

# ソフトウェア工学との類似性に着目した立法支援方法（四・完）

角 田 篤 泰

## 第一章 はじめに

### 第一節 概要

### 第二節 背景

## 第二章 ソフトウェア工学の応用例

### 第一節 要求分析

### 第二節 仕様記述

### 第三節 設計方法論

### 第四節 テキスト処理

### 第五節 バージョン管理

### 第六節 再利用

### 第七節 簡易開発

## 第三章 ソフトウェア開発と立法過程

第一節 プログラミング

第二節 ソフトウェア開発過程

一 ソフトウェア開発概要

二 要求分析

三 設計

四 製作（コーディング）

五 試験

六 提供

七 保守

第三節 立法過程との比較と類似性

第四節 非落水型の開発方法

第四章 要求分析－政策の検討と立法事実の収集

第一節 ソフトウェア工学における要求分析の手法

一 要求分析の目的

二 要求獲得手法の概要

三 シナリオに基づく要求獲得

四 要求定義の手法

第二節 政策検討段階への応用

（以上二三五号）

- 一 要求分析手法の応用
  - 二 シナリオを用いた立法過程への要求分析
  - 三 要求表現方法の応用
- 第三節 立法事実の収集
- 第五章 設計－要綱の作成
- 第一節 ソフトウェア開発における設計手法
- 第二節 オブジェクト指向に基づく設計手法
- 一 オブジェクト指向パラダイム
  - 二 オブジェクト指向のプログラム例
  - 三 仕様記述方法
- 第三節 構造の表現
- 第四節 政策立案への応用
- 一 要綱作成過程への応用
  - 二 要綱表現への応用
- 第六章 製作－法令・条例案の作成
- 第一節 ソフトウェア開発における製作
- 第二節 立法過程における製作
- 一 条文作成の支援

（以上二三六号）

二 条文作成の自動化

第三節 非落水型開発の応用

第七章 試験―意見収集と議会審議

第一節 ソフトウェア開発における試験技法

一 ソフトウェア試験の目的

二 ソフトウェア試験の種類

第二節 作成された法令の検査

一 立法過程における試験

二 立法過程におけるテストケース

三 高階テスト技法

四 立法過程における試験支援

五 試験としての議会

第八章 提供と保守―公布・施行・改廃

第一節 公開とバージョン管理

第二節 法制執務支援システムの事例

第九章 立法支援システムと政策知識ベース

第一節 ソフトウェア開発における統合的開発支援ツール

第二節 立法支援システム

(以上二三七号)

(以下本号)

一 立法支援システムのプロトタイプ

二 条例検索システム

### 第三節 法政策知識ベース

一 法政策知識ベースの利用目的

二 法政策知識表現へのアプローチ

三 法政策知識の利用例

四 法政策知識の内部表現例

五 法政策知識ベースの課題

### 第十章 まとめ

## 第九章 立法支援システムと政策知識ベース

本章では前章までに示した各段階のツールを統合するような立法支援システムを提案する。さらに、そのようなシステムの内部で用いられる法政策の構造データを蓄えておくための知識ベースについても言及する。本章第二節と第三節においてこれらを順に示すが、その前に、第一節において、ソフトウェア開発においても、各開発段階のツールを統合したソフトウェアが非常にポピュラーなツールとして利用されているので、それらを簡単に紹介しておく。

## 第一節 ソフトウェア開発における統合的開発支援ツール

ソフトウェア開発において、その開発プロセスを全体的に支援するツールとしては、典型的には、①プログラムを中心に全プロセスを支援するものと、②プロジェクトを中心に全プロセスを支援するものの大きく二種類がある。本節の以降では、これらを順に説明する。

まず、①のツールとしては、「統合開発環境」システムとして、IDE (Integrated Development Environment) と呼ばれる類いのシステムが典型的である。最近では、フリー・ソフトウェアであるにも関わらず、高性能で、広く利用されているEclipse<sup>①</sup>が著名であるが、既に二〇年前には、プログラミング言語処理系のソフトウェアにはIDEが附属しているケースが多かった。例えば、ボーランド社の Borland C++ やマイクロソフト社の Visual Basic など、ウインドウズが実用的に利用されるようになった頃には、既にIDEを取り入れた製品として、パッケージ販売されていた。IDEでは、一個のソフトウェア開発と考えられる場合、実際には複数のプログラムから構成されている、すべてをまとめて一つのプロジェクトとして実施しているものと考え、そのプロジェクトに対応するコンピュータ上の実体としてのプロジェクト・ファイルを作成し、そのプロジェクトが管理すべき、すべてのプログラム、各プログラムの属性情報、プログラム間の参照・制御関係、共有データの状況、開発過程のログなどが、そのプロジェクト・ファイルやそのプロジェクトのフォルダに保持されるように実現されている。さらに、IDEでは、プログラマがソースコードを記述する時は、編集画面内にガイドとなる強調表示やテンプレート表示が現れ、また、既存のプログラムやデータを参照する時も、一覧から選ぶだけで良いようになっていて。こうして、ソースコードを書いたら、ボタンを押すだけで、後は自動で、ソースコードが機械語に変換され、関連プログラムと結合

されてソフトウェアが作成される。その途中でエラーがあれば、そのエラー情報を細かく出力してくれる。上手くソフトウェアが出来上がった場合は、自動的に実行することも可能で、その上でエラーがあれば、その箇所も特定してくれる。このようにIDEは支援ソフトウェアと言っても、あたかも開発環境そのものが提供されるような形で構成されているので、名前に「環境（Environment）」と付くのである。

次に、②に対しては、「プロジェクト管理システム」というタイプのシステムがポピュラーである。このタイプにも様々なシステムが存在しており、商用製品のソフトウェアもあれば、フリー・ソフトウェアでも高機能を備えるものがある。<sup>(3)</sup>これらは、主に、プロジェクトの進捗管理と資源管理を行うものである。ソフトウェア開発に特化したプロジェクト管理システムの場合には、生産物のバージョン管理機能を含んでいるシステムも存在する。プロジェクトを進める際に、WBS（Work Breakdown Structure）と言って、作業を大きなまとまりから、細かい作業へと段階を追って細分化した木構造の表現形態が用いられることが多い。このWBSはプロジェクト管理方法の世界的な標準であるPMBOKの中でも定義されている。<sup>(4)</sup>そこで、多くのプロジェクト管理システムにおいても、このWBSの各作業に対してスケジュール、進捗状況、次の関連する作業へのリンク等を付した図によって、その全体的な状況を示すことが中心的な画面となっている。なお、このようなスケジュールを記載した部分は「ガントチャート」と呼ばれる。このWBSと連動するガントチャートのイメージの例を図13に示す。これらによって、進捗管理を行う。

これらのプロジェクト管理システムにおいて想定されている資源管理とは、作業をする人材や必要な人材、時間、コスト、設備、物品などの利用や消費について、その予定と状況を把握し、割り当てることである。これらは無尽蔵に与えられる訳ではなく、必ずプロジェクトごとに上限のあるものであり、様々な外的要因などから、その調達

がボトルネックとなることもあるものである。そこで、これらの状況も作業進捗と並行に管理する必要があるため、プロジェクト管理システムにも資源管理を支援する機能が広く導入されている。なお、「資源」という見方に立てば、進捗管理の図も、各作業に対して、スケジュールという形で時間を割り当てて、実際にどれくらい消費したか、つまり、時間を要してしまったか、ということを表示しているのであるから、これも資源管理の一種と考えることができる。

注意すべきことは、プロジェクト管理システムは、元々はソフトウェア開発に限らず、どの分野でも、プロジェクトでさえあれば、一般的に広く利用可能なように設計されているので、立法過程の作業もプロジェクト作業の一種であるから、当然利用可能であるということである。さらに、ソフトウェア開発に特化されたプロジェクト管理システムの機能の中にも、バージョン管理機能などにおいて、立法過程の作業に活用可能な技術が含まれている。

