterLawLink	xsd:string	
47			revisionType		LegalInstrument	xsd:string	
48			sanctionType		Sanction	xsd:string	
49			scope		LinkEntity	xsd:string	
50			sharedElement		InterLawLink	xsd:string	

순서	FP	전방 추론	프로퍼티명	한글	DOMAIN	RANGE	설명
51			sourceSystem		LegalInstrument	xsd:string	
52			tableType		AnnexTable	xsd:string	
53			text		TextBearingEntity		
54			title		TitledEntity		
55	●		unitType		StructuralUnit	xsd:string	열거형 UnitType
56			validFrom			xsd:date	
57			validTo			xsd:date	
58			value		Condition	xsd:string	
59		★	violatedNorm	위반규범	LegalEffect	xsd:string	위반된 규범의 조항 텍스트
60	●	★	violationCount	위반건수	LegalRole	xsd:integer	사업장 총 위반 건수 (추론 결과)
61	●	★	weeklyWorkHours	주근로시간	LegalRole	xsd:integer	1주간 실제 근로시간 (법§50,§53)

2) ObjectProperty

ObjectProperty는 58개로 구성되었으며, 법령의 기본적인 추적, 연계, 시계열 등의 중요한 기능을 지원한다.

순서	FP	전방 추론	프로퍼티명	한글	DOMAIN	RANGE	설명
1	●	★	appliesToWorkplace	적용사업장	Condition	LegalRole	해당 합의가 적용되는 사업장
2			authorityMayDo		NormStatement	AuthorityAction	
3	●		basisSource		ConstitutionalBasis	StructuralUnit	
4	●		basisTarget		ConstitutionalBasis	LegalInstrument	
5	●	★	basisViolation	근거위반	Sanction	LegalEffect	벌칙 부과의 근거 위반 사항
6			conflictingNorm		NormConflictCase	NormStatement	
7	●		delegationSource		DelegationLink	StructuralUnit	
8			delegationTarget		DelegationLink	ReferencableEntity	
9			derivedFrom		ProvisionVersion	ProvisionVersion	
10			enforcedByAuthority		NormStatement	CompetentAuthority	
11			expressedInUnit		NormStatement	StructuralUnit	
12			hasAction		NormStatement	LegalAction	
13			hasAddendumChild		Addendum	StructuralUnit	
14			hasAppendix		LegalInstrument	Appendix	
15			hasAuthority		LegalInstrument	CompetentAuthority	
16			hasChildUnit		StructuralUnit	StructuralUnit	
17			hasCondition		NormStatement	Condition	
18			hasConstitutionalBasis		LegalInstrument	ConstitutionalBasis	
19			hasDelegationLink		StructuralUnit	DelegationLink	
20			hasEffect		NormStatement	LegalEffect	
21			hasHierarchyLink		LegalInstrument	NormativeHierarchyLink	
22			hasInstrumentVersion		LegalInstrument	InstrumentVersion	
23			hasInterLawLink		LegalInstrument	InterLawLink	
24	●		hasModality		NormStatement	DeonticModality	

순서	FP	전방 추론	프로퍼티명	한글	DOMAIN	RANGE	설명
25			hasModificationLink		NormStatement	ModificationLink	
26			hasNormStatement		StructuralUnit	NormStatement	
27	●		hasNormativeLevel		LegalInstrument	NormativeLevel	
28			hasObject		NormStatement	LegalObject	
29			hasParentUnit		StructuralUnit	StructuralUnit	
30		★	hasPenalty	벌칙	LegalRole	Sanction	부과 가능한 벌칙 (추론 결과)
31			hasPlainExplanation		LegalInstrument	PlainExplanation	
32			hasProvisionVersion		StructuralUnit	ProvisionVersion	
33			hasReferenceLink		StructuralUnit	ReferenceLink	
34			hasSanction		NormStatement	Sanction	
35			hasServiceFeature		LegalInstrument	ServiceFeature	
36			hasStructuralUnit		LegalInstrument	StructuralUnit	
37			hasSubjectRole		NormStatement	LegalRole	
38			hasSummary		LegalInstrument	SummaryView	
39			hasVersion		VersionedEntity		
40		★	hasViolation	위반사항	LegalRole	LegalEffect	탐지된 법 위반 사항 (추론 결과)
41	●		hierarchySource		NormativeHierarchyLink	LegalInstrument	
42	●		hierarchyTarget		NormativeHierarchyLink	LegalInstrument	
43			instrumentsAtLevel		NormativeLevel	LegalInstrument	
44	●		modifiesTarget		ModificationLink	NormStatement	
45	●		partOfInstrument		StructuralUnit	LegalInstrument	
46	●	★	penaltyArticle	벌칙조문	Sanction	Article	벌칙 규정이 명시된 조문
47	●		preferredNorm		ResolutionDecision	NormStatement	
48	●		referenceSource		ReferenceLink	StructuralUnit	
49	●		referenceTarget		ReferenceLink	ReferencableEntity	
50			relatedCase			CaseLawLink	
51			resolutionPrinciple		ResolutionDecision	ResolutionPrinciple	
52	●		resolvesConflict		ResolutionDecision	NormConflictCase	
53			subordinateLevel		NormativeLevel	NormativeLevel	
54			superiorLevel		NormativeLevel	NormativeLevel	
55	●	★	violatedArticle	위반조문	LegalEffect	Article	위반된 법 조문 인스턴스
56		★	violatingWorker	위반관련근로자	LegalEffect	LegalRole	위반과 관련된 근로자
57		★	violatingWorkplace	위반사업장	LegalEffect	LegalRole	위반이 발생한 사업장
58	●	★	worksAt	근무사업장	LegalRole	LegalRole	근로자가 소속된 사업장

2. 법령 온톨로지 인스턴스

가. NormativeLevel 인스턴스 (TBox 내 고정)

NormativeLevel 인스턴스는 TBox에 고정되는 인스턴스인데, 이는 예: 헌법, 법률, 대통령령 등 시스템 운영 중 임의로 증감하는 데이터가 아니라, 온톨로지 설계 단계에서 미리 정의해 둔 기준값(관리형 코드값) 으로 유지한다는 의미이다. 이 인스턴스들은 법규범의 위계(hierarchyRank) 와 제정 주체(issuingAuthority) 를 함께 표현하는 기준 체계로서, 각 법령을 일관되게 분류・연계하는 참조 축 역할을 한다.

순서	인스턴스	한글명	hierarchyRank	issuingAuthority
1	LV_CONSTITUTION	헌법	1	—
2	LV_ACT	법률	2	NATIONAL_ASSEMBLY
3	LV_PRESIDENTIAL_DECREE	대통령령	3	PRESIDENT
4	LV_PM_DECREE	총리령	4	PRIME_MINISTER
5	LV_MINISTERIAL_ORD	부령	4	MINISTER
7	LV_MUNICIPAL_ORD	조례	5	LOCAL_COUNCIL

나. ABox 인스턴스 클래스별 분류

NormativeLevel 인스턴스는 TBox에 고정되는 인스턴스인데, 이는 예: 헌법, 법률, 대통령령 등 시스템 운영 중 임의로 증감하는 데이터가 아니라, 온톨로지 설계 단계에서 미리 정의해 둔 기준값(관리형 코드값) 으로 유지한다는 의미이다. 이 인스턴스들은

카테고리	클래스	개수	인스턴스
법령문서	LegalInstrument	3	CONST, LSA, LSA_ED
문서구조	Chapter	2	LSA_Ch1, LSA_Ch12
	Article	12	CONST_Art32~33, LSA_Art1~114, LSA_ED_Art7
	Paragraph	1	LSA_Art2_P1
	Item	2	LSA_Art2_P1_I1, I2
규범위계	NormativeHierarchyLink	3	HL_01(헌법→법), HL_02(위임), HL_03(상하위)
헌법적 근거	ConstitutionalBasis	2	CB_01(헌법§32), CB_02(헌법§33)
참조링크	ReferenceLink	3	RL_01~03
위임링크	DelegationLink	1	DL_01 (§11→시행령)
규범명제	NormStatement	5	NS_Art3, Art6, Art7, Art10, Art11
양상성	DeonticModality	3	Mod_PROHIBITION, OBLIGATION, PERMISSION
법적역할	LegalRole	5	Role_Employer/Worker/Party, CompanyB★, KimCS★
법적행위	LegalAction	4	Act_LowerConditions 외 3개

카테고리	클래스	개수	인스턴스
법적객체	LegalObject	1	Obj_WorkConditions
조건	Condition	6	Cond_Art6~11, MinWage2024★, OvertimeAg★
법적효과	LegalEffect	5	Eff_NullityBelow 외 4개
제재	Sanction	2	SN_Art107, SN_Art114
합계		**60**	

제5절. 소결 및 시사점

1. 소결

제3장은 노동법(근로기준법 중심) 영역을 대상으로, 법령지식을 온톨로지로 구조화하고 이를 실제 시스템에서 활용 가능한 형태로 구축하는 과정을 제시하였다. 먼저 노동관계 법령의 범위를 정리하고, 법령 텍스트를 단순 문장 집합이 아니라 구조적 위계(편・장・절・조・항・호・목), 문장 성분, 의미론적 요소(주체・행위・객체・요건・효과・제재)로 분해하는 지식 구조화 기준을 설정하였다. 이를 통해 법령을 추론 가능한 지식 단위로 재구성하기 위한 개념적 기반을 마련하였다.

이후 법령 온톨로지 모델은 추상 클래스 기반 공통 메타데이터, 법령위계 레이어, 문서구조 레이어, 규범명제 레이어, 링크/참조 레이어를 중심으로 설계되었으며, 특히 **위계(레벨)-법령(문서)-링크(근거/위임/참조)**를 분리하여 규범의 법체계적 위치와 관계 근거를 함께 표현할 수 있도록 구성하였다. 또한 규범명제를 조문과 연결(추적성)하고, 링크 엔티티를 1급 객체로 관리함으로써 향후 설명가능한 검색・추론・영향분석이 가능한 구조를 확보하였다.

구축 측면에서는 law.go.kr Open API의 XML 응답을 RDF(Turtle)로 변환하는 **5단계 파싱 파이프라인**을 설계하고, 문서구조/법령위계/규범명제 레이어별 매핑 규칙을 정의하였다. 이 과정에서 조문 패턴 기반의 규칙 라이브러리를 적용하여 인스턴스 자동 생성을 수행하였고, 일부 의미 해석이 필요한 규범명제 요소는 반자동 방식으로 처리하도록 구성하였다. 특히 패턴 기반 추출은

총 8개 대상에 대해 70개 이상의 패턴으로 정리되어 자동화 범위와 한계를 함께 제시했다는 점에서 실무적 의의가 있다.

구축 결과, TBox는 48개 클래스, 61개 DatatypeProperty, 58개 ObjectProperty(총 119개 프로퍼티), 7개 제약(Restriction)으로 구성되었으며, 규범위계 인스턴스(NormativeLevel)도 별도로 정리되었다. 클래스는 상위 추상, 법령문서, 문서구조, 규범명제, 링크/참조, 법적효과/제재, 규범위계/충돌, 서비스/뷰 등으로 체계화되었다. ABox는 예시 구축 기준으로 총 60개 인스턴스가 생성되었고, 법률・시행령・시행규칙, 조문 단위, 위계 링크, 참조/위임 링크, 규범명제 구성요소(주체・행위・객체・요건・효과・제재)까지 포함되어, 실제 추론 엔진의 입력 기반으로 활용 가능한 수준의 데이터 구조를 확인하였다.

2. 시사점

제3장의 가장 큰 시사점은 법령 온톨로지가 단순한 분류체계가 아니라, **법령의 구조·관계·규범성·시간성**을 함께 담는 실행 가능한 지식 스키마로 기능할 수 있음을 보여준 점이다. 특히 조문 단위 구조와 규범명제, 그리고 위임・참조・타법연계를 분리하면서도 연결 가능한 형태로 설계함으로써, 이후 장에서 구현된 전방추론/후방추론, 개정영향 탐지, 법적 판단 자동도출의 기술적 기반이 자연스럽게 성립한다.

또한 본 장은 “법령 검색” 중심 접근에서 벗어나, 법령을 **지식그래프 기반의 추론 자산으로 전환**하는 실질적 경로를 제시한다. 즉, 법령 텍스트를 RDF 인스턴스로 변환하고, 규범명제 및 링크 관계를 명시적으로 모델링함으로써, 기존 키워드 검색으로는 어렵던 근거 추적, 영향 범위 탐지, 판단 과정의 설명 가능성을 확보할 수 있다.

아울러 구축 결과를 수치(클래스/프로퍼티/인스턴스 수)로 제시한 점은 연구의 재현성과 확장 가능성을 높인다. 이는 향후 다른 법령군(산업안전보건법, 고용보험법, 남녀고용평등법 등)으로 범위를 확대할 때, 동일한 스키마 원칙과 파이프라인을 적용하여 점진적으로 확장할 수 있는 기반이 된다.

제4장 (노동)법령 온톨로지 활용

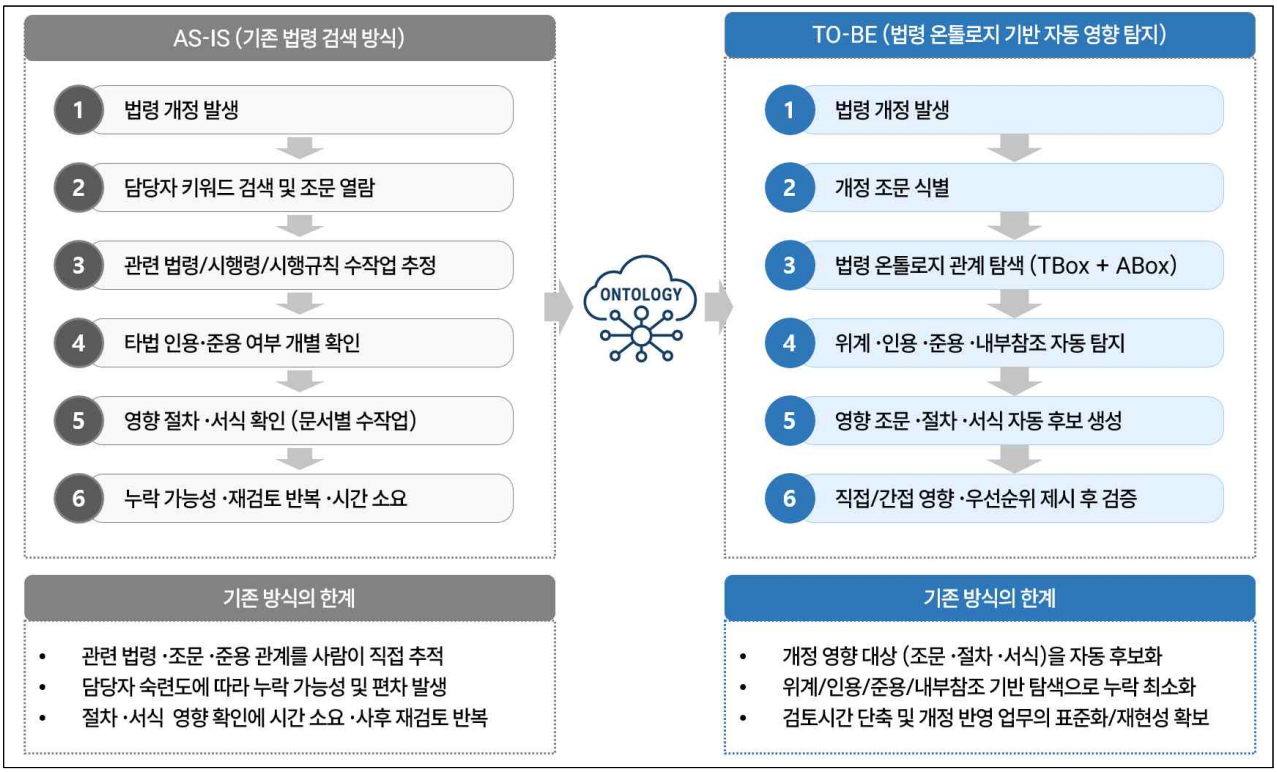
제1절. (노동)법령 온톨로지 효과 검증

1. 법령 개정의 영향 탐지 체계 검증

가. 검증 개요

본 절은 법령 온톨로지를 이용하여, 법령 개정 시 영향을 받는 조문·절차·서식을 자동으로 탐지하여 기존 법령 검색의 한계를 극복할 수 있는지 검증하였다.

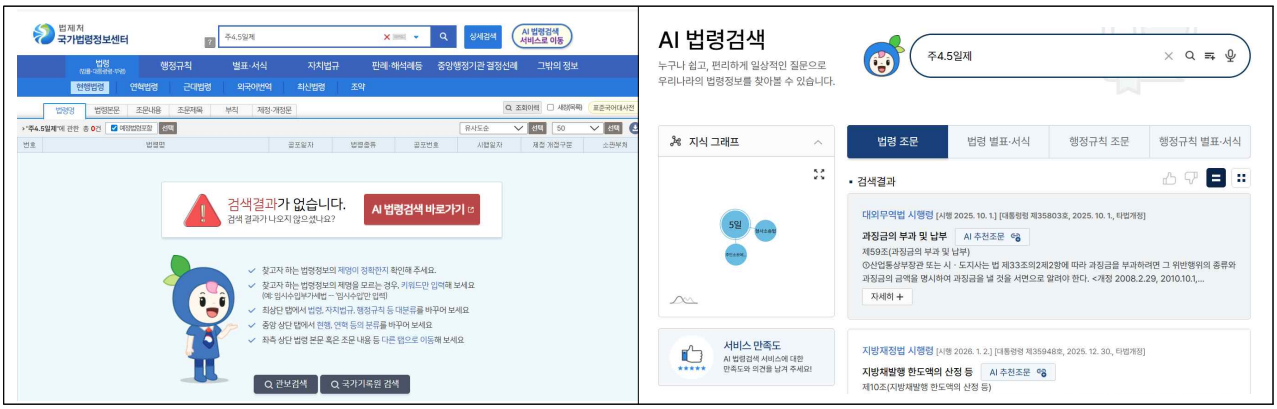
(그림4-1)은 법령 온톨로지 기반 개정 영향을 탐지하는 장단점을 비교한 것으로 기존 방식은 개정 법령 발생 시 담당자가 연관 법령, 조문, 절차, 서식을 수작업으로 검색·확인해야 하므로 누락과 시간 지연의 한계가 있다. 반면, 법령 온톨로지 기반 방식은 규범 위계, 타법 인용·준용, 내부 참조 및 절차·서식 연계를 자동 탐지하여 개정 영향 대상을 체계적으로 제시함으로써 기존 법령 검색의 한계를 보완할 수 있다.



(그림4-1) 법령 온톨로지 기반 개정 영향 탐지 체계

검증을 위한 사례는 최근 '주4.5일제'라는 정책이 시행될 때, 법령온톨로지를 활용했을 때 어떤 법령이 관련되고 해당 법령의 조문과 관련된 법령을 자동으로 탐지할 수 있는지를 검증하였다.

(그림4-2)는 기존 국가법령정보센터와 지능형법령검색시스템의 '주4.5일제'에 대한 검색 결과이다. 그림에서 알 수 있듯이 관련 '주4.5일제'에 대한 법령을 제공하지 못하는 한계가 있다.



(그림4-2) 기존 법령정보검색시스템의 검색 결과 (법제처)

나. 법령 온톨로지 검증 방법

검증 테스트는 법령온톨로지 TBox와 ABox를 대상으로 하여 (1) 검색어 관련 법령 추론 (2) 추론된 법령을 기준으로 한 연관 법령(조문, 타법, 상하위법, 서식 등)의 추론 (3) 추론된 결과의 수작업 확인 순서로 검증한다.

위 내용은 SPARQL을 사용하여 법령 온톨로지의 추론 결과를 확인한다.

1) 검색어 관련 법령 추론 방법

검색어 관련 법령 추론은 아래 설계 원리에 따라 쿼리를 구성하였다.

검색어 관련 추론 단계	
1단계 (설계 원리)	① 검색어가 현행 법령에 명시적으로 존재하는 용어인지 확인 ② 검색어를 구성 요소(근로시간·탄력적·선택적·연장·단시간·통상임금· 209시간·실근로시간 등)로 분해 ③ ABox의 5종 텍스트 속성 (title,text, anchorText, sharedElement, delegationScopeText)을 REGEX로 탐색하여 관련 법령을 추출
2단계 (쿼리 구성)	① 키워드 REGEX 텍스트 탐색 SPARQL 실행

검색어 관련 추론 단계	
	② 클래스 확장 후 의미기반 태그 탐색 SPARQL 실행 ③ ①과 ②을 통합하고 VALUES 파라미터화로 범용 정책 키워드 탐색 SPARQL 실행

2단계 REGEX 매칭[48]	
3단계 (쿼리)	① ABox 텍스트 속성에서 REGEX 매칭

```
PREFIX ont:  <http://legalontology.kr/ontology#>
PREFIX inst: <http://legalontology.kr/instance#>
SELECT DISTINCT ?tier ?entity ?title ?entityType ?matchSource
?matchDetail
WHERE {
  {
    # ── Tier A: Article.title 키워드 매칭 ──
    # 조문 제목에서 직접 키워드를 탐색
    ?entity a ont:Article ;
           ont:title ?title .
    FILTER(REGEX(STR(?title),
      "근로시간|소정근로|탄력적|선택적|연장.*근로|단시간|통상임금", "i"))
    BIND("A_조문제목" AS ?tier)
         ~~~~~~~~~
    BIND("LegalInstrument" AS ?entityType)
    BIND("ont:title(LegalInstrument)" AS ?matchSource)
    BIND(STR(?title) AS ?matchDetail)
  }
}ORDER BY ?tier ?entityType ?entity
```

2) 의미기반 태그 탐색

의미기반 태그 탐색은 TBox와 ABox를 기반으로 직접 태깅→Link 확장→반복 과정을 구성하여 전체 파급범위를 도출하는 단계이다.

