

公共業務システム開発における法令からの 要件の抽出／検証手法の提案

伊藤信治[†] 秦野康生[†] 來間啓伸[†] 宮崎邦彦[†]

官公庁や自治体などの公共業務の多くは、法令に基づいて行われているため、公共業務を支援するシステムは法令と整合していることが必須である。法令は、「法律」「政令」「省令」など複数の資料があり、要件はこれら複数の資料に分散して記載され、複雑に絡み合っている。そのため、法令から漏れなく正確に要件を抽出することは非常に手間のかかる作業となっている。本稿では、法令とシステムの整合性確保を目的とした、法令からの要件抽出手法、および、抽出した要件の無矛盾性を形式検証技術により検証する手法について提案する。

Proposal of Methodology for Extraction and Verification of Legal Requirements on Development of Public Sector System

SHINJI ITOH[†] YASUO HATANO[†]
HIRONOBU KURUMA[†] KUNIHICO MIYAZAKI[†]

Most of operations of public offices are performed based on laws. Therefore, it is essential that the system supporting the operations is consistent with the laws. The laws have plural documents including “act”, “cabinet order” and “ordinance of the ministry”. Requirements disperse and are listed in these plural documents and are connected with each other complicatedly. Therefore, it becomes the very time-consuming work to extract requirements from laws exactly without omission. In this paper, we propose methodology for extracting requirements from laws and verifying consistency of the requirements using formal verification technology in order to consistency between laws and system.

1. はじめに

年金、税金、住民情報管理など官公庁や自治体などの公共業務の多くは、法令に基づいて行われている。そのため、公共業務は、法令に記載されている要件（以下、法令要件）に基づいて設計することが必須であり、必然的に、その業務を支援するシステムも法令要件に基づいて設計することが必須である。

法令に基づいたシステム開発は、関連する法令の分析（以下、法令分析）に始まり、要件定義、設計、製造、テストと続く。起点となる法令分析は、当然のことながら非常に重要である。法令分析の結果は、その後の要件定義、設計、製造、テストのすべてに影響を与える。テストは、法令分析の結果を前提として行われるため、法令分析に不備があった場合には、顧客先での運用が始まるまで、システムの不備が表面化しない恐れがある。特に、公共業務システムでは、顧客先での運用開始後に、その不備が表面化した場合、新聞や雑誌などに掲載され社会的問題となる場合がある。

このように、法令分析は非常に重要であるが、現状は、法令の解釈誤りや要件抽出漏れなどの法令分析の不備による事故が発生している。

事故の要因の1つとして、法令の複雑さの問題がある。法令は、「法律」「政令」「省令」「告示」「通知」「事務連

絡」など（以下、法令資料）、多数の法令資料に法令要件が分散して記載されており、それらが複雑に絡み合っている。例えば、介護保険の例では、法律、政令、省令の3つの法令資料のみで445条の条文がある。ネット上に公開されている条文をそのまま印刷すれば500頁相当のボリュームがある。他にも、「告示」や「通知」など介護保険に関連する法令資料は200以上存在する。

また、法令は、他の条項へのリファレンスや、括弧書きが入れ子構造になっているなど、非常に読みづらい構造となっている。

このような法令の複雑さの問題から、法令要件を抽出する作業だけでなく、レビュアーが、抽出された法令要件の正確さや網羅性を確認することも大きな手間となっている。

このような背景の中、著者らは、法令とシステムの整合性確保を実現する開発方法論「法令ベースシステム開発技術」の研究開発を進めている。法令ベースシステム開発技術の確立により、属人性の排除と、複雑な法令の分析の容易化を実現し、法令とシステムの整合性確保の実現を目指している。

本稿では、これまで検討してきた、法令資料から要件を抽出する手順、および、抽出した要件の無矛盾性について形式検証技術[1][2]を活用して検証する手法について提案する。

[†](株)日立製作所
Hitachi, Ltd.

