法令工学の可能性と課題

片山卓也，寺町康昌

Seagaia Meeting, 2023/5/19

法令工学の可能性と課題

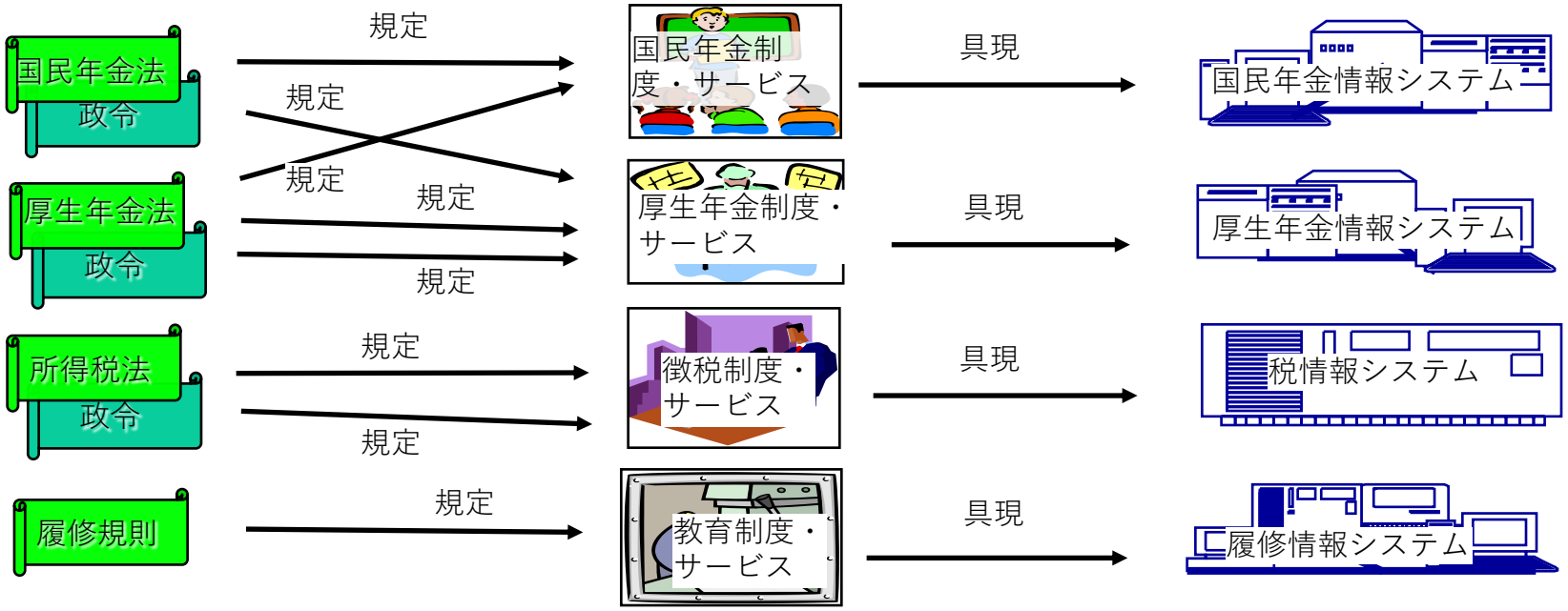
- 法令工学による千年行政ITシステム-

片山卓也，寺町康昌

Seagaia Meeting, 2023/5/19

法令は社会を動かすソフトウェア

社会を支える制度やサービスは法令により決められている.



- 法令を正しく作り，進化させることが，社会の安心の源
- 法律の数２０００本，複雑に絡み合ったスパゲティ状態
- 法令の作り方は変わっていない：基本的には職人的手作業，莫大な作成・保守コスト

労働者派遣法改正法案の条文の誤り

- 職業安定局（平成 26 年 5 月 26 日）

http://www.mhlw.go.jp/file/05-Shingikai-10201000-Daijinkanbousoumuka-Soumuka/0000046988.pdf

- 国会に提出した労働者派遣事業の適正な運営の確保及び派遣労働者の保護等に関する法律等の一部を改正する法律案（閣法第 56号）において、「一年以下の懲役」とすべきところ「一年以上の懲役」としてしまったもの。

「附則第 6 条

6 前二項の規定による処分に違反した者は、一年以上の懲役又は百万円以下の罰金に処する。」

- 6月20日廃案、厚労省事務次官以下6名訓告
- 最近では，コロナに関連した多くの法案で誤り

立法爆発

- 榎並博利＠富士通総研経済研究所
  - “近年の「立法爆発」で法律は「スパゲティ状態」の限界に”
    - 法と経済のジャーナル　２０１５/９/１８
  - 法律の数１９５０本、８０００本（省政令を含める）
  - 内容的にも、複雑に絡み合ったスパゲティ状態
    - 新聞に報道された幾つかの誤り
  - 立法制度の大きな見直しが必要
    - IT利用、ソフトウェア開発技術の利用
    - 人間によるチェックの限界

社会保険オンラインシステムの巨額保守コスト

- 初期構築後４０～５０年以上を経たシステム
  - メインフレーム上のCOBOLプログラムの塊
- 度重なる法令改正による改修の結果，内部がスパゲティ化
- 年間運用費用
  - 785億円/年「厚生労働省のIT投資状況について」，厚生労働省　H25/11/21
  - 変更・機能追加等による費用　ソフトウェア　337億円/年
    「衆議院議員内山晃君質問に対する答弁書」，衆議院答弁本文情報　H16/4/27
  - 小黒 一正：IT予算の8割が維持管理費…「デジタル政府」が延々と進まない … Business Journal（2020/10/21）
- フェーズ1､フェーズ2に分けたシステム改革実施中　R1～R8
  - データベースの統合とRDB化，計算機システム=>オープン系
  - COBOLによる計算ロジックは不変

