

NMDA 電子申請フレームワーク
電子申請用 XML 様式的设计ガイドライン
ー タグ设计方針と申請書共通ボキャブラリ ー
[第 1 版]

財団法人 ニューメディア開発協会

目 次

0. 電子申請における XML 技術の背景知識の解説.....	1
0.1. 本ガイドラインで使用する XML 関連用語.....	1
0.2. 様式の電子化	2
0.3. 電子申請において行政機関側ソフトウェアと申請者側ソフトウェア間で共有すべき規約	3
1. はじめに	5
1.1. 本ガイドラインの目的.....	5
1.2. ガイドラインが対象とする範囲.....	6
1.3. 様式の電子化と本ガイドラインへの適合度.....	7
2. 申請書のタグ設計の考え方	8
2.1. 申請書のタグ設計への基本要件.....	8
2.2. 申請書の論理構造モデル.....	9
2.3. 申請書のタグ設計の方針.....	10
2.3.1. 「専用タグ付け方式」の採用について.....	10
2.3.2. 申請書の論理構造モデルを反映したタグ設計方針.....	11
2.4. 申請書共通ボキャブラリ作成の方針.....	12
2.4.1. 申請書共通ボキャブラリがカバーする範囲.....	12
2.4.2. 分野別共通ボキャブラリ.....	13
2.4.3. 分野別共通ボキャブラリの作成.....	13
2.4.4. 申請書の記載項目の考え方.....	14
2.5. 申請書のスタイル	16
2.6. 申請書のタグ設計規約および申請書共通ボキャブラリの主な内容	17
3. 申請書のタグ設計規約	18
3.1. 様式個別部分の設計規約.....	18
3.1.1. 要素の構造化	18
3.1.2. 典型的な申請書構造パターンの提示.....	19
3.1.3. 要素内容と属性値の使い分け.....	20
3.1.4. 要素の命名規則.....	20
3.1.5. 申請書共通ボキャブラリの利用方法.....	20
3.1.6. 設計にあたっての注意点.....	21
3.2. 符号化文字集合、文字符号化方式、XML 宣言・文書型宣言等に関する規約.....	21
3.2.1. 符号化文字集合・文字符号化方式等に関する規約.....	21
3.2.2. XML 宣言・文書型宣言等の XML ファイルとしての形式や宣言に関する規約.....	21
4. 申請書共通ボキャブラリ	22
4.1. 申請書共通ボキャブラリとは.....	22
4.2. 申請書共通ボキャブラリ.....	26

4.2.1. 様式識別要素群.....	27
4.2.1.1. 文書名に関する記載項目.....	27
4.2.1.2. 文書番号（様式番号）に関する記載項目.....	28
4.2.1.3. 様式を規定している法令に関する記載項目.....	29
4.2.1.4. あて先に関する記載項目.....	30
4.2.1.5. 申請届出根拠に関する記載項目.....	31
4.2.1.6. 法令名に関する記載項目.....	32
4.2.1.7. 手続名に関する記載項目.....	33
4.2.2. 記載内容要素群.....	34
4.2.2.1. 個人情報または組織情報に関する記載項目.....	34
A 氏名に関する記載項目.....	34
B 法人団体名に関する記載項目.....	36
C 住所に関する記載項目.....	37
D 連絡先に関する記載項目.....	39
E その他の個人情報または組織情報に関する記載項目.....	41
4.2.2.2. 時間に関する記載項目.....	41
A 日付に関する記載項目.....	41
B 時刻に関する記載項目.....	46
C 日付・時刻に関するその他のバリエーション.....	50
4.2.2.3. 数量に関する記載項目.....	51
A 金額に関する記載項目.....	52
B 期間（数量）に関する記載項目.....	55
C その他の数量に関する記載項目.....	56
4.2.2.4. 範囲に関する記載項目.....	57
A 期間（範囲）に関する記載項目.....	57
B 区間に関する記載項目.....	60
4.2.2.5. その他の記載内容要素群.....	62
A 振り仮名に関する記載項目.....	62
B 場所に関する記載項目.....	64
C 理由に関する記載項目.....	65
D 符号に関する記載項目.....	66
4.2.3. 記載支援要素群.....	67
4.2.3.1. 記載説明に関する記載項目.....	67
5. 申請書共通ボキャブラリの XML Schema への適用.....	68
5.1. XML Schema の導入方針.....	68
5.2. XML Schema 版申請書共通ボキャブラリ.....	69
5.2.1. 様式識別要素群.....	69
5.2.1.1. 文書名に関する記載項目.....	69
5.2.1.2. 文書番号（様式番号）に関する記載項目.....	71

5.2.1.3.	様式を規定している法令に関する記載項目	73
5.2.1.4.	あて先に関する記載項目	75
5.2.1.5.	申請届出根拠に関する記載項目	77
5.2.1.6.	法令名に関する記載項目	79
5.2.1.7.	手続名に関する記載項目	81
5.2.2.	記載内容要素群	83
5.2.2.1.	個人情報または組織情報に関する記載項目	83
A	氏名に関する記載項目	83
B	法人団体名に関する記載項目	85
C	住所に関する記載項目	87
D	連絡先に関する記載項目	90
E	その他の個人情報または組織情報に関する記載項目	92
5.2.2.2.	時間に関する記載項目	93
A	日付に関する記載項目	93
B	時刻に関する記載項目	100
C	日付・時刻に関するその他のバリエーション	104
5.2.2.3.	数量に関する記載項目	105
A	金額に関する記載項目	105
B	期間（数量）に関する記載項目	109
C	その他の数量に関する項目	111
5.2.2.4.	範囲に関する記載項目	112
A	期間（範囲）に関する記載項目	112
B	区間に関する記載項目	116
5.2.2.5.	その他の記載内容要素群	118
A	振り仮名に関する記載項目	118
B	場所に関する記載項目	120
C	理由に関する記載項目	122
D	符号に関する記載項目	124
5.2.3.	記載支援要素群	126
5.2.3.1.	記載説明に関する記載項目	126