2. 法令とシステム

2.1 法令とシステムの関係

図 1 は、法令とシステムの関係を示している。法令は、「社会の仕様」としての役割があり[3]、社会の組織や構造、目標や目的、組織の活動や手続きなどを規定している。組織から見れば、法令は、「業務の仕様」とみなせる。顧客要望に基づく仕様は、優先度やコストの点などから変更可能であるが、法令で規定している仕様は変更できない。そのため、組織の業務は、必ず法令と整合性を確保しなければならない。当然のことながら、業務の一部をシステム化した場合には、その部分については、法令との整合性確保は必須となる。そのため、法令に基づいたシステムを構築する上では、法令から正確かつ網羅的に、対象システムに関連する要件を抽出し、その要件を正確にシステムに実装する必要がある。

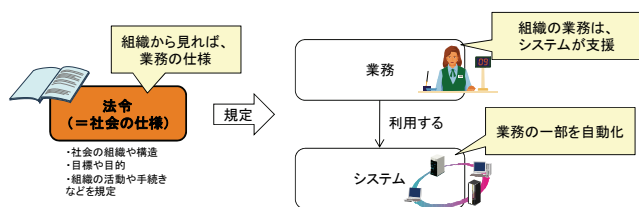


図 1 法令とシステムの関係

2.2 法令の条文構成

膨大な量の法令資料を分析する上では、詳細に分析すべき場所がどこかを絞り込むことが効率化の観点で重要であり、法令の条文構成を知っておくことは有用である。

表 1 は、法令の条文構成を示している。法令条文は、「本則」と「附則」に分かれ、本則はさらに「総則的規定」「実体的規定」「雑則的規定」「罰則規定」から構成されている[4]。

表 1 法令の条文構成

大分類	小分類	説明
本則	総則的規定	法令全体に通ずる基本的な事項
	実体的規定	法令の主体をなす事項
	雑則的規定	法令全般に通ずる手続的、技術的な事項
	罰則規定	違反行為に対する刑罰又は過料に関する事項
附則		施行期日、他法令の廃止、経過措置などに関する事項

このうち、総則的規定と罰則規定は、システム化の観点では重要性は低い。総則的規定には、法律の目的や市町村長等の責務などが規定されており、市町村が行わなければならない事務などの規定は記載されていない。ただし、用語の定義情報を含む場合は、その部分は分析する必要がある。用語の定義は、データ項目や業務ルールに関連する場合が多いからである。また、罰則規定には、違反行為に対する刑罰又は過料に関する事項が規定されている。通常、「△△に違反した場合、懲役〇年」などの事項はシステム

化しないため、分析する必要はない。

その他の規定については、法令の目次に記載されている章や節の見出しなどから、詳細に分析すべきかどうか判断できる場合がある。

2.3 法令の構造

図 2 に法令の条文構造を示す。法令は、「第〇条・・・」という条文構造を持ち、条の下に項、号が続く形式をとっている。「条」は1つの意味のかたまりであり、「項」は段落に相当し、法令が規定する内容の最小単位である。「号」は、事柄の列挙に用いられ、「項」の一部とみなすことができる。

なお、法令には、他にも、章や節などがあるが、これらは、法令の読み手が内容の理解と検索を容易にするために用いられる。

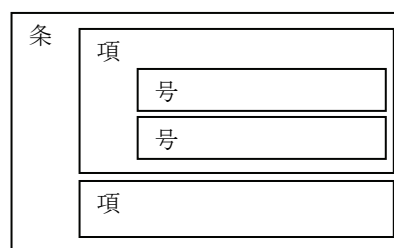


図 2 法令の構造

2.4 法令条文の構造

法令条文は、標準的には、「主部」「条件部」「述部」から構成される[4]。

- 主部：規定する権利・義務の適用を受ける主体
- 条件部：「～の場合」「～のとき」といった規定したい権利・義務が成立する条件
- 述部：「～できる」「～しなければならない」といった規定したい権利・義務