行政ITシステムの巨額運用経費

- 行政ITシステム（年金，税，戸籍，．．．）
  - 根拠法令に従って構築され，社会基盤として長期間に亘る安定動作が要求される．
- 制度改正 ⇒ 法令改正 ⇒ システムの改修 ⇒ 内部構造の劣化
- 過大な保守コスト
- 保守開発業者がいなくなる危惧
  - 保守作業が困難，保守作業の失敗による訴訟
  - 社会の停滞

2 種類の法令

- 法令 I ：六法全書に載っている法令など
  - 憲法、民法、商法、刑法など
  - 条文の解釈の幅が非常に大きい.
  - 判例や社会的コンセンサスにより条文を解釈，裁判で決着

  ⇨ 法推論研究(AI)

- 法令 II ：年金法などの行政サービスに関する法令
  - 法令という形をしたソフトウェア
  - 基本的には解釈の幅はない．必要に応じて，補完法令で詳細を規定
  - 最終的には，行政組織や法令施行ITシステムにより法令を解釈実行

  ⇨ 法令工学研究 (SE)

法令工学研究の経緯

- 文科省21世紀COEプログラム「検証進化可能電子社会」(2004-2008、北陸先端科学技術大学院大学(JAIST))で提案
- 名大法情報研究センターで法制執務システムの研究
  - 例規データベース(自治体条例や規規のテキストレベルの再利用)
  - 多くの自治体で利用
- 中央大学研究開発機構法令工学研究ユニット(2016ー2020)
  - セコム研究助成，文科省CREST
  - 例規データベースの高度化
  - 法令工学セミナー（阪大，中央大，東工大，ＪＡＩＳＴ，筑波大）
- 興味を持ついくつかの企業（特に，Atrris社）

法令工学

（１）正しさの保障された法令作成の方法論

- 現在の法令は，目と手だけで作られ，テストをせずに運用されている.
  - 正しさの確保に多くの汗と時間
  - 新しい社会制度の実現に必要な法令作成に迅速に対応出来ない.
- 手法：法令の論理式記述＋定理証明技術によるテスト・検証

（２）正しい法令から法令施行ITシステムを作成する方法論

- 現在の開発方法論は，法令の構造を無視している.
  - 法令の改正により内部が劣化する可能性
- 手法：法令指向開発方法論

法令工学（１）

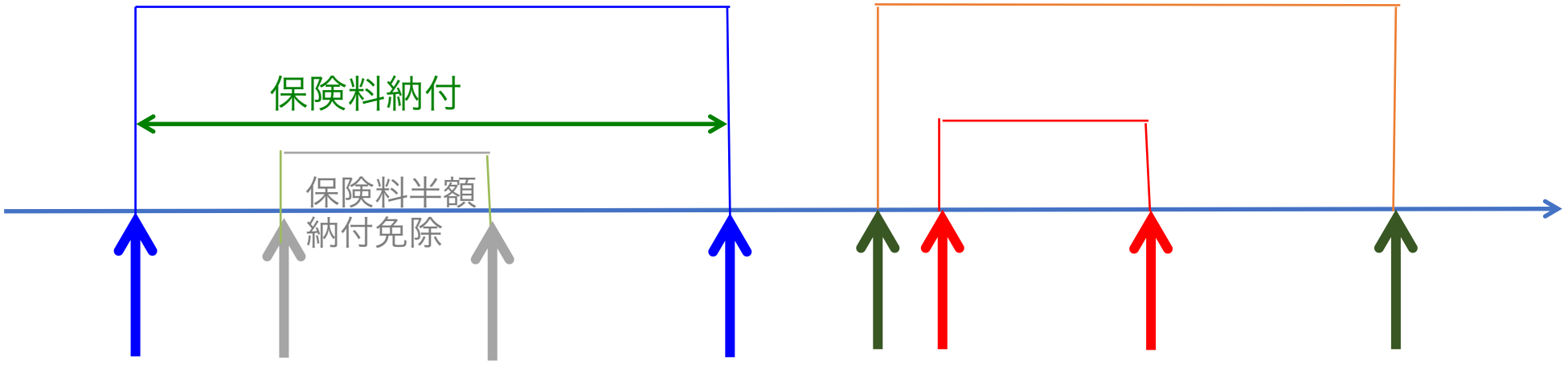
条文の論理式化による正しい法令の作成

Z3Pyによる国民年金法の論理式化

- 国民年金法の主要部分約40条の論理式記述実験
  - 被保険者
  - 給付：老齢基礎年金，障害基礎年金，遺族基礎年金
- 述語論理体系Z3Pyにより論理式化・検証
  - Z3Py：マイクロソフト社製，高速定理自動証明システムZ3のPython版

  （Coq ：遺族基礎年金のみ）
- 結論
  - 条文の論理式による記述は可能である．
  - 定理証明システムによる解釈実行により，条文の正しさを確認
  - 条文は，文章上は複雑であるが、論理的深度は浅く定理証明システムは高速で動く．

老齢基礎年金の概略



Or
・日本国内に住む
　20才以上60才未満
・厚生年金の被保険者
・第2号非保険者の配偶者

And
・本人配偶者世帯主が困窮
・残りの半額の納付
・他の納付免除なし

Or
・被保険者資格の不成立
・死亡

And
・保険料納付期間＋保険料免除期間≧10年
・65才に達する
・被保険者による申請

Or
・併給の調整
・受給権者の申し出

・受給権者の死亡

論理式化の方法と記述体系

- 条文単位で論理式化
  - 各条文で定義される概念を，Z3Py論理式として記述
  - Python lambda式による述語定義
  - 殆どの述語は，日付(日または月) ⇒ {True,False}

- Z3Py論理式
  - 一階論理
    - And, Or, Not, Implies, ForAll, Exists, ==
  - Int, Real, Array, bit列などのドメイン＋それ上の解釈された演算子
  - 抽象データ型