付録

- A：申請書様式サンプル集
- B：紙様式の電子化にあたっての課題
- C：ロードマップ
- D：申請書共通ボキャブラリの作成手順
- E：様式用語シソーラス
- F：申請書共通ボキャブラリ辞書
- G：申請書共通ボキャブラリ一覧
- H：申請書共通ボキャブラリ・リファレンス
- I：電子申請用 XML 様式の作成手順 ーチュートリアルー
- J：基本物理量ボキャブラリ

0. 電子申請における XML 技術の背景知識の解説

0.1. 本ガイドラインで使用する XML 関連用語

XML の関連用語については、基本的には、本ガイドラインの公開時点での W3C(World Wide Web Consortium: <http://www.w3.org/>)の規格に準ずるものとする。但し、必ずしも W3C の規格が規定していないが、一般的に通用していると思われる XML 関連用語として、以下に示す「タグ」と「ボキャブラリ」を使用する。

タグ

本ガイドラインでは、誤解や混乱が生じないと思われる範囲で、XML の「要素」のことを、「タグ」と呼ぶことがある。例えば、XML の初心者にとってわかりやすいと思われる場合や、「タグ設計」や「タグ付け」等技術者の間で一般的に通用していると思われる表現を利用する場合等である。

ボキャブラリ

ある利用目的で用いられている XML 文書を構成している要素や属性を集めたものをボキャブラリと呼ぶ。例えば、WWW ブラウザにマルチメディア文書を表示するために用いられる XHTML 文書を構成している「table(表組み)」、「p(パラグラフ)」、そして「img(イメージ)」等の要素や属性を集めたものを、XHTML ボキャブラリと呼ぶ。本ガイドラインでは、行政分野の申請書で共通に用いられ、再利用が見込まれる要素やデータ型の定義を集めたものを「申請書共通ボキャブラリ」として規定し、申請書用 XML 様式的设计の際に利用できるライブラリとして提供する。

0.2. 様式の電子化

申請者側から行政機関側に提出される申請書(申請書、届出書、報告書等を総称して「申請書」と呼ぶ)およびその添付文書は、その申請手続を定める法令に規定されている様式に従う必要がある。本ガイドラインでは、現行の法令に規定されている様式を「紙様式」と呼ぶ。紙様式は、様式の電子化という観点からみた場合、「文書のスタイル」、「文書の論理構造」、「文書の意味制約」の3つを規定しているといえる(「図1 紙様式と電子様式」を参照)。ここで「文書のスタイル」は「提出する紙としての申請書の見栄えを規定する部分」を、「文書の論理構造」は「申請書内の記載項目や記載項目間の関連を規定する部分」を、「文書の意味制約」は「項目の内容に関する記載要領を規定する部分」を、それぞれ意味するものとする。

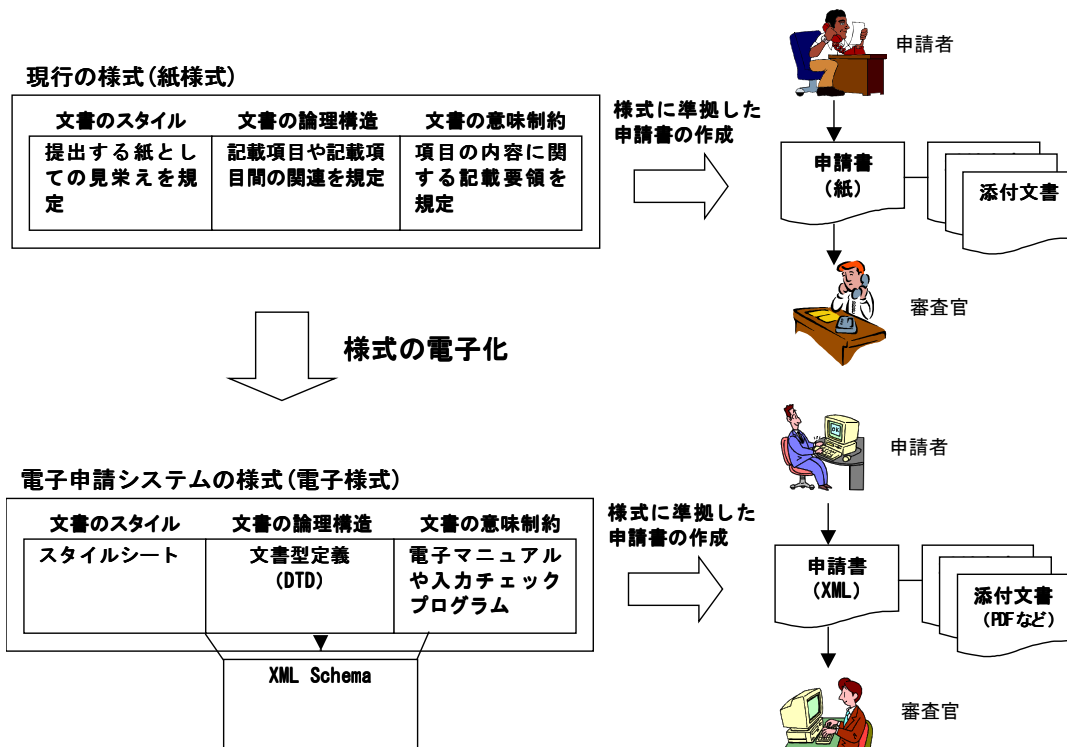


図1 紙様式と電子様式

申請書を XML 形式で電子化するにあたっては、様式を構成するそれぞれの規定を電子化する必要がある(紙様式を電子化したものを「電子様式」と呼ぶ)。本ガイドライン策定時の XML 技術では、「文書のスタイル」を規定する記述は、スタイルシートと呼ばれる。これは、例えば XSLT と呼ばれる XML 文書の変換言語により、XML を HTML に変換するための記述ファイルを提供し、XML 文書を WWW ブラウザ等で表示できるようにするものである。また、「文書の論理構造」は文書型定義(DTD: Document Type Definition)や XML Schema 等のスキーマによって表現できる。また、「文書の意味制約」は、数値や番号等のデータ型は XML Schema で記述できる。ただし、紙様式中の記入要領や備考等自然言語で書かれた説明の大半は、申請書の本文やスタイルシート中にテキストとしてそのまま挿入されるか、申請書入力支援ツール等から呼び出される入力チェックプログラムや電子マニュアルとして実現される。