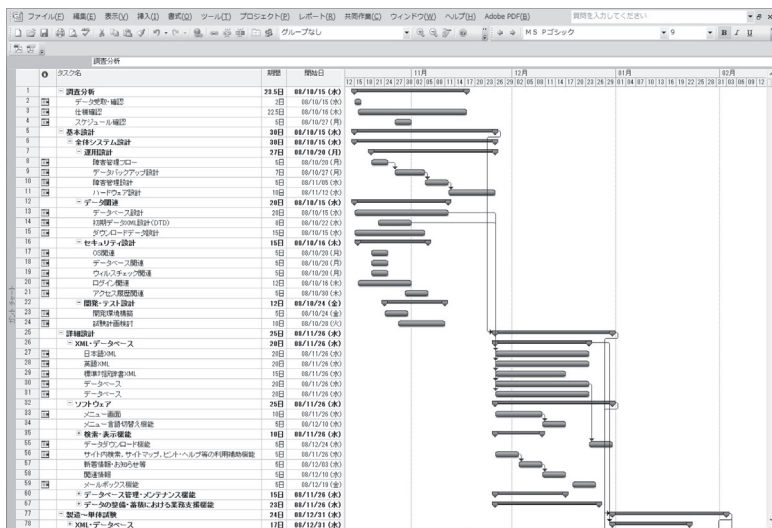


図13 WBSに連動するガントチャートの例



## 第二節 立法支援システム

### 一 立法支援システムのプロトタイプ

本稿で提案するような立法支援システムのプロトタイプとして、筆者は既に立法支援機能を統合的に提供するシステム（eLe: e-Legislation environment）を試作している<sup>6)</sup>。本システムはウェブ上のシステムとして提供されており、そのウェブ上の画面イメージを図14に示す。

このシステムの背景にある着想は、第三章第三節で述べたソフトウェア開発過程との類似性に着目した点である。本システムは、前節で紹介したような開発環境を提供するシステムであり、言わば、立法向けのIDEである。従って、開発の初期に対応する政策設計段階の支援から、プログラミングに対応する条文への書き下し作業までの一連の作業をガイドするものである。そして、各作業局面に応じた様々な支援機能を統合する形で提供され、それらの作業がシームレスに行

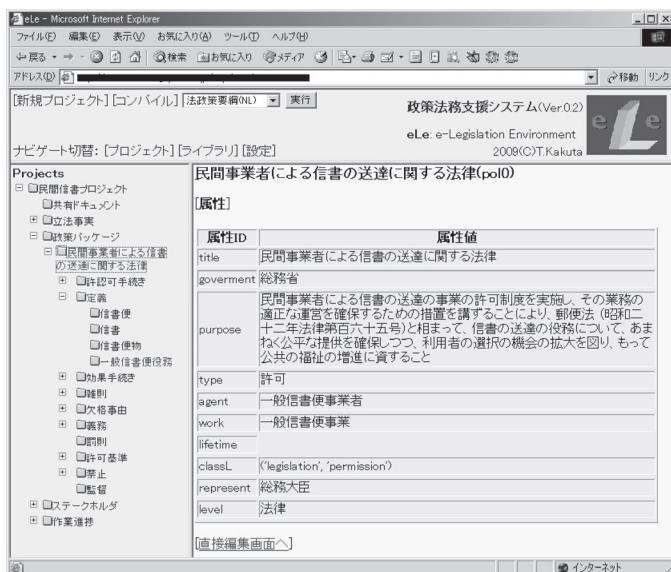


図14 eLeの画面

われるように支援するものである。現在のところ、それらのすべての支援機能が実現されている訳ではなく、開発途上にあるので、新たな支援方法や実現手法が確立されれば、随時それらを導入して、改良を進めるつもりである。なお、本小節では、実現された部分の報告だけでなく、設計部分も含めて、本システムの全体像を示す。

まず、本システムの全体構成であるが、本システムは、外見上は、次の三つのビュー（小画面）から構成されている。

・メニュー（図14の上部の横方向に細長い矩形部分）

この部分には、本システムの基本操作のメニュー項目やシステムのラベルが表示されている。メニュー項目としては、新規プロジェクトの作成・切り替え、要綱形式や条文形式へのコンパイル、各種ファイルへの出力、ナビゲート切り替え等が用意されている。

・ナビゲート・ビュー（図14の下部左側の縦方向に細長い矩形部分）

メニューで切り替えられた各種のナビゲート・ビューを表示する部分である。ナビゲート・ビューには、プロジェクト・ビュー、ライブラリ・ビュー、オプション・ビューの三種の画面が用意されている。プロジェクト・ビューは、プロジェクトごとに、現在立法過程にある法政策に関するデータやその政策の構造を表示する画面である。ライブラリ・ビューは、法政策のテンプレート自体を作成・登録し、編集を行う際に表示される画面である。オプション・ビューは本システムに対する様々なオプションを設定するための画面である。なお、各ナビゲート・ビュー内で選択された項目に応じて、ワークスペースには、対応する様々な作業用の画面が表示される。

・ワークスペース（図14の下部右側の広い矩形部分）

ナビゲート・ビューで選択された項目に応じた作業画面が表示される部分である。実際の作業の中心は、テキストの書き込みや編集であり、これらの作業は、すべてこの画面の中で行われる。

これらのメニュー部分やナビゲート・ビュー部分に表示されている項目が本システムの主要機能であるので、それらを示すと次の通りである。

・プロジェクトの作成・切り替え

一つの法令に関する立法作業が始まると、その一連の作業は「プロジェクト」として捉えられ、これに対応するシステムの内部の実体が作成される。なお、この内部の実体も「プロジェクト」と呼ばれる。これを一旦作成すると、内部的にその実体が保持され、現在進行中の複数のプロジェクトや過去のプロジェクトとともにシステム内部に登録される。そしていつでも、どのプロジェクトにでも、切り替えることができる。このようなプロジェクトの作成や切り替え操作の機能が提供されている。

・コンパイル

テンプレートなどを用いた法政策の設計が終了した後、その設計状態を基に、要綱を自動作成したり、さらに、それを条文形式の法令案に自動変換したりする機能である。（本稿の執筆時点では開発途中の機能である。）

・ファイル出力

内部のプロジェクトの状況や要綱・法令案等をファイルとして出力する機能である。

・ワークスペース

ナビゲート・ビューで選択された項目によって、ワークスペースが切り替わり、その項目ごとに、提供機能は様々である。多くの作業では、その項目に対応するデータを書き込んだり、修正したりすることが中心的操作となる。それらの操作機能を提供している。

・立法事実、ステークホルダのデータベース

本データベースの各データに関連条文の情報を持たせるようにして、高階の試験段階や説明資料を作成する際の作業を円滑にすることが、このデータベースの主要な利用目的である。従って、単にデータを蓄積しているだけではなく、内部では構造化されており、高度な検索にも対応できる。

・ドキュメント・フォルダ

各プロジェクトで共有されるドキュメント類を蓄積しておくフォルダである。

・知識編集・ライブラリ登録機能

現状では、知識ベースへデータを蓄積する際には、操作性の良いインターフェースが用意されておらず、本システムの基盤システムとして動作している Zope<sup>(6)</sup> システムが提供する基本機能だけを用いて直接実現し、蓄積している。将来、エンドユーザにも、ある程度の知識記述ができるような環境を提供するために、現在、設計も含めて、開発中である。

・作業進捗管理機能

ガントチャートを用いたタスクごとの簡単な進捗管理機能である。あるタスクを細かくサブタスクに分けた木構造状のタスクのグラフに添えて、タスクごとのガントチャートを表示するものである。この機能の基本的な