法令文の論理式化の例

（被保険者の資格）

第七条　次の各号のいずれかに該当する者は、国民年金の被保険者とする。

一　日本国内に住所を有する二十歳以上六十歳未満の者であつて次号及び第三号のいずれにも該当しないもの（厚生年金保険法（昭和二十九年法律第百十五号）に基づく老齢を支給事由とする年金たる保険給付その他の老齢又は退職を支給事由とする給付であつて政令で定めるもの（以下「厚生年金保険法 に基づく老齢給付等」という。）を受けることができる者を除く。以下「第一号被保険者」という。）

二　厚生年金保険の被保険者（以下「第二号被保険者」という。）

三　第二号被保険者の配偶者であつて主として第二号被保険者の収入により生計を維持するもの（第二号被保険者である者を除く。以下「被扶養配偶者」という。）のうち二十歳以上六十歳未満のもの（以下「第三号被保険者」という。）

2　前項第三号の規定の適用上、主として第二号被保険者の収入により生計を維持することの認定に関し必要な事項は、政令で定める。

3　前項の認定については、行政手続法（平成五年法律第八十八号）第三章（第十二条及び第十四条を除く。）の規定は、適用しない。

```
# f7.py  第７条被保険者の資格, d：日を表す変数

被保険者=(lambda d :
                    Or(第一号被保険者(d), 第二号被保険者(d), 第三号被保険者(d)))
第一号被保険者=(lambda d :
                      And(日本国内に住所を有する(d),
                          二十歳以上六十歳未満(d),
                           Not(第二号被保険者(d)),
                           Not(第三号被保険者(d)),
                           Not(厚生年金保険法老齢等受給可能(d))))
第二号被保険者=(lambda d :
                    厚生年金保険の被保険者(d))
第三号被保険者=(lambda d :
                     And(被扶養配偶者(d),
                          二十歳以上六十歳未満(d)))
被扶養配偶者=(lambda d :
                     And(第二号被保険者の配偶者(d),
                         主に第二号被保険者の収入により生計維持(d),
                          Not(第二号被保険者(d))))
```

論理式の検証：充足性判定

- Z3Pyシステムによる推論：充足性判定
  論理式P(x,y,z,…)に含まれる変数のx,y,z,…の値の自動決定

  - P(x,y,z,…)＝True になるようなx,y,z,…の値を求める.
    存在する　⇒ sat (satisfiable)　解x,y,..,zを出力
    存在しない ⇒ unsat
    分からない ⇒ unknown　または　停止せず

  - 命題論理式の充足性判定(DPLL)＋解法理論(線形方程式・不等式など)
    SMT(Satisfiability Modulo Theory)
  - ネットワーク，OS 等の検証で実用化

検証方法

```
# v7_ladyA_date.py
# 同じ日には異なる種別の被保険者にはなれないことの確認

import dummy
dummy.honnin='honnin_ladyA'  # 年金原簿の設定
dummy.definitions='definitions'  # 用語定義の設定
from essentials import *
from f7 import * # 第7条論理式
d=Int('d')

s=Solver()
p=Or(And(第一号被保険者(d),第二号被保険者(d)),
        And(第一号被保険者(d),第三号被保険者(d)),
        And(第二号被保険者(d),第三号被保険者(d)))
s.add(p)
print(s.check())  # unsat
```

年金原簿

被保険者・受給権者の年金関連個人記録を論理式で記述したもの

honnin_ladyA_date.py　A女史の年金原簿

誕生日=通日(1955,5,10)
二十歳以上六十歳未満=区間(年齢到達日(20,誕生日),年齢到達日(60,誕生日)-1)
日本国内に住所を有する=DD((1955,5,10),(2020,12,31))
厚生年金保険法老齢等受給可能=FALSE
厚生年金保険の被保険者=OR(DD((1977,4,1),(1998,9,30)),DD((2008,2,1),(2013,1,31)))
第二号被保険者の配偶者=OR(DD((1985,6,1),(1998,11,20)),DD((2013,2,1),(2020,12,31)))
主に第二号被保険者の収入により生計維持
　　=AND(第二号被保険者の配偶者,NOT(厚生年金保険の被保険者))
…

dfinitions.py　基本用語定義

DD=(lambda x1,x2:lambda d: And(通日(x1[0],x1[1],x1[2])<=d,d<=通日(x2[0],x2[1],x2[2])))