本ガイドラインでは、「文書の論理構造」を電子的に表現するための設計を中心に扱う。「文書の論理構造」の表現には、現在運用中の電子申請システムからの継続性を重視して DTD(文書型定義)を採用しつつ、世の中の技術動向を踏まえて、XML Schema への対応についても記述する。

0.3. 電子申請において行政機関側ソフトウェアと申請者側ソフトウェア間で共有すべき規約

電子申請では、行政機関側と申請者側の間でインターネットを用いて申請書類一式が送信される。以下に、電子申請システムにおいて行政機関側ソフトウェアと申請者側ソフトウェアの間で共有すべき規約の概要と、本ガイドラインが対象とする申請書の論理構造の規定の位置付けについて説明する。

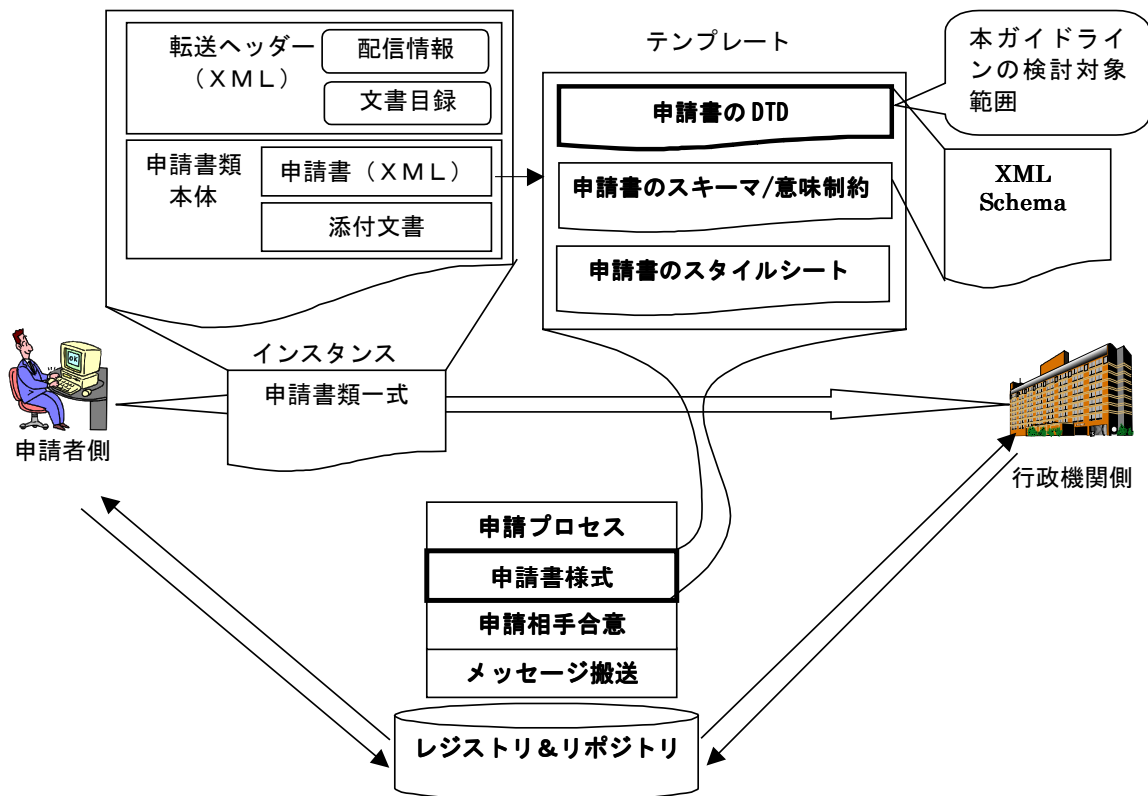


図2 電子申請において行政機関側ソフトウェアと申請者側ソフトウェア間で共有すべき規約

「図2 電子申請において行政機関側ソフトウェアと申請者側ソフトウェア間で共有すべき規約」に示すように、行政機関側ソフトウェアと申請者側ソフトウェアの間で共有すべき規約は、「申請プロセス」、「申請書様式」、「申請相手合意」、「メッセージ搬送」、「レジストリ&リポジトリ」の規約から構成される。以下に、これらの構成要素について順に簡単に説明する。

(1) 申請プロセス

申請届出、申請受理、内容問合せ、進捗確認、許可書発行等、申請業務の手続を規定したものである。

(2) 申請書様式

「0.2. 様式の電子化」で説明した内容がこれに該当する。本ガイドラインは、申請書様式のうち申請書の XML、DTD および XML Schema に相当する部分を検討対象とする。

(3) 申請相手合意

利用する認証や通信プロトコル等、コンピュータによる申請自動化に必要な合意事項の記述形式を規定したものである。申請相手合意に相当する記述が、提供される申請書入力支援ソフトウェアや送信ソフトウェア等に含まれることもある。

(4) メッセージ搬送

送信相手の指定や、信頼性の高い文書交換に必要とされる情報を記述する「配信情報」や、交換するビジネス

文書一式に含まれる文書を記した「文書目録」等の記述形式を規定したものである。申請書と添付文書を含めた申請書類一式のヘッダー情報が、これに相当する。

(5) レジストリ&リポジトリ

各種のソフトウェア（申請書入力支援ソフトウェア、送信ソフトウェア等）、申請書の電子様式、申請を規定する法令について、その登録方法や参照方法を規定したものである。

電子様式以外のさまざまな規定については、現在公開している NMDA 電子申請フレームワーク (<http://www.nmda.or.jp/nmda/soc/egov.html>) の改訂や、必要に応じて別ガイドラインの策定、公開を検討していく。