의미기반 태크 설계 원리	
1단계 (설계 원리)	① 전체 TBox에 PolicyTag 클래스, ABox에 PT_4_DayWeek 인스턴스 등록 ② TBox와 ABox 확장 ③ 파급 전략: Hop 0 (직접 태깅) → Hop 1 (Link 확장) → 반복으로 전체 파급 범위
2단계	SELECT DISTINCT ?hop ?entity ?title ?entityType ?pathway

48) SPARQL의 쿼리 구문은 연구 보안 목적에 따라 전체 게시를 하지 않음

의미기반 태크 설계 원리	

(쿼리 구성)	WHERE {
	  # ── Step 0: 사용자 입력으로 PolicyTag 식별 ──
	  ?tag a ont:PolicyTag .
	  {
	  ?tag skos:prefLabel ?lbl .
	  FILTER(REGEX(STR(?lbl), "4\\.?5일", "i"))
	  }
	  UNION
	  {
	  ?tag skos:altLabel ?alt .
	  FILTER(REGEX(STR(?alt), "4\\.?5일|근로시간.*단축", "i"))
	  }
	  ~~
	  FILTER(?entity != ?tagged)
	  FILTER(!EXISTS { ?entity ont:hasPolicyTag ?tag })
	  BIND("Hop1_타법확장" AS ?hop)
	  BIND("LegalInstrument" AS ?entityType)
	  BIND(CONCAT("InterLawLink[", STR(?relType), "]") AS ?pathway)
	  }
	}ORDER BY ?hop ?entityType ?entity

실행 결과	Hop 0: LSA_Art50, Art51, Art51_2, Art52, Art53, Art56, Art69,
	  LSA_ED_Art6, ED_Art9, RealWorkHourReductionAct,
	  MinWageAct, GenderEqAct (직접 태깅된 12건)
	Hop 1: LSA_Art2(DEFINITION_REF→50), LSA_Art74(PROVISION_REF→50),
	  LSA_Art110(PENALTY_REF→50), LSA_ED_Art22(Delegation),
	  OSHA, RetireBenefitAct, FixedTermAct, EmpInsuranceAct,
	  VocTrainingAct, DispatchAct, EmpPolicyAct (확장 11건)
	총 23건

3) 통합 파급 쿼리 탐색

1)과 2)과정을 통합하여 입력하는 키워드를 일반화하여 범용적으로 사용할 수 있도록 구성한 프로세스이다.

의미기반 태크 설계 원리	
1단계 (설계 원리)	① Q5(텍스트 REGEC) + Q6(PolicyTag)를 결합 ② 키워드 입력 ③ 키워드 분해 → VALUES 절에 패턴 등록 ④ 6개 UNION (A-F) 각각에서 패턴별 매칭 ⑤ DISTINCT로 중복 제거 ⑥ TIER 별 정렬로 발견 경로 추적

2단계
(쿼리 구성)
	SELECT DISTINCT ?tier ?entity ?title ?entityType ?source ?detail
	WHERE {
	  # ────────────────────────────────────
	  # 사용자 입력 → 키워드 분해 (이 부분만 수정하면 범용 사용)

의미기반 태크 설계 원리	

```
  # ────────────────────────────────────────
  VALUES (?searchPattern ?patternLabel) {
    ("근로시간"          "법정근로시간")
    ("40시간|36시간"      "주당 시간 기준")
    ("소정근로"          "소정근로시간 정의")
                  ~~~~
    ("포괄임금"          "포괄임금제")
  }
  {
    # ── A: Article.title 매칭 ──
    ?entity a ont:Article ;
           ont:title ?title .
    FILTER(REGEX(STR(?title), ?searchPattern, "i"))
                     ~~~~
  UNION
  {
    # ── F: LegalInstrument.title 매칭 (신규 입법 포함) ──
    ?entity a ont:LegalInstrument ;
           ont:title ?title .
    FILTER(REGEX(STR(?title), ?searchPattern, "i"))
    BIND("F_법률명" AS ?tier)
    BIND("LegalInstrument" AS ?entityType)
    BIND("title" AS ?source)
    BIND(?patternLabel AS ?detail)
  }
}ORDER BY ?tier ?entityType ?entity
```

실행 결과

의미기반 태크 설계 원리

Tier별 발견경로 추적 결과 33 results · 0.047s

JSON CSV TSV XML

A_조문제목 **조문제목 매칭** 키워드 패턴이 조문 제목에 직접 매칭 9

발견경로 keyword → title CONTAINS → Article

#	?ENTITY	?TITLE	?ENTITYTYPE	?MATCHEDPATTERN
1	inst:LSA_Art50	"근로시간"@ko	Article	"근로시간"
2	inst:LSA_Art51	"탄력적 근로시간제"@ko	Article	"탄력적 근로"
3	inst:LSA_Art51_2	"탄력적 근로시간제"@ko	Article	"탄력적 근로"
4	inst:LSA_Art52	"선택적 근로시간제"@ko	Article	"선택적 근로"
5	inst:LSA_Art53	"연장 근로의 제한"@ko	Article	"연장 근로"
6	inst:LSA_Art56	"연장·야간 및 휴일근로"@ko	Article	"휴일근로"
7	inst:LSA_Art69	"근로시간(연소자)"@ko	Article	"연소자"
8	inst:LSA_ED_Art6	"통상임금"@ko	Article	"통상임금"
9	inst:LSA_ED_Art9	"단시간근로자 결정기준"@ko	Article	"근로시간"

B_조문본문 **조문본문 매칭** 키워드 패턴이 조문 본문 텍스트에서 발견 1

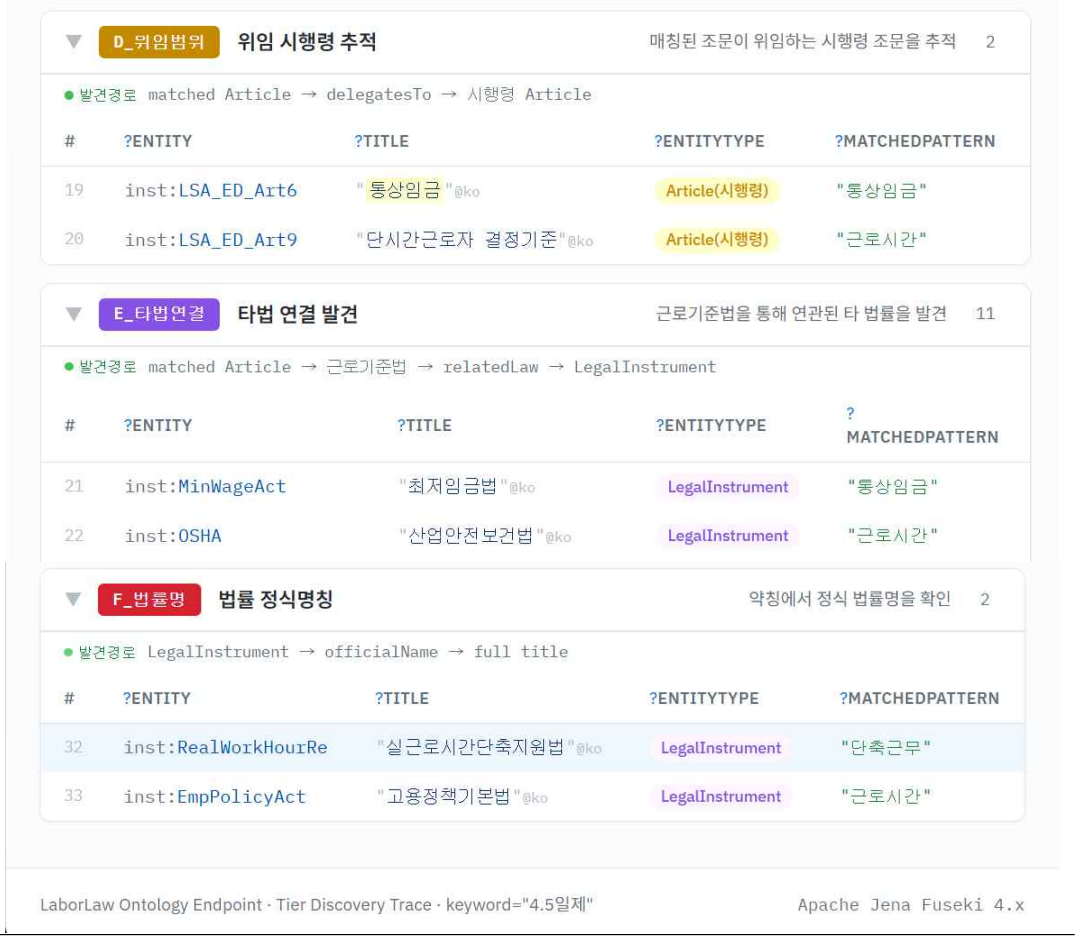
발견경로 keyword → body text CONTAINS → Article

#	?ENTITY	?TITLE	?ENTITYTYPE	?MATCHEDPATTERN
10	inst:LSA_Art50	"근로시간(연소자)"@ko	Article	"근로시간"

C_참조텍스트 **참조텍스트 발견** 본문에서 참조된 다른 조문들을 추적 8

발견경로 matched Article → refersTo → referenced Article

#	?ENTITY	?TITLE	?ENTITYTYPE	?MATCHEDPATTERN
11	inst:LSA_Art2	"정의"@ko	Article	*참조 추적*
12	inst:LSA_Art51	"탄력적 근로시간제"@ko	Article	"탄력적 근로"
13	inst:LSA_Art52	"선택적 근로시간제"@ko	Article	"선택적 근로"
14	inst:LSA_Art53	"연장 근로의 제한"@ko	Article	"연장 근로"
15	inst:LSA_Art56	"연장·야간 및 휴일근로"@ko	Article	"휴일근로"
16	inst:LSA_Art69	"근로시간(연소자)"@ko	Article	"연소자"
17	inst:LSA_Art74	"임산부의 보호"@ko	Article	*참조 추적*
18	inst:LSA_Art110	"벌칙"@ko	Article	*참조 추적*
23	inst:RetireBenefitAct	"퇴직급여보장법"@ko	LegalInstrument	"근로시간"
24	inst:FixedTermAct	"기간제법"@ko	LegalInstrument	"근로시간"
25	inst:EmpInsuranceAct	"고용보험법"@ko	LegalInstrument	"근로시간"
26	inst:VocTrainingAct	"근로자직업능력개발법"@ko	LegalInstrument	"근로시간"
27	inst:GenderEqAct	"남녀고용평등법"@ko	LegalInstrument	"근로시간"

의미기반 태크 설계 원리	
	

D_위임범위 위임 시행령 추적 매칭된 조문이 위임하는 시행령 조문을 추적 2

발견경로 matched Article → delegatesTo → 시행령 Article

#	?ENTITY	?TITLE	?ENTITYTYPE	?MATCHEDPATTERN
19	inst:LSA_ED_Art6	"통상임금"@ko	Article(시행령)	"통상임금"
20	inst:LSA_ED_Art9	"단시간근로자 결정기준"@ko	Article(시행령)	"근로시간"

E_타법연결 타법 연결 발견 근로기준법을 통해 연관된 타 법률을 발견 11

발견경로 matched Article → 근로기준법 → relatedLaw → LegalInstrument

#	?ENTITY	?TITLE	?ENTITYTYPE	?MATCHEDPATTERN
21	inst:MinWageAct	"최저임금법"@ko	LegalInstrument	"통상임금"
22	inst:OSHA	"산업안전보건법"@ko	LegalInstrument	"근로시간"

F_법률명 법률 정식명칭 약칭에서 정식 법률명을 확인 2

발견경로 LegalInstrument → officialName → full title

#	?ENTITY	?TITLE	?ENTITYTYPE	?MATCHEDPATTERN
32	inst:RealWorkHourRe	"실근로시간단축지원법"@ko	LegalInstrument	"단축근무"
33	inst:EmpPolicyAct	"고용정책기본법"@ko	LegalInstrument	"근로시간"

LaborLaw Ontology Endpoint · Tier Discovery Trace · keyword="4.5일제" Apache Jena Fuseki 4.x

2) 법령온톨로지 검증 테스트 결과

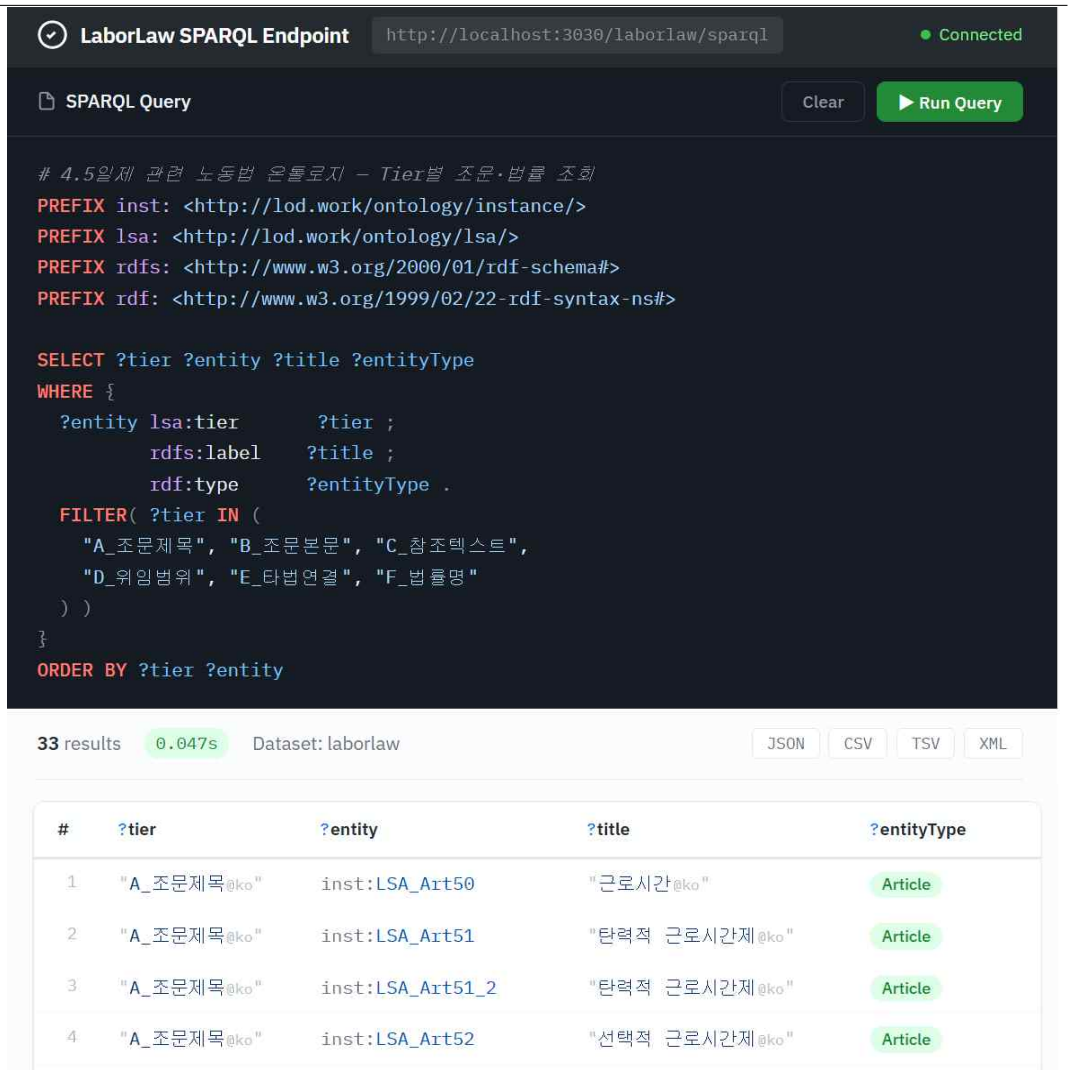
가) 통합 파급 쿼리 탐색

추론된 법령 조를 기준으로 각 TIER별로 발견된 객체를 전체적으로 추론

의미기반 태크 설계 원리	
1단계 (설계 원리)	① Q5(텍스트 REGEC) + Q6(PolicyTag)를 결합 ② 키워드 입력 ③ 키워드 분해 → VALUES 절에 패턴 등록 ④ 6개 UNION (A-F) 각각에서 패턴별 매칭 ⑤ DISTINCT로 중복 제거 ⑥ TIER 별 정렬로 발견 경로 추적

2단계
(쿼리 구성)
	SELECT DISTINCT ?tier ?entity ?title ?entityType ?source ?detail
	WHERE {
	  # ────────────────────────────────
	  # 사용자 입력 → 키워드 분해 (이 부분만 수정하면 범용 사용)

의미기반 태크 설계 원리	

	``
	  # ────────────────────────────────────
	  VALUES (?searchPattern ?patternLabel) {
	  ("근로시간" "법정근로시간")
	  ("40시간|36시간" "주당 시간 기준")
	  ("소정근로" "소정근로시간 정의")
	  ~~
	  ("포괄임금" "포괄임금제")
	  }
	  {
	  # ── A: Article.title 매칭 ──
	  ?entity a ont:Article ;
	  ont:title ?title .
	  FILTER(REGEX(STR(?title), ?searchPattern, "i"))
	  ~~
	  BIND("F_법률명" AS ?tier)
	  BIND("LegalInstrument" AS ?entityType)
	  BIND("title" AS ?source)
	  BIND(?patternLabel AS ?detail)
	  }
	}ORDER BY ?tier ?entityType ?entity
	``
	 

실행 결과	

의미기반 태크 설계 원리

5	"A_조문제목@ko"	inst:LSA_Art53	"연장 근로의 제한@ko"	Article
6	"A_조문제목@ko"	inst:LSA_Art56	"연장·야간 및 휴일근로@ko"	Article
7	"A_조문제목@ko"	inst:LSA_Art69	"근로시간(연소자)@ko"	Article
8	"A_조문제목@ko"	inst:LSA_ED_Art6	"통상임금@ko"	Article
9	"A_조문제목@ko"	inst:LSA_ED_Art9	"단시간근로자 결정기준@ko"	Article
10	"B_조문본문@ko"	inst:LSA_Art50	"근로시간(연소자)@ko"	Article
11	"C_참조텍스트@ko"	inst:LSA_Art2	"정의@ko"	Article
12	"C_참조텍스트@ko"	inst:LSA_Art51	"탄력적 근로시간제@ko"	Article
13	"C_참조텍스트@ko"	inst:LSA_Art52	"선택적 근로시간제@ko"	Article
14	"C_참조텍스트@ko"	inst:LSA_Art53	"연장 근로의 제한@ko"	Article
15	"C_참조텍스트@ko"	inst:LSA_Art56	"연장·야간 및 휴일근로@ko"	Article
16	"C_참조텍스트@ko"	inst:LSA_Art69	"근로시간(연소자)@ko"	Article
17	"C_참조텍스트@ko"	inst:LSA_Art74	"임산부의 보호@ko"	Article
18	"C_참조텍스트@ko"	inst:LSA_Art110	"벌칙@ko"	Article
19	"D_위임범위@ko"	inst:LSA_ED_Art6	"통상임금@ko"	Article(시행령)
20	"D_위임범위@ko"	inst:LSA_ED_Art9	"단시간근로자 결정기준@ko"	Article(시행령)
21	"E_타법연결@ko"	inst:MinWageAct	"최저임금법@ko"	LegalInstrument
22	"E_타법연결@ko"	inst:OSHA	"산업안전보건법@ko"	LegalInstrument
23	"E_타법연결@ko"	inst:RetireBenefitAct	"산업안전보건법@ko"	LegalInstrument
24	"E_타법연결@ko"	inst:FixedTermAct	"기간제법@ko"	LegalInstrument
25	"E_타법연결@ko"	inst:EmpInsuranceAct	"고용보험법@ko"	LegalInstrument
26	"E_타법연결@ko"	inst:VocTrainingAct	"근로자직업능력개발법@ko"	LegalInstrument
27	"E_타법연결@ko"	inst:GenderEqAct	"남녀고용평등법@ko"	LegalInstrument
28	"E_타법연결@ko"	inst:DispatchqAct	"파견근로자보호법@ko"	LegalInstrument
29	"E_타법연결@ko"	inst:RealWorkHourRe	"파견근로자보호법@ko"	LegalInstrument
30	"E_타법연결@ko"	inst:RealWorkHour	"실근로시간단축지원법@ko"	LegalInstrument
31	"E_타법연결@ko"	inst:EmpPolicyAc	"고용정책기본법@ko"	LegalInstrument

2. 법령 온톨로지의 핵심 역할 및 성과

가. 핵심 역할 개요

본 검증은 앞에 기술한 바와 같이 “주4.5일제”를 중심 키워드로 하여 법령 온톨로지 기반 추론을 수행한 결과, 단순 키워드 검색을 넘어 **법률-시행령-타법**까지 연결된 영향 범위를 네트워크 형태로 구조적으로 탐지할 수 있음을 확인하였다.

특히 본 결과에서는 네트워크 상에서 다음과 같은 구조가 확인되었다.

• 노드 유형 5종과 연결 유형 4종으로 구성되어 핵심 키워더를 중심으로 근로기준법 조문, 시행령 조문, 타법과 각 Tier 허브를 기준으로 조문 소속, 위임관계, 타법 연결 사항을 확인할 수 있다.

• Tier 허브 6개(A~F)를 통해 추론 경로를 구분하여 법령의 개정과 관련성의 원천을 추적할 수 있는 근거를 제시해주었다.

• 연관 대상 법령으로 조문/시행령 조문 12건과 타법 11건을 확인할 수 있었다.