システムの観点では、主部はアクタ、条件部はチェック条件、述部は処理内容とみなすことができる。

法令条文は、自然文で記述されているが、一定の規則に基づいて記述されている。そのため、法令条文の構造と法令要件との関係を整理することで、法令からの要件抽出をパターン化できる。

2.5 法令条文の種類

図 3 は、法令条文の種類とシステム構成要素との関係を示している。図 3 に示す通り、著者らは、法令に記載されている条文を「外部環境（内部への影響なし）」「外部環境（内部への影響あり）」「内部環境」の3つに分類した。例えば、市町村のシステムを考えた場合、「市町村」の義務や権利を規定した条文が「内部環境」に該当し、それ以外が「外部環境」に該当する。また、「外部環境」のうち、市町村のシステムに影響を及ぼすものは、「外部環境（内部への影響あり）」であり、それ以外が「外部環境（内部への影響なし）」となる。例えば、「第一号被保険者は、・・・(略)・・・

必要な事項を市町村に届け出なければならない。」という条文を考えてみる。この条文は、第一号被保険者の義務を規定しており、市町村の義務ではないため、「外部環境」と判断できる。そして、「市町村に届け出」とあるので、市町村に影響があることがわかり「外部環境(内部への影響あり)」となる。

これら3つの分類のうち、対象システムに関連する条文は、「外部環境(内部への影響あり)」と「内部環境」である。「外部環境(内部への影響あり)」の条文は、対象システムに必要なインターフェースや入出力データの情報を主に含んでいる。また、「内部環境」の条文は、対象システムに関するさまざまな情報を含んでいる。

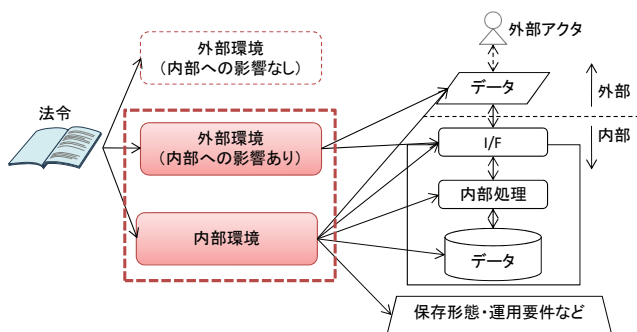


図 3 法令条文の種類とシステム構成要素との関係

法令条文の種類を知ることは、法令の条文構成を知っておくことと同様に、詳細に分析すべき場所がどこかを絞り込む上で有用である。

3. 法令分析のアプローチ

法令に基づいたシステムを構築する上では、法令から抽出した法令要件の品質が重要となる。著者らは、標準規格となっている IEEE 830[5]の要求仕様書の品質特性を、以下のように、法令要件に当てはめ、解決アプローチを検討した。(他にも、「順位付け」があるが、法令分析の段階では順位付けはしない前提で省略した)

- 正確性：解釈に誤りがないこと
- 無曖昧性：一意に意味を特定できること
- 完全性：法令要件に漏れがないこと
- 一貫性：法令要件間に矛盾がないこと
- 検証可能性：法令要件の正しさを評価できること
- 保守性：法改正時に一貫性を保って修正できること
- 追跡性：法令条文と法令要件の対応がわかること