1. はじめに

1.1. 本ガイドラインの目的

経済産業省では、民間への行政サービス向上を目的に、電子政府の実現の一環として、インターネットによる申請・届出手続の電子化を推進している（「経済産業省 国の行政機関等の申請・届出等手続の電子化推進に関するアクション・プラン」 <http://www.meti.go.jp/topic/data/e10717aj.html>）。申請・届出手続の電子化にあたっては、共通的な利用が期待される電子申請様式のデータフォーマットについて世界的なインターネット関連の技術動向やシステム構築時の適応性に配慮し、申請書の文書形式としてインターネットでの次世代文書形式である XML を採用している。申請書を XML 化することにより、省庁内の業務システムでの自動処理が促進されるだけでなく、他省庁や他事業者とのデータ交換が容易になることが期待される。

XML は、業務要求に応じてタグを柔軟に定義できる。これは XML の大きな特長である反面、約 3,000 件におよぶ経済産業省の申請手続に対して、無方針にタグの設計を進めると、「申請者に対し、様式データごとに別のシステム環境を要求するおそれがある」、「他省庁等他機関との間に業務横断的な申請の仕組みが実現しにくくなる」、「特定の人しか保守できない」といった問題が生じる可能性がある。本ガイドラインの策定は、これらの問題への対処を目的としている。

本ガイドラインは、電子申請システム開発における申請書様式の作成者が、XML 形式の申請書のタグを設計する際に参考にできる情報を提供することを目的とする。なお、本ガイドラインの作成にあたっては、経済産業省および国土交通省の電子申請システム開発関係者が検討作業に参画し、実際の申請書様式を考慮した上で内容をまとめている。

1.2. ガイドラインが対象とする範囲

本節では、申請に関する文書交換フローと本ガイドラインの本版の検討範囲について説明する。

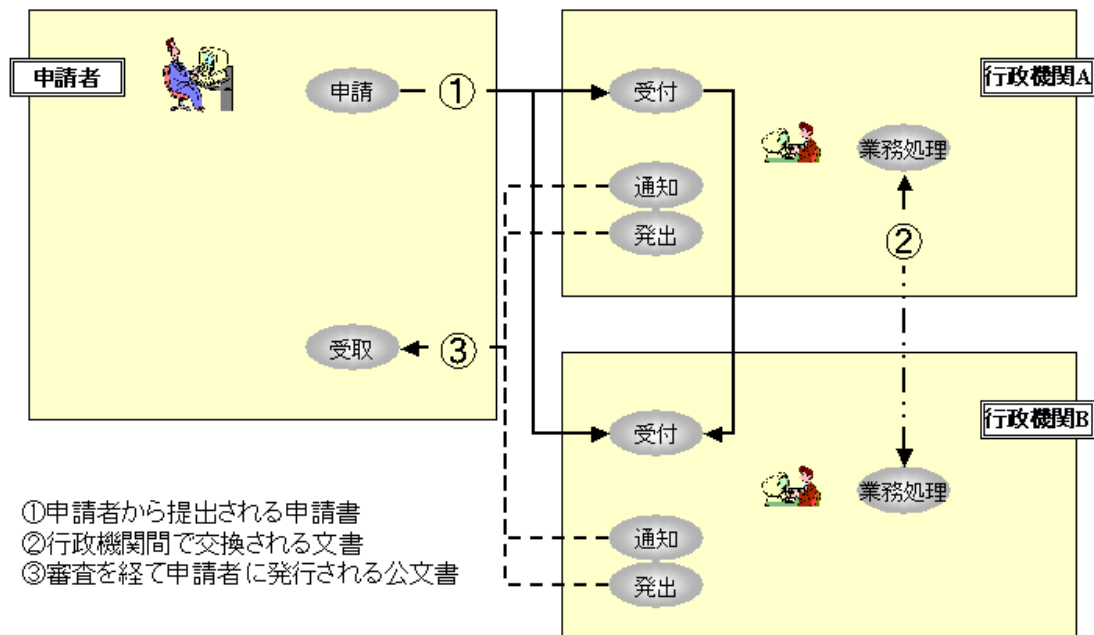


図3 申請に関する文書交換フロー

本ガイドラインの本版では、申請者の利便性に配慮し、申請者側から行政機関側に提出される申請書(申請書、届出書、報告書等を総称して「申請書」と呼ぶ)を検討対象とする(「図3 申請に関する文書交換フロー」の①を参照)。これにより、異なる行政機関に同一内容の申請を行う場合に、各々別のシステムを利用し、異なる申請書を作成しなければならない状況を避けることを狙いとする。また、タグ設計におけるガイドラインを示すことにより、当該様式の作成者以外にもその内容を理解できるように配慮し、保守性の向上を図るものとする。

行政機関間における申請書に関わる文書交換(「図3 申請に関する文書交換フロー」の②を参照)については、行政事務効率化の観点から、標準的な文書交換手法の確立が求められる。これについては、申請書自体ではなく、既存システムの実装等に大きく依存する部分であることを考慮し、本ガイドラインの次版以降での検討課題とする。ただし、提出された申請書がそのままの形式で交換される場合については、①と同等の扱いと考え、連携の容易さ等を考慮してタグの設計を進めるものとする。

行政機関から発行される公文書(「図3 申請に関する文書交換フロー」の③を参照)については、公文書の流通性の観点からも標準化が求められる部分であるが、電子署名法や政府認証基盤等の社会インフラの整備が並行して検討されている状況を鑑み、次版以降での検討課題とする。

1.3. 様式の電子化と本ガイドラインへの適合度

本ガイドラインを参照して様式を電子化する場合には、ガイドラインに対する適合度が考えられる。下記に示す例においては、各々の文中の前者のケースが比較的強くガイドラインに適合しており、後者のケースは少し緩いレベルでガイドラインに適合していると考えられることができる。

- ・ 「4. 申請書共通ボキャブラリ」に示す申請書共通ボキャブラリにおいて、モジュールの要素で定義されているタグ名を必ず用いるケースと、エンティティによる参照によって各々の項目の意味に応じて新たな名前のタグを定義するケース
- ・ 申請書共通ボキャブラリで示されている項目についてはガイドラインに全て従うケースと、システムの実装との関係から部分的に申請書共通ボキャブラリで用いていない名称でシステム固有のタグを定義して用いるケース（例：時刻の表現に関して、新たな構造のモジュール“時刻-X”を定義して使用する）
- ・ 典型的な申請書構造パターン（「3.1.2. 典型的な申請書構造パターンの提示」参照）にあてはまるケースと、そうでないケース