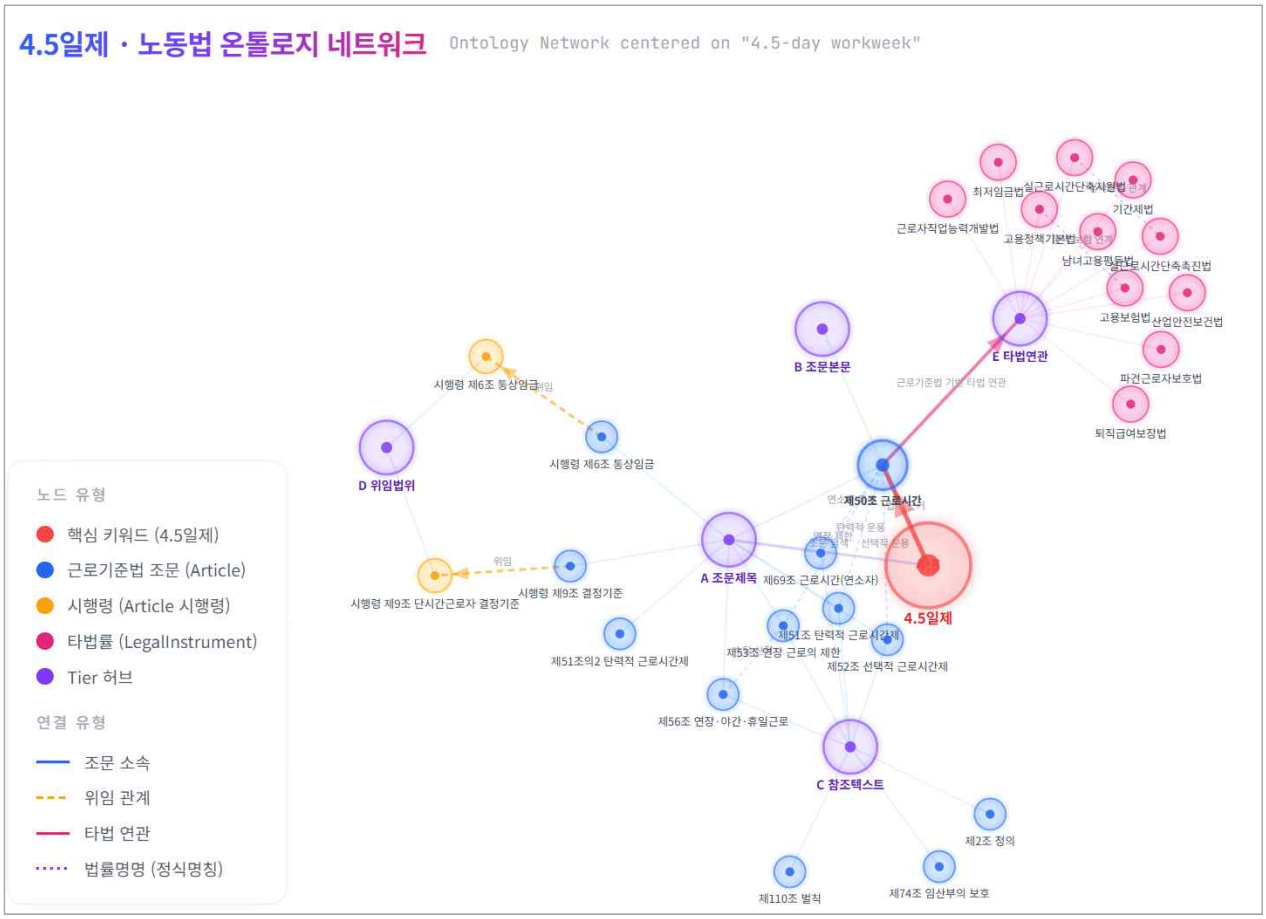
즉, 하나의 검색 키워드에서 출발해 직접 관련 조문뿐 아니라 시행령 및 타법까지 연결된 영향 후보를 한 번에 추론할 수 있음을 보였으며, 이는 기존 법령검색 방식의 한계 (문서 단위 검색, 수작업 확인, 누락 가능성)를 실질적으로 보완할 수 있는 근거가 된다.

나. 핵심 성과

• 탐지 범위의 확장성: 기존 방식은 보통 “키워드가 직접 포함된 조문”중심으로 결과가 제한되지만, 본 방식은 다음 경로를 통해 탐지 범위를 확장한다. 이런 구조는 결과적으로 “관련 법을 다 찾아야 하는 문제”를 관계 기반 추론으로 전환했다는 점에서 의미가 크다. 즉, 담당자가 법률.시행령.타법을 일일이 손으로 찾는 방식이 아니라, 온톨로지 관계를 따라 연관 법령 후보를 체계적으로 전수(후보군) 추론할 수 있다.

• 법률 개정의 영향 식별: 네트워크 결과는 단순히 근로기준법 조문만 보여주는 것이 아니라 직접영향(근로시간, 탄력근로, 선택근로, 연장근로 등)을 받는 핵심조문과 간접영향(시행령, 타법 등)을 함께 제시하기 때문에 실제 개정 검토에서 중요한 “빠트리기 쉬운 간접 영향”을 사전에 후보와 한다는 점에

서 효과가 기대된다.



(그림4-3) 주 4.5일제 관련 법령 온톨로지 기반 연관 법령 추론 네트워크

다. 기대 효과

- 법령 개정 영향 분석의 전수성(coverage) 강화

법령 온톨로지를 적용하면 개정 조문 기준으로 연결된 관계(위계.위임.참조.준용.타법연결)를 따라 관련 법령을 전수에 가깝게 추론 후보화할 수 있다. 특히 기존 검색에서 누락되기 쉬운 시행령.시행규칙.타법까지 함께 탐지 가능하다는 점이 가장 큰 장점이다.

- 검토 시간 단축 및 반복 작업 감소

기존에는 담당자가 개정 시점마다 (1) 관련 조문 찾기, (2) 시행령 확인, (3) 타법 영향 확인, (4) 절차/서식 확인을 반복했으나, 온톨로지 기반 체계에서는 이를 초기 후보 목록으로 자동 생성할 수 있어 검토 시간을 줄일 수 있다.

- 검토 품질의 표준화

담당자 숙련도에 따라 결과 편차가 발생하던 문제를 줄이고 동일한 입력(개

정 조문/키워드)에 대해 동일한 탐지 경로와 기준을 제공할 수 있어 조직 차원의 재현성과 일관성이 높아진다.

- 향후 자동화 기능으로 확장 검토 품질의 표준화

현재는 법령 중심 추론이지만, 향후에는 '업무절차', '행정서식', '민원안내문/지침', '내부 업무매뉴얼' 까지 연결하면 개정 영향분석을 법령 수준을 넘어 업무 운영 영향 분석으로 확장될 수 있다.

라. 법령 온톨로지 기반 개정 영향 탐지 검증 결과 및 시사점

본 검증은 앞에 기술한 바와 같이 "주4.5일제"를 중심 키워드로 하여 법령 온톨로지 기반 추론을 수행한 결과, 단순 키워드 검색을 넘어 **법률-시행령-타법**까지 연결된 영향 범위를 네트워크 형태로 구조적으로 탐지할 수 있음을 확인하였다.

검증 결과, 통합 파급 쿼리에서는 근로기준법의 직접 관련 조문(예: 근로시간, 탄력적·선택적 근로시간제, 연장근로 관련 조문), 시행령 조문, 참조 조문, 위임 범위 조문뿐 아니라 최저임금법, 산업안전보건법, 기간제법, 고용보험법, 남녀고용평등법 등 타법까지 연계하여 추론할 수 있었고, 문서에 제시된 결과 기준으로 총 23건의 연관 대상을 식별하였다. 이는 법령 개정 영향 탐지에서 직접 영향뿐 아니라 간접 영향(타법 연결, 위임·참조 관계)을 함께 제시할 수 있음을 보여준다.

종합하면, 법령 온톨로지 기반 접근은 기존의 키워드 중심 법령 검색을 대체하기보다, 규범 위계·타법 인용·준용·내부 참조·위임 범위를 구조적으로 탐지하는 방식으로 보완함으로써 개정 영향 분석의 정확성과 효율성을 높일 수 있는 가능성을 확인하였다. 특히 개정 영향 대상의 후보를 체계적으로 제시함으로써 담당자의 검토 범위를 줄이고, 누락 위험을 낮추는 데 실질적인 효과가 있을 것으로 판단된다.

다만 본 검증은 '주4.5일제' 사례 중심의 검증으로 수행되었으므로, 향후에는 다양한 정책·개정 시나리오에 대한 추가 검증과 함께, 탐지 정확도(정밀도/재현율), 검토 시간 단축 효과, 누락 감소율 등 정량적 성과지표를 포함한 후속 검증이 필요하다. 이를 통해 법령 개정 영향 탐지 체계의 실무 적용 가능성을 보다 객관적으로 입증할 수 있을 것이다.

제2절. (노동)법령 온톨로지 기반 규범 명제 추론 검증

1. 규범명제 기반 법적 판단 자동도출 개요

가. 근로기준법의 적용 가능성 탐색

본 절에서는 법령 온톨로지에 구조화된 규범명제(요건→효과)를 실제 사업장 사실관계에 적용하여, 법적 판단을 자동으로 도출할 수 있는지를 검증하였다. 본 검증의 핵심은 단순히 법조문을 검색하는 것이 아니라, 사업장의 구체적 사실(상황)을 ABox 형태로 입력하고, 이를 규범명제 규칙에 따라 전방추론(Forward Chaining) 방식으로 순차 적용함으로써 위반 여부, 제재 가능성, 감독 대상 여부까지 단계적으로 도출하는 데 있다. 즉, 본 절은 법령 온톨로지의 활용이 **"검색 보조"**를 넘어 **"법적 판단 지원"**으로 확장될 수 있음을 실증적으로 제시하는 검증 사례에 해당한다.

나. 검증 목적 및 시나리오

본 검증의 목적은 특정 사업장의 근로 사실관계가 주어졌을 때, 법령 온톨로지에 구조화된 규범명제를 전방추론 방식으로 적용하여 적용 가능한 법 위반 및 후속 법적 효과를 자동으로 탐지할 수 있는지를 확인하는 데 있다. 검증 시나리오는 "한 사업장의 근로 사실관계가 주어졌을 때, 적용 가능한 모든 근로기준법 위반을 탐색"하는 것으로 설정하였다.

다. 법적 판단 자동도출 프로세스

1) 입력 사실 구성

입력은 사업장 규모, 근로시간, 임금 지급 수준, 서면계약 여부, 최저임금 기준, 근로자대표와의 서면합의 여부 등 총 8개의 사실관계(ABox)로 구성하였다. 이들 사실은 개별적으로 존재하는 정보가 아니라, 추론 과정에서 결합되어 법적 요건 충족 여부를 판단하는 근거로 사용된다. 예를 들어, 주 근로시간 정보와 서면합의 사실은 연장근로 적법성 판단에 결합되며, 월급여·소정근로시간·최저임금 정보는 최저임금 위반 여부 판단에 함께 사용된다.

2) 규범명제(규칙) 구성

규칙은 근로기준법 및 관련 규정의 규범명제를 요건(If)-효과(Then) 형태로 구조화한 것으로, 문서에서는 총 8개의 규칙(R1~R8)을 제시하였다. 규칙은 크게 (1) 위반 여부 판단 규칙, (2) 적법성 판단 규칙, (3) 벌칙·제재 도출 규칙, (4) 복합 위반에 따른 행정감독 대상 판단 규칙으로 구분할 수 있다. 이러한 구성은 법조문 텍스트를 단순 저장하는 수준을 넘어, 법적 판단 로직을 기계가 적용 가능한 형태로 표현했다는 점에서 의미가 있다

3) 전방추론 수행 절차

전방추론은 입력된 사실(ABox)에서 시작하여, 규칙의 요건을 충족할 때마다 새로운 결론을 생성하는 방식으로 수행하였다. 문서에서는 총 8개 사이클(Cycle 1~8)로 추론 과정을 제시하고 있으며, 이 과정에서 사실관계의 결합과 규범명제의 적용이 단계적으로 진행된다.

- **사실관계 → 위반 판단**: 서면 근로계약서 미작성 사실로부터 제17조 위반 도출
- **사실관계 결합 → 적법성 판단**: 주 52시간 근로 사실과 서면합의 존재로 연장근로 적법 판단
- **사실관계 결합 → 임금 관련 위반 판단**: 연장근로 가산율 부족으로 제56조 위반 도출
- **수치 연산 포함 판단**: 월급여/소정근로시간 계산을 통해 최저임금 위반 도출
- **파생 결론 → 제재 도출**: 위반 결론을 근거로 벌금/징역 가능성 도출
- **복합 판단 → 감독 대상 도출**: 복수 위반 누적 결과를 바탕으로 특별근로감독 대상 도출

이와 같이 전방추론은 개별 사실을 순차적으로 결합하면서 위반, 제재, 행정조치까지 단계적으로 확장하는 구조로 구성된다.

2. 규범명제 기반 법적 판단 자동도출 구현

가. 법적 판단 자동도출 구현 개요

법령 온톨로지에 구조화된 규범명제(요건→효과)를 기반으로, 사업장 사실관계에 대한 법적 판단을 자동으로 도출하는 전방추론 엔진을 구현하고 그

동작 결과를 확인하였다. 구현 시스템은 사실 입력(Facts), 규칙 자동 구성, 전방추론 실행, 최종 결론 도출의 단계로 구성되며, 입력된 사실관계(ABox)와 규범명제 규칙을 결합하여 위반 여부, 적법성, 제재 가능성 등의 판단 결과를 순차적으로 생성하도록 설계하였다. 본 구현을 통해 규범명제 기반 법적 판단 구조가 단순 개념 설계 수준을 넘어 실제 실행 가능한 형태로 구현될 수 있음을 확인하였으며, 추론 과정과 결과를 화면 기반으로 검증함으로써 법령 온톨로지의 실무 적용 가능성을 제시하였다.

나. 법적 판단 자동도출 구현 범위

구현 범위는 근로기준법 위반 탐색 시나리오를 대상으로 설정하였으며, 사업장 사실관계 8건을 입력 사실로 구성하고, 이를 처리하는 규범명제 규칙 8개를 적용하여 전방추론을 수행하도록 구현하였다. 입력 사실에는 사업장 규모, 주 근로시간, 연장근로 가산수당 지급수준, 서면 근로계약서 작성 여부, 월 급여, 최저임금 기준, 월 소정근로시간, 근로자대표와의 서면합의 여부 등 법적 판단에 필요한 핵심 요소를 포함하였다.

시스템은 이들 사실을 Working Memory에 적재한 후 규칙 매칭과 추론 사이클을 반복 수행하며, 중간 결론을 누적하여 최종적으로 위반 여부, 적법성 판단, 제재 가능성, 감독 대상 여부 등의 결론을 도출하도록 구성하였다. 본 사례에서는 전방추론 사이클을 통해 총 6개의 최종 결론이 생성됨을 확인하였으며, 이를 통해 사실관계의 종합적 결합에 기반한 법적 판단 자동도출이 구현 수준에서도 유효하게 작동함을 검증하였다.

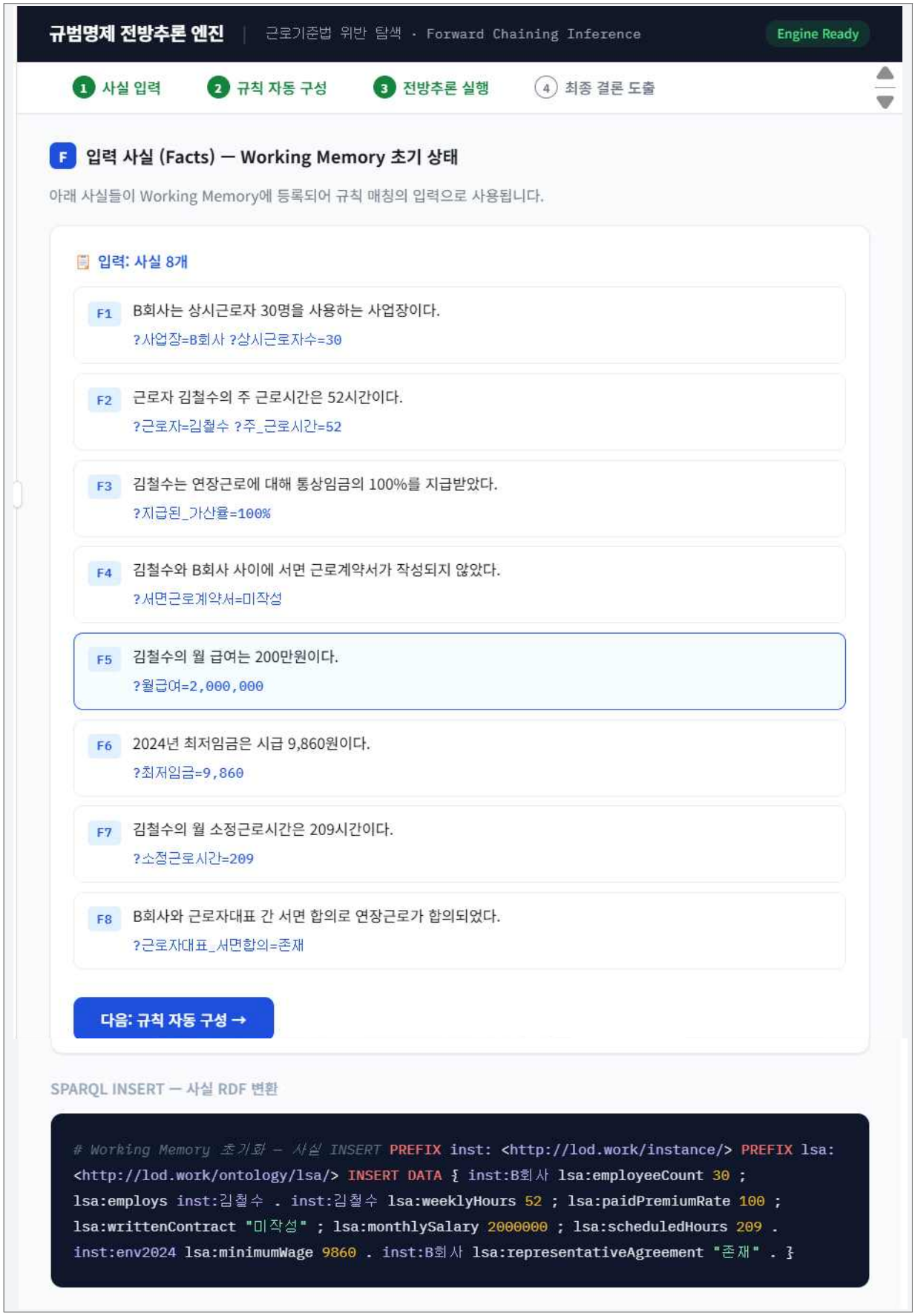
다. 법적 판단 자동도출 구현 화면

법적 판단 자동 도출 구현 화면으로, 추론 절차를 사실 입력 → 규칙 자동 구성 → 전방추론 실행 → 최종 결론 도출의 4단계로 구분하여 제시하였다.

1) 사실입력

사실입력 화면은 Working Memory 초기 상태를 보여주며, 사업장 규모・근로시간・임금・계약서 작성 여부・최저임금・서면합의 등 법적 판단에 필요한 입력 사실 8건이 구조화되어 등록된 상태이다.

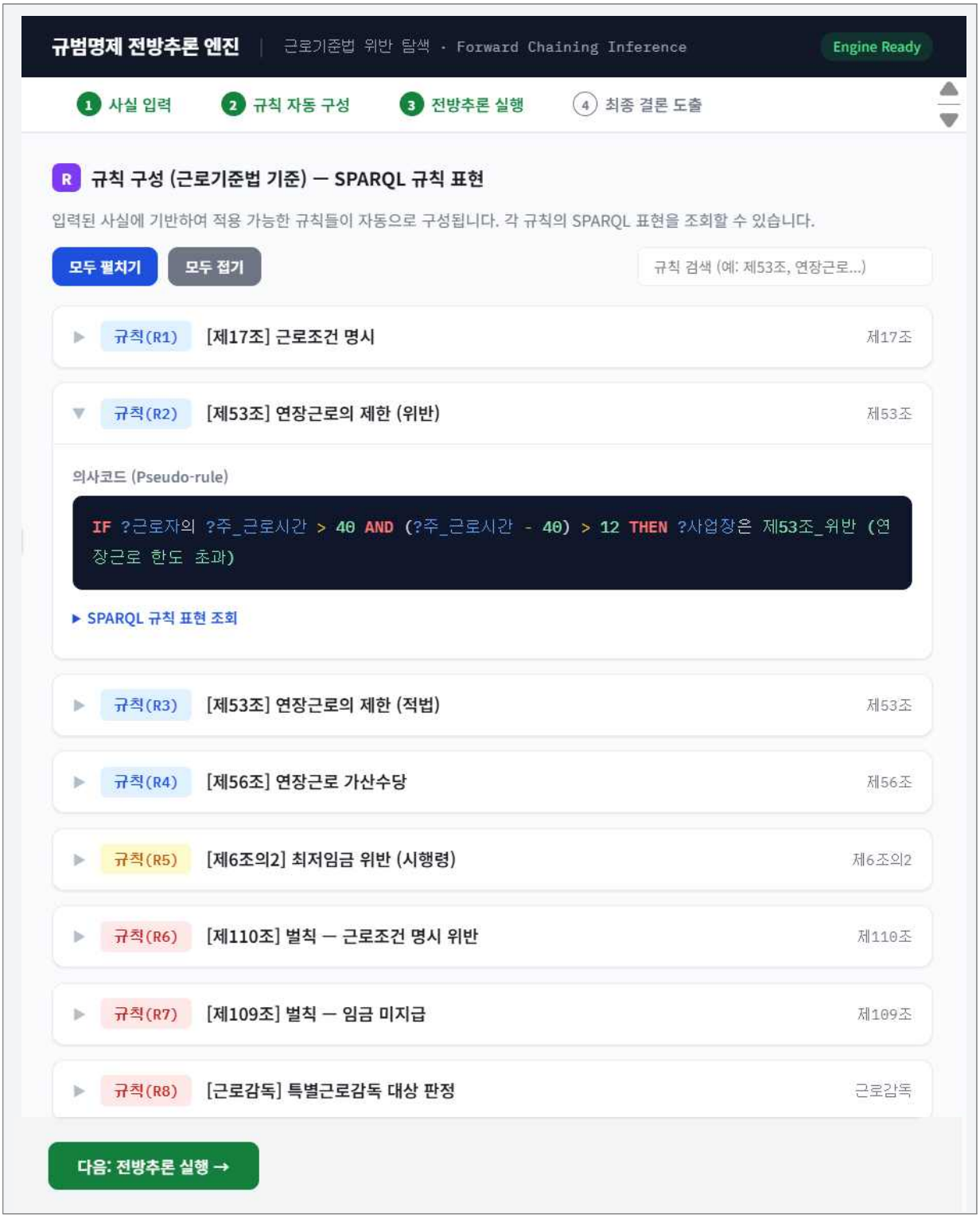
각 입력 사실은 이후 규칙 매칭의 기초 데이터로 사용되며, 하단의 단계 이동 버튼을 통해 규칙 구성 및 추론 실행 결과를 순차적으로 확인할 수 있도록 구성하였다.