システム構成については、多くのプロジェクト管理システムと同じ考え方を採用している。

次に、本システムが想定している利用イメージの概要を作業順に示す。通常、ユーザは本システムにアクセスする初回にメニュー項目「新規プロジェクト」を選択して、新たなプロジェクトの内部的な実体を作成する。なお、今後はこの内部的な実体も「プロジェクト」と呼ぶことにする。作成時には、ライブラリとして既に登録されている法政策のテンプレート（法政策クラス）の一覧の中から、今回立案する法令のタイプを選んでおく。プロジェクトが作成されたら、メニューの「ナビゲート切替」で「プロジェクト」を選ぶと、ナビゲート・ビューの部分が「プロジェクト・ビュー」に切り替わる。このビューの中で、先に選択しておいたテンプレートから自動的に政策立案に必要な項目が「政策パッケージ」下のサブ項目として並ぶ。また、このビューには、立法事実のデータベースやステークホルダのデータベースへの切り替えメニューも現れる。作業進捗を管理する項目もこのプロジェクト・ビュー内に現れる。例えば、ユーザが政策パッケージの部分で「定義」を選ぶと、法令の中で用語定義をするための画面がワークスペースの部分に表示される。その部分に実際に用語を定義していく。このようにして、各項目の作業が終了したら、メニュー項目の「コンパイル」を実行する。こうすると、その時点までに行った作業対象の内部データを基に、自動的に要綱や条文形式の法令が作成され、ワークスペースに表示される。条文に近い形の法令案要綱やより粗い形の法政策要綱が必要な場合は、プルダウンメニューの中から「法令案要綱」や「法政策要綱」を選んで「実行」を押すと、これらの要綱ファイルが自動的に作成される。このようにして、さらに適宜修正したり、項目を増やしたりして、推敲を重ねることで法令を作成する。なお、この利用イメージは落水型の立法支援を想定したものであるが、実は、非落水型でも利用可能であるし、あるいは、落水型と非落水型の中間的な作

業状態でも利用可能である。

最後に、本システムの実現手法の概要について示す。本稿は、技術者向けの研究報告ではないので、詳細は省略し、実現のためのアプローチと内部構成の概略を簡単に示すだけに留める。本システムは、Zopeというコンテンツ管理機能を備えたウェブサーバ・システムを利用して、このZopeが提供する機能を組み合わせて各機能を実現したものである。内部データはXMLと互換性が保てるような、XMLにおける「エレメント」や「属性」に対応する形式のデータ構造で蓄積され、フォルダのような入れ物とコンテンツとなるようなデータ本体を基本データとして組み合わせられた、構造的なデータとして形成されている。なお、フォルダにもコンテンツにも、「プロパティ」と呼ばれる属性情報をいくつでも付加することができるので、知識表現の中で実体を持つ属性は、このプロパティ付与機能を用いて実現されている。

## 二 条例検索システム

ある程度のまとまった条例に対して単に検索を試みたり、各自治体の条例を入手しただけであれば、既に、インターネットのウェブ上のサービスとして、いくつも提供されている。例えば、『洋々亭の法務ページ』のような著名なサイトもあり、また、多くの自治体もウェブ上で条例を公開している。もちろん、Google等の検索エンジンをを用いて、それらのサイトも含めて様々な形で条例データを検索する方法もある。これらに対して、本研究で提案する条例検索システムでは、単なる検索や条例を入手することに留まるのではなく、立法支援に、より役立つような条例検索機能を提供することを目指している。特に、非落水型のアプローチによって条例作りを進める局面での支援を想定している。

非落水型のアプローチでは、ゼロから法政策を設計する訳ではなく、既存の条例を参考にしたり、それらをテンプレートのよう利用したりして、立法作業を進めることが中心となる。このような作業方法は、いくつかの自治体でのインタビュアーから、国内ではポピュラーなアプローチであると考えられる。さらに、韓国の法制処でのインタビュアーにおいても、この手法に基づいていることを前提に法制執務の支援システムが開発されているという説明を受けている。従って、既存条例を参照・利用する手法は、単なるアドホックな方法と言うよりも、むしろ、広く採用され、一定の効果も期待されている、普遍性のある方法論であると予想される。もし、そうであるならば、我々が開発しているような落水型の手法を意識したe-leのようなシステムだけでなく、このような非落水型のアプローチについてもフォローする必要があるだろう。

そこで、本研究で提案する条例検索システムとして、現在、このような既存の条例を参照して作業を進める際に便利な機能を備えた条例検索システムの開発を進めている。この条例検索システムは、開発開始時点でインターネットのウェブ上で公開されていた例規をできる限り集め、形式を揃えて、自治体横断的に検索できるようにしたものである（第六章の図10を参照）。本稿執筆時には、このシステムに収録されている自治体数は八〇三団体で、収録されている例規数は約四九二、〇〇〇本である。従来、各自治体の条例は様々なウェブサーバに分散して存在しており、さらに、データ形式にも統一性がなかったもので、なかなか思うような、自治体横断的な検索を行うことができなかったが、本システムではそれが可能となっている。なお、『条例Web検索システム<sup>(9)</sup>』というサイトで、ウェブ上で公開されているほとんどの例規を対象にして、横断的な検索を行う機能が提供されている。さらに、このサイトで提供される検索システムはキーワード検索時のノイズや検索漏れを低減する工夫も施されている。<sup>(10)</sup>しかしながら、このシステムは例規の本文を自らのサーバ上に置いておらず、本研究で開発されたシステムのように、

すべての例規データを加工して保持しているものとは実現方式の点で異なる。このような実現方式ゆえに、本研究で開発された条例検索システムは、さらに次のような機能も備えている。

・ 類出項目メニュー

法令の中に類出する項目名や言い回しが整理されてメニュー化されており、これを選択することで、類似状況で、どのような表現が使われているか、ユーザがキーワードを考えて迷ってしまうことなく検索操作を進めることができる。例えば、「欠格事由」や「許可基準」のような典型的な項目名や、「目的」という項目であれば「することを目的とする」や「もって、目的とする」のような言い回しが事前に整理され、メニューに一覧されており、これらを選んで検索し、出力された条文群を比較参照しながら、表現を検討することができる。

・ 文脈検索

文脈検索とは、検索結果を表示する際に、検索キーワードを中心に揃えて、そのキーワードを含む前後を表示するものである。例えば、「事業」というキーワードで検索すると、次のような表示形式で検索結果を返すような検索方式が文脈検索である。

＜法令名 (条文番号)＞

＜検索結果＞

航空法 (134)	： 航空業務、航空運送事業、航空機使用事業又は航空
介護保険法 (194)	：、これらの規定中「事業」とあるのは、「事業 (介
石綿による健康…法律 (69)	： 項中「及び労働福祉事業」とあるのは「労働福祉事



金融商品取引法（附則5）

：その他これに類似する事業に係る資産の経理の状況そ

ガス事業法（2）

：において「簡易ガス事業」とは、一般の需要に応じ

このように表示させることによって、単にその語を含む条文が出力されるのではなく、その前後の文字列が表示されることで、どのような形式的な文脈で使われているかという情報が取得できるため、ユーザはその言葉の使われ方の傾向を直ちに知ることができる。

・自治体属性検索

自治体の持つ属性情報に基づいた検索も可能である。自治体の属性情報とは、例えば、人口、予算規模、地理的条件などを指す。自分達の自治体とあまりにも規模や状況の異なる自治体の条例を調べるよりも、自分達の自治体と似た自治体に絞って検索した方が、実際に利用できるような条例が含まれる可能性が高くなる場合があるからである。また、近隣の自治体と同じような条例を制定する可能性も大きいため、そのような絞込みも可能になっている。さらに、まだ実装はされていないが、設計としては、地理的な条件の中で、ある国道やある河川に沿った自治体に絞って検索できるような機能も加えている。これは、単に地理的な意味での距離的尺度で「近い」だけではなく、このような共通の資源（道路、河川、観光資源、農林・鉱業資源等）を持つことで、足並みを揃えたり、競合する利益を調整したり、あるいは、同じ問題を共有したりすることになり、その意味で「近い」自治体となり得るからである。もし、このように共通の資源がある自治体同士であれば、似たような政策の実施が可能となるケースも多くなるので、似たような条例を考えることも可能であると思われる。そもそも非落水型のアプローチは類似性に基づくものであるから、判例法のように、類似ケースには類似の帰結

をもたらすような法制度を検討することは、自然なことであろう。

このような検索システムを用いることで、発想段階で様々な新たな着想を得たり、記述段階で表現の適切性を多くの自治体の表現の傾向から判断したり、あるいは、表記ミスを発見したりすることも可能になる。さらに、他の自治体がどのような条例を制定しているかという点が、重要な立法事実として提示されるケースが多いので、このように網羅的でシステムティックに類似条例や関連自治体の条例を提示するシステムが提供されれば、そのような立法事実の収集・整理作業は格段に効率を高めることができると考えられる。