これら適合度については今後明確にしていく方針であるが、本ガイドラインを適用する人は、各々の事情に応じてガイドラインに対する適合度を考慮しながらガイドラインを活用されたい。

2. 申請書のタグ設計の考え方

本章では、本ガイドラインの対象である申請書の特徴を把握する目的で、「2.1. 申請書のタグ設計への基本要件」で、申請書のタグ設計の要件について述べ、続く「2.2. 申請書の論理構造モデル」で、現行の紙申請書様式の分析に基づく申請書の論理構造モデルについて述べる。そして、「2.3. 申請書のタグ設計の方針」で、この要件と論理構造モデルに基づいた申請書のタグ設計方針について述べ、続く「2.4. 申請書共通ボキャブラリ作成の方針」で、申請書共通ボキャブラリおよび分野別共通ボキャブラリの作成方針について説明する。また、「2.5. 申請書のスタイル」では、申請書のスタイルについて説明する。最後の「2.6. 申請書のタグ設計規約および申請書共通ボキャブラリの主な内容」では、申請書のタグ設計を支援するための本ガイドラインの構成について述べる。

2.1. 申請書のタグ設計への基本要件

電子申請システムで用いられる申請書のタグを設計するにあたっては、以下の3つの基本要件がある。

- (1) 申請書の法令準拠性
申請書は、法令が規定する様式に従うこと。
- (2) 申請書の業務処理容易性
申請書内のデータは、電子申請システムの業務処理機能から利用しやすいものであること。
- (3) 行政機関間の申請書データ交換容易性
申請書は、行政機関間で交換しやすいものであること。

「申請書の法令準拠性」は、電子化にあたっての必須要件であり、「申請書の業務処理容易性」、「行政機関間の申請書データ交換容易性」は、申請書の電子化の効果を向上させるための要件である。以下に、これら3つの基本要件について説明する。

(1) 申請書の法令準拠性（申請書は、法令が規定する様式に従うこと）

申請者側から行政機関側に提出される申請書は、法令が規定する様式に従うことが必要である。したがって、様式設計者は、この様式が規定する「文書の論理構造（申請書内での記載項目や記載項目間の関連）」を表現できるように申請書のタグを設計することが必要である。

(2) 申請書の業務処理容易性（申請書内のデータは、電子申請システムの業務処理機能から利用しやすいものであること）

電子申請によって申請者側への行政サービスを向上させるには、申請者にとって使いやすい申請書作成支援機能を提供したり、行政機関の審査官が使いやすい審査支援機能を提供したり、あるいはシステム機能が法令の改正に伴う様式変更迅速かつ柔軟に対応できることが求められる。こうした要求に応えるために、電子申請システムの申請書内のデータは、電子申請システムの業務処理機能から利用しやすいものであることが求められる。

(3) 行政機関間の申請書データ交換容易性（申請書は、行政機関間で交換しやすいものであること）

電子政府構想の進展に伴い、行政機関間における様々な業務連携の重要性が今後ますます高まっていくと思われる。そのため、申請書やそれに付随する各種のデータを、行政機関間で交換しやすいものにする必要がある。本ガイドラインでは、「1.2. ガイドラインが対象とする範囲」で説明したように、異なる機関に同一の内容の申請を行う場合に、申請書内のタグが同一であるか、相互の対応関係が取りやすくなるようなタグの設計を検討している。

2.2. 申請書の論理構造モデル

「2.1. 申請書のタグ設計への基本要件」で説明したタグ設計の基本要件「(1)申請書の法令準拠性」を把握する目的で、経済産業省や国土交通省等の省庁の申請書様式を分析した。この結果得られた申請書の論理構造モデルについて説明する。

「図4 申請書の論理構造モデル」に示すように、申請書の記載項目の論理構造は、申請書の管理用部分と内容部分に分かれており、そのうち申請書の内容部分は、申請書様式ごとに異なる「様式個別部分」とそれぞれの申請書様式に共通する「様式共通部分」から構成されている。