(그림4-4) 근로기준법 위반 탐색을 위한 입력 사실 등록 화면

2) 규칙 자동 구성

규칙 자동 구성 화면으로, 입력된 사실관계를 바탕으로 적용 가능한 근로기준법 규칙들을 자동으로 선별·구성한 결과를 보여준다.



(그림4-5) 근로기준법 위반 탐색을 위한 규칙 자동 구성 화면

각 규칙은 조문 기준(예: 제17조, 제53조, 제56조, 최저임금, 벌칙, 근로감독)

으로 목록화되어 있으며, 선택 시 의사코드(Pseudo-rule) 및 SPARQL 규칙 표현을 확인할 수 있도록 구성하였다.

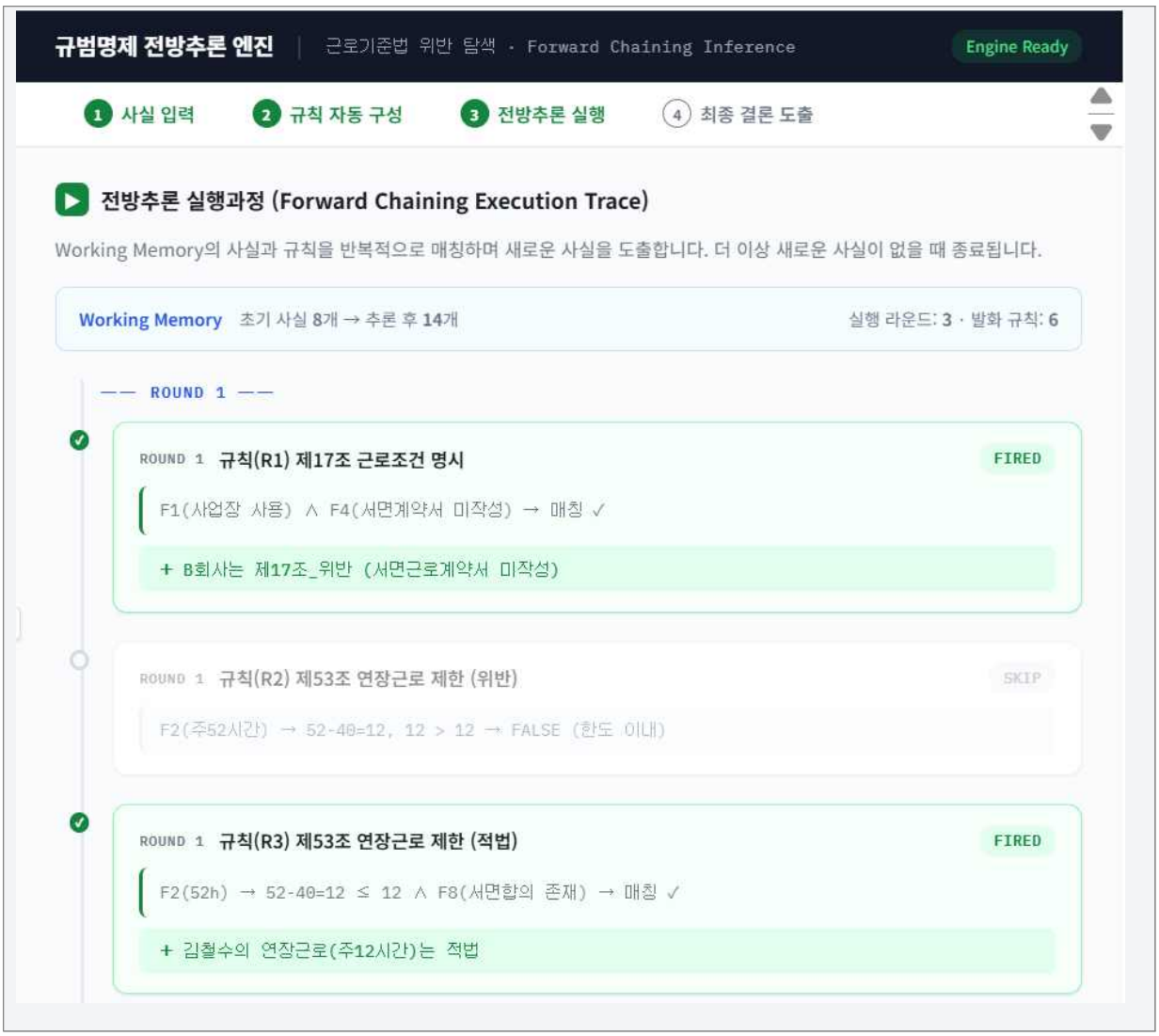
이를 통해 사용자는 추론 실행 전, 어떤 법적 판단 규칙이 적용되는지와 그 논리 구조를 사전에 검토할 수 있다.

3) 전방추론 실행

전방추론 실행과정(Execution Trace) 화면으로, 입력 사실과 규칙의 매칭 결과를 라운드(Round) 단위로 순차적으로 보여준다.

각 규칙에 대해 발화(FIRED)/미발화(SKIP) 여부와 매칭 근거(예: 근로시간, 서면합의, 임금·최저임금 비교)가 함께 표시되며, 규칙이 발화될 때마다 Working Memory에 새로운 결론이 추가되는 과정을 추적할 수 있다.

본 화면에서는 제17조 위반, 제56조 위반, 최저임금 위반, 벌칙 적용, 특별근로감독 대상 판정까지 단계적으로 도출되는 흐름을 확인할 수 있다.



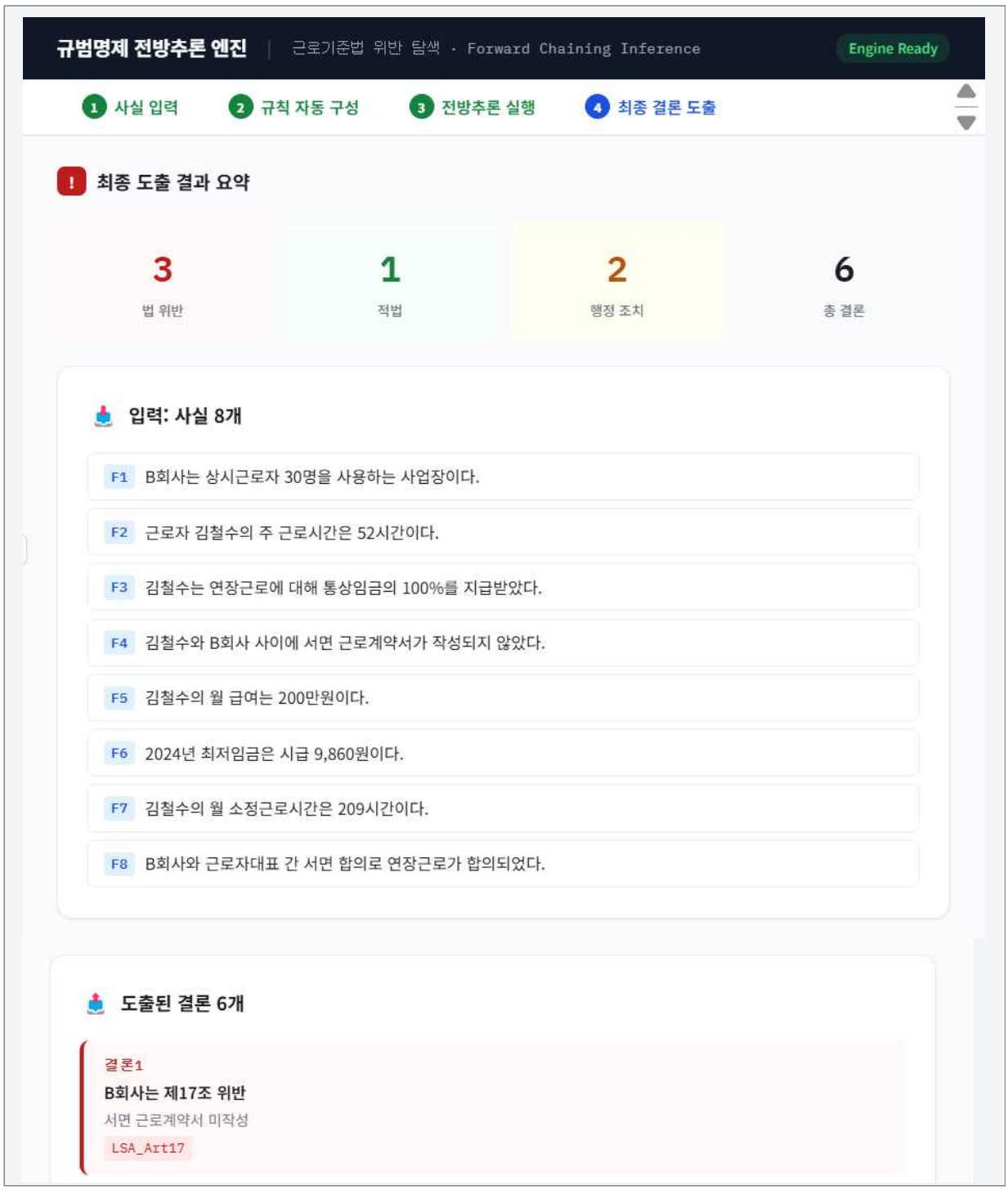
(그림4-6) 근로기준법 위반 탐색을 위한 전방추론 실행 화면

4) 전방추론 최종 결론 도출

규범명제 전방추론 엔진의 최종 결론 도출 화면으로, 입력 사실 8건과 추론 결과를 종합하여 위반・적법・행정조치 결과를 한눈에 요약해 제시한다.

상단에는 결과 유형별 집계(법 위반 3건, 적법 1건, 행정조치 2건, 총 6건)를 표시하고, 하단에는 각 결론별로 적용 조문·판단 근거(예: 제17조 위반, 제56조 위반, 최저임금 위반, 벌칙 가능, 특별근로감독 대상)를 카드 형태로 정리하였다.

이를 통해 사용자는 전방추론의 최종 판단 결과와 법적 근거를 직관적으로 확인할 수 있으며, 개별 결론의 검토 및 후속 조치 판단에 활용할 수 있다.



(그림4-7) 근로기준법 위반 탐색을 위한 최종 결론도출 화면

바. 구현 결과

본 구현은 규범명제(요건→효과) 기반 법적 판단 자동도출 구조를 실제 화면과 실행 절차로 구현하고, 그 결과를 단계별로 검증했다는 점에서 의미가 크다. 구현 화면은 ① 사실 입력(Facts) → ② 규칙 자동 구성 → ③ 전방추론 실행(Execution Trace) → ④ 최종 결론 도출의 4단계로 구성되어, 법적 판단 과정의 전 흐름을 사용자가 순차적으로 확인할 수 있도록 설계되었다.

특히 본 사례에서는 입력 사실 8건을 Working Memory에 적재하고, 이에 대응하는 규칙 8개를 자동 구성한 뒤, 전방추론을 수행하여 총 3개 라운드, 발화 규칙 6개, 최종 결론 6건을 도출하였다. 최종 결과는 법 위반 3건, 적법 1건, 행정조치 2건으로 요약되며, 각 결론에 대해 적용 조문과 판단 근거가 함께 제시되어 결과의 해석 가능성과 검토 편의성을 높였다.

또한 전방추론 실행 화면에서 각 규칙의 발화(FIRED)/미발화(SKIP) 여부와

매칭 근거가 함께 제시됨으로써, 단순 결과 제시에 그치지 않고 "왜 해당 결론이 도출되었는지"를 추적할 수 있는 구조를 구현하였다. 이는 규범명제 기반 법적 판단 자동화에서 중요한 투명성, 설명가능성, 검증가능성을 확보한 구현 형태로 평가할 수 있다.

3. 규범명제 기반 후방추론을 통한 적법성 판단 구현

가. 후방추론 기반 적법성 판단 개요

본 절에서는 규범명제(요건→효과)를 후방추론(Backward Chaining) 방식으로 적용하여, 특정 사업장 사실관계(ABox)에 대한 법적 판단을 목표 지향적으로 도출하는 구현 내용을 제시한다. 후방추론은 "어떤 법적 결론이 성립하는가"를 먼저 목표(goal)로 설정한 뒤, 해당 결론을 성립시키기 위해 필요한 요건 사실들을 역으로 추적·검증하는 방식이다. 따라서 본 구현은 다수의 규칙을 일괄적으로 발화시키는 전방추론과 달리, 질의 중심(Goal-driven) 법적 판단에 적합한 구조를 갖는다.

나. 후방 추론의 특징

후방추론은 규범명제 기반 법적 판단에서 다음과 같은 특징과 장점을 가진다.

1) 목표 지향성(Goal-driven)

후방추론은 "해고가 적법한가?", "해당 행위가 위반인가?"와 같이 명확한 법적 질의가 주어졌을 때 효율적이다. 결론에서 출발하여 필요한 요건만 추적하므로, 모든 규칙을 동시에 평가할 필요가 없다.

2) 필요 사실 중심 검증

전방추론이 입력된 사실 전반에서 가능한 결론을 폭넓게 생성하는 방식이라면, 후방추론은 특정 결론을 입증(또는 반증)하는 데 필요한 사실만 확인한다. 따라서 실무 질의응답형 시스템, 법률상담 보조, 적법성 점검과 같은 목적에 적합하다.

3) 법적 판단 구조와의 정합성

법률 실무에서는 통상 "이 결론이 성립하려면 어떤 요건이 필요한가?"라는 방식으로 판단한다. 후방추론은 이러한 법적 검토 방식과 동일하게, 결론

→ 요건 → 세부 사실의 구조로 추론을 전개하므로 규범명제의 논리 구조와 잘 부합한다.

4) 설명가능성(Explainability)

후방추론은 목표를 어떤 규칙으로 분해했고, 어떤 하위요건이 충족/미충족 되었는지를 추적할 수 있어 판단 근거를 설명하기 용이하다. 특히 법적 판단 시스템에서는 "왜 적법/위법으로 판단되었는지"를 제시해야 하므로 중요한 특성이다.

5) 누락 사실 식별에 유리

후방추론 과정에서 특정 하위요건을 검증할 사실이 없으면, 해당 요건이 "미확인" 상태로 남는다. 이는 곧 추가 확인이 필요한 사실(질문 항목) 을 자동으로 식별하는 기능으로 활용될 수 있다.

다. 검증 시나리오

검증 시나리오는 "B회사가 김철수를 해고하려고 할 때, 해당 해고가 적법한가"라는 질문을 목표로 설정하고, 이에 필요한 규범명제를 순차적으로 적용하여 적법 여부를 판단하는 구조로 구성하였다. 즉, 법적 결론(해고의 적법/위법)을 먼저 설정한 후, 이를 판단하는 핵심 요건(해고 사유의 정당성, 절차의 적법성, 보호대상 여부, 예외사유 해당 여부 등)을 하위 목표로 분해하고, 입력 사실관계가 각 요건을 충족하는지 역방향으로 확인하도록 설계하였다.

4. 후방추론 기반 적법성 판단 구현 화면

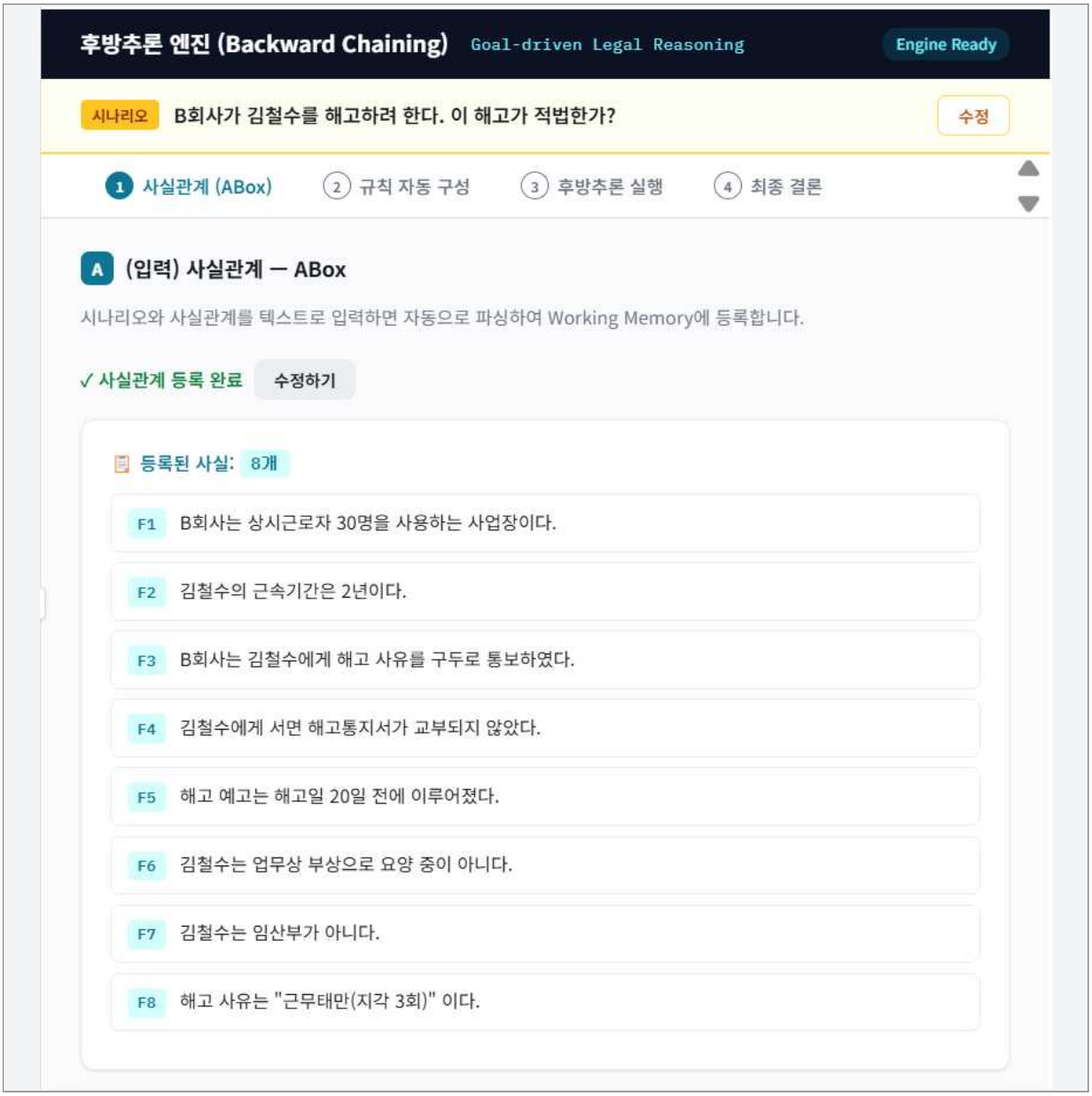
가. 적법성 자동 판단 구현 개요

본 구현에서는 해고 적법성 판단을 최상위 목표로 설정하고, 이를 규범명제 기반 규칙에 따라 하위요건(정당사유, 절차적 요건, 보호사유, 예외사유)으로 분해하는 후방추론 구조를 적용하였다. 엔진은 각 하위요건을 충족하는지 여부를 입력 사실관계(ABox)에서 확인하고, 필요한 경우 추가 확인이 필요한 사실 항목을 식별하며, 최종적으로 해고의 적법성 여부 또는 위법 가능성을 판단 근거와 함께 제시하도록 설계하였다. 이를 통해 규범명제 기반 법적 판단 자동도출이 단순 결과 산출을 넘어, 요건 중심의 검토 절차와 설명 가능한 판단 흐름으로 구현될 수 있음을 확인하였다.