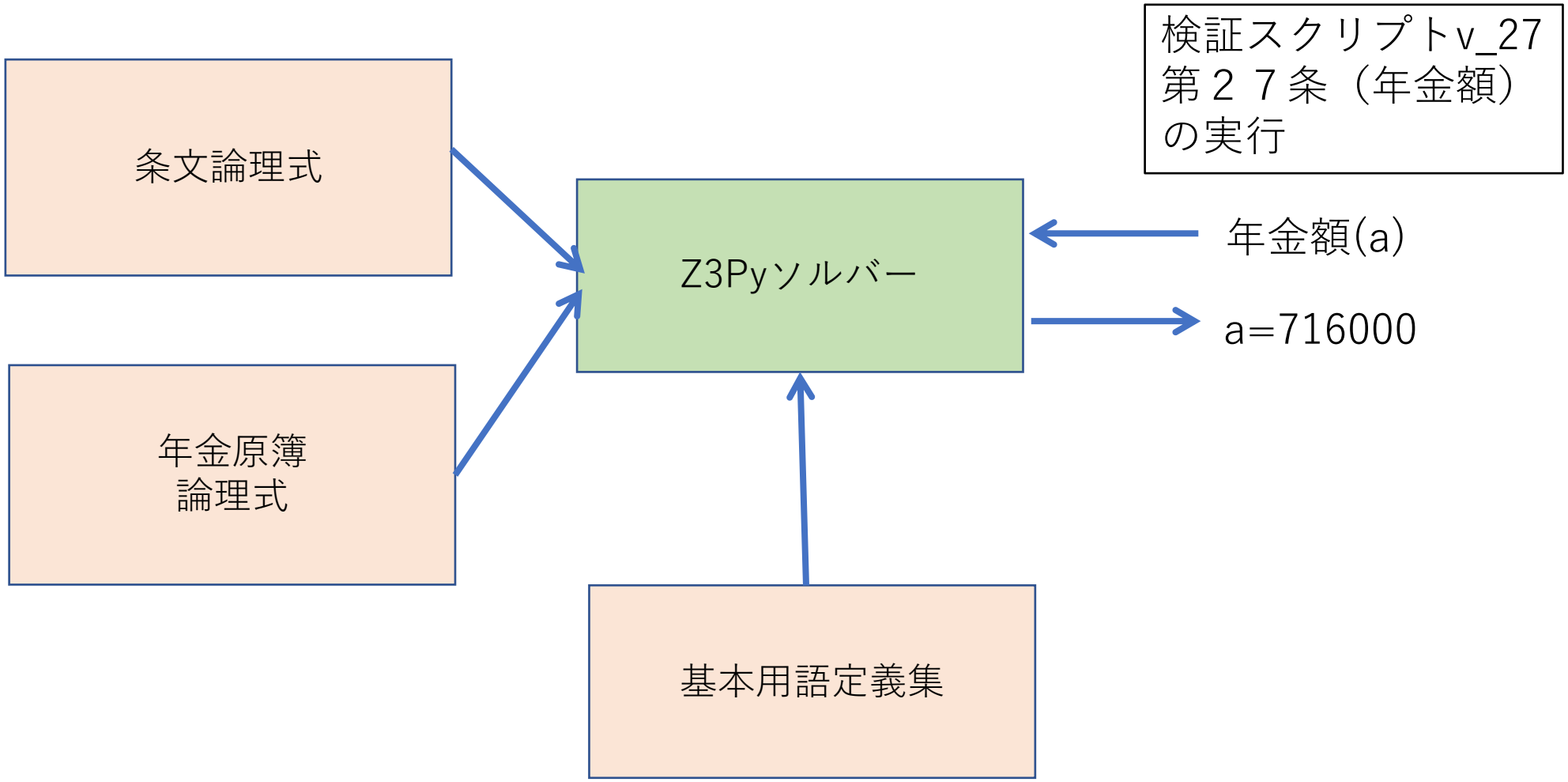
基本用語定義式

- 条文や年金原簿論理式で使われる用語の定義
- 日付関係
  - 日付は1858年11月17日を原点とした日数で計る：修正ユリウス通日
  - 修正ユリウス通日と西暦年月日との相互変換，属月：通日⇒通月，
  - 日区間 DD((2020,5,2),(2030,10,1))，月区間 MM
- 月の集合
  - 期間(f,g)：fが成立する日の属する月の次の月からgが成立する月までの月の集合
  - 月数(f)：fが成立する月数
- 年齢の計算　年齢は誕生日の前日に加算される．
- イベントの成立の時間的前後関係
  - 成立に至る(f)(d)，前に成立(d)(g)，間で成立(d1,d2)(h)，月内で最後に成立(f,g,h),…
- 日付軸上に拡張された論理演算子
  - OR, AND, NOT, …

検証例　A女史の年金額の計算

- 関連条文
  27条(年金額)
    5条(用語の定義)
      11条(被保険験者期間の計算)
        7条(被保険験者の資格))
      89条(法定免除), 90条(全額免除), 90条の2(一部免除),
      90条の3(学生特例免除)
- 年金原簿　honnin_ladyA, haiguusaha_ladyA, setainushi_ladyA
- 基本用語定義集　definitions
- 検証スクリプト　v27（第27条論理式f27を起動）

条文の実行


検証スクリプトv_27
第２７条（年金額）
の実行
条文論理式
年金原簿
論理式
Z3Pyソルバー
年金額(a)
a=716000
基本用語定義集

論理式化＋検証により発見された条文の誤りや不備

- 基本的に条文は正しく作られているが，
  - アルゴリズム的側面は，目視だけによる正しい条文の作成には限界がある．
  - 第20条の誤りは，事務手続きにより運用上回避されている．

第２０条(併給の調整)：誤りの発見とその検証

第８条(資格取得の時期)：不備の発見とその検証

第９条(資格喪失の時期)：不備の発見とその検証

第２８条(支給の繰下)：不備の発見

条文の論理式化

- 行政サービスのための法令は，原理的には論理式化可能
  - 年金法論理式化のこれまでの経験
  - サービスが実際に実施され，計算機化されている．
    （政策，財政などは除く）
- 論理式化の方法
  - しばらくは人間の手作業
    - 法務実務家，ITエンジニア
  - 自動論理式化
    - 条文は用語や表現のぶれが少く，内容的にも幅の狭い文章なので，ツールの整備を行えば可能か？
    - とりあえずは，文章構造解析システムがほしい．

施行令，施行規則，附則

- 実用システムの構築には，「法律」部分のみでなく，施行令（政令），施行規則（省令）の論理式化が必要
  - 法律：原理的側面の記述
  - 施行令：法律の条文の詳細化･具体化
  - 施行規則：法律，施行令の条文の詳細化･具体化
  - これら論理式化が法律部分に比べて困難ということはない.
- 附則中での条文のバージョン管理記述
  - 施行日の指定
  - 経過措置（「なお従前の例による」，「なおその効力を有する」）
  - 条文中の語句の読み替え