法令要件が上述の性質を満たすための手段として、著者らが提案するアプローチの概要を図 4 に示す。

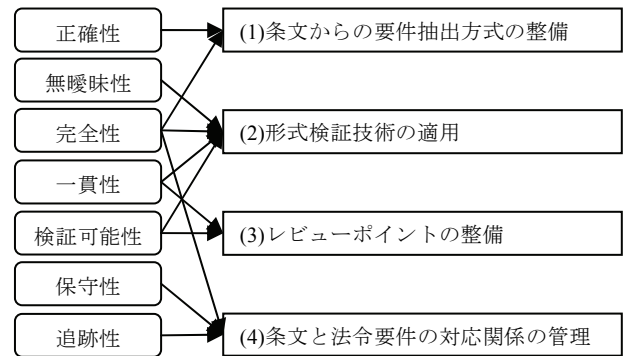


図 4 提案アプローチの概要

以下、各アプローチの概要について説明する。

(1) 条文からの要件抽出方式の整備

「条文からの要件抽出方式の整備」では、法令から要件を抽出する手順を確立し、法令の解釈の誤りと要件の抽出モレを軽減する。法令は、自然文で記述されているものの、通常の文章と異なり、一定の規則に基づき記述されている。そのため、法令の条文構造に基づき、法令から要件を抽出することで、解釈の誤りや要件の抽出モレの軽減が期待できる。

(2) 形式検証技術の適用

「形式検証技術の適用」では、曖昧さの排除と要件間の矛盾の機械的チェックを実現する。また、要件の抽出モレにより要件間の矛盾が発生する場合も考えられるため、形式検証による無矛盾性の検証により、要件の抽出モレの発見も期待できる。

(3) レビューポイントの整備

「レビューポイントの整備」では、形式検証技術では検証できない性質について、人によるレビューを支援するための観点(チェックリスト)を整備する。

(4) 条文と法令要件の対応関係の管理

「条文と法令要件の対応関係の管理」では、条文と、そこから抽出した法令要件を紐づけて管理する。これにより、どの条文から法令要件が抽出されたか、または、抽出されていないかがわかり、法令要件の抽出漏れの確認が容易になる。また、法改正の際の影響範囲の特定や修正の効率化も期待できる。

以下の章では、まず、法令分析の全体像を説明した後、図 4 に示したアプローチのうち、「(1)条文からの要件抽出方式の整備」「(2)形式検証技術の適用」について、詳しく述べる。

4. 法令分析の手法

4.1 法令分析の作業の流れ

図 5 は、著者らが提案する法令分析の作業の流れの概要を示している。また、表 2 は、主要成果物の概要を示して

いる。

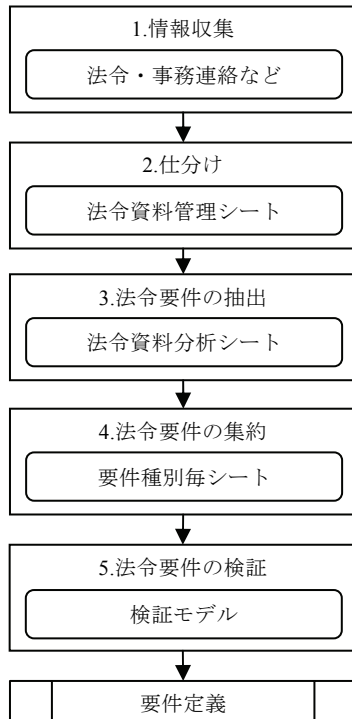


図 5 法令分析の作業の流れ

表 2 法令分析における主要成果物

#	成果物名	説明
1	法令資料管理シート	入手した法令資料に関する情報を記述する。
2	法令資料分析シート	各法令資料から抽出した法令要件を記述する。
3	用語定義一覧	法令で定義されている用語について、その用語名と定義内容を記述する。
4	法令データ要件一覧	データ要件に関する法令要件を記述する。
5	法令ユースケース一覧	ユースケース要件に関する法令要件を記述する。
6	検証モデル	法令要件を形式仕様記述言語で記述する。

以下、法令分析の各作業ステップの作業内容について説明する。

(1) 情報収集

分析対象となる法令資料を Web や顧客の官庁や自治体から入手する。

(2) 仕分け

入手した法令資料を簡易的にチェックし、詳細な分析が必要かどうか（対象システムに影響するかどうか）を判断し、その結果を法令資料管理シートに記載する。

(3) 法令要件の抽出

詳細な分析が必要と判断した法令資料について、法令資料単位に詳細に分析し、抽出した法令要件を法令資料分析シートに記載する。法令資料分析シートには、法令の条文、抽出した法令要件、要件の分類（用語、データ、ユースケ