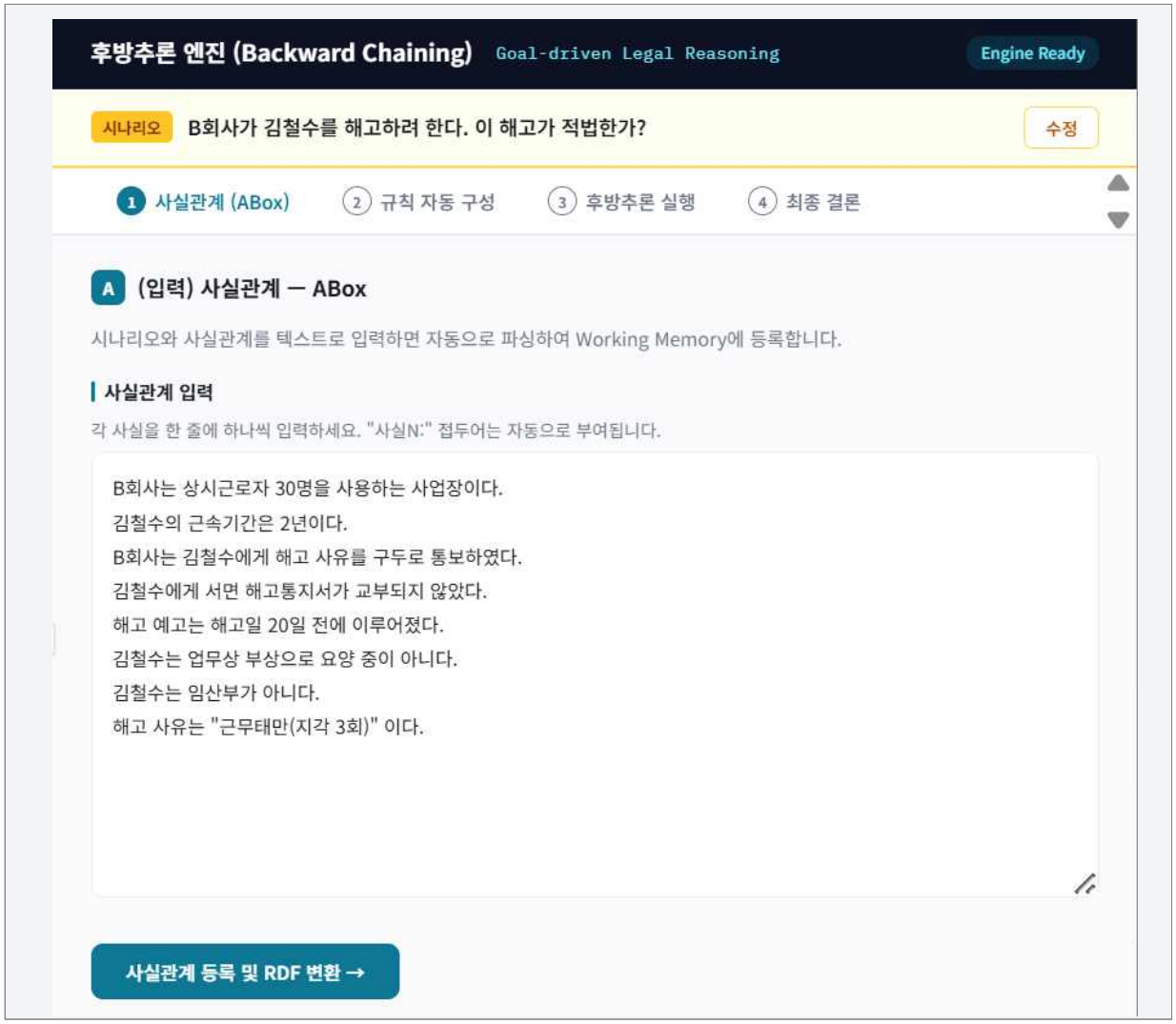
1) 시나리오 입력 및 사실관계 등록

후방추론 엔진의 사실관계(ABox) 입력 화면으로, 해고 적법성 판단 시나리오에 필요한 사실관계를 입력·관리하는 단계이다. 첫 번째 화면에서는 등록된 사실 8건이 Working Memory에 구조화되어 저장된 상태를 보여주며, 시나리오 기반 후방추론의 입력값으로 사용된다.

또한 '수정하기' 버튼을 통해 두 번째 화면으로 전환하여 사실관계를 자유롭게 수정·추가할 수 있도록 하였고, 수정된 내용은 다시 등록(RDF 변환)되어 후속 규칙 구성 및 후방추론 실행에 반영되도록 구성하였다.



(그림4-8) 적법성 판단을 위한 시나리오 및 사실관계 입력 화면



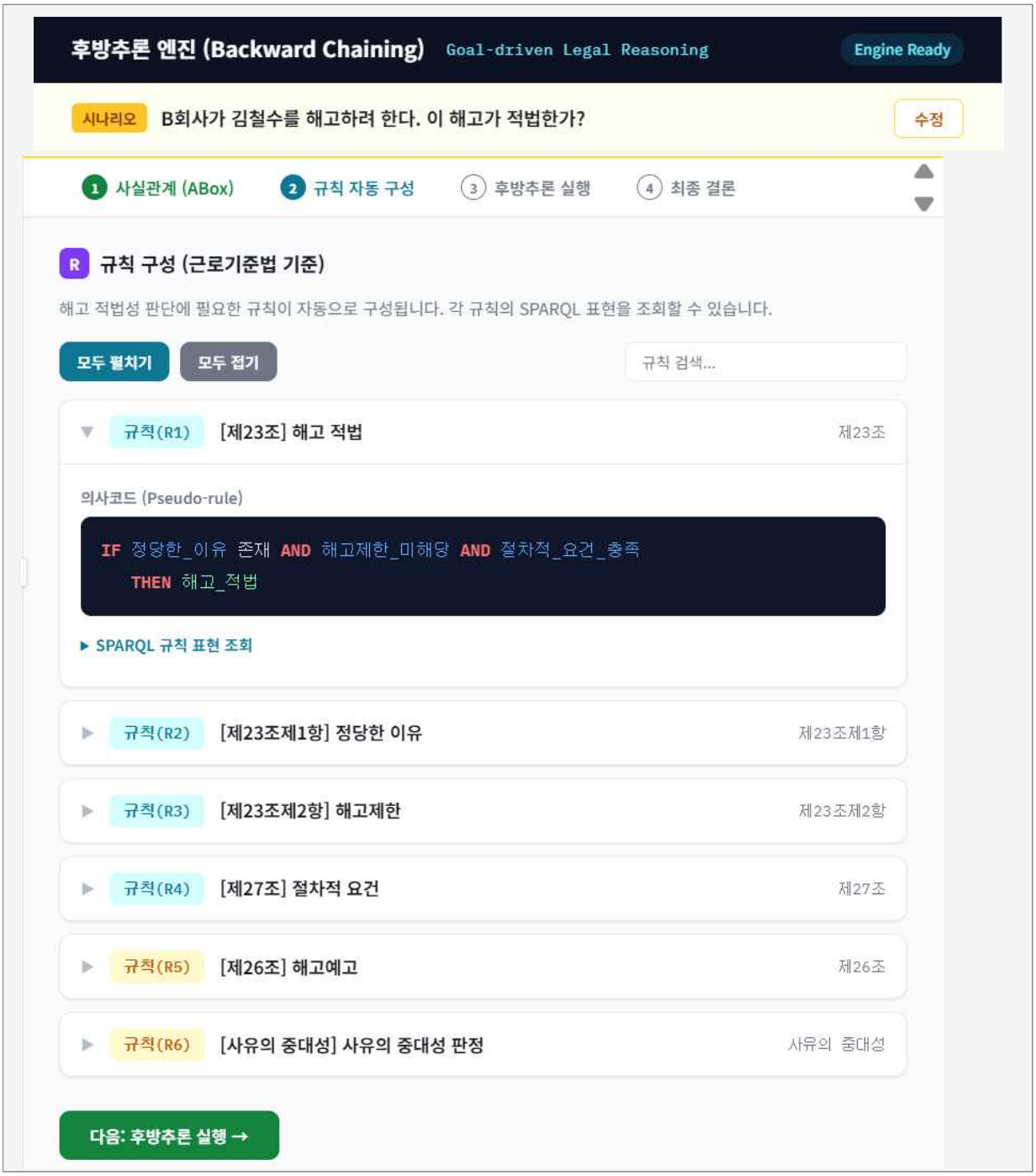
(그림4-9) 적법성 판단을 위한 시나리오 및 사실관계 입력 화면

2) 규칙 자동 구성

후방추론 엔진의 규칙 자동 구성 화면으로, 해고 적법성 판단에 필요한 규범명제 규칙(근로기준법 기준)을 자동으로 선별·구성한 결과를 보여준다. 각 규칙은 조문 단위(예: 제23조 해고 적법, 제23조제1항 정당한 이유, 제23조제2

항 해고제한, 제27조 절차적 요건)로 목록화되며, 펼치기 기능을 통해 의사코드(Pseudo-rule) 와 SPARQL 규칙 표현을 확인할 수 있도록 구성하였다.

이를 통해 사용자는 후방추론 실행 전에 최상위 목표(해고 적법)와 이를 구성하는 하위 요건 규칙의 논리 구조를 사전에 검토할 수 있다.



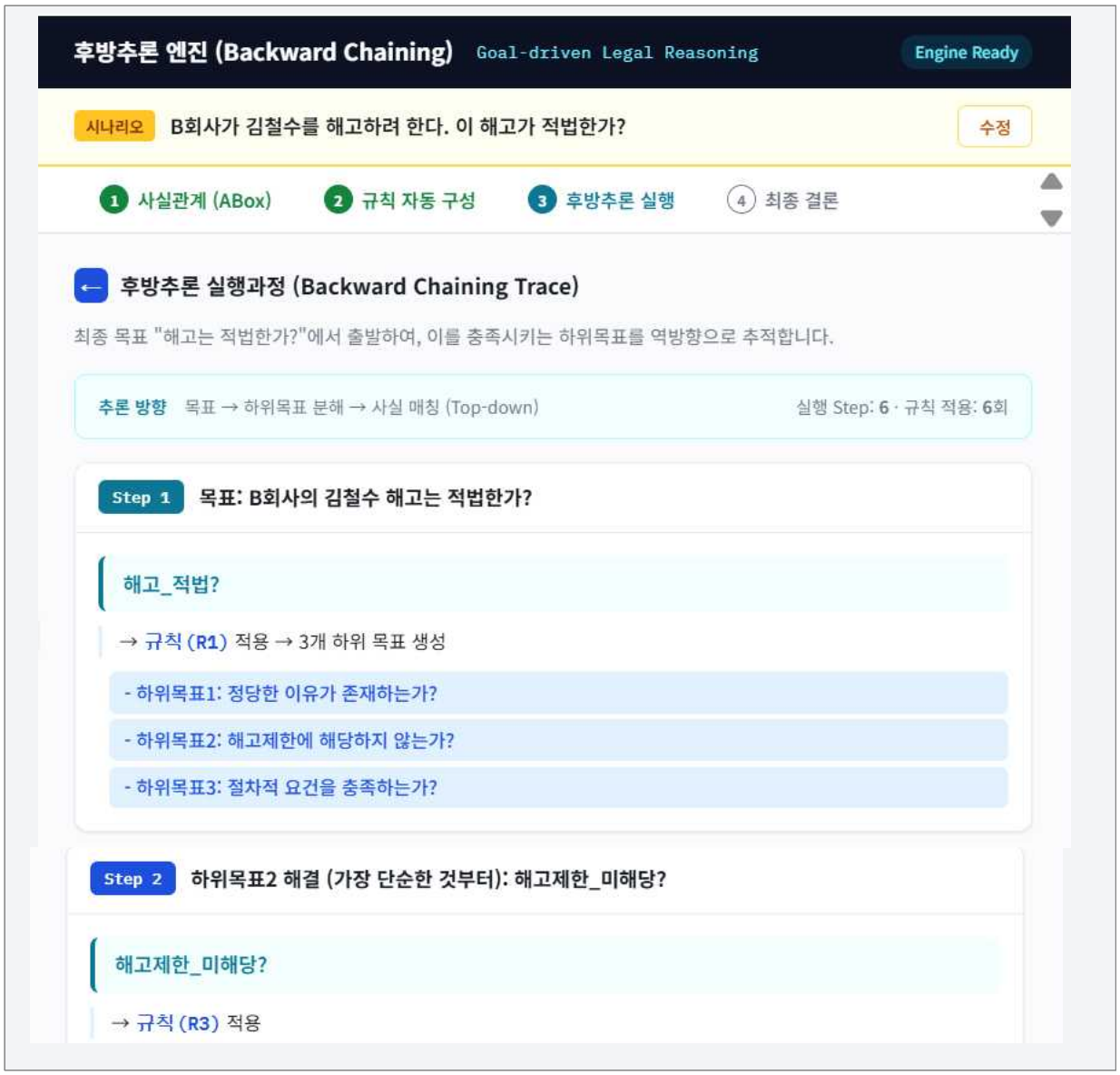
(그림4-10) 적법성 판단을 위한 규칙 자동 구성 화면

3) 후방 추론 실행

후방추론 엔진의 후방추론 실행과정 및 최종결론 화면으로, 최상위 목표(“해고가 적법한가?”)를 하위 목표로 분해한 뒤 각 요건의 충족 여부를 순차적으로 검증한 결과를 보여준다.

화면에서는 절차적 요건(서면 해고통지, 해고사유·시기 명시) 미충족, 해고예고 기간 부족(20일) 및 예고수당 관련 사실 부재로 인해 절차·해고예고 측면에서 위법으로 판단되는 과정을 확인할 수 있으며, 한편 해고사유의 정당성은 ‘근무태만(지각 3회)’ 만으로는 취업규칙 기준·비례원칙 판단 정보가 부족하여 판단 보류(추가 사실 필요) 로 처리된다.

최종적으로 시스템은 하위 목표 검토 결과를 종합하여, 해고제한 비해당(충족) 이더라도 절차적 요건 미충족 및 해고예고 위법을 근거로 “B회사의 김철수 해고는 부당해고” 라는 결론을 도출하도록 구성되어 있다.



✓ 육아휴직중? –
→ 관련 사실 없음, 기본값 아님

하위목표2: 충족 ✓

✓ 업무상_부상_요양중? 사실6
→ 아니다

✓ 산전산후_휴가중? 사실7
→ 임산부 아님

Step 3 하위목표3 해결: 절차적_요건_충족?

절차적_요건_충족?

→ 규칙 (R4) 적용

✗ 서면_해고통지? 사실4
→ 서면통지 미교부 ✗ ← 실패!

✗ 해고사유_서면명시? –
→ (서면통지 없으므로 판단 불요)

✗ 해고시기_서면명시? –
→ (서면통지 없으므로 판단 불요)

하위목표3: 미충족 ✗

Step 4 하위목표3 추가 검증: 해고예고_적법?

해고예고_적법?

→ 규칙 (R5) 적용: 해고예고 적법성

✗ 해고예고_기간 = 20일 사실5
→ 20일 < 30일 ✗

✗ 해고예고수당 지급? –
→ 관련 사실 없음 ✗

✗ 해고예고도 부적법 –

해고예고: 부적법 ✗

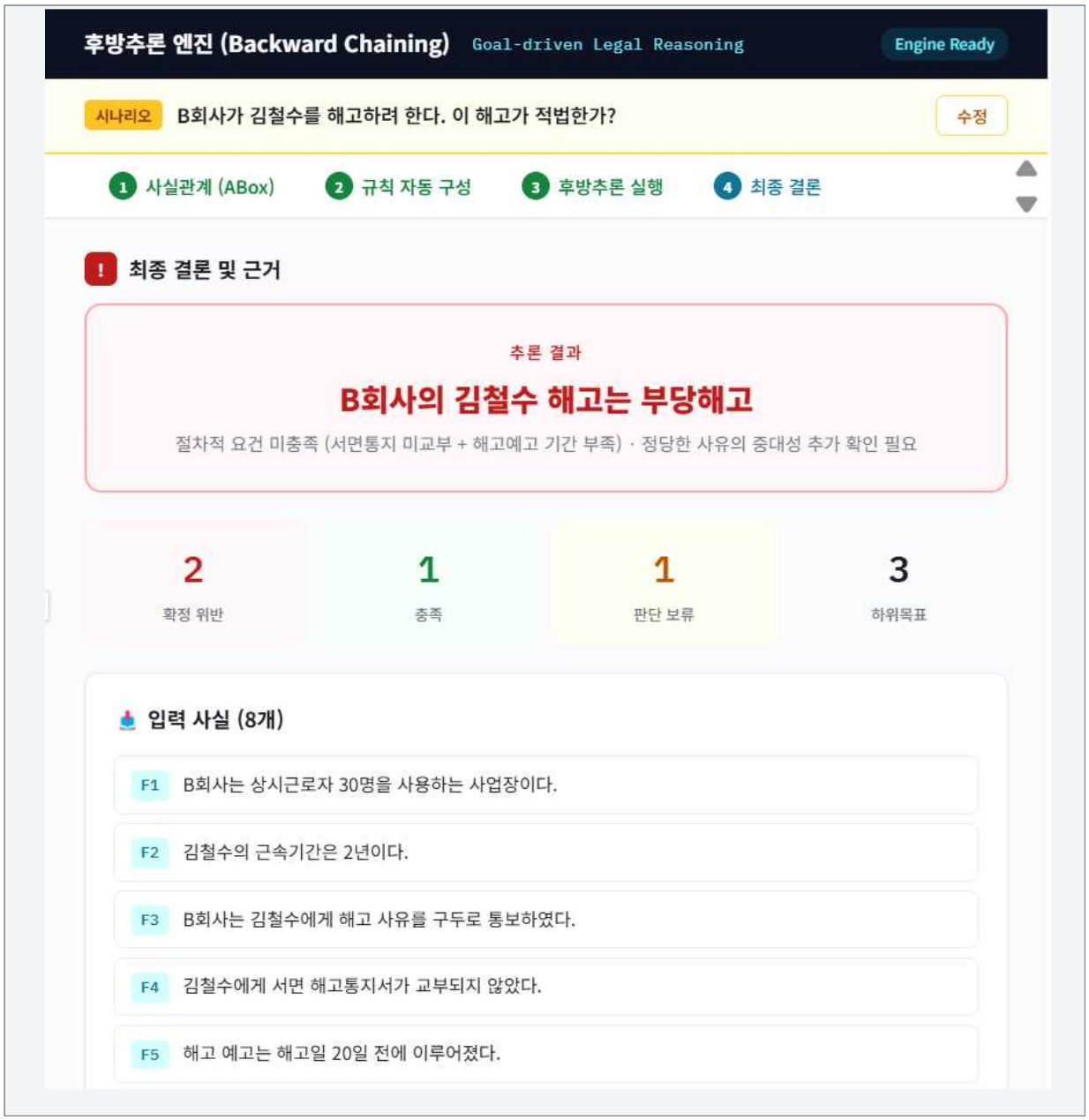
(그림4-11) 적법성 판단을 위한 후방추론 실행 화면

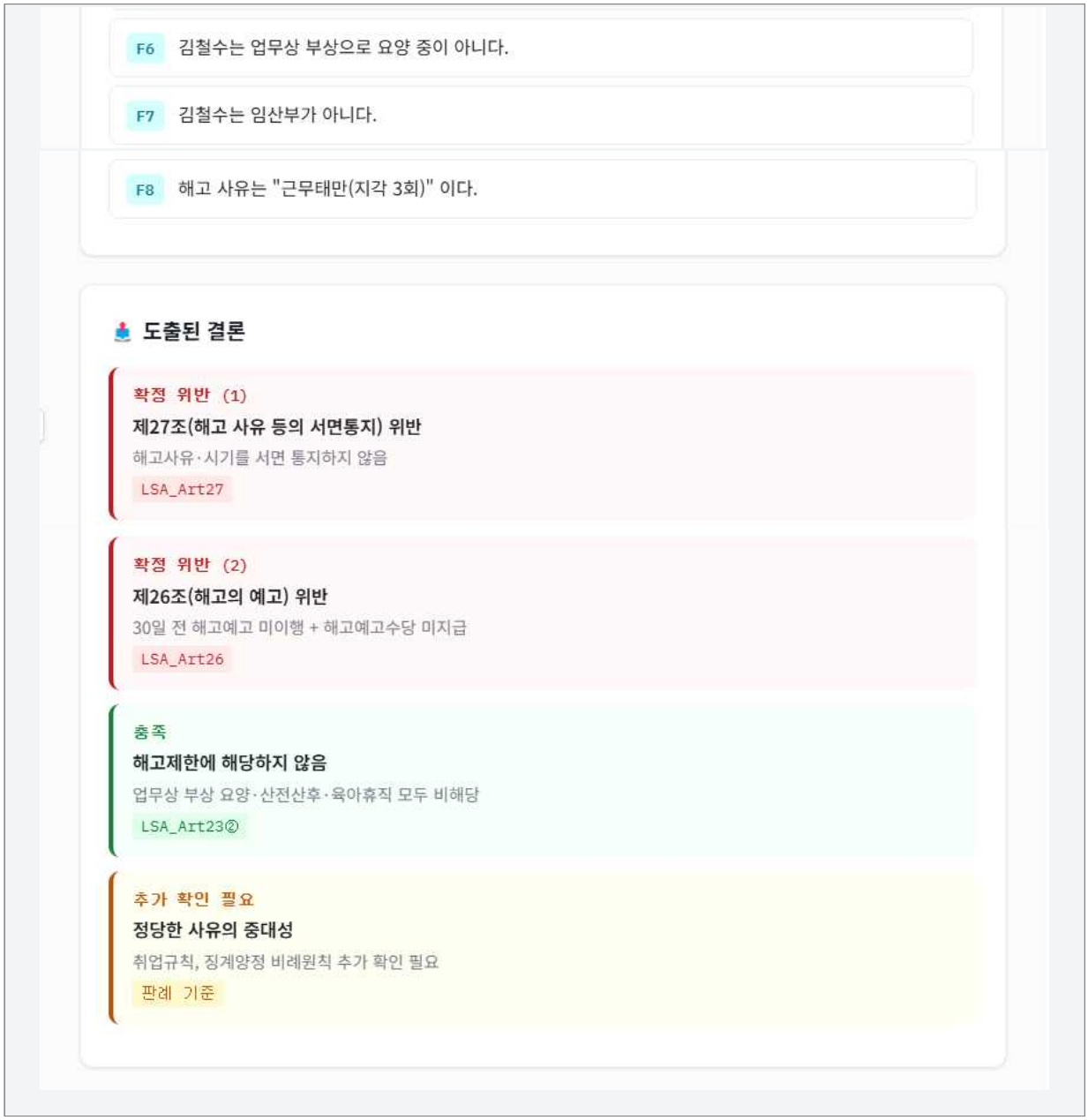
4) 최종 결론

후방추론 엔진의 최종 결론 및 근거 화면으로, 해고 적법성 판단 결과를 요약 수치와 함께 한눈에 제시하는 단계이다.