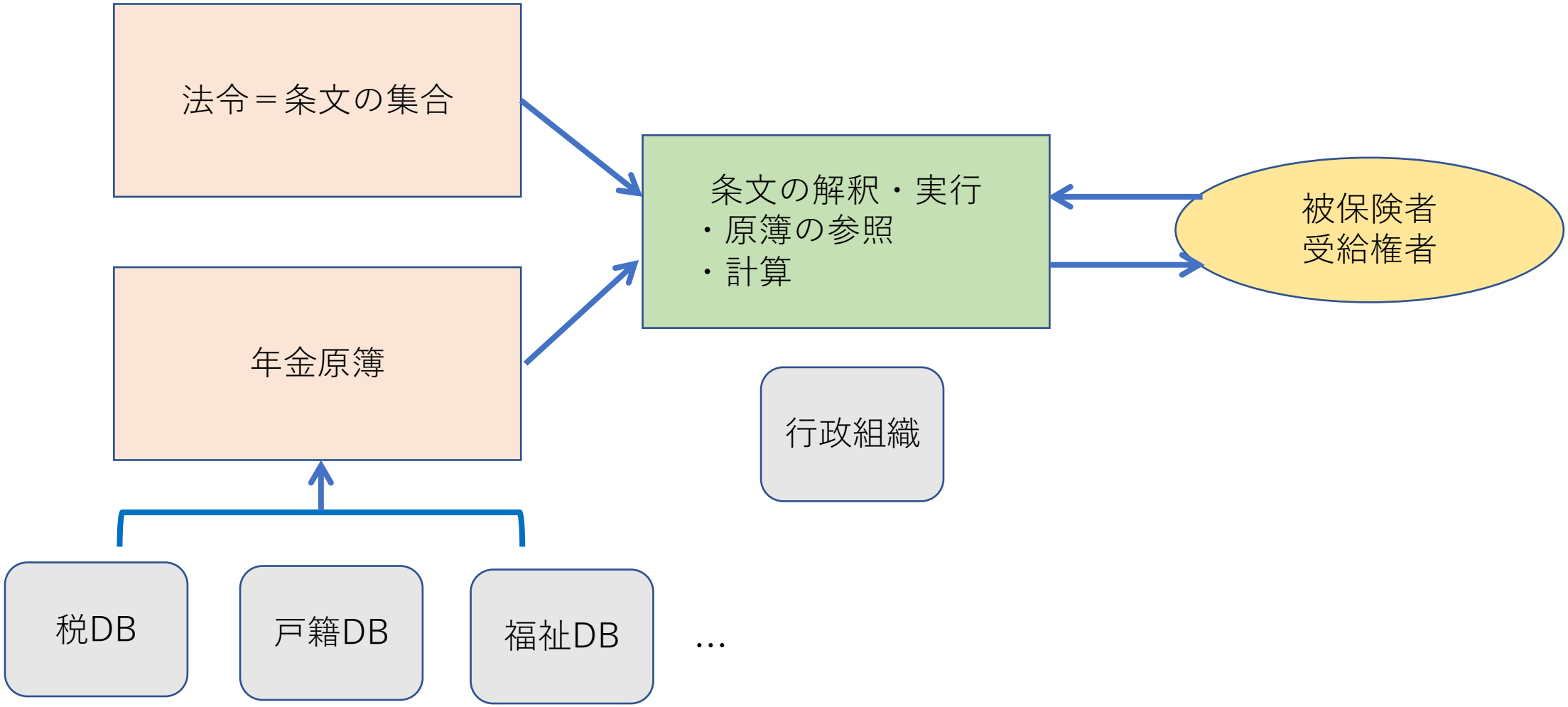
行政全体の形式記述と検証

- 今回の論理式化は，年金額の計算に直結する部分のみ
  - 基本的には述語論理で十分
- 行政組織によるサービスや業務の分担や委譲，被保険者による届出などの側面の形式的記述するには，別のメカニズムが必要
  - 多くの論理体系が既に型式手法の分野で提案，研究され，仕様記述などの分野で使われている．