ース）、他の法令要件に関連する場合はその法令要件の ID などを記載する。法令条文と法令要件の対応関係が明確になるようにする。

(4) 法令要件の集約

法令資料毎に作成した法令資料分析シートに記載されている法令要件をその分類（用語、データ、ユースケース）に応じて、集約する。これにより、複数の資料に分散して記載されていた法令要件が、1ヶ所に集約され、複数の法令資料を行ったり来たりする必要がなくなり、法令要件の把握が容易になる。

(5) 法令要件の検証

抽出した法令要件から検証モデルを作成し、法令要件の整合性を検証する。これにより、開発者による解釈の誤りや要件の抽出漏れなどの不備を機械的に検証できる。

以下、4.2 節で法令要件の抽出方式について述べ、4.3 節で法令要件の検証方式について述べる。

4.2 法令要件の抽出方式

法令要件の抽出では、法令資料分析シートを作成する。図 6 は、法令資料分析シートの作成の流れを示している。また、表 3 に法令資料分析シートの主要な記載項目を示す。法令資料分析シートは、法令条文と法令要件の対応関係が明確となるフォーマットとしている。これにより、開発者とレビュー者が、それぞれ法令資料のどこから法令要件を抽出して、どこから抽出していないのかが視覚的にわかり、要件漏れを低減できる。

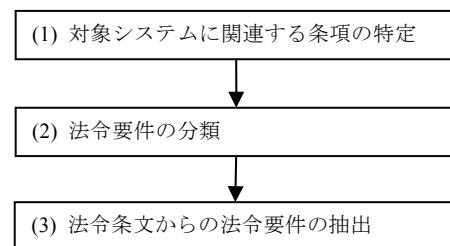


図 6 法令資料分析シートの作成の流れ

表 3 法令資料分析シートの主要記載項目

#	記載項目	説明
1	条文	法令に記載されている条文を「項」または「号」単位に記載する。
2	分析対象	分析対象か否かを記載する。
3	法令要件 ID	法令要件の識別子を記載する。
4	要件種別	法令要件の種別を記載する。
5	関連法令要件 ID	関連する法令要件の ID を記載する。
6	名称	法令要件の名称を記載する。
7	法令要件	条文から抽出した法令要件を記載する。

以下では、各作業について詳細を述べる。

4.2.1 対象システムに関連する条項の特定

効率的に法令を分析する上では、詳細に分析すべき範囲の絞り込みが重要となる。そのために、詳細な分析が必要な条項を、まず、章・節単位で絞り込み、続いて、項単位で絞り込む。

章・節単位での絞り込みは、「2.2 法令の条文構成」で述べた条文構成と章や節タイトルから判定する。

項単位での絞り込みは、「2.5 法令条文の種類」の考えに基づき行う。具体的には、事前に、対象システムを扱うのは「だれか」を特定しておき（特定した者を、以下では「対象システム主体」という）、法令条文中に対象システム主体を含むか否かで分析対象か否かにより、外部環境と内部環境のいずれに関する条文であるかを判断する。また、対象システム主体で判断できない場合、法令条文中に含まれるリファレンス情報から、参照先の法令条文が分析対象であるか否かで判断する。

4.2.2 法令要件の分類

分析対象とした法令条文は、表 4 に示す種別のいずれかに分類する。分類は、「項」単位に行うが、「項」によっては2つ以上に分類される場合もある。

表 4 法令要件の種別

種別	説明
用語	用語の定義に関する情報である。
データ	ユースケースの入出力となるデータに関する要件である。帳票に記載するデータ項目やデータ項目の制約条件などが該当する。
ユースケース	業務手続きに関する要件である。

法令要件の分類にあたっては、法令要件の種別毎に法令条文の記載パターンを整理した。法令条文は、「2.4 法令条文の構造」で述べた通り、自然文で記述されているが、一定の規則に基づいて記述されているため、パターン化しやすいという特徴がある。これにより、開発者による法令要件の分類の判定を支援する。