## 第三節 法政策知識ベース

### 一 法政策知識ベースの利用目的

まず、本稿における「法政策知識ベース」の定義を示しておく。本稿における「法政策知識ベース」とは、法政策を立案し、条文を作成するまでの間に利用される知識を客観的な形式によって電子的に蓄積したものである。元々「知識ベース」とは、人工知能のような、より知的な情報処理を行う際に、そのためのソフトウェアが参照するデータを構造化して蓄えたものである。通常のデータベースとの直観的な違いを述べておくと、高度（知的）な利用を意図している点と特定のソフトウェアに依存しないデータ表現を採用している点である。例えば、典型的な知識ベースのデータ表現形式、すなわち、「知識表現形式」としては、論理式が挙げられるが、実際、論理式は特にソフトウェアを選ぶものではないし、そもそもコンピュータの出現以前から存在していた知識の表現形式である。

このような表現形式によって記述された知識を蓄えたものが知識ベースである。最近では、セマンティック・ウェブ用のデータとして蓄積されている知識の表現形式には、RDF形式やOWL形式などのXML系の表現形式が利用されている。

次に、本研究で提供する法政策知識ベースの目的と意義について述べる。まず、その目的が本稿で提案しているような立法支援システムの中で活用される点を確認しておく。その意義については、そもそも知識ベースと呼ばれるものは、ソフトウェアを選ばない独立性を持つものであったので、法政策知識ベースの場合も、法政策知識の構造自体が電子的に扱えれば、本研究で提供されるシステムに限らず、様々なソフトウェアで利用される局面が増えるため、自動化できる作業も増え、立法作業の効率化に寄与できることに意義があると考えられる。また、たとえ完全な自動化でなく、半自動化の場合でも、機械的处理を介在させることになるので、人手による手間やミスを軽減することに貢献できるであろう。この他にも、法政策の知識の可視化や客観化にも貢献できる点には大きな意義があると思われる。例えば、通常の条文は、一般の人々にとって読み難いケースが多いが、これは、自動的な図式表示などによる可視化が可能となれば、軽減できるであろう。また、そのような条文を記述する作業自体も、従来はベテラン職員でないと難しいものであったが、これも同様の可視化によって、作業の困難さを軽減でき、より多くの職員に法制執務の一旦を担ってもらえるようになるであろう。さらに、そのような法制執務の作業の技能伝承は、多くの場合、暗黙知的なものになっているので、そのような暗黙知の部分も、知識ベースとして客観化することによって、伝承しやすくなるであろう。このように、各自治体の法制執務作業支援や住民への情報提供という観点からも、可視化や客観化の基盤となる法政策知識ベースを構築する意義はあるであろう。また、より一般的に見ても、客観化が進めば、知識の再利用の可能性を高めることになるので、他の自治体への例規の移植や、

外国語化と組み合わせることによって法制度の海外への移殖も容易になるであろう。もちろん、学術的な分析も円滑になるであろう。

最後に、本節の構成について述べる。本節の以降では、法政策知識一般の議論を行うのではなく、本研究で試作し、現在も改良や拡張を行っている立法支援システム（eLe）の中で用いられる法政策知識ベースについて、その表現方法、システムの外部から見た利用例、システムの内部における表現例、及び課題について、順に示す。

## 二 法政策知識表現へのアプローチ

前節で示した立法支援システム、eLe（以降「本システム」と記す）で用いられる法政策知識ベース（以降「本知識ベース」と記す）において、法政策知識を表現するための手法は、本稿の中でも何度か紹介した、古典的な意味ネットワークや述語論理などとも互換性のあるものを採用する。これは、数学的な基盤や様々な知識表現への変換可能性を担保しているからである。ただし、本稿執筆時の開発段階では、本システム自体がプロトタイプであり、今後の研究・開発の余地が大きい。いきなり決定版の知識表現を提案する訳ではなく、まず、一時的、中間的な知識表現手法を独自に考案しておき、それを用いて知識表現を行うことを目指す。このようなアプローチをとっても、互換性の高い方式を採用することになっているため、後に、表現変換を行うことは容易であり、大きな問題は生じないと予想されるからである。むしろ、この時点で無理に表現方法を固定しようとして、不確定な要素の多い事実を根拠にしたままで設計を進める方が様々な問題を引き起こす危険性が高くなる。

本知識ベースは立法支援システムで利用されるもので、当然その利用局面としては、条文の解釈や法律概念の議論のためではなく、主に条文の生成処理に供するケースを想定している。そこで、本知識ベースの設計に当

たつては、条文を記述する際に必要となるような情報については敏感であるが、一般の法学のような解釈に力点を置くための情報については、あまり関心を払っていないことを断わっておく。つまり、立法者側、あるいは行政の側から道具立てとして使える制度として法令を捉えており、その道具立てとしての機能や属性に着目する立場をとっている。

本知識ベースにおける知識表現は、前述のように、単純な形式変換によって、意味ネットワークや述語論理とも互換性を保てるようなものであり、それゆえ、セマンティック・ウェブなど最近のシステムでも利用できるRDF等とも互換性を保ちやすいものとなっている。本知識ベースでは、それらの多くの知識表現手法と同じように、概念階層の表現も導入している。すなわち、各政策の内部構造の表現だけではなく、ある政策を一つの概念、あるいは一つのクラスと見たとき、他の政策との関係が体系的に分るような、概念階層（クラス階層）の表現である。例えば、「許可制度」については、通常、図15に示したような階層の中で考えられている。<sup>(13)</sup>なお、この図は政策ツールとしての階層を示したものであり、「行政行為」という観点からは、別の分類階層が形成される。図15の中で、「許可制

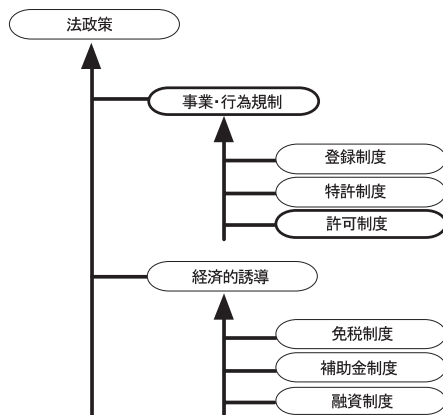


図15 法政策の概念階層図（一部）

度」という制度の上位の概念である「事業・行為規制」であれば、通常は、例えば「所管となる行政機関」や「対象事業・行為」という属性を持っているので、「許可制度」については、これら属性を記さなくても、本システムでは自動的にこれら属性を持っているものと見なされる。これが、第五章第二節でも紹介した「属性継承」というメカニズムである。論理的に考えるなら、上位概念で成り立つ性質が下位概念で成り立つのは当然のことであり、それを機械的にエミュレート（擬制）したものと考えれば良い。

一方、本知識ベースが他の知識ベースと多少異なる点についても記しておく。これは、本知識ベースが開発途上のシステムで用いられることにも起因するものである。本質的な相違点ではないかも知れないが、現状報告として記す。その相違点とは、現時点では、大きく二つのタイプの知識を持つことである。そのタイプとは、法政策のクラス（概念）構造の知識と法令テンプレートの知識である。前者はより深層の知識と見ることができ、後者はより表層の知識と見ることができる。法政策のクラス構造の知識は、各政策の属性や政策間の関係性などが記述されたものである。一方、法令テンプレートの知識は、法令の条文構成としての表現上の構造が記述されたものである。後に議論するように、これらの関係性は、直観的には一貫性が保たれており、見え方の違い程度のはずであるが、もし、そうならば、本来は相互に変換可能なはずである。しかしながら、まだ、その実証が示された訳ではない。そこで、現在のところ、どちらか一方にすることはしないで、利用者の便宜を図って、両者を用いるという二本立てのアプローチを採用している。ただし、このアプローチでは、定型的なものの場合、特に問題がないが、新規性の高い政策を立案する場合にはネットワークとなる問題を孕んでいる。例えば、新たな属性を付加したとき、条文構成上は、どこにその属性を反映した記述が現れるべきか、単純な法則性がある訳ではなさそうなので、アドホックに決める必要性が生じるかも知れないからである。その結果、表現の一貫性がなくなるかも知れない。幸い、実