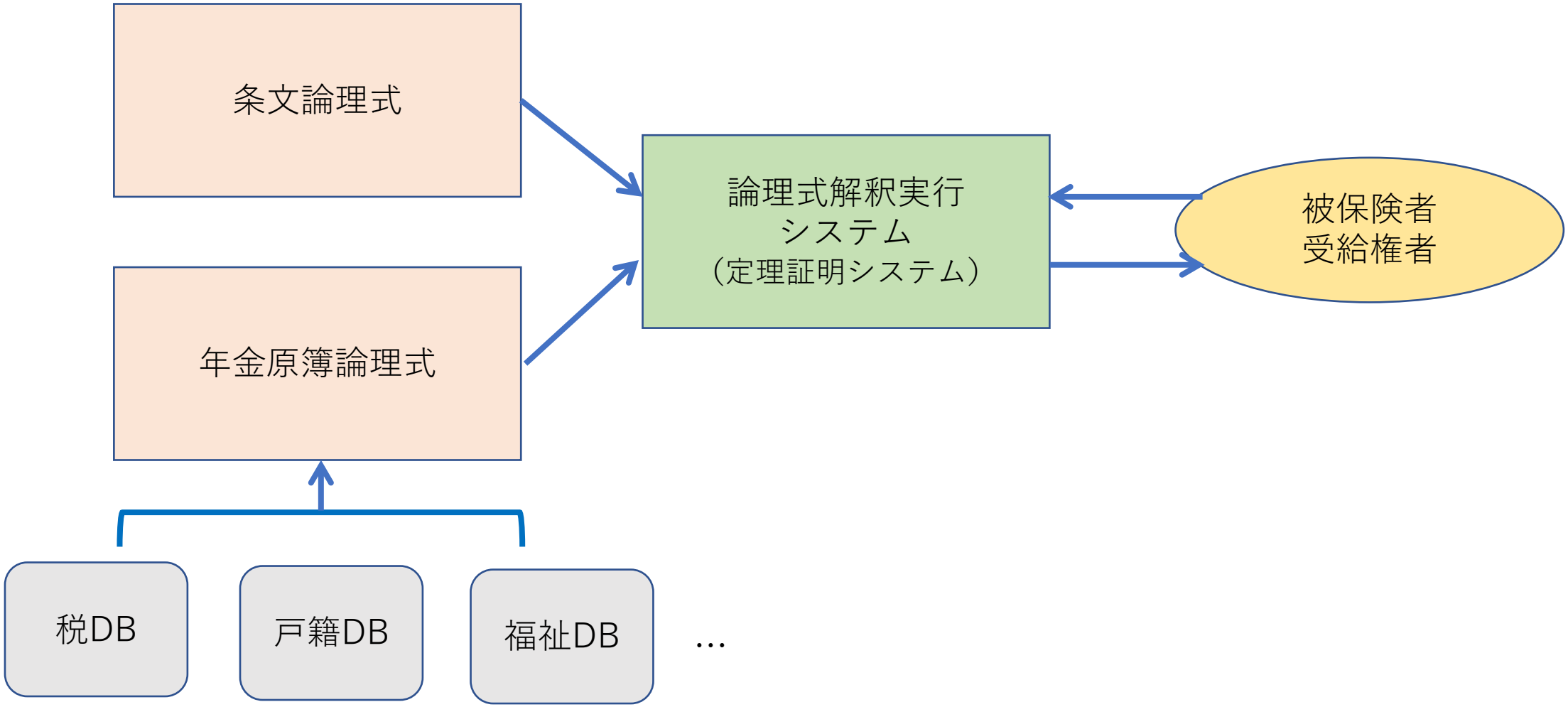
法令工学（２）

法令施行ITシステムの法令指向開発

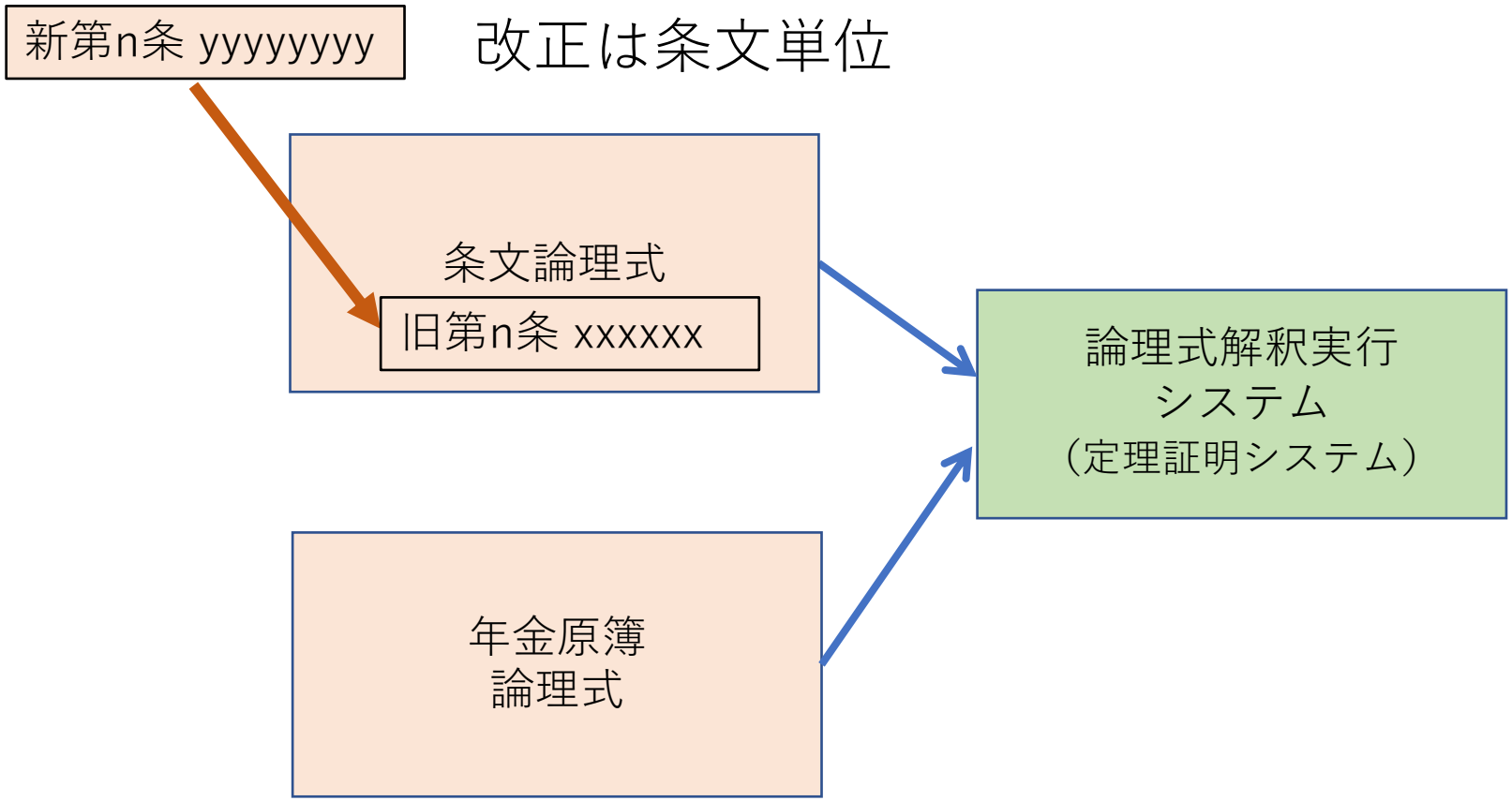
法令と行政サービス



法令施行ITシステムの法令指向開発

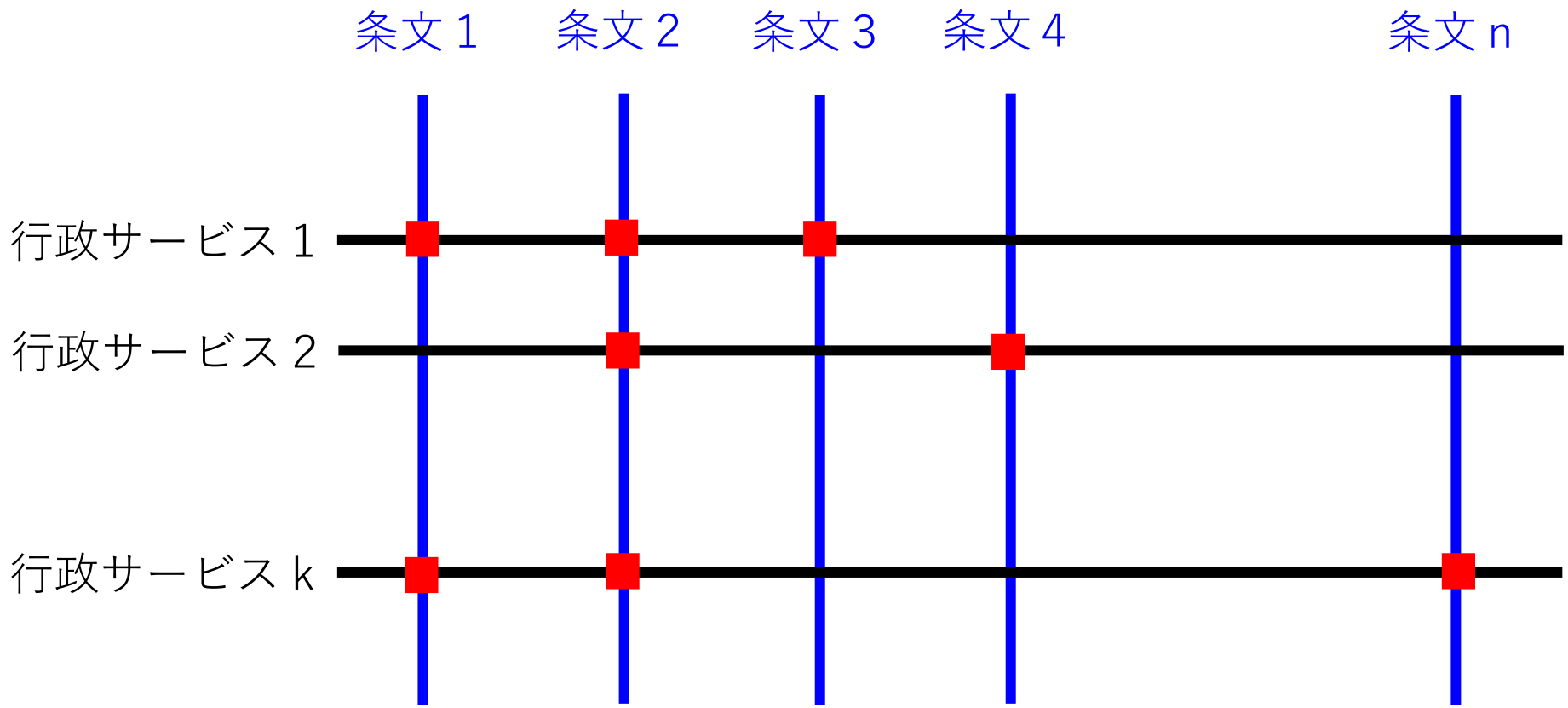


法令改正に対応するITシステムの変更
➡ 改正条文論理式のみを変更

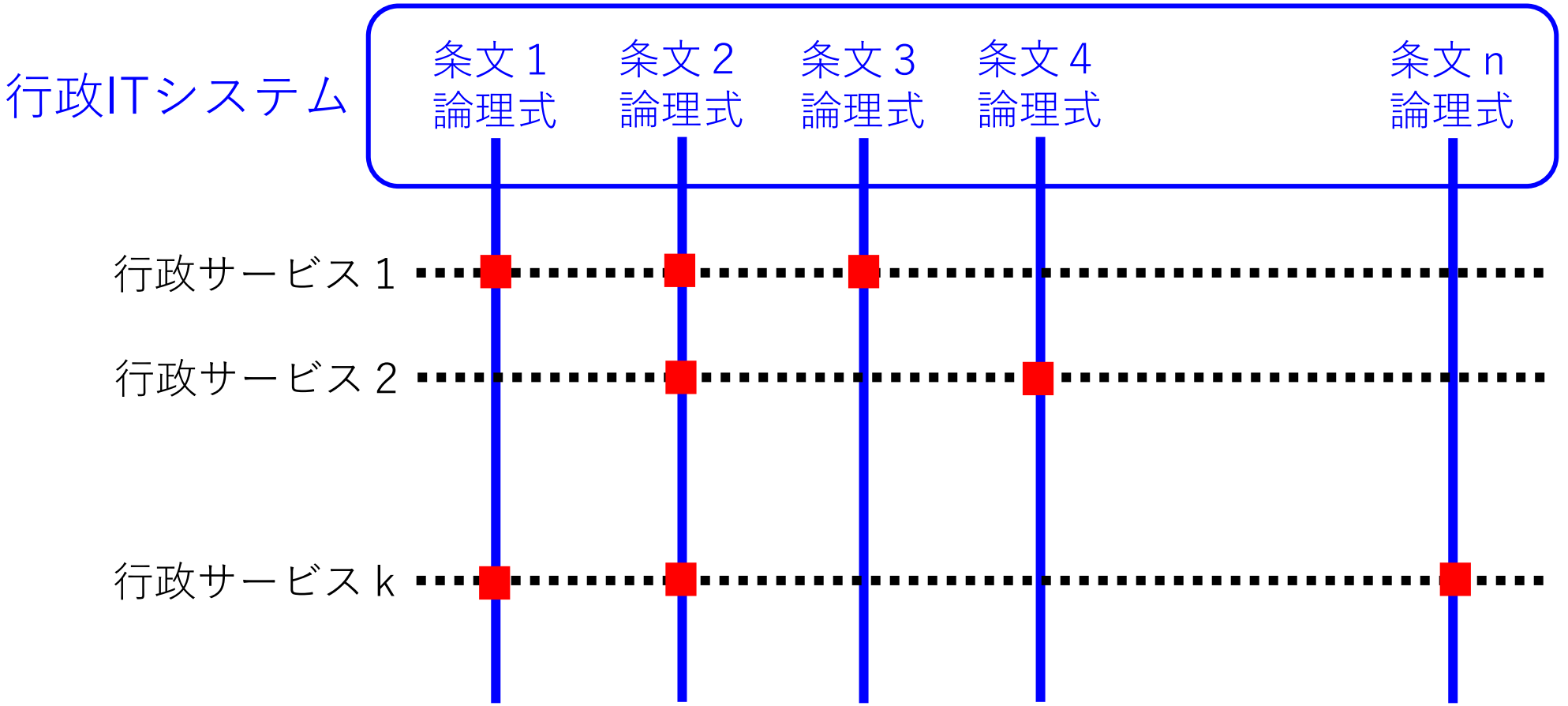


千年システムとしての法令

- 法令＝条文の集合として行政サービスを表現
- 法令作成の長い伝統から適切な条文の設定
- 行政サービスの変更 ⇒ 少数の条文の改正

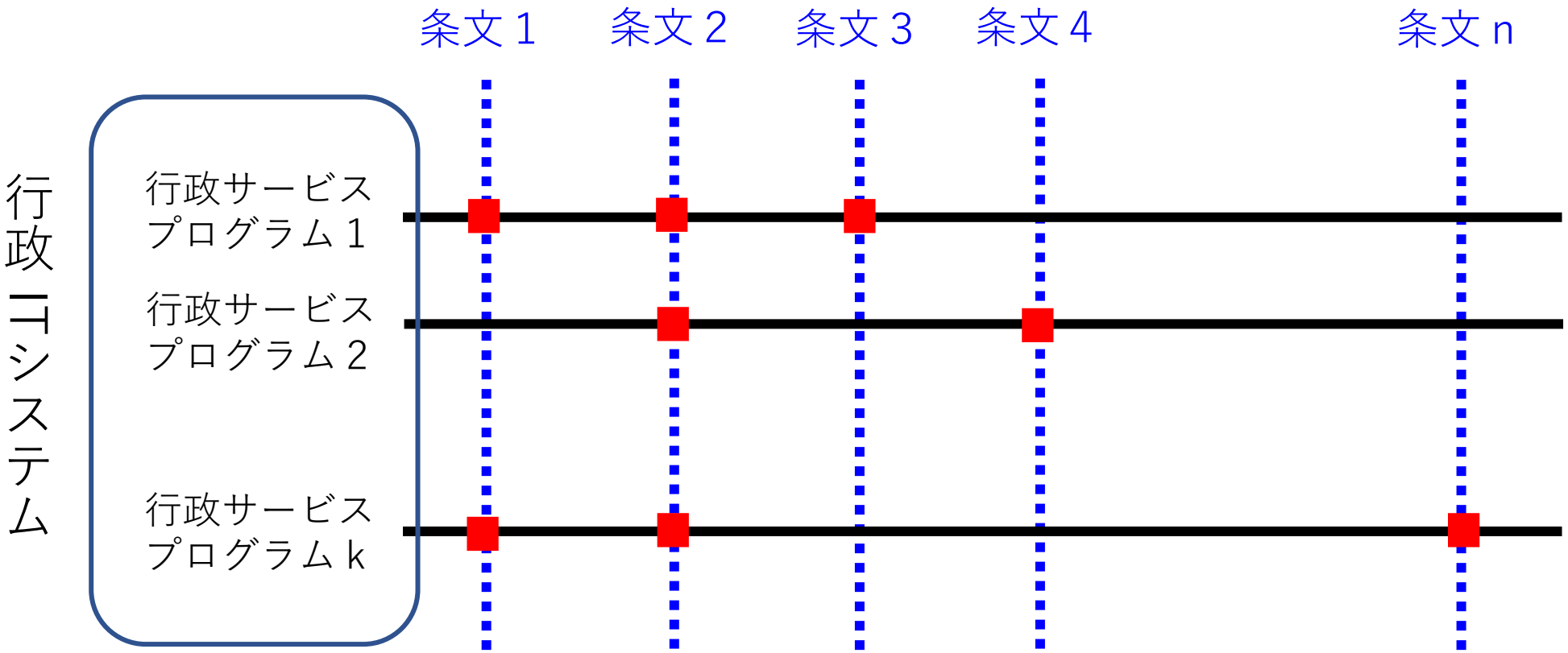


千年システムとして行政ITシステム



・条文を論理式としてコーディング
・行政サービス：定理証明器内の条文論理式の相互関連により動的に実現