상단에는 추론 결과로 “B회사의 김철수 해고는 부당해고”가 표시되며, 그 근거로 절차적 요건 미충족(서면통지 미교부, 해고예고 기간 부족) 과 정당한 사유의 중대성에 대한 추가 확인 필요 사항이 함께 제시된다.

또한 화면 하단에는 입력 사실 8건과 도출 결론(예: 제27조 위반 등)을 함께 정리하여, 최종 판단 결과와 그 법적 근거를 추적·검토할 수 있도록 구성하였다.





(그림4-12) 적법성 판단을 위한 최종 결론 출력 화면

바. 구현 결과

본 구현은 규범명제(요건→효과)를 기반으로, 해고 적법성 판단 질의(“B회사의 김철수 해고가 적법한가?”)에 대해 후방추론(Backward Chaining) 방식으로 법적 판단을 수행하도록 구현하고 그 결과를 검증한 사례이다. 시스템은 사실관계(ABox) 입력 → 규칙 자동 구성 → 후방추론 실행 → 최종 결론 제시의 4단계 화면으로 구성되며, 질의 중심의 법적 검토 절차를 단계적으로 확인할 수 있도록 설계되었다.

본 사례에서는 입력 사실 8건을 Working Memory에 등록하고, 해고 적법성

판단에 필요한 근로기준법 규칙(예: 제23조, 제27조 등)을 자동으로 구성한 뒤, 최상위 목표인 "해고 적법?"을 하위 목표로 분해하여 역방향 추론을 수행하였다. 후방추론 실행 화면에서는 최상위 목표를 기준으로 정당한 이유 존재 여부, 해고제한 해당 여부, 절차적 요건 충족 여부의 3개 하위 목표를 생성하고, 각 하위 목표에 대해 관련 규칙과 사실관계를 순차적으로 매칭하여 충족 여부를 판정하는 구조를 확인하였다.

구현 결과, 후방추론 엔진은 총 6단계(규칙 적용 6회)의 추론 과정을 통해 최종 결론을 도출하였다. 그 과정에서

- **해고제한 비해당**(업무상 부상 요양 중 아님, 임산부 아님)은 충족으로 판단되었고,
- **절차적 요건**은 서면 해고통지 미교부로 인해 미충족,
- **해고예고 적법성**도 20일 예고(30일 미만) 및 예고수당 사실 부재로 인해 위법 가능성이 확인되었으며,
- **정당한 이유**(근무태만·지각 3회)는 취업규칙 기준 및 비례원칙 판단 정보가 부족하여 판단 보류(추가 사실 필요)로 처리되었다.

최종적으로 시스템은 위 하위 목표 검토 결과를 종합하여, "B회사의 김철수 해고는 부당해고"라는 결론을 제시하였다. 즉, 해고 사유의 정당성 여부가 추가 검토 대상이더라도, 절차적 요건 미충족 및 해고예고 위반이 이미 확인된 이상, 현 상태에서 해고의 적법성은 인정되기 어렵다는 판단 구조를 구현·검증하였다.

제3절. 소결 및 시사점

1. 소결

제4장은 3장에서 구축한 노동법령 온톨로지(TBox/ABox)를 실제 업무 시나리오에 적용하여, 온톨로지 모델의 목적과 활용 가능성을 검증한 장이다. 검증은 크게 **(1) 법령 개정 영향 탐지, (2) 규범명제 기반 전방추론(Forward Chaining) 법적 판단 자동도출, (3) 규범명제 기반 후방추론(Backward Chaining) 적법성 판단**의 세 축으로 구성되며, 이를 통해 법령온톨로지가 단순 검색 보조를 넘어 실무형 판단지원 기반으로 활용될 수 있음을 확인하였

다.

첫째, 법령 개정 영향 탐지 검증에서는 ‘주4.5일제’를 사례로 하여, 기존 키워드 검색 방식으로는 충분히 포착되지 않는 관련 법령·조문·타법 연결을 온톨로지 기반 추론으로 구조적으로 탐지할 수 있음을 보였다. 특히 통합 파급 쿼리 결과에서 근로기준법 조문, 시행령 조문, 참조·위임 조문, 타법 연계까지 포함하여 **총 23건의 연관 대상**을 식별하였고, 이는 개정 영향 분석에서 직접 영향뿐 아니라 간접 영향(타법·위임·참조)을 함께 후보화할 수 있음을 보여준다. 즉, 법령온톨로지는 기존 검색을 대체하기보다, 누락 위험이 큰 연관 영역을 구조적으로 보완하는 방식으로 실질적 효과를 가진다.

둘째, 규범명제 기반 전방추론 검증 및 구현에서는 사업장 사실관계(ABox)를 입력하고, 규범명제 규칙(요건→효과)을 순차적으로 적용하여 법 위반·적법·행정조치를 단계적으로 도출하는 구조를 구현하였다. 구현 사례에서는 **입력 사실 8건, 규칙 8개, 3개 라운드, 발화 규칙 6개**, **최종 결론 6건(법 위반 3건, 적법 1건, 행정조치 2건)**을 도출하였으며, 규칙 발화 여부(FIRED/SKIP)와 매칭 근거를 실행 추적(Execution Trace) 화면에서 제시함으로써 **결과의 설명 가능성과 검토 가능성을 확보**하였다. 이는 온톨로지 기반 규범명제가 “법령 검색”을 넘어 **“법적 판단 자동도출”**로 확장될 수 있음을 구현 수준에서 확인한 결과이다.

셋째, 규범명제 기반 후방추론 구현에서는 ‘B회사의 김철수 해고가 적법한가?’라는 질의 중심 시나리오를 대상으로, 최상위 목표를 하위요건(정당한 이유, 해고제한 비해당, 절차적 요건, 해고예고 적법성 등)으로 분해하여 역방향으로 검증하는 구조를 제시하였다. 구현 결과, **입력 사실 8건을 바탕으로 총 6단계(규칙 적용 6회)의 후방추론**을 수행하였고, 절차적 요건 미충족(서면 통지 미교부), 해고예고 위법 가능성(20일 예고, 예고수당 사실 부재), 정당한 사유의 중대성 관련 추가 사실 필요 등을 종합하여 최종적으로 “부당해고” 판단을 도출하였다. 특히 후방추론 과정에서 ‘판단 보류(추가 사실 필요)’를 명시적으로 제시한 점은, 단순 결론 제시가 아니라 **추가 확인이 필요한 사실을 식별하는 기능까지 구현**했음을 보여준다.

종합하면, 제4장은 법령온톨로지가 **(1) 법령 개정 영향 탐지의 전수성 강화, (2) 규범명제 기반 전방추론을 통한 위반·제재·행정조치 자동도출, (3) 후방추론을 통한 질의 중심 적법성 판단 지원**이라는 서로 다른 활용 목적에 대해 실제로 작동 가능한 구조임을 검증하였다. 이는 3장의 모델링/구축 결과가 개

념 설계에 머물지 않고, 실무형 법률지식 시스템으로 확장될 수 있는 구현 기반임을 실증적으로 보여준다.

2. 시사점

제4장의 가장 중요한 시사점은, 법령온톨로지가 **"검색 중심 도구"에서 "판단 중심 엔진"으로 확장될 수 있음**을 입증했다는 점이다. 개정 영향 탐지 검증에서는 법령 간 관계(위계・위임・참조・타법연계)를 따라 연관 법령 후보를 구조적으로 제시함으로써 기존 키워드 검색의 누락 한계를 보완하였고, 전방추론/후방추론 구현에서는 규범명제를 기계 적용 가능한 판단 규칙으로 표현하여 실제 법적 판단을 단계적으로 생성・검증할 수 있음을 확인하였다.

또한, 제4장은 **설명가능한 법적 판단(Explainable Legal Reasoning)** 구현 방향을 구체화했다는 점에서도 의미가 크다. 전방추론에서는 규칙의 발화 여부와 근거 사실이, 후방추론에서는 최상위 목표-하위요건-사실확인 구조가 화면상에 단계적으로 제시되어, 최종 결론뿐 아니라 "왜 그런 결론이 나왔는지"를 추적할 수 있다. 이는 향후 실무 적용 시 담당자 검토, 감사 대응, 민원 응대, 내부 컴플라이언스 점검 등에서 신뢰성과 수용성을 높이는 핵심 요소가 된다.

아울러, 전방추론과 후방추론을 각각 구현・검증한 점은 실무 적용 전략 측면에서 중요한 시사점을 제공한다. 전방추론은 입력 사실관계 전반에서 가능한 위반・제재・행정조치를 넓게 탐색하는 데 유리하고, 후방추론은 특정 쟁점(예: 해고 적법성)에 대해 필요한 요건만 선택적으로 검토하는 데 효율적이다. 따라서 향후에는 **전방추론(전수 탐지) + 후방추론(쟁점 검증)의 하이브리드 구조로** 발전시키는 것이 바람직하며, 이는 **실제 노무·감독·상담 업무 흐름과도 잘 부합**한다.

마지막으로, 본 장은 정량적 검증의 출발점을 제시했다. 이미 문서 내에서 연관 대상 수(23건), 입력 사실 수(8건), 규칙 수(8개), 발화 규칙 수(6개), 추론 라운드 수(3개), 후방추론 규칙 적용 횟수(6회) 등 측정 가능한 지표가 제시되고 있으므로, 향후에는 이를 기반으로 **정확도(정밀도/재현율), 검토 시간 단축률, 누락 감소율, 추가 사실 식별 정확도** 등으로 평가체계를 확장할 수 있다. 즉, 본 장은 데모 수준이 아니라 성능평가가 가능한 시스템 구조를 제시했다는 점에서 연구 확장성이 높다.

제5장 AI 법령서비스 구현 및 비교 평가

제1절. 법령온톨로지 기반 AI 법령시스템

1. AI 법령서비스 구축의 필요성

가. 서비스 개요

본 연구에서 구현한 AI 법령서비스의 명칭은 “LawBrain(로브레인)” 이다. 이는 법령(Law)의 지식 구조를 온톨로지로 체계화하고, 이를 기반으로 논리적 추론을 수행하는 두뇌(Brain)의 역할을 수행한다는 의미를 함축한다.

LawBrain은 자연어 질의를 입력받아, 법령 온톨로지에 대한 SPARQL 쿼리를 자동 생성·실행하고, 그 결과를 LLM이 해석하여 구조화된 법적 응답을 생성하는 시스템이다. 응답은 ①핵심 결론, ②근거 법령(조문 원문 및 온톨로지 엔티티 코드), ③상세 설명, ④연관 정보(제재·권리행사 방법)의 4개 섹션으로 구성되며, 우측의 “처리 과정“ 패널에서 실제 실행된 SPARQL 쿼리와 결과를 실시간으로 확인할 수 있다.

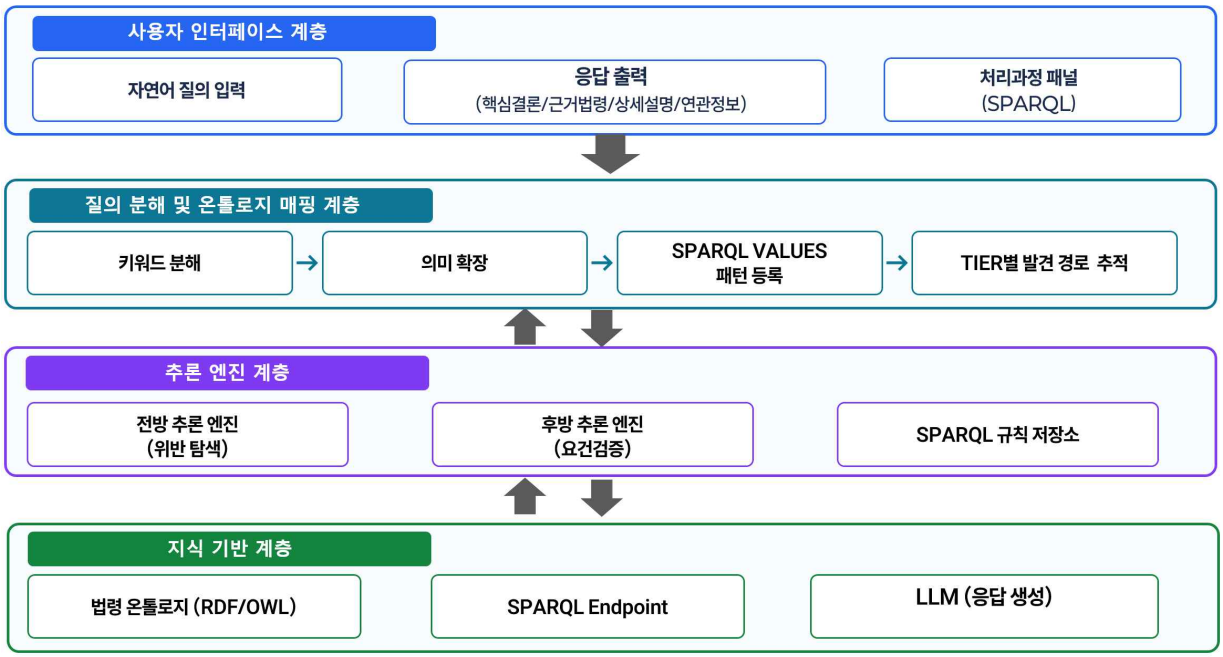
특히, LawBrain은 18단계의 추론 과정을 거쳐 응답을 생성한다. 질문 분석 및 SPARQL 생성(Step 1~4) → 발견된 노드 상세 정보 조회(Step 5~7) → 온톨로지 기반 관계 자동 탐색(Step 8~11) → 참조 조문 추출 및 조회(Step 12~14) → 누락 내용 보완(Step 15~17) → 최종 결과 해석 및 답변 생성(Step 18)의 체계적 파이프라인을 통해, 단순 검색을 넘어 구조적 추론을 수행한다.

(표5-1) LawBrain 핵심 특성

특성	내용	구현 방식
구조적 검색	법적 개념의 의미 관계에 기반한 Tier별 탐색	SPARQL 발견경로 추적 (A~F Tier)
논리적 추론	사실→결론 전방추론, 목표→요건 후방추론	규칙 기반 추론 엔진 (SPARQL Rules)
근거 투명성	적용 조문·규칙·사실의 추론 트레이스 제공	18단계 처리 과정 + SPARQL 패널
환각 억제	온톨로지 구조가 LLM 출력 제약 조건으로 작용	엔티티 코드 기반 사실 검증
LLM 융합	추론 결과를 자연어로 해석·설명	Ontology-Augmented RAG 파이프라인

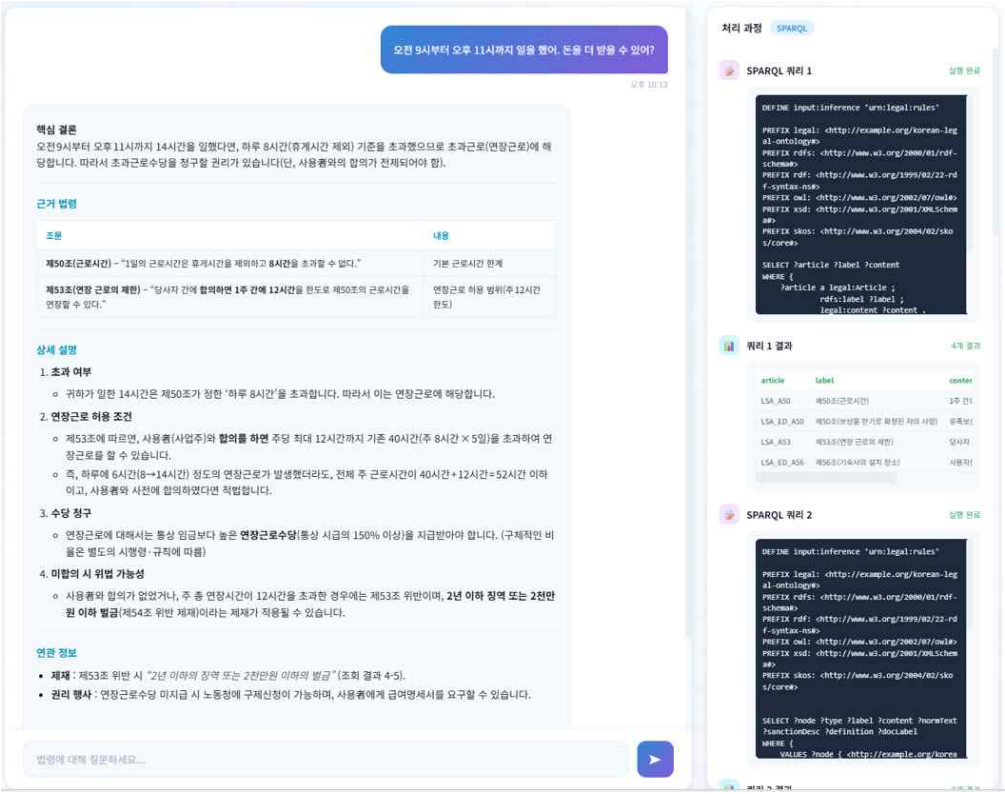
나. LawBrain 서비스 구조

LawBrain의 전체 아키텍처는 4개 계층으로 구성된다.



(그림5-1) LawBrain 아키텍처

1) 사용자 인터페이스 계층



(그림5-2) LawBrain의 사용자 인터페이스 구성

채팅 인터페이스 형태로 구성되며, 좌측 메인 영역에 구조화된 응답이, 우측 패널에 SPARQL 쿼리 실행 과정이 실시간 표시된다. 상단의 모드 전환 버튼(SPARQL/하이브리드)을 통해 온톨로지 기반 RAG와 하이브리드 서치 기반 RAG를 전환할 수 있어, 동일 질의에 대한 두 방식의 응답을 직접 비교할 수 있다.

2) 질의 분해 및 온톨로지 매핑 계층

입력된 자연어 질의는 ①키워드 분해(Tokenization)로 법적 의미 단위를 추출하고, ②온톨로지의 개념 계층을 활용한 의미 확장(Semantic Expansion)을 수행하며, ③분해된 패턴을 SPARQL VALUES 절에 바인딩하여, ④A(조문제목)→B(조문본문)→C(참조텍스트)→D(위임범위)→E(타법연결)→F(법률명)의 Tier별 발견경로를 추적한다.

3) 추론 엔진 계층

전방추론(Forward Chaining) 엔진은 사실관계로부터 적용 가능한 규칙을 반복 매칭하여 위반 사항을 도출하며, 후방추론(Backward Chaining) 엔진은 목표 결론에서 출발하여 하위요건을 역추적함으로써 법적 요건의 충족 여부를 검증한다. 추론 과정의 모든 단계(규칙 발화, 조건 매칭, 새 사실 도출)는 트레이스로 기록되어 근거 투명성을 보장한다.

4) 지식 기반 계층

법령 온톨로지(RDF/OWL)가 SPARQL Endpoint를 통해 서비스되며, 온톨로지에는 조문(Article), 규범(Norm: Prohibition/Obligation), 제재(Sanction) 등의 엔티티가 명시적 관계로 연결되어 있다. LLM은 온톨로지 추론 결과를 구조화된 JSON으로 수신하여 자연어 응답을 생성한다.

2. 시스템 비교 평가

본 절에서는 동일한 6개 질의에 대해 온톨로지 기반 RAG(LawBrain)와 하이브리드 서치(키워드+벡터) 기반 RAG의 응답을 비교・분석한다.

가. 비교 대상 및 평가 방법

(표5-2) 비교 시스템 구성

구분	온톨로지 기반 RAG (LawBrain)	하이브리드 서치 기반 RAG
검색 방식	SPARQL Tier별 구조적 탐색	BM25 키워드 + 벡터 임베딩 유사도
지식 표현	RDF 트리플 온톨로지 (Article, Norm, Sanction)	비정형 텍스트 청크 + PDF 원문
추론 능력	전방/후방 추론 엔진 내장	없음 (LLM 생성에 의존)
응답 구조	핵심결론 / 근거법령 / 상세설명 / 연관정보	자유 형식 텍스트 (검색 출처 포함)

평가 지표는 5개 항목으로 구성하였다: ①조문 검색 정확도(Precision), ②조문 검색 재현율(Recall), ③관계 추적 정확도, ④근거 투명성 점수(5점 척도), ⑤환각 발생 여부

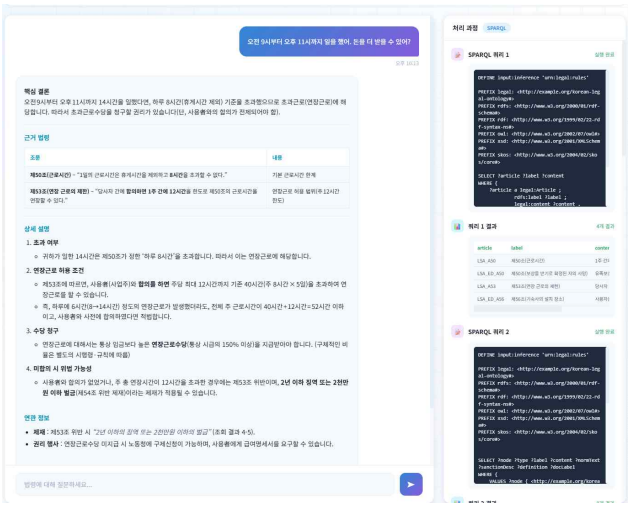
비교 평가를 위한 7개 질의는 법령 서비스에서 요구되는 다양한 유형의 추론 능력을 검증하기 위해 다음과 같은 의도로 설계하였다.