際には、自治体の例規においては、定型なものが多いので、当面は、そのような定型のテンプレートを参照すること、ある程度は対応できると思われる。そこで、当面はこれらの二つのタイプの知識を前提としながら、研究・開発を進めている。

表層的な部分に着目している法政策のテンプレートも、概念構造の知識と同様に、本システム上では「クラス」と呼ばれ、ライブラリの一部として蓄積されている。このライブラリからユーザが用いる法政策のクラスを選択すれば、そのテンプレートがワークスペースに用意される、という形で作業が進められる。こうして、内部的には前述のように知識が二重化されていても、そのようなことをエンドユーザが意識することはなく、作業を進めることができる。

なお、エンドユーザ向けには、当面、先に筆者らで用意した典型的なテンプレートをいくつかライブラリに登録した状態で本システムを提供するつもりである。エンドユーザがクラス定義自体やテンプレート自体を作成しようとしても、簡単な作業とはならないことが予想されるからである。

このテンプレートは、ユーザにとっては一種の準則のようなものとして提供されることになる。もちろん、余力のある人達が協力し合って、法政策のクラスを作成することができれば、それが最も望ましい。さらに、このような共同作業を行う環境やデータの共有環境が提供されれば、法政策のクラス作成を促進することができるであろう。このような共有環境をウェブ上に提供することも本支援システムの重要な側面である。そして、このような環境が用意されれば、他の自治体、中央省庁、研究者、ボランティア等が協力し合うための客観的な共有基盤が構築されることにもなる。

## 三 法政策知識の利用例

まず、本システムにおける、法政策の概念階層の表示に付いて記す。前小節で説明したように、法制度には概念階層があり（図15参照）、さらに、各制度は様々な属性を持っている。それらの様々な属性は、本システムのナビゲート・ビューにも示されている（図14の画面の左下の縦長のエリア参照）。このような構造がウェブ画面上に自動的に表示されるのは、システム内部に法政策の知識ベースを持っているからである。もちろん、開発に余力があれば、さらに見やすいグラフィック表示を行うことも可能である。いずれにせよ、人手によって直接、図式を描画したのではなく、システム側で、内部の知識ベースを参照して、その知識表現に基づいて自動的に描いている点が重要である。

次に、本システムにおける、定義構造の取扱いについて示す。本稿の事例としては、「民間事業者による信書の送達に関する法律」の一部を想定したものを利用している。ここでは、その中に現れる「一般信書便役務」の概念定義を例にして説明する。この法律内の該当部分（第二条第四項）は次の通りである。ただし、括弧書き部分は省略している。

この法律において「一般信書便役務」とは、信書便の役務であつて、次の各号のいずれにも該当するものをいう。

一 長さ、幅及び厚さがそれぞれ四十センチメートル、三十センチメートル及び三センチメートル以下であり、かつ、重量が二百五十グラム以下の信書便物を送達するもの

二 国内において信書便物が差し出された日から三日以内に当該信書便物を送達するもの



本システムにおいて、もし、この部分に対応する「一般信書便役務」という概念が定義されていた場合、ナビゲート・ビュー上で定義項目として「一般信書便役務」を選択すると、ワークスペース上には図16のように、その定義構造のテンプレートが表示される。これは内部の知識表現を基にユーザに分りやすく可視化した画面である。

この画面の中で、「三」記号の右側が、その定義内容である。持つべき属性とその値の制約をすべて満たすものが「三」記号の左側の「概念名」で表現される概念であるという定義関係を表している。「：」記号の左側が「属性名」であり、右側が「属性値」である。注意すべき点として、属性値に具体的値が入る場合もあるが、個体の属性記述ではなく、概念の定義であるため、ほとんどの場合、属性値としては、具体値ではなく、その値として可能な範囲（変域やタイプ）が記入されている点である。すなわち、実際にこの法令の適用を受ける個々の一般信書便役務にとっては、例え

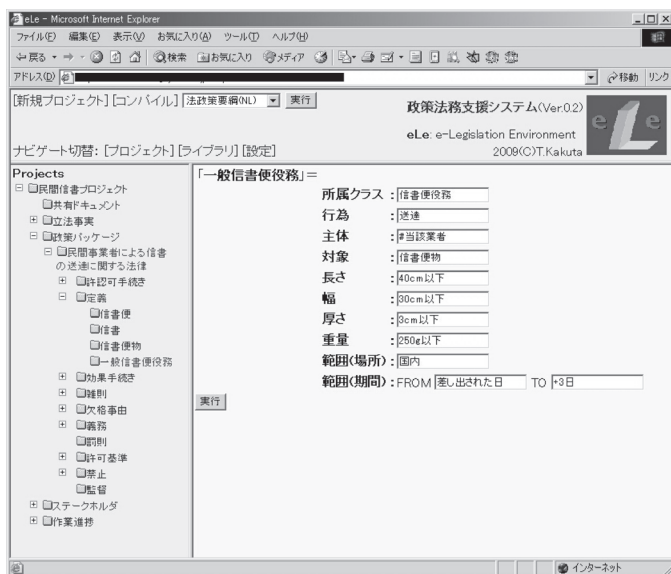


図16 eLeにおける「一般信書便役務」定義画面

ば、「対象」という属性ならば、「信書便物」という概念ではなく、個々の信書便物の実体を表す何らかの識別子が入ることになるし、「長さ」の属性も、その個々の信書便物の実際の長さ、例えば、それが25cmならば、その値が入ることになる。ここには、信書に該当する個体として可能な性質の範囲を記述することになるので、通常は、固定的な数値や個体を示すような識別子、あるいは固有名詞が入るケースは少ない。なお、本システムでは、このような定義内容の中に現れる属性項目欄自体を追加することも、削除することもできる。

ここで、この定義内容の中で、特殊な箇所について説明しておく。

・「所属クラス」の属性については、他の属性と異なり、上位クラスとなるクラスの名前を記入する必要がある。しかも、上位クラスであれば何でも良い訳ではなく、最もその概念に近い（下の）上位クラスでなければならぬ。例えば、「ヒト」を「知的な生物」「知的な動物」「知的な哺乳類」と定義させるのではなく、「知的な霊長類」と定義させるようなものである。

・「主体」の属性値の中に「#」という記号が含まれているが、これは、予約記号としてシステムに認識される記号であり、変数のように、他の個所で定義されたり、値が挿入されたりしている言葉について、その内容を参照する、という指示書きを意味している。このように直接言葉を参照する場合だけでなく、例えば、他の法令を参照する場合などにもこの表記法を用いる。

・「範囲（期間）」となっている部分のように、細かく記述するための定型的な様式については、システムの方で事前に用意しており、それを選べば、このように詳細な定型的記述も可能になる。

最後に、実体的な規定の部分で、テンプレートとして扱うことができるような定型構造の一部の例を示す。これは、本システムのワークスペースに表示される許可制度のテンプレートの画面の内容であるが、システムのハードコピーを用いると見難くなってしまうので、次のように、その内容を取り出してテキスト化して示しておく。

```
所管：[総務省      ]
許可対象事業：[一般信書便事業  ]
許可対象事業者：[一般信書便事業者  ]
許可条文：[#6  ]

事業の許可：
  許可を受ける者：[一般信書便事業を営もうとする者  ]
  許可主体：[総務大臣  ]
  許可の申請：[#7  ]
  申請書提出による：[YES]
  申請書提出先：[#許可主体  ]
  申請書内容：
    ・氏名・名称：
    ・住所
    ・法人の場合、その代表者の氏名：
```