現行の行政ITシステムの構成法＝ユースケース指向



・行政サービスを直接コーディング，条文との関連は明確に意識されない．
・１つの条文の変更により複数のプログラムの変更が必要

法令施行ITシステムの「法令指向」開発

- 法令指向開発

  法令施行ITシステム：法令自身を動かす.

  法令施行ITシステム

  ＝法令論理式＋定理自動証明エンジン＋UI＋DB
- 技術的には可能
  - 論理式記述体系，証明エンジン，ハードウェア
- 法令改正への対応
  - 基本的には，改正条文の論理式の変更のみ
  - 法令の改正を重ねても，システムの構造が劣化することはない.

法令工学の社会実装の可能性

- x 法令工学１：条文の論理式化による正しい法令の作成
  - 法曹界の現状を考えると困難と思われる.
    - 法学部では法律の作り方を教えていない.
    - 伝統や安定性を重要視する業界
  - 暫くは新しい技術の導入をしなくても，法令の質は守られる.

- ○ 法令工学2：法令指向方法論
  - 政府ITシステムの保守困難性の解決は緊急課題
  - 正しい法令の存在を仮定すれば，論理式化は技術的には可能
    - 基盤技術は既に存在
  - 技術以外の問題

出版物

1. 国民年金法の述語論理による記述と検証：SMTソルバーZ3Pyを用いたケーススタディ
   - 日本ソフトウェア科学会誌Vol.36, No.3 (2019)
2. 法令工学の実践：国民年金法の述語論理による記述と検証
   - JAIST学術研究成果リポジトリ(JAIST図書館)
   - 第 1 版 2019/9，第 2 版 2019/11，第 3 版 2020/11，第4版 2021/12
   - ダウンロード数 約40/月

- COE Research Monograph Series, Vol. 2 : 法令工学の提案
  - 2007/9 出版
  - 2019/9～2020/11 ダウンロード数 1852(月平均 154)

法令工学（３）

Atrris社の取り組み