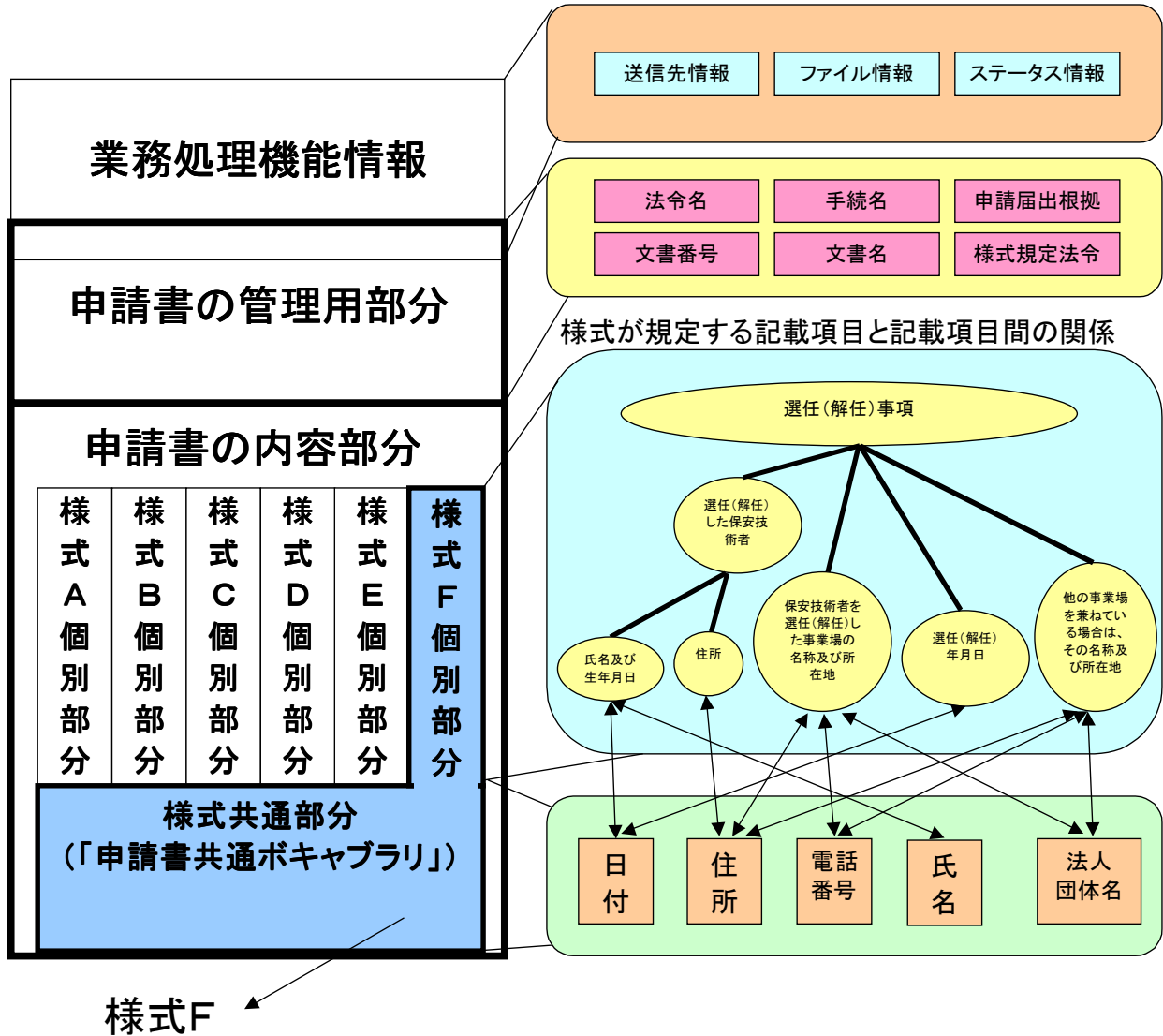


図4 申請書の論理構造モデル

(1) 申請書の管理用部分

「申請書の管理用部分」とは、文書番号（様式番号）や文書名等、申請書を管理・識別するために必要な、様式に共通する部分である。

(2) 申請書の内容部分

「申請書の内容部分」は、申請書様式によって異なる様式個別部分と、各申請書様式に共通する様式共通部分から構成される。

・ 様式個別部分

「様式個別部分」は、「図 4 申請書の論理構造モデル」の場合、「選任（解任）事項」、「選任（解任）した保安技術者」等、申請の内容に依存して決まる記載項目や、それらの項目が成す階層構造の部分である。

・ 様式共通部分

「様式共通部分」は、「日付」、「住所」、「電話番号」、「氏名」等、申請書様式に依存せず共通して現れる記載項目や、それら記載項目のデータ型の部分である。本ガイドラインでは、申請書様式に共通して用いられる要素や属性という意味で、この「様式共通部分」を「申請書共通ボキャブラリ」と呼ぶ。

(3) 業務処理機能情報

電子申請システムの業務処理機能（申請書作成支援、申請書送受信、申請審査支援、申請書管理等の各機能）に必要な情報や申請書以外の管理用ファイルやデータベース項目に必要な情報は、申請書の管理用部分と密接に関係すると考えられるが、実装の仕方により大きく処理の仕方が異なるため、申請書の管理用部分としては取り扱わないこととする。

2.3. 申請書のタグ設計の方針

本節では、最初に申請書のタグ設計の方針として、「2.1. 申請書のタグ設計への基本要件」に基づいて、業務処理容易性を満たすために「専用タグ付け方式」を採用した背景について説明する。次に、「2.2. 申請書の論理構造モデル」に応じたタグ設計方針について説明する。

2.3.1. 「専用タグ付け方式」の採用について

一般に、文書のタグ設計では、どの程度の抽象度で設計を実施するかという問題がある。申請書様式の場合で説明すると、「章」、「節」、「項」等の階層構造、表の階層構造、フォントサイズ等の表示構造を始め、すべての申請書が備えた抽象度で汎用のタグ付けを行う方式（「汎用タグ付け方式」）と、申請書様式ごとに決まる記載項目である「選任（解任）事項」、「技術者の要件を備えていることの証明」等、個々の様式が備えた専用のタグ付けを行う方式（「専用タグ付け方式」）がある。

いずれのタグ付け方式を採用するかは、文書の電子化の目的によって異なる。本ガイドラインでは、「2.1. 申請書のタグ設計への基本要件」で説明したタグ設計の基本要件「(2) 申請書の業務処理容易性」の観点から、業務処理機能から申請書毎の記載項目の内容を容易に参照できるようにするため、後者の「専用タグ付け方式」を採用する。