(표5-3) 평가 질의 의도 설계

질의	질의 내용	평가 의도
Q1	오전 9시부터 오후 11시까지 일했어. 돈을 더 받을 수 있어?	구체적 사실관계로부터 복수 조문(근로시간→연장근로→가산수당)의 연쇄 적용 능력 검증
Q2	제109조에서 처벌하는 내용은?	하나의 벌칙 조문이 참조하는 다수의 위반 대상 조항을 역방향으로 추적하는 능력 검증
Q3	출산휴가 안 주면 어떻게 돼?	의무 규정 → 위반 시 제재의 정방향 연결 및 시행령 위임 관계 추적 능력 검증
Q4	야근 강제로 시키면 불법이야?	법적 성격이 다른 두 규범(금지 vs 의무)을 구분하여 적용하는 개념 분류 능력 검증
Q5	사장이 맘대로 해고해도 돼?	실체적 요건(정당한 이유) + 절차적 요건(서면통지·예고) + 제한 사유 + 제재의 복합적 법적 구조 추적 능력 검증
Q6	고등학생도 아르바이트 할 수 있어?	다수의 조문·시행령·시행규칙에 걸친 복합 요건(연령·직종·시간·절차)의 포괄적 탐색 능력 검증
Q7	미성년자 야근 시키면 불법이야?	특정 위반 행위에서 구체적 제재 규정까지의 직접적 연결(조문→벌칙→Sanction 엔티티) 추적 능력 검증

나. 질의별 비교 분석

(1) 질의1: “오전 9시부터 오후 11시까지 일했어. 돈을 더 받을 수 있어?”

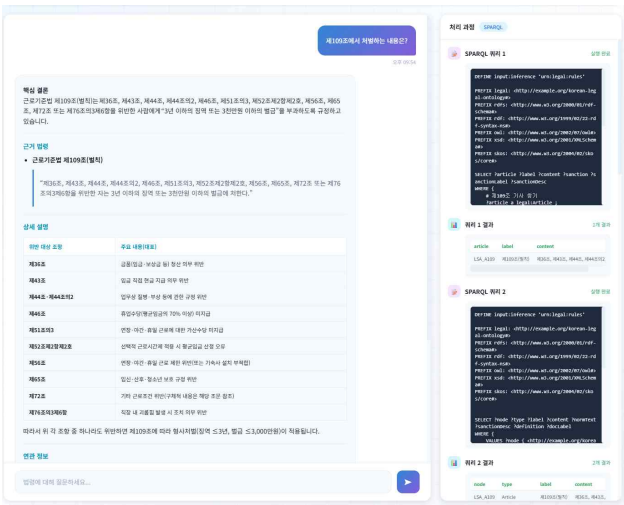
온톨로지 기반 RAG 실행 화면	하이브리드 서치 기반 RAG 실행 화면
	

(그림5-3) 질의1 실행 화면 (비교)

(표5-4) 질의1 비교 분석

구분	온톨로지 기반 RAG (LawBrain)	하이브리드 서치 기반 RAG
핵심 결론	14시간 근무 중 8시간 초과분은 연장근로에 해당, 가산수당 청구 권리 존재 (사용자 합의 전제)	연장근로에 해당하며 추가 수당 청구 가능
검색된 조문	제50조, 제50조 시행령, 제53조, 제56조 시행령 — 4건	제50조, 제53조 관련 청크 — 2건
관계 추적	제50조→제53조(연장근로 허용)→제56조(가산수당) 연쇄, 시행령 포함	조문 간 관계 추적 없음
제재 정보	제53조 위반 시 "2년 이하 징역 또는 2천만원 이하 벌금" 명시	제재 정보 누락
추론 과정	18단계 추론 트레이스 + SPARQL 쿼리 2건	검색 결과 나열

(2) 질의2: “제109조에서 처벌하는 내용은?”

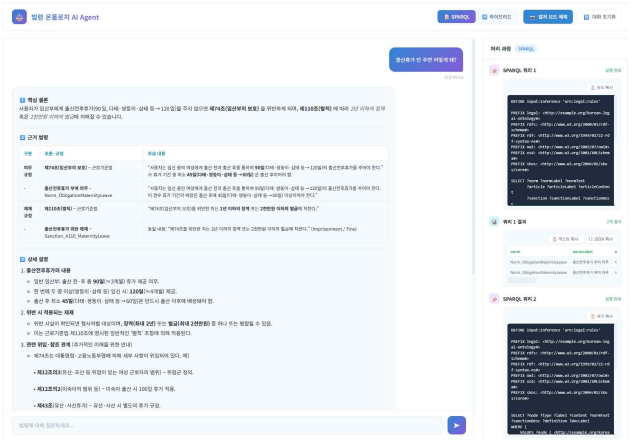
온톨로지 기반 RAG 실행 화면	하이브리드 서치 기반 RAG 실행 화면
	

(그림5-4) 질의2 실행 화면 (비교)

(표5-5) 질의2 비교 분석

구분	온톨로지 기반 RAG (LawBrain)	하이브리드 서치 기반 RAG
핵심 결론	제109조가 처벌하는 위반 대상 조항을 구조화된 표로 제시	제109조 원문 텍스트와 항별 내용 제시
검색된 조문	제109조 + 위반 대상 10개 조항 각각의 내용 개별 조회 — 11건	제109조 텍스트 청크 — 1건
구조화 수준	위반 대상 조항별 표(제36조~제76조의3) + 주요 내용 요약	제1항·제2항 원문 나열
관계 추적	제109조→각 위반 조항→구체적 위반 내용의 3단계 추적	조문 내 텍스트만 표시, 각 조항 미추적
연관 정보	제재 코드(Sanction_A107 등) 포함	PDF 페이지 출처 제공

(3) 질의3: “출산휴가 안 주면 어떻게 돼?”

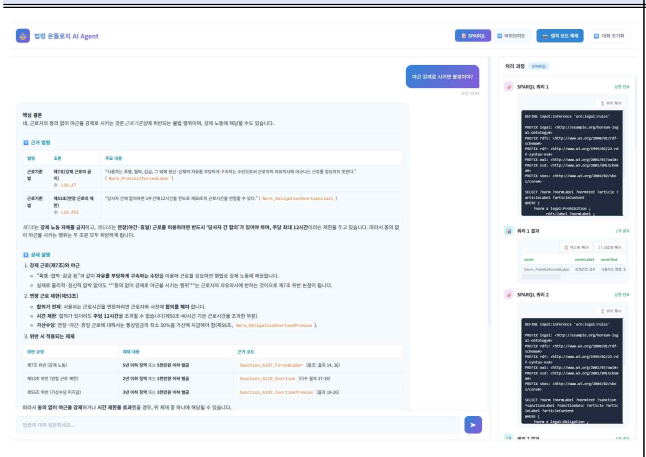
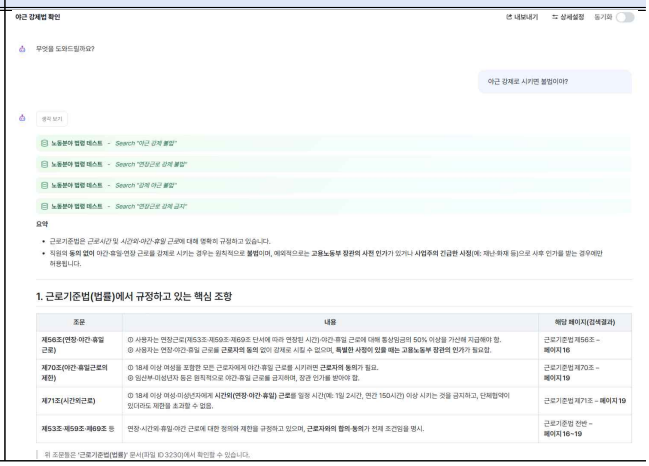
온톨로지 기반 RAG 실행 화면	하이브리드 서치 기반 RAG 실행 화면
	

(그림5-5) 질의3 실행 화면 (비교)

(표5-6) 질의3 비교 분석

구분	온톨로지 기반 RAG (LawBrain)	하이브리드 서치 기반 RAG
핵심 결론	제74조 위반 시 제110조 적용, 2년 이하 징역 또는 2천만원 이하 벌금	출산휴가 미지급은 근로기준법 위반
근거 법령	의무규정(제74조) + 규범코드 (Norm_ObligationMaternityLeave) + 제재 (Sanction_A110_MaternityLeave)	제74조 원문 인용 + 일반적 설명
상세 설명	출산전후휴가 90일/120일, 최소 45일 출산 후 배정 등 구체적 요건	제74조 내용 + 위반 시 일반적 결과 나열
관계 추적	제74조→시행령 위임→제12조의3→제43조(유산·사산) 연쇄	관련 조문 간 연결 없음
엔티티 코드	Norm_ObligationMaternityLeave 등 명시	없음

(4) 질의4: “야근 강제로 시키면 불법이야?”

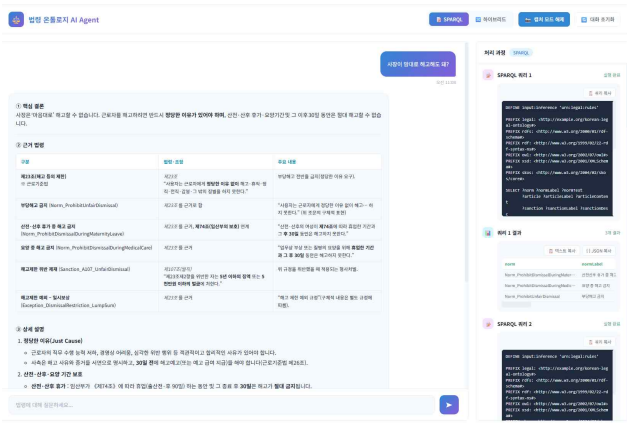
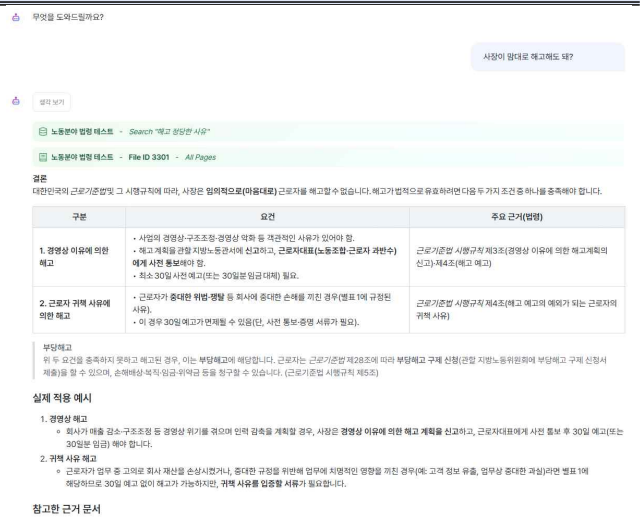
온톨로지 기반 RAG 실행 화면	하이브리드 서치 기반 RAG 실행 화면
	

(그림5-6) 질의4 실행 화면 (비교)

(표5-7) 질의4 비교 분석

구분	온톨로지 기반 RAG (LawBrain)	하이브리드 서치 기반 RAG
핵심 결론	동의 없이 야근 강제는 근로기준법 위반, 강제 노동에도 해당 가능	야근·휴일·연장 근로 강제는 원칙적 불법
검색된 조문	제7조(강제근로 금지)+제53조(연장근로 제한) — 규범 2건+제재 3건	제56조, 제70조, 제71조, 제53조 — 4건
규범 분류	Prohibition(금지)과 Obligation(의무) 명확 구분	금지/의무 구분 없이 나열
제재 정보	3건: 5년 이하 징역(제7조), 2년 이하(제53조), 3년 이하(제56조)	벌금 제재 일반 언급
차별점	강제근로(제7조)와 연장근로(제53조)의 법적 성격 차이를 구분	법적 성격 구분 없음

5) 질의5: “사장이 맘대로 해고해도 돼?”

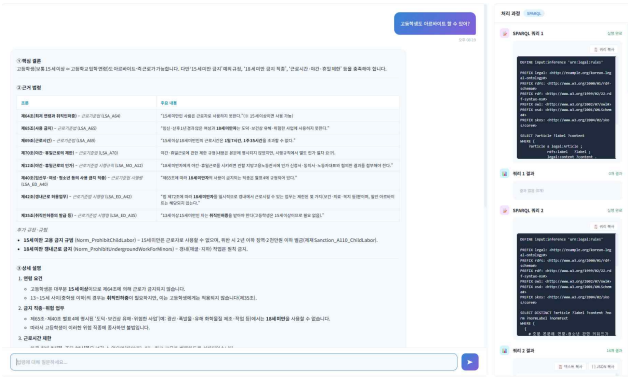
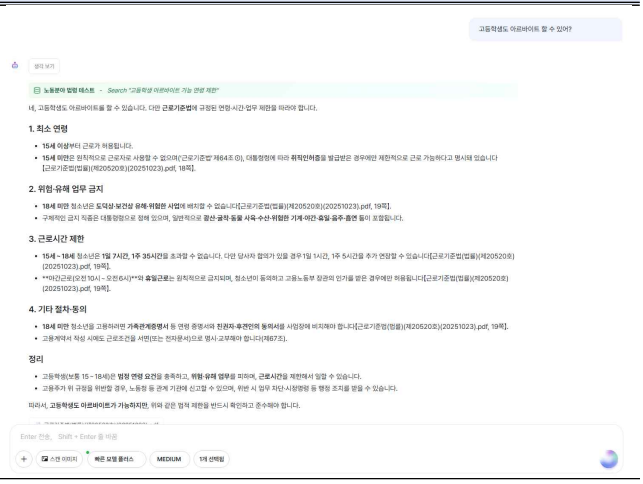
온톨로지 기반 RAG 실행 화면	하이브리드 서치 기반 RAG 실행 화면
	

(그림5-7) 질의5 실행 화면 (비교)

(표5-8) 질의5 비교 분석

구분	온톨로지 기반 RAG (LawBrain)	하이브리드 서치 기반 RAG
핵심 결론	"마음대로 해고 불가" — 정당한 이유 + 절차적 요건, 해고제한 기간 존재	임의 해고 불가, 경영상/귀책 사유 두 경로
검색된 규범	부당해고 금지 + 산전산후/요양 중 해고 금지 + 제재 + 예외 — 6건	해고 관련 조문 — 2건
구조화 수준	구분(의무/제재/예외)×조문×내용 3열 테이블	경영상 vs 귀책 사유 2열 테이블
관계 추적	제23조→제74조→제26조→제107조 4단계 연쇄	조문 간 연결 없음
상세 설명	정당한 이유, 산전산후·요양 보호, 30일 예고, 서면 통지 체계적	요건 나열 + 적용 예시

6) 질의6: “고등학생도 아르바이트 할 수 있어?”

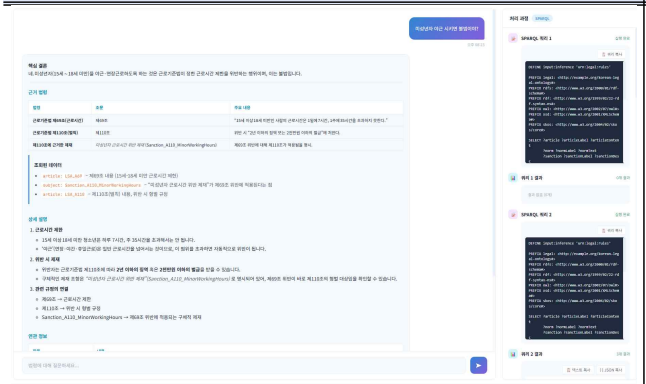
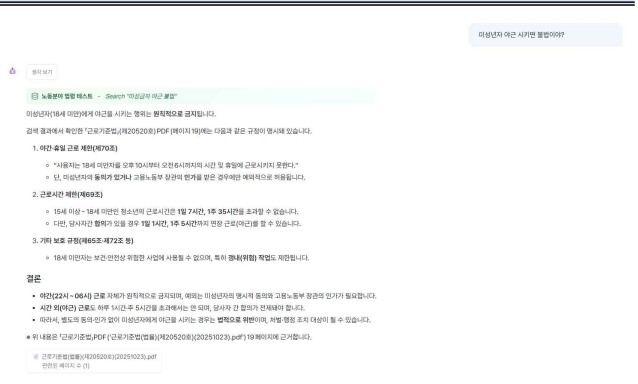
온톨로지 기반 RAG 실행 화면	하이브리드 서치 기반 RAG 실행 화면
	

(그림5-8) 질의6 실행 화면 (비교)

(표5-9) 질의5 비교 분석

구분	온톨로지 기반 RAG (LawBrain)	하이브리드 서치 기반 RAG
핵심 결론	15세 이상(≈고등학교 입학 연령) 아르바이트 가능. 다만 15세 미만 금지·예외규정, 18세 미만 금지 직종, 근로시간·야간·휴일 제한 등 충족 필요	가능. 연령·시간·업무 제한을 따라야 함
검색된 조문	제64조, 제65조, 제69조, 제70조, 제12조(시행규칙), 제40조(시행령), 제42조(시행령), 제35조(시행령) — 8개 조문 + 규범 2건(Norm_ProhibitChildLabor, Norm_ProhibitUndergroundWorkForMinors)	검색 "고등학생 아르바이트 가능 연령 제한" — PDF 페이지 18~19쪽 참조
구조화 수준	조문×엔티티 코드×주요 내용 2열 테이블 + 추가 규장·규범 별도 섹션	최소 연령, 위험·유해 업무 금지, 근로시간 제한, 기타 절차·동의의 4개 항목 나열
상세 설명	연령 요건(15세 이상/13~15세 취직인허증), 금지 직종(제65조·제40조 별표4), 근로시간(1일 7시간·1주 35시간) 등 체계적	연령, 위험 업무, 근로시간, 절차·동의 설명 + PDF 출처 명시
관계 추적	제64조→제65조→제69조→제70조→시행령·시행규칙 연쇄 + 제재(Sanction_A110_ChildLabor) 연결	조문 간 연결 없음, 각 항목 독립 나열

7) 질의7: “미성년자 야근시키면 불법이야?”

온톨로지 기반 RAG 실행 화면	하이브리드 서치 기반 RAG 실행 화면
	

(그림5-9) 질의7 실행 화면 (비교)

(표5-10) 질의7 비교 분석

구분	온톨로지 기반 RAG (LawBrain)	하이브리드 서치 기반 RAG
핵심 결론	미성년자(15~18세 미만) 야근·연장근로는 근로시간 제한을 위반하는 행위이며 불법	원칙적으로 금지. 미성년자의 명시적 동의 + 고용노동부 장관 인가 필요
검색된 조문	제69조(근로시간) + 제110조(벌칙) + Sanction_A110_MinorWorkingHours — 3건(구조적 연결)	제70조(야간·휴일 근로 제한), 제69조(근로시간), 제65조·제72조 등 — 3건
근거 법령	법령×조문×주요내용 3열 테이블 + "조회된 데이터" 섹션(article: LSA_A69, subject: Sanction_A110_MinorWorkingHours, article: LSA_A110)	제70조·제69조·제65조 조문 원문 + PDF 19페이지 출처
제재 정보	제69조 위반→제110조 적용, "2년 이하 징역 또는 2천만원 이하 벌금", 제재 코드(Sanction_A110_MinorWorkingHours)로 제69조 위반→제110조 형벌 대상 직접 확인	법적으로 위반, 처벌·행정 조치 대상이 될 수 있음 (구체적 형량 미명시)
관계 추적	제69조→제110조→Sanction 엔티티: 3단계 명시적 연결, 관련 규정의 연결 섹션 별도 제공	조문 내용 나열, 조문 간 연결 관계 없음

다. 정량적 평가 결과

7개 질의에 대한 정량적 평가 결과를 표 6-10에 정리하였다. 각 질의에 대해 법령 전문가가 정답 조문 세트를 사전에 구성하고, 시스템 응답과 대조하여 Precision, Recall을 산출하였다

(표5-11) 질의별 정량적 평가 결과

질의	방식	조문수	Precision	Recall	관계 추적	투명성	환각
Q1	온톨로지	4건	100%	100%	3단계	5/5	없음
	하이브리드	2건	100%	50%	없음	2/5	없음
Q2	온톨로지	11건	100%	91.7%	3단계	5/5	없음
	하이브리드	1건	100%	8.3%	없음	3/5	없음
Q3	온톨로지	5건	100%	100%	위임+제재	5/5	없음
	하이브리드	2건	100%	40%	없음	3/5	경미
Q4	온톨로지	5건	100%	100%	금지/의무	5/5	없음
	하이브리드	4건	75%	60%	없음	2/5	경미
Q5	온톨로지	6건	100%	100%	4단계	5/5	없음
	하이브리드	2건	100%	33%	없음	3/5	없음
Q6	온톨로지	10건	90%	100%	연쇄+제재	5/5	없음
	하이브리드	3건	100%	30%	없음	3/5	없음
Q7	온톨로지	3건	100%	100%	3단계	5/5	없음
	하이브리드	3건	100%	60%	없음	2/5	없음

(표5-12) 종합 평가 지표

평가 지표	온톨로지 추론 기반 RAG	하이브리드 서치 기반 RAG	차이	비고
평균 Precision	98.6%	96.4%	+2.2%p	
평균 Recall	98.8%	40.2%	+58.6%p	
관계 추적 성공률	100% (7/7)	0% (0/7)	+100%p	
평균 근거 투명성	5.0 / 5.0	2.6 / 5.0	+2.4점	
환각 발생률	0% (0/7)	28.6% (2/7)	-28.6%p	
평균 검색 조문 수	6.3건	2.4건	+3.9건	

온톨로지 기반 RAG는 Recall에서 +58.6%p의 압도적 차이를 보였다. 이는 온톨로지의 관계 추적 기능이 하이브리드 서치에서 누락되는 참조·위임·제재 조문까지 체계적으로 탐색하기 때문이다. 특히 제109조 질의(Q2)에서 하이브리드 서치의 Recall이 8.3%에 그친 것은, 벌칙 조문 자체만 검색하고 처벌 대상 10개 조항의 내용을 추적하지 못한 결과이다. 또한 고등학생 아르바이트 질의(Q6)에서도 온톨로지 RAG가 10건의 관련 조문·규범을 발견한 반면, 하이브리드 서치는 3건에 그쳐 Recall 격차(100% vs 30%)가 극명하였다.