表 5 に、法令要件の種別が「ユースケース」であるパターンの一例を示している。

表 5 ユースケースの記載パターン例

パターン説明	法令条文例
法令条文の主部が対象システム主体であり、かつ、条件部をもち、かつ、述部が手続き(処理)を表す	市町村長は、新たに市町村の区域内に住所を定めた者その他新たにその市町村の住民基本台帳に記録されるべき者が <u>あるときは</u> 、次項に定める場合を除き、その者の住民票を作成しなければならない。 ※対象システム主体＝市町村長
法令条文の主部が対象システム主体ではない者であり、目的語(誰に)に対象システム主体を含む	転入をした者は、転入をした日から十四日以内に、次に掲げる事項を <u>市町村長</u> に届け出なければならない。 ※対象システム主体＝市町村長

4.2.3 法令条文からの法令要件の抽出

法令要件の分類後は、法令条文に記載されている法令要件を法令資料分析シートの名称欄と法令要件欄に記載する。記載に当たっては、記載ルール（AND 結合の要件は箇条書き、OR 結合の要件はその旨を記載など）と法令要件抽出のパターンを事例集として用意し、開発者による記載のばらつきを軽減する。

4.3 法令要件の検証方式

法令要件の検証では、法令から抽出した自然言語ベースの法令要件を、形式手法の専門知識を有する者が形式仕様記述言語で記述し、ツールによる検証処理を実行する。

以下では、採用した形式仕様記述言語、および、検証モデルの作成方針について述べる。

4.3.1 採用した形式仕様記述言語

形式仕様記述言語には、Event-B、B 言語、Z 記法、VDM、Promela など数多く存在し、言語により記述できる仕様の抽象度や検証可能な性質が異なる。本研究では、抽象的な記述の多い法令からの要件抽出を対象としているため、抽象的な仕様の記述に適した Event-B[6][7]を採用した。

Event-B は、1990 年代に J. -R. Abrial によって考案されたシステム全体の仕様のモデリングと分析を行うための形式仕様記述言語の 1 つである。Event-B では、コンテキスト(CONTEXT)と抽象機械(MACHINE)の 2 つにより仕様を記述する。コンテキストでは、抽象機械が前提とする概念（システムの観点では「型」に相当）や定数などを記述する。一方、抽象機械では、変数（状態）を宣言し、変数が常に満たすべき「不変条件」と変数を変化させる「イベント」を記述することで、システムの振舞いを記述する。Event-B では、「イベント」の発生により、「不変条件」が常に満たされることを検証する。

4.3.2 検証モデルの作成方針

図 7 は、法令要件と Event-B の構成要素の対応関係の概要を示している。法令から抽出した用語とデータについては、用語名とデータ名を Variables 節に記載し、用語定義とデータ要件を Invariants 節に記載した。また、ユースケースについては、ユースケースの入出力を Any 節に記載し、事前条件を Where 節、アクションを Then 節に記載した。

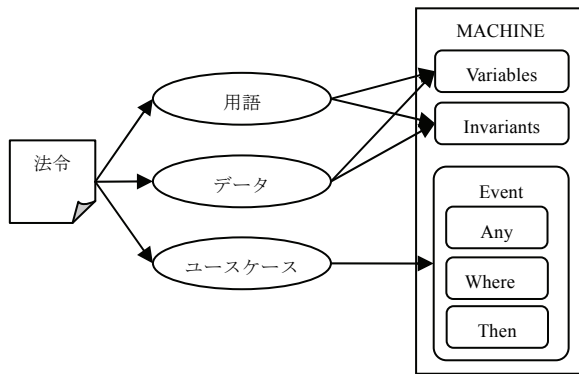


図 7 法令要件と Event-B との関係

用語とデータに関する要件は、個別の事務処理には依存しない常に満たされるべき要件であることから、Variables 節と Invariants 節に記載することとした。また、ユースケースに関する要件は、事務処理を行う事前条件や処理内容（アクション）である。事務処理は、データ（Event-B では、状態に相当）に対する処理であることから、ユースケースに関する要件は、Event-B の「イベント」に記載することとした。