2.3.2. 申請書の論理構造モデルを反映したタグ設計方針

申請書様式のタグ設計は、申請書の論理構造モデルを構成する「申請書の管理用部分」、「様式個別部分」、「様式共通部分」ごとに進めるものとする。

申請書様式のタグ設計方針を、「図5 申請書の論理構造モデルを反映したタグ設計方針」に示す。

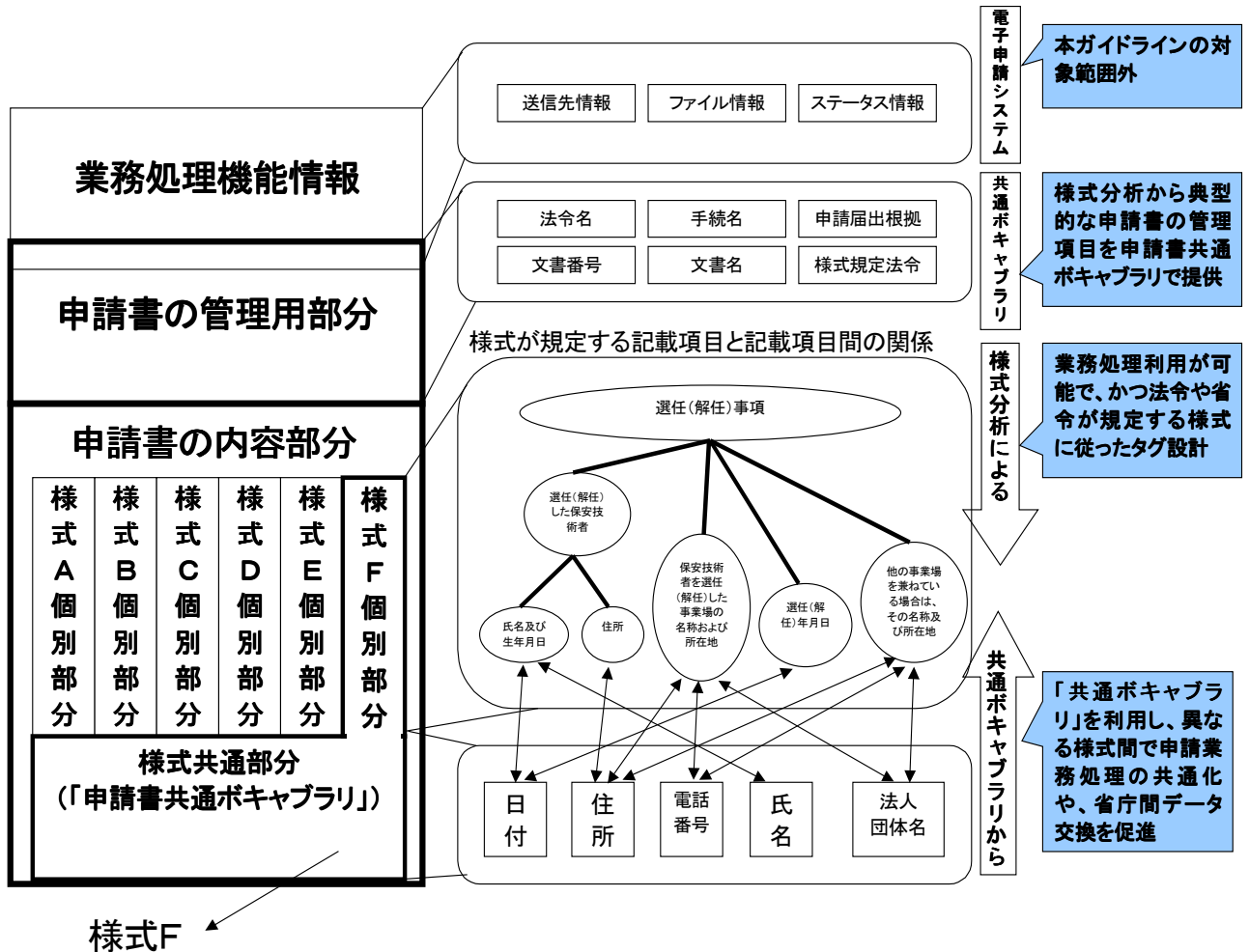


図5 申請書の論理構造モデルを反映したタグ設計方針

以下に、申請書の管理用部分と様式個別部分の設計方針と、申請書共通ボキャブラリの役割について説明する。

(1) 申請書の管理用部分のタグ設計方針

申請書の管理用部分のタグ設計では、経済産業省及び国土交通省等の様式の分析により、大半の様式に共通的に適用されている文書番号（様式番号）、文書名、申請届出根拠等の様式を管理・識別する項目について抽出を行い、申請書共通ボキャブラリとして提供している。

この申請書の管理用部分について、申請書共通ボキャブラリにて対応できない管理項目が出てきたときには、様式上の管理・識別の意味を包含している項目かどうかを検討したうえで、タグ設計を行うこととする。

(2) 様式個別部分のタグ設計方針

様式個別部分のタグ設計では、法令が規定する様式を分析した上で、個々の記載項目の内容が業務処理機能から参照できるように、専用タグ付け方式に従って、タグを設計する。

様式の分析にあたっては、「〇〇項目に××を記入する」等の記載すべき項目の規定や、様式上での記載項目枠間における表示上の包含関係から、「記載項目」と「記載項目間の関連」を抽出する。記載項目間の関連については、この抽出作業の上で、個々の記載項目に対応するタグを設計し、タグ間の階層関係が成す構造としてDTD等を用いて表現する。

また、抽出された記載項目に対応するタグを決めるにあたっては、本ガイドラインの申請書共通ボキャブラリが利用できるかどうかを配慮しつつ、設計を進める。

(3) 申請書共通ボキャブラリの役割

申請書様式の間で共通して用いられている「記載項目」や「記載項目の内容のデータ形式」を抽出して、申請書共通ボキャブラリを作成し申請者用XML様式の設計の際に利用できるライブラリとして提供する。これにより、様々な様式の統一的な管理が可能となり、更に様式設計が容易になったり、行政機関間のデータ交換を促進したりすることができる。

2.4. 申請書共通ボキャブラリ作成の方針

申請書共通ボキャブラリの作成にあたっては、各省庁の申請書様式中に共通して出現する記載項目を収集・分析した上で、申請書様式に共通して用いられる可能性の高い要素や属性を集め、「申請書共通ボキャブラリ」として定義した（申請書共通ボキャブラリの作成手順については、「付録D 申請書共通ボキャブラリの作成手順」を参照されたい）。

2.4.1. 申請書共通ボキャブラリがカバーする範囲

本ガイドラインにおける申請書共通ボキャブラリは、行政機関への申請において特有な項目や、共通的に出現する項目に着目する。申請書の内容によっては、金融分野や科学分野、物理量や単位に関する項目等が含まれる場合があるが、これらは各分野や業界毎のXMLに関する活動での標準に合わせるべき範囲とする。

ただし、事業の届け出を行う企業の概要を示すための「資本金」のような、行政機関への申請に共通的に現われるいくつかの項目については、様式の電子化の際の利便性を考えて例外的に本ガイドラインにて扱うこととした。上記の考え方を「図6 申請書共通ボキャブラリがカバーする範囲」に示す。