- ・次に掲げる事項に関する事業計画（対象となる方法を列挙）
  - (1)：[信書便物の引受けの方法] ]
  - (2)：[信書便物の配達の方法] ]
  - (3)：[(1)及び(2)に掲げるもののほか、信書便物の送達の方法]
- ・他に事業を行っている場合、その事業の種類
  - 氏名等の変更：[#7]
  - 欠格事由：[#8]
  - 禁固・懲役：[1年以上]
  - 禁固・懲役経過：[2年以内]
  - 取消経過：[2年以内]
  - 許可の基準：[#9]
  - 氏名等の変更：[#10]
  - 事業計画遵守義務：[#11]
  - 事業計画変更方法：[#12]
  - 事業譲渡：[#13]

この画面の中で、角括弧の部分のみがユーザによって記入される部分である。そこだけ書き込めば、対応する条文は自動的に作成される。なお、「#」記号は参照する項目名や条文番号を表す。このテンプレートに対応して自

動出力される部分の条文は次の通りである。

（事業の許可）

第六条 一般信書便事業を営もうとする者は、総務大臣の許可を受けなければならない。

（許可の申請）

第七条 前条の許可を受けようとする者は、次に掲げる事項を記載した申請書を総務大臣に提出しなければならない。

- 一 氏名又は名称及び住所並びに法人にあっては、その代表者の氏名
  - 二 次に掲げる事項に関する事業計画
    - イ 信書便物の引受けの方法
    - ロ 信書便物の配達の方法
    - ハ イ及びロに掲げるもののほか、信書便物の送達の方法
    - ニ その他総務省令で定める事項
  - 三 他に事業を行っているときは、その事業の種類
- ２ 前項の申請書には、事業収支見積書その他総務省令で定める事項を記載した書類を添付しなければならない。

(欠格事由)

第八条 次の各号のいずれかに該当する者は、第六条の許可を受けることができない。

- 一 一年以上の懲役又は禁錮の刑に処せられ、その執行を終わり、又は執行を受けることがなくなつた日から一を経過しない者
- 二 一般信書便事業又は特定信書便事業の許可の取消しを受け、その取消しの日から二年を経過しない者（当該許可を取り消された者が法人である場合においては、当該取消しに係る聴聞の通知が到達した日（行政手続法（平成五年法律第八十八号）第十五条第一項の通知が到達した日（同条第三項により通知が到達したものとみなされた日を含む。）をいう。）前六十日以内にその法人の役員であつた者で当該取消しの日から二年を経過しないものを含む。）
- 三 法人であつて、その役員のうちに前二号のいずれかに該当する者のあるもの

(氏名等の変更)

第十条 一般信書便事業者は、第七条第一項第一号又は第三号に掲げる事項に変更があつたときは、遅滞なく、その旨を総務大臣に届け出なければならない。

(事業計画の遵守義務)

第十一条 一般信書便事業者は、その業務を行う場合には、第六条の許可に係る事業計画（以下この章において単に「事業計画」という。）に定めるところに従わなければならない。

（事業計画の変更）

第十二条 一般信書便事業者は、事業計画の変更（第三項に規定するものを除く。）をしようとするときは、総務大臣の認可を受けなければならない。

- 2 第九条の規定は、前項の認可について準用する。
- 3 一般信書便事業者は、総務省令で定める軽微な事項に関する事業計画の変更をしたときは、遅滞なく、その旨を総務大臣に届け出なければならない。

（事業の譲渡し及び譲受け等）

第十三条 一般信書便事業の譲渡し及び譲受けは、総務大臣の認可を受けなければ、その効力を生じない。

2 一般信書便事業者たる法人の合併及び分割は、総務大臣の認可を受けなければ、その効力を生じない。

ただし、一般信書便事業者たる法人と一般信書便事業を営まない法人が合併する場合において一般信書便事業者たる法人が存続するとき、又は一般信書便事業者たる法人が分割をする場合において一般信書便事業を承継させないときは、この限りでない。

- 3 第八条及び第九条の規定は、前二項の認可について準用する。
- 4 第一項の認可を受けて一般信書便事業を譲り受けた者又は第二項の認可を受けて一般信書便事業者たる法人が合併若しくは分割をした場合における合併後存続する法人若しくは合併により設立された法人若しくは分割により一般信書便事業を承継した法人は、第六条の許可に基づく権利義務を承継する。

ユーザの記入量に比較して、かなり多くの条文を自動作成できることが分るであろう。なお、この条文中では他の許可制度に置き換えることができる可変部分には下線を引いている。その部分を拾い出して、ワークスペース部分のテンプレートが構成されていることが分るであろう。

ここで、「氏名等の変更」「事業計画遵守義務」「事業計画変更方法」「事業譲渡」等の条文についてはテンプレートでは、ほとんど何も書いていないのに、条文が作成できることに気が付くであろう。これは、「許可対象事業者」と「許可主体」を参照する以外に可変部分はなく、後の部分は、自動的に決まってくるものであるから、条文番号以外は、特にユーザがテンプレートに書き込まなくても、自動的に作成できるのである。条文番号自体も、ユーザがテンプレート上で割り付ければ、後は相対的な参照関係から自動的に付すことができる。なお、本来第九条となる「許可の基準」に関する条文が出力されていないが、これだけは許可される事業の内容に大きく依存してしまい、このような表層的なテンプレートから自動生成することは困難なため、ここでは出力されていない。

#### 四 法政策知識の内部表現例

本システムの内部で用いている法政策知識の表現方法について示す。例として、仮に「民間事業者による信書の送達に関する法律」の定義の前半部分を内部表現形式で書き下した場合を考える。まず、その部分の条文は次の通りである。なお、第三項の括弧書きと第四項第二号の括弧書きの部分、及び第五項以降は省略している。

第二条 この法律において「信書」とは、郵便法第四条第二項に規定する信書をいう。

2 この法律において「信書便」とは、他人の信書を送達すること（郵便に該当するものを除く。）をいう。



3 この法律において「信書便物」とは、信書便の役務により送達される信書をいう。

4 この法律において「一般信書便役務」とは、信書便の役務であつて、次の各号のいずれにも該当するものをいう。

一 長さ、幅及び厚さがそれぞれ四十センチメートル、三十センチメートル及び三センチメートル以下であり、かつ、重量が二百五十グラム以下の信書便物を送達するもの

二 国内において信書便物が差し出された日から三日以内に当該信書便物を送達するもの

この条文に対し、本システム用に独自に設計した内部表現形式（中間言語による表現形式）に従うとすれば、次のように書き下すことができる。なお、各項について、内部表現形式と対応する述語論理式による表現形式を順に記している。また、「※」印の部分に簡単な説明も付した。

【第一項：内部表現】

“信書” := @信書 <- 郵便法第五条第二項

※ 「:=」の左辺の概念を右辺の記述によって定義している。

※ 「@」記号を頭に付した概念が定義概念の所属するクラスである。

※ 「<-」表記の部分は、その右側に参照先を記すことで、その参照先の中にある指定概念の定義を流用することを表す。

【第一項：論理式表現】

信書 (x)  $\leftrightarrow$  郵便法第五条第二項の信書 (x)

※「 $\leftrightarrow$ 」は論理的な同値関係を表す論理結合子とする。述語論理式では、括弧内の主語を先頭の述語で言明する命題の形をとるので、左辺は、x という個体が「信書」である、という状態を表し、それと同値な命題として、その x という個体が「郵便法第五条第二項の信書」である、と表記しており、これによって「信書」の定義を与えている。

※左辺の「信書」は「本法律における」という修飾語が省略されていると考える。この文脈の違いは、システムの方で判断させるものとする。

※本論理式以降、煩雑な表記を避けるため、変数はすべて全称限量されているものとする。

【第二項：内部表現】

"信書便":=@ [送達する (to[信書 of 他人])]!郵便

※角括弧で括られた部分は、述語で表された行為や状況を表現する命題自体を実体化して英文の T H A I 節のように、名詞化するための表記である。

※「!」記号の右側には、例外となる概念を記す。

※「|」記号の左側には、その述語の持つ該当する格の識別名を記し、右側には、その格の値を記す。

※ of の左側の対象の所有者を右側で表す。

【第二項：論理式表現】

信書便 (x)  $\longleftrightarrow$  送達する (x)  $\wedge$  to(x,y)  $\wedge$  信書 (y)  $\wedge$  主体(x,a)  $\wedge$  他人(a,b)  $\wedge$  of(b,y)  $\wedge$   $\neg$  郵便 (x)