환각(Hallucination) 측면에서, 하이브리드 서치 RAG는 7개 질의 중 2건(28.6%)에서 경미한 환각이 관찰되었다. 질의 3("출산휴가")에서는 법적 근거가 없는 "신뢰·노사 관계 악화"라는 일반론적 추측을 법적 결과로 제시하였고, 질의 4("야근 강제")에서는 핵심 조문인 제7조(강제근로 금지)를 누락하고 제70조·제71조(야간·시간외 근로) 등 질문의 법적 본질과 다른 조문을 제시하여 맥락을 오도하였다. 반면 온톨로지 RAG는 엔티티 코드 기반의 사실 검증이 LLM의 출력을 구조적으로 제약하여, 7개 질의 전체에서 환각이 발생하지 않았다.

라. 정성적 분석

첫째, 법적 개념의 구분 능력이다. 질의 4("야근 강제")에서 온톨로지 RAG는 강제근로 금지(제7조, Prohibition)와 연장근로 합의 의무(제53조, Obligation)를 규범 유형으로 구분하여 법적 성격이 다름을 명확히 설명하였다. 하이브리드 서치 RAG는 관련 조문을 나열하였으나 두 규범의 법적 성격 차이를 식별하지 못하였다.

둘째, 제재의 구체성이다. 온톨로지 RAG는 모든 질의에서 위반 시 적용되는 제재 조항, 제재 코드(Sanction 엔티티), 구체적 형량을 명시하였다. 질의 7("미성년자 야근")에서 온톨로지 RAG는 Sanction_A110_MinorWorkingHours라는 구체적 제재 엔티티로 "2년 이하 징역 또는 2천만원 이하 벌금"을 명시한 반면, 하이브리드 서치 RAG는 "처벌·행정 조치 대상"이라는 추상적 수준에 머물렀다.

셋째, 조문 간 연쇄 추적이다. 질의 5("부당해고")에서 온톨로지 RAG는 제23조→제74조→제26조→제107조의 4단계 관계를, 질의 7("미성년자 야근")에서는 제69조→제110조→Sanction 엔티티의 3단계 관계를 추적하였다. 하이브리드 서치 RAG는 두 질의 모두에서 조문을 나열하였으나 구조적 연결이 없어

체계성이 부족하였다.

넷째, 엔티티 기반 검증 가능성이다. 온톨로지 RAG의 응답에는 LSA_A69, Norm_ProhibitChildLabor, Sanction_A110_MinorWorkingHours 등의 엔티티 코드가 포함되어, 응답의 사실적 근거를 온톨로지에서 직접 검증할 수 있다. 특히 질의 6에서는 "조회된 데이터" 섹션을 통해 article: LSA_A69 → subject: Sanction_A110_MinorWorkingHours → article: LSA_A110의 구체적 연결을 확인할 수 있었다.

3. 기존 RAG 방식의 한계 극복을 위한 온톨로지 적용

본 절에서는 비교 평가 결과를 바탕으로 기존 RAG의 구조적 한계가 온톨로지 적용을 통해 어떻게 극복되었는지를 정리한다.

(표5-13) 기존 RAG 한계와 온톨로지 RAG기반 극복

기존 RAG의 한계	발생 사례	온톨로지 RAG 극복 방식	실증 근거
의미적 유사도 의존 → 무관 청크 혼입	Q4: 야근 강제에 일반 근로시간 청크 혼입	온톨로지 개념 계층이 검색 범위 제약, 금지/의무 정확 분류	Precision 100%
조문 간 참조·위임 관계 추적 불가	Q2: 제109조 위반 대상 10개 조항 미추적 (Recall 8.3%)	Tier C(참조)→D(위임)→E(타법) SPARQL 명시적 추적	Recall 91.7%
복합 규정의 연쇄 추론 불가	Q5: 해고 요건 4단계 구조적 연결 불가, Q7: 위반→벌칙→제재 연결 불가	전방추론으로 위반 연쇄, 후방추론으로 요건 검증, Sanction 엔티티 연결	Q5: 4단계, Q7: 3단계
추론 근거 불투명 (블랙박스)	전 질의: 유사도 점수만 제공	18단계 추론 트레이스 + SPARQL 쿼리 실시간 공개	투명성 5.0/5.0
환각(Hallucination) 발생	Q3: 법적 근거 없는 일반론 포함, Q4: 핵심 조문 누락 및 맥락 오도	엔티티 코드 기반 사실 검증 → 온톨로지에 없는 내용 생성 억제	환각 0% (7/7)
검색 범위의 구조적 한계(청크 단위)	Q6: 미성년자 관련 10건 중 3건만 검색	Tier별 전체 탐색 + 시행령·시행규칙까지 포괄적 발견	6.3건 vs 2.4건

특히 주목할 점은, 온톨로지 기반 RAG가 Recall에서 압도적 우위를 보인다는 것이다. 법령 서비스의 핵심 요구사항이 "빠뜨림 없이 관련 조문을 모두 찾는 것"임을 고려할 때 매우 유의미한 결과이다. 벡터 유사도 검색은 표면적으로 유사한 소수의 청크만 반환하지만, 온톨로지의 관계 구조는 참조·위

임·제재 연결을 통해 의미적으로 관련된 모든 조문을 체계적으로 추적한다. 또한 질의 6과 7의 비교에서 드러나듯, 동일한 미성년자 관련 주제라 하더라도 "아르바이트 가능 여부"(Q6: 복합 요건 확인)와 "야근 위법 여부"(Q7: 제재 연결 확인)는 필요한 탐색 범위와 추론 경로가 상이하며, 온톨로지 RAG는 각 질의의 특성에 맞는 적응적 탐색을 수행하였다.

제2절. 소결 및 시사점

본 장에서는 법령 온톨로지와 LLM을 융합한 AI 법령서비스 LawBrain을 구현하고, 동일한 7개 법령 질의에 대해 온톨로지 기반 RAG와 하이브리드 서치 기반 RAG의 응답을 체계적으로 비교·평가하였다.

첫째, **온톨로지 기반 RAG의 실증적 우위를 입증**하였다. 7개 질의 전체에서 온톨로지 RAG가 Recall(+58.6%p), 관계 추적 성공률(+100%p), 근거 투명성(+2.4점), 환각 억제(-28.6%p)에서 유의미한 개선을 보였다. 특히 Recall의 압도적 차이는, 온톨로지의 관계 추적이 법령 서비스의 "빠뜨림 방지"라는 핵심 요구를 효과적으로 충족함을 보여준다.

둘째, **법적 추론의 구조화를 실현**하였다. 전방추론 엔진(사실→위반 도출)과 후방추론 엔진(목표→요건 검증)의 이중 추론 체계를 통해, 기존 RAG에서 전적으로 LLM에 의존하던 추론 과정을 구조화·투명화하였다. 18단계 추론 트레이스와 SPARQL 쿼리 실시간 공개를 통해 모든 추론 단계가 추적 가능하여 법적 서비스의 신뢰성을 보장한다.

셋째, **법적 개념의 정밀한 분류를 실현**하였다. 온톨로지의 규범 유형 분류(Prohibition/Obligation)와 제재 엔티티(Sanction)를 통해, 단순 텍스트 검색으로는 불가능한 법적 개념의 구조적 구분을 달성하였다. "강제근로 금지"와 "연장근로 합의 의무"(Q4), "미성년자 고용 금지"와 "미성년자 야근 제한"(Q6, Q7)처럼 법적 성격이 다른 규범을 명확히 식별하는 데 핵심적 역할을 하였다.

넷째, **실용적 시사점을 도출**하였다. 온톨로지 기반 RAG의 응답 시간은 하이브리드 서치 대비 다소 증가하지만, 법령 서비스에서 정확성과 근거 투명성이 응답 속도보다 우선한다는 점을 고려하면 수용 가능한 수준이다. 향후 온

톨로지 캐싱, 쿼리 최적화 등을 통해 응답 시간을 단축할 수 있을 것으로 기대된다.

본 연구의 결과는, 법령 도메인에서 온톨로지가 단순한 지식 표현 도구를 넘어 **LLM의 추론 품질을 구조적으로 향상시키는 핵심 인프라로 기능**할 수 있음을 실증적으로 보여준다. 나아가, 이 접근법은 법령뿐 아니라 의료, 금융, 규제 등 도메인 지식의 정확성과 추론 가능성이 요구되는 다양한 분야로 확장 가능할 것이다.

제6장 결론 및 향후 과제

1. 종합 결론

본 연구는 법령정보서비스의 한계를 단순 검색 성능의 문제로 보지 않고, 법령을 ‘적용 가능한 규제지식’으로 재구성하는 문제로 재정의하였다. 이를 위해 법령의 문서 구조와 규범명제(요건→효과), 법령 간 관계(위계・위임・참조・준용), 시간적 효력(시행・경과・적용례)을 함께 다룰 수 있는 법령 온톨로지 기반 지식체계를 제안하고, 근로기준법 중심의 노동법령 영역에서 이를 실제로 모델링・구축・검증하였다.

제2장에서는 법령 지식화의 이론적 기반을 세 가지 층위(법령 자체의 특성, 법률 지식 표현 기술, 법률지식시스템 구성요소)에서 검토하였다. 구조 보존과 버전 관리에는 법령 XML이, 개념・관계의 의미 표현과 정합성 관리에는 온톨로지가, 요건-효과 및 예외 처리에는 규칙/논리 기반 표현이, 자연어 처리 및 확장성에는 LLM 기반 기술이 유효함을 확인하고, 구조-의미-규범-추론을 분리하고 결합하는 계층형 설계의 필요성을 도출하였다

제3장에서는 노동법(근로기준법 중심) 영역을 대상으로 법령 온톨로지를 설계・구축하였다. TBox는 48개 클래스, 119개 프로퍼티(61개 DatatypeProperty, 58개 ObjectProperty), 7개 제약(Restriction)으로 구성되었으며, ABox는 예시 구축 기준 60개 인스턴스가 생성되었다. 특히 law.go.kr Open API의 XML 응답을 RDF(Turtle)로 변환하는 5단계 파싱 파이프라인을 설계하고, 총 8개 대상에 대해 70개 이상의 패턴 기반 추출 규칙을 적용하여 자동화 범위와 한계를 함께 제시하였다.

제4장에서는 구축된 온톨로지를 실제 업무 시나리오에 적용하여, 법령 개정 영향 탐지, 규범명제 기반 전방추론, 규범명제 기반 후방추론의 세 가지 목적에 대해 작동 가능성을 검증하였다. 첫째, 법령 개정 영향 탐지에서는 ’주4.5 일제’ 사례를 통해 총 23건의 연관 대상(직접 영향 및 간접 영향)을 구조적으로 식별하였다. 둘째, 전방추론(Forward Chaining)에서는 입력 사실 8건, 규칙 8개를 적용하여 3개 라운드에 걸쳐 최종 결론 6건(법 위반 3건, 적법 1건, 행정조치 2건)을 도출하였다. 셋째, 후방추론(Backward Chaining)에서는 해고 적법성 판단을 위해 6단계 역방향 검증을 수행하여 ’부당해고’ 판단을 도출하고,

추가 확인이 필요한 사실까지 명시적으로 제시하였다. 이런 입증을 통해, 법령 온톨로지가 검색 보조를 넘어 실무형 판단지원 기반으로 확장될 수 있음을 확인하였다.

제5장 실증에서는 법령 온톨로지와 LLM을 결합한 AI 법령서비스 LawBrain을 구현하고, 동일한 7개 질의에 대해 온톨로지 기반 RAG와 하이브리드 서치 기반 RAG를 비교·평가하였다. 그 결과, 온톨로지 기반 RAG는 Recall, 관계 추적 성공률, 근거 투명성, 환각 억제 측면에서 유의미한 개선을 보였으며, 전방·후방추론 엔진과 결합한 구조적 추론 체계를 통해 LLM 응답의 설명가능성과 신뢰성을 높일 수 있음을 실증적으로 확인하였다.

2. 결론

본 연구의 주요 성과는 다음과 같이 정리할 수 있다.

첫째, 법령의 구조·관계·규범성·시간성을 함께 표현하는 노동법령 온톨로지 모델을 설계하고, TBox 기준 48개 클래스, 61개 DatatypeProperty, 58개 ObjectProperty, 7개 제약으로 구성된 지식 스키마를 구축하였다. 또한 예시 구축 기준으로 ABox 60개 인스턴스를 생성하여 추론 엔진 입력 기반으로 활용 가능한 수준의 데이터 구조를 확보하였다.

둘째, 법령 개정 영향 탐지 검증에서 '주4.5일제' 사례를 통해 총 23건의 연관 대상을 식별함으로써, 키워드 검색으로 누락되기 쉬운 위임·참조·타법 연결까지 포함한 구조적 영향 탐지 가능성을 확인하였다. 이는 개정 영향 분석의 전수성과 누락 방지 측면에서 온톨로지 기반 접근의 실질적 가치를 보여준다.

셋째, 규범명제 기반 전방추론 구현에서는 입력 사실 8건, 규칙 8개, 3개 라운드, 발화 규칙 6개를 통해 최종 결론 6건(법 위반 3건, 적법 1건, 행정조치 2건)을 도출하였다. 이는 규범명제를 기계 적용 가능한 규칙으로 구성하고, 실행 추적을 통해 근거를 제시하는 법적 판단 자동도출 구조가 실제로 구현 가능함을 보여준다.

넷째, 규범명제 기반 후방추론 구현에서는 해고 적법성 시나리오에 대해 입력 사실 8건, 규칙 적용 6회의 역방향 추론을 수행하여, 절차적 요건 미충족과 해고예고 위법 가능성, 추가 사실 필요를 종합해 "부당해고" 결론을 도

출하였다. 특히 판단 보류(추가 사실 필요)를 명시적으로 제시함으로써, 결론 도출뿐 아니라 추가 확인사항 식별 기능까지 구현하였다.

다섯째, LawBrain 비교평가에서는 온톨로지 기반 RAG가 하이브리드 서치 기반 RAG 대비 Recall(+58.6%p), 관계 추적 성공률(+100%p), 근거 투명성(+2.4점), 환각 억제(-28.6%p) 측면에서 개선을 보였고, 전방·후방추론 및 SPARQL 기반 근거 공개를 통해 법령 서비스에 요구되는 설명가능성과 신뢰성을 강화할 수 있음을 보여주었다.

3. 종합 시사점

본 연구의 가장 큰 시사점은, 법령 온톨로지가 "검색 중심 도구"를 넘어 "판단 중심 엔진"의 기반 인프라로 기능할 수 있음을 입증했다는 점이다. 즉, 법령을 조문 단위 텍스트로 저장·조회하는 수준을 넘어, 규범명제·위임·참조·타법연계·시간효력까지 포함한 구조화된 지식으로 재구성함으로써, 개정 영향 탐지와 법적 판단 자동도출이 가능한 체계로 확장할 수 있음을 보였다.

또한 본 연구는 LLM의 활용 방식에 대해 중요한 설계 원칙을 제시한다. 법령 분야에서 LLM은 단독 판정 주체로 사용하는 것보다, 온톨로지·규칙 기반 판단 계층과 결합하여 질의 해석·용어 정규화·설명 생성을 담당하게 하는 것이 정확성과 설명가능성 측면에서 더 적절하다. 이는 공공 법률행정 서비스에서 요구되는 신뢰성과 감사가능성을 확보하는 데 핵심적인 접근이다.

아울러 전방추론과 후방추론을 함께 구현·검증한 결과는 향후 실무 적용 전략에도 직접적인 시사점을 제공한다. 전방추론은 감독·점검·컴플라이언스 진단처럼 가능한 위반·조치 대상을 넓게 탐지하는 데 유리하고, 후방추론은 상담·쟁점검토·적법성 판단처럼 특정 목표를 중심으로 요건 충족 여부를 점검하는 데 효율적이다. 따라서 실제 시스템에서는 **전방추론(전수 탐지) + 후방추론(쟁점 검증)**의 하이브리드 운영 구조가 바람직하다.

마지막으로, 본 연구는 법령온톨로지 기반 법적 추론 체계가 법률 분야에만 국한되지 않고, 의료·금융·규제 등 도메인 지식의 정확성과 추론 가능성이 중요한 분야로 확장될 수 있는 가능성을 보여준다. 즉, 본 연구는 법령 도메인을 넘어 설명가능한 AI 기반 규제지식 시스템의 일반적 설계 원형으로도 의미를 갖는다.

4. 한계

본 연구는 모델 설계-구축-검증-서비스 실증까지 일관된 흐름을 제시하였으나, 다음과 같은 한계를 보여줬다.

첫째, 검증 사례가 근로기준법 중심(근로시간·연장근로·가산임금, 해고 적법성 시나리오, 주4.5일제 영향 탐지)에 집중되어 있어, 다른 법령군에 대한 일반화 가능성은 추가 검증이 필요하다. 둘째, 규범명제 추출 과정에서 일부 요소(예외·특례·중대성 판단 등)는 여전히 반자동 또는 전문가 검토를 요구하며, 완전 자동화에는 한계가 있다. 셋째, 현재 정량 성과는 기능 검증 중심 지표(연관 대상 수, 규칙 수, 발화 수, 추론 단계 수, 질의 비교 결과)에 초점이 맞춰져 있으므로, 향후에는 전문가 정답셋 기반의 정밀도·재현율 검증이 추가되어야 한다

5. 향후 과제 및 발전 방향

첫째, **법령 범위 확장**이 필요하다. 근로기준법 중심 모델을 산업안전보건법, 고용보험법, 남녀고용평등법 등으로 확대하여, 스키마와 규범명제 추론 구조의 재사용성과 일반화 가능성을 검증해야 한다. 특히 타법 연계가 많은 영역을 포함할수록 개정 영향 탐지의 실효성을 더 명확히 입증할 수 있다.

둘째, **정량 평가체계 고도화**가 필요하다. 현재 제시된 수치 지표를 기반으로, 정밀도·재현율·F1, 검토 시간 단축률, 누락 감소율, 추가 사실 식별 정확도, 설명가능성 평가(근거 추적 성공률 등)를 포함하는 성능평가 프레임을 구축해야 한다. 이를 통해 시스템의 실무 적용 가능성을 객관적으로 제시할 수 있다.

셋째, **규범명제 추출 및 유지보수 자동화를** 강화해야 한다. 패턴 기반 추출과 반자동 검토 구조를 고도화하고, 법령 개정 시 규범명제·링크·시간효력 정보를 자동 갱신할 수 있는 파이프라인(버전 관리, 변경점 탐지, 제약 검증)을 구축할 필요가 있다. 이는 법령온톨로지를 정적 데이터셋이 아닌 동적 지식체계로 운영하기 위한 핵심 과제이다.

넷째, **하이브리드 추론 엔진의 고도화**를 통해 실제 노무·감독·상담 업무 흐름에 부합하는 통합 추론 엔진을 개발해야 한다. 전방추론으로 입력 사실관계에서 잠재적 위반·제재·행정조치를 넓게 탐색한 뒤, 후방추론으로 특정

쟁점에 대한 요건을 선택적으로 검증하는 이중 추론 워크플로를 자동화하고, 추론 과정의 단계별 설명 생성을 강화해야 한다.

다섯째, **ABox 자동 구축 및 사실관계 입력 정규화**를 통해 대규모 법령 인스턴스의 자동 생성과 LLM 기반 규범명제 자동 추출 및 분류 기술을 연구해야 한다. 또한 사용자의 비정형 사실관계 입력을 온톨로지 ABox 인스턴스로 자동 정규화하는 기술을 개발하여, 전문가가 아닌 일반 사용자도 자신의 상황에 적합한 규범 적용 여부를 확인할 수 있는 접근성을 확보해야 한다. 이 과정에서 LLM의 자연어 이해 능력과 온톨로지의 구조적 정합성 검증 기능을 결합하는 파이프라인의 정확성과 효율성을 평가할 필요가 있다.

여섯째, **LLM-온톨로지 통합 서비스의 운영 성능 개선**이 필요하다. 온톨로지 기반 RAG는 정확성과 근거 투명성 측면에서 강점을 보였으나, 응답 시간 측면에서는 최적화 여지가 있다. 따라서 온톨로지 캐싱, SPARQL 쿼리 최적화, 검색 인덱스 개선 등을 통해 성능을 개선하고, 동시에 근거 공개와 추론 트레이스 제공 기능을 유지하는 방향으로 고도화해야 한다.

일곱째, **실무 적용형 사용자 인터페이스와 업무 연계**를 강화해야 한다. 감독・점검・상담・민원 응대 등 실제 업무 흐름에 맞추어 전방/후방추론 화면, 근거 제시 방식, 추가 사실 입력 템플릿을 표준화하고, 법령 개정 알림・영향 대상 추천・검토 체크리스트 생성 등 업무지원 기능으로 확장할 필요가 있다.

6. 맺음말

본 연구는 법령 온톨로지가 단순한 지식 표현 도구를 넘어, **LLM의 추론 품질을 구조적으로 향상시키는 핵심 인프라로 기능할 수 있음을 실증적**으로 보여주었다. 법령은 사회의 보편적 규범이자 공공 지식이며, 모든 시민과 기업이 자신에게 적용되는 규범을 정확히 이해하고 준수할 수 있어야 한다는 점에서, '**적용 가능한 법지식**'으로의 전환은 단순한 기술적 진보를 넘어 법치주의의 실질적 구현에 기여하는 과제이다.

본 연구에서 제시한 온톨로지 기반 접근법은 법령뿐 아니라 의료, 금융, 규제 등 도메인 지식의 정확성과 추론 가능성이 요구되는 다양한 분야로 확장 가능하다. 향후 연구를 통해 법령 온톨로지의 범위를 확대하고, 성능평가를 체계화하며, 실무 적용의 기반을 강화함으로써, 법령 정보서비스가 진정한 의미의 **'법지식 서비스**'로 발전하는 데 기여할 수 있을 것으로 기대한다.

고용노동부
Ministry of Employment and Labor