このように検証モデルを作成することで、用語・データに関する要件とユースケースに関する要件との間の矛盾の有無が検証可能となる。

なお、CONTEXT には、用語やデータの型を記述する。用語の型としては、「人」「人の属性」など用語の抽象的な概念を記述する。また、データの型としては、データ名を記述する。

5. 評価

法令要件の抽出・集約、および、法令要件の検証について、評価を実施した。法令要件の抽出・集約については、公共業務システムの開発者に協力して評価頂き、法令要件の検証については、著者らが住民基本台帳法を題材として評価した。

5.1 法令要件の抽出・集約の評価結果

評価は、各評価者が実際に担当している法令を題材として、評価を実施した。評価の結果を整理すると以下の通りである。法令要件の抽出・集約の成果物が、要件定義のインプットとして有効であることを確認できた。

- 網羅的な要件抽出に役立つ
- 要件の抽出漏れの軽減に役立つ
- 法改正時の影響範囲調査に役立つ

一方で、法令要件の抽出・集約の作業については、「法令要件の種別の判定が難しい」「リファレンス情報や括弧書きが多く含まれる法令条文の分析が難しい」などの意見を頂いた。これらの意見に対しては、自然言語処理技術を活用した法令要件の種別の候補表示や、法令条文の可視化方法などを検討中である。

5.2 法令要件の検証の評価結果

評価は、住民基本台帳法（法律、政令、省令）から法令分析の作業の流れに従い法令要件を抽出し、Event-B で検証モデルを作成し、検証を行った。表 6 に評価作業の概要を示す。

表 6 法令要件の検証の評価作業の概要

項目	内容
法令要件の数	563
検証モデルに取り込んだ法令要件の数	23
作業工数（記述・検証・修正）	2.0 日
非明示要件の数	35

検証モデル作成・検証により、法令に明示的に記載されていない要件（以下、非明示要件という）を 35 個抽出することができた。表 7 は非明示要件の一例を示している。

表 7 非明示要件の一例

名称	要件
住民票 (データ)	・住民票を世帯ごとに編成して住民基本台帳を作成しなければならない
住民票の作成 (ユースケース)	[事前条件] ・新たに市町村内に住所を定めた者がある [アクション] ・そのものに対する住民票を作成する ・住民票を所属世帯に登録する

※下線の要件が非明示要件

6. まとめ

本稿では、法令資料から要件を抽出する手順、および、抽出した要件の無矛盾性について形式検証技術を活用して検証する手法について述べた。提案方式を活用することで、法令に基づいた公共業務システムの品質向上が可能であることを確認した。

提案方式は、一部、ツール化しているが、手作業による部分が大半をしめており、開発者のスキルに左右される部分が多く残っている。また、法令要件の検証については、検証可能な性質が限定的であり、検証可能な性質を拡張することが今後の課題である。

参考文献

- 1) 中島震: ソフトウェア工学の道具としての形式手法 - 彷徨える形式手法 -, NII Technical Report, 2007/7
- 2) 荒木啓二郎: フォーマルメソッドの過去・現在・未来 - 適用の実践に向けて -, 情報処理, Vo.49 No.5, 2008/5
- 3) 片山卓也他: 電子社会と法令工学, 人工知能学会誌, 23 巻 4 号, 2008/7
- 4) 磯崎陽輔: 分かりやすい法律・条例の書き方, ぎょうせい, 2006/5
- 5) IEEE Computer Society: IEEE Std 830-1998 Recommended Practice for Software Requirements Specifications, 1998.
- 6) Event-B.org, <http://www.event-b.org/>
- 7) J.-R. Abrial, "Modeling in Event-B: System and Software Engineering", Cambridge University Press, 2010.