※ 「信書便」という行為自体を表す x が、この定義から「送達する」こと自体を指すので、「送達」という概念の一実例として x を考え、その送達行為 x の対象 (toで表記) が当該信書 y であり、同様に x の「主体」である a にとって、他人であるところの b の所有物 (ofで表記) が当該信書 y である、ということを示している。最後に当該行為は「郵便」概念の一例とはならないことが条件として加えられている。「 $\neg$ 」は否定を表す論理演算子である。なお、当該事業者を指すと考えられる主体 a は、条文の中には直接出現しない。実際に、述語論式に変換する際には、このような補足も自動化する必要がある。

【第三項：内部表現】

"信書便物" := @信書\* with ([送達する (to信書\*)] as 信書便；役務)

※ with は、その左側の概念を右側の名詞化された命題によって、関係代名詞的に限定するための表記である。

※ as は、その左右の概念が等価関係であることを示すための表記である。

※ 「:」記号によって、その左側の概念が所属するクラスを右側に付記するための記法である。

※ 「\*」記号は、その左側の概念が、同一定義文中で、同じ個体を指すことを表す。直観的には、指示語のような使い方であり、指示される側にも、「\*」を付して明示的に同じものであることを表現して

いる。同一文中で同じ概念が、別の個体に具体化され、さらに指示される場合には、「\*1」や「\*2」のようにIDとなるような番号を振っておけば良い。

### 〔第三項：論理式表現〕

信書便物 (x)  $\longleftrightarrow$  信書 (x)  $\wedge$  送達 (y)  $\wedge$  役務 (y)  $\wedge$  信書便 (y)  $\wedge$  to (y, x)

※ 「信書便」はその定義から「役務」の一種なので、「役務 (y)」の条件は、意味的に冗長であるが、中間言語に現れている制約なので、一応表記している。

### 〔第四項：内部表現〕

"一般信書便役務" := @信書便\* : 役務 & @ [送達する (to|信書便物\* with 信書便\* &

[長さ  $\leq 40\text{cm}$  & 幅  $\leq 30\text{cm}$  & 厚さ  $\leq 3\text{cm}$  & 重量  $\leq 250\text{g}$ ]]

& @ [送達する (to|信書便物\* with [差し出し (in|国内, time|日\*)],

within| (3日 from 日\*))]

※ 「@」記号を用いた定義文を複数「&」で結合することによって、複数の文によって条件を規定することが表記される。そこで、この場合は、第一号と第二号となる各部分が「&」によって分けられている。

※ with によって修飾する側の表記が複数存在する場合も「&」で結ぶ。

※ 意味的に命題となる条件が加わる場合も「&」記号で結ぶ。

※「 $\leq$ 」は中置きされた述語と考える。

※ from はその左側に記述される時間や距離に対して、その右側に、その起点を記述する。

#### 〔第四項：論理式表現〕

一般信書便役務(x)  $\longleftrightarrow$  信書便(x)  $\wedge$  役務(x)  $\wedge$  送達(x)  $\wedge$  to(x,e)  $\wedge$  信書便物(e)  $\wedge$  長さ(e,y1)  $\wedge$

y1  $\leq$  40cm  $\wedge$  幅(e,y2)  $\wedge$  y2  $\leq$  30cm  $\wedge$  厚さ(e,y3)  $\wedge$  y3  $\leq$  3cm  $\wedge$  重量(e,y4)  $\wedge$

y4  $\leq$  250g  $\wedge$  差出(z)  $\wedge$  起点(z,国内)  $\wedge$  時点(z,d1)  $\wedge$  時点(x,d2)  $\wedge$  d2 = d1 + 3日

※「 $\leq$ 」「 $=$ 」は中置きされた二項述語、「 $+$ 」は中置きされた二項関数と考える。

※自然数の加法の実計算は、第一階述語論理の公理系から外れるものであるが、ここでは、表現のために導入している。

このような内部表現形式であれば、条文表記上のレトリック情報が保てるので、機械的操作のみでも、ほぼ元の条文に戻すことができる。また、このような内部表現形式から、述語論理式に機械的に変換することも可能である。ただし、一旦述語論理式に変換してしまった場合は、条文表記上のレトリック情報が落ちてしまうので、述語論理式から元の条文形式のような日本語文を自動的に作成することはできず、冗長で読み難い条文となるか、条文表現の選択肢が多くなってしまう、事実上、自動化できないことになる可能性もある。そこで、当面、このような中間言語による内部表現を媒介にして、述語論理式にも、自然言語による条文にも、自動変換可能な状態を保つ方式を採用しておくことにした。

ここで注意すべきことがある。本来、述語論理式化した場合には、数学的な意味論から見ると、述語の引数の値となるものは、個体として扱われるべきものである。しかしながら、引数に現れる名詞を見ると、明らかに、特定の個体（固有名詞や、英語で定冠詞の付いた名詞）を指している訳ではなく、不特定の個体（英語で不定冠詞の付いた名詞）であるので、意味的に特定の個体を指している訳ではない。むしろ、先に出現した同一名詞は不定冠詞的な修飾を受ける個体であり、その後は特に修飾が付かない限り、その名詞と同じ名詞に対しては、最初に現れた不特定の個体を指すと考えるべきである。すなわち、あたかも変数のような役割で用いられていると考えることができる。しかしながら、このような解釈方法は通常の述語論理の体系にはないので、運用する側の解釈でその点を補う必要性が生じる。

また、この表現形式においては、述語論理式化する際の特徴として、所属クラスを表す述語には一項述語を用いそれ以外の述語は二項述語で表現するようにしておくものとしている。なぜならば、二項述語であれば、その述語を「関係」リンクと見なして、二つの引数が示す実体をノードとして結び付けていることになり、意味ネットワークや RDF への変換が容易になるからである。ただし、簡単に二項述語にならないケースもある。それは、例えば「当該事業開始前に当該事業者が総務大臣に当該事業許可を申請する」を表す「申出」（出願）（提出）の辞書情報として、述語の「格」情報が登録してあれば、その格のラベルを述語名とすることで、二項述語化を図ることができる。ここでの「格」とは、元々 C・J・フィルモアによつて提唱された「格文法」に現れる概念である。<sup>(5)</sup>簡単に言えば、中心となる動詞に対して、どのような役割を持つかという情報である。この格情報が事前に与えられているのであれば、この例の場合も、次のように二項述語化できる。

行為者（当該申請、当該事業者）

対象（当該申請、当該事業許可）

到達先（当該申請、総務大臣）

時点（当該申請、当該事業開始前）

この「行為（agent）」「対象（object）」「到達先（goal）」「時点（time）」が、元々の「申請する」という述語の持つ格を表しており、これらを述語化して表現することで、二項述語化を図っているのである。このためには、元々の述語のどの引数がどの格に該当するかという一種の辞書情報が事前に必要である。その情報によって、自動的にこのような二引数の述語の命題群に変換させることも可能になる。なお、実際の変換時には、これら四つの命題に加え、「世間（当該行為）」という命題も付加されて、「当該行為」という個体が所属する概念が「申請」であることを示す。また、このような格情報は、リンク表現用の二項述語化のためにのみ保持されている訳ではない。自然言語文を機械的に生成する際にも、格助詞を自動的に選択させる必要があるが、その時に参照する情報源ともなる。

注意すべき点は、このような方法で二項述語に変換する場合、元々の述語が、「関係」的存在から「実体」的存在に変換されて扱われることになるので、解釈上の配慮が必要になってくる点である。なぜなら、関係を実体化するということは、意味的には高階の操作をしていることになるからである。先の例で言えば、「申請する」という述語で表された命題の状態が、「申請」という概念の一個体として扱われるようになったことであり、意味的には、命題自体を命題の操作対象にしていることになるので、矛盾する自己言及のようなパラドキシカルな状況も含まれ

てしまう可能性があるからである。

最後に、このような二項述語を用いて意味ネットワーク化する場合についても簡単に示す。述語をリンクとし、引数の定数をノードと考えて、そのノードを引数とする別の述語をさなるリンクとして結ぶことによって、徐々にネットワークを構成することができる。例えば、 $p(a,b)$ と $q(b,c)$ という二つの述語表現であれば、最小のネットワークとして、図17のように結合できる。

このように二項述語を用いることで、意味ネットワーク表現だけでなく、RDF表現の基本単位を構成する三つの要素である、主語（リソース）、述語（プロパティ）、目的語（プロパティ値）に対応付けることもできる。これら三つから成り立つ基本単位を「RDFトリプル」と呼ぶ。二項述語表現も、RDFトリプルに当てはめて、第一引数を主語として、述語をまさに述語として、第二引数を目的語として見れば、この三つを基本単位として構成されているので、RDF表現への変換も容易に可能となることが分る。

## 五 法政策知識ベースの課題

現在、本知識ベースの開発途上で直面している二つの課題について記しておく。これらは、技術的に回避できる可能性もあるが、設計段階での前提事項にも関わる核心的な問題なので、ここで示しておく。

第一の問題は、個体と概念（クラス）の区別に関する問題である。この問題は、現在、自然



図17 二項述語に基づくネットワーク



な日本語条文に容易に変換できるように、表層的にほとんどの単語を概念として扱っているため、実は、個体と概念の区別を曖昧なままにしているというものである。将来、正確な推論を行わせたり、自動的に正確な文章を生成したりするためには、素朴な見通しとしては、この区別をはっきりさせた方が良いと思われる。一方で、このような曖昧なままの概念でも、日本語が便利に利用されているメリットと考えることができるので、何らかの有用性を内在しているのかも知れない。そこで、本当に、このような曖昧なものをはっきりした表現に単純に置き換えてしまう方向性が良いものなのか、現在のところ明確に判断することができない。

第二の問題は概念構造と定型文のようなテンプレートとの関係に関する問題である。これは、定型文の構造に基づいて表現しているテンプレートでは、深層の意味構造ではなく、比較的表層に近い部分の構文的な制約を扱っているので、本当に、意味ネットワークのような、より深層の概念構造から、自動的にシームレスな変換を実施することで、該当するテンプレート通りの条文が整合的に生成できるか、という疑問である。かつてN・チョムスキーの生成文法<sup>(46)</sup>において、深層構造からの形式的な変換のみで、すべてを解決することができなかったように、<sup>(47)</sup>多分、何らかの別の情報やヒューリスティックを導入しないと、深層の意味に対して、整合的に条文を生成するのは難しいかも知れない。すると、今後、知識ベース内の知識表現の記述から条文生成までのプロセスを自動化するようなシステムを実現しようとしても、シームレスに連続的な自動変換で進めることができず、結局人手を介在させたり、アドホックなプロセスを挟み込んだりすることにならざるを得ない可能性がある。これでは、システム設計の立場からは、好ましい実現方式とは言えない。すなわち、見通しが悪く、整った設計にならないと思われるのである。

もし、知識ベースから定型的な条文が生成されることが保証されている上で、その一種のショートカットとして、定型文のテンプレートを蓄積しておき、それを利用するのであれば、知識ベース内の知識やシステムの整合性は維

持されていることになる。しかしながら、その保証がない場合、すなわち、概念構造とテンプレートとの間の関係性が分断されて二重化されていると、本質的に同じものがショートカットされているとは言えなくなり、そのテンプレートが知識ベースを反映しているか否かが分らなくなる。当然、整合性に疑問を持たざるを得ないことになる。そこで、この第二の問題は、この保証を確保しなければならない、という問題設定と考えることができる。言い換えると、もし、定型文のテンプレートが素朴に概念構造からの変形で生成できるものではなく、特別な操作が入り込んで、その作用があることによつて、ようやく生成できるものであるとしたら、その作用は、いわゆるヒューリスティックによるものであり、それは法制執務の暗黙知の部分である。これが本当に存在するのか否か、存在するのであれば、それを特定しなくてはならないという課題でもある。もちろん、実際に、そのような暗黙知が存在したとしても、本研究によつて、それが間接的に浮き彫りになるのであれば、法制執務の技能の客観化に貢献できるかも知れないので、この問題設定自体は意義深いものになると考えられる。いずれにせよ、本研究の今後の大きな課題の一つである。

本節で示した二つの課題は、元々述語論理への表現変換手続きを保証することによつて、数学的な意味論を担保しようという企図から生じたものである。しかしながら、述語論理との関係性を保つのではなくて、条文で使われる語彙や表現自体の整合性を保証するに留めたとしても、本研究のように、表現の客観化や表現の支援をすることが目標であれば、実は、問題となるケースは少ないかも知れない。この点の議論についても、今後の課題である。

## 第十章 まとめ

本稿はソフトウェア開発の手法を立法過程の作業支援に応用するための方法を提案したものである。各段階の作業局面、あるいは、全体を通した作業局面で応用可能な手法を紹介し、必要に応じて、情報処理分野の知見をヒントに新たな方法論を提案した。これらの中で、本稿執筆時の段階では、自治体や中央省庁で実際に導入されている技術や手法はまだ僅かである。しかしながら、本研究に協力して頂くことになっている自治体は現在も増えており、我々が提供するシステムのプロトタイプの動作や要求仕様についての意見を頂くなど、この手法を広める試みを推進中である。

本稿で示したように、多くの応用手法は具体的に実現可能なものであるが、実際に現場で容易に利用できるようなものになっているか否かは、現段階では不明な点も多い。そこで、自治体の現場の方々の協力を得ることで、より実践的な研究開発を行うことができると考えている。これが本研究プロジェクトの当面の最大の課題である。

本稿のように、我々が展開しているプロジェクトのより具体的な目標や技術的な説明が示され、さらに広めることができれば、今後、この分野に、新たな研究者や開発者も参入できる可能性が広がり、ひいては、立法過程の作業支援環境が豊かに発展していくことにつながると思われる。

### 注

- (1) <http://www.eclipse.org/> 参照。

- (2) 例えば、マイクロソフト社製の Microsoft Project というソフトウェアがある。

- (3) 例えば、Trac、Redmine、OpenProj等のフリー・ソフトウェアがある。
- (4) Project Management Institute, "A guide to the project management body of knowledge (PMBOK guide) 3rd ed." (2004).
- (5) 角田篤泰、齋藤大地、外山勝彦「ソフトウェア開発過程との類似性に基づく立法支援システム」人工知能学会第二三回論文集(二〇〇九年) 2F2、4・1・4・4頁。
- (6) <http://www.zope.org/> 参照。
- (7) <http://www.hi-ho.ne.jp/tonita/> 参照。
- (8) 大韓民国の「法制処」とは日本の内閣法制局に似た組織である。二〇一〇年一〇月一八日にインタビューを実施した。英語版サイトは<sup>4)</sup><http://www.moleg.go.kr/english/> 参照。
- (9) <http://jorel.sis.keio.ac.jp/> 参照。このシステムでは、二〇一〇年三月六日現在で、一三四二自治体の九七四、三四一件の例規の検索が可能である。
- (10) 原田隆史、青木淳一「例規の構造を利用した例規横断検索の高度化」情報ネットワークレビュー九巻二号(二〇一〇年)三九一五三頁。
- (11) <http://www.w3.org/RDF/> 参照。
- (12) <http://www.w3.org/TR/owl-features/> 参照。
- (13) 大島稔彦『法令起案マニュアル』ぎょうせい(二〇〇四年)二四二一三三七頁。
- (14) 幸田雅治、安念潤司、生沼裕「政策法務の基礎知識」改訂版、第一法規(二〇〇八年)六二頁。
- (15) C.J.Fillmore, "The Case for Case", *Universals in Linguistic Theory* (1968) 1-88.
- (16) N. Chomsky, "Aspects of the Theory of Syntax", M.I.T.Press (1965).

(17)

長尾真『言語工学』昭晃堂（一九八七年）一五―一七頁。