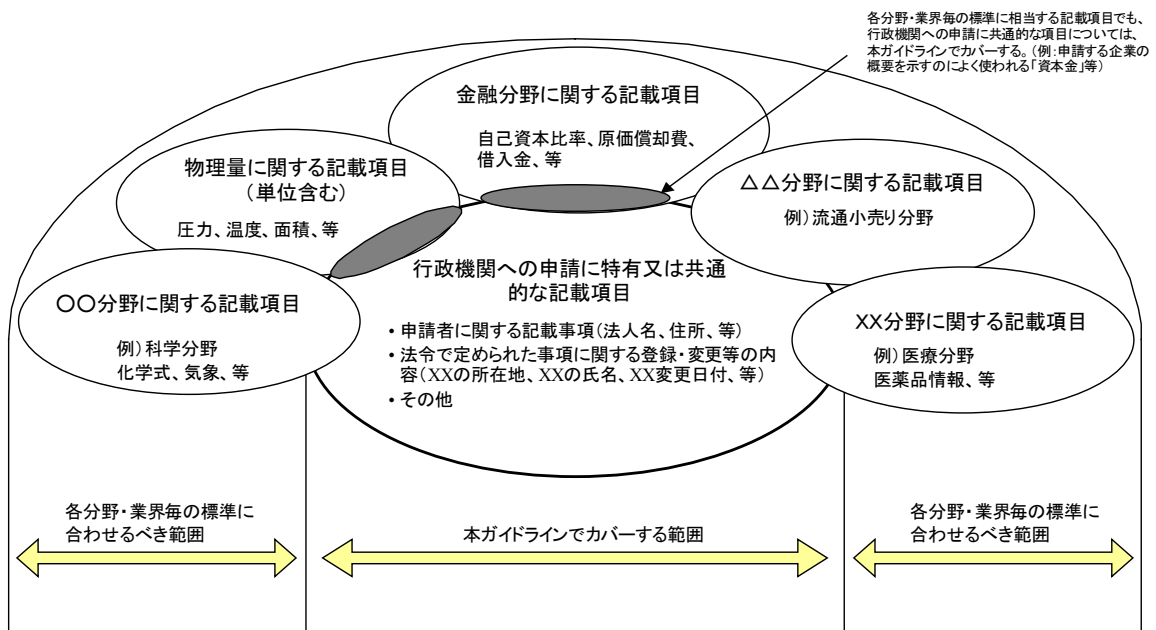


図6 申請書共通ボキャブラリがカバーする範囲

2.4.2. 分野別共通ボキャブラリ

行政機関への申請において特有な項目や共通的に出現する項目を抽出してまとめたものが「申請書共通ボキャブラリ」である。ところで、様式を電子化する際には、これに加え、一定の分野で出現頻度の高い項目をまとめ、「分野別共通ボキャブラリ」として（ライブラリ等の形態をとって）実装用に提供することにより、DTD や XML の設計の利便性が増すことが予想される。

例えば面積や温度、体積といった一般に単位と関係した物理量は、行政機関へのあらゆる申請に共通的に出現するわけではないが、出現頻度は比較的高い（※）。そこで、これらを「物理量に関する記載項目」としてまとめれば、様式電子化の際に定義済みのボキャブラリを利用できるため、タグ設計の負担を軽減できると考えられる（「付録 J 基本物理量ボキャブラリ」を参照されたい）。

また、法令等で規定されている様式には、一連の法令に共通する言い回しや、固有の意味をもった専門的な記載項目が使われている場合がある。例えば「〇〇事業の開始申請」「〇〇事業の変更届出」「〇〇事業の終了届出」といった、一連の法令で規定された関連性のある手続を見ると、申請者を表す記載項目や許認可番号等が、その法令内で一定の共通性を備えていることが多い。そこで、法令や法律ごとにボキャブラリ（「〇〇分野に関する記載項目」）を作成すれば、「物理量に関する記載項目」の場合と同様、タグ設計の負担を軽減できると考えられる。あわせて、法改正時のメンテナンス性の向上も期待できる。

※ 経済産業省をはじめとする複数省庁の約 1000 様式を対象とした調査による。

2.4.3. 分野別共通ボキャブラリの作成

分野別共通ボキャブラリの作成にあたっては、「申請書共通ボキャブラリ」をその基盤とすることが求められる。申請書共通ボキャブラリは、前述のように、行政機関への申請において特有な項目や共通的に出現する項目を、さまざまな様式の統一的な管理を実現し、様式設計を容易にし、申請書のデータ交換容易性を高めることを想定して作られたものである。

したがって、分野別共通ボキャブラリは基本的にこの考え方に依拠しつつ、申請書共通ボキャブラリを補完し、「2.4.2. 分野別共通ボキャブラリ」に述べた効果を実現するように設計することが大切である。逆に、申請書共通ボキャブラリに該当するタグ名があるにもかかわらず、それを使わずに分野別共通ボキャブラリとして新たに独自のタグを定義するようなことは、申請者と行政機関、行政機関間でのデータ交換容易性を下げのおそれがあり、本ガイドラインの趣旨にはそぐわない。

2.4.4. 申請書の記載項目の考え方

行政機関への申請・届出手続は、その手続の根拠となる法令が必ず存在しており、「その法令に該当する者が、該当する組織へ、一定の期間内に、必要事項を提出する」という性質を持っている。そのため、申請書の様式では、「どの法令を根拠として(法令名・様式を規定している法令)」、「誰が(提出者)」、「どこへ(あて先)」、「いつ(提出日)」、そして「何を(内容)」等の記載項目をもつことになる。これらの記載項目のことを、申請書の様式が法令を満たすために必要な構成要素という意味で、「様式構成要素」と呼ぶことにする。様式構成要素は、以下の4つに分類できる。

- ・様式識別要素群 …………… 根拠となる法令との対応関係を示すための構成要素
- ・記載内容要素群 …………… 審査に必要な情報等、申請・届出の内容を記載するための構成要素
- ・記載支援要素群 …………… 申請書の作成を支援するための情報を付与するための構成要素
- ・行政使用要素群 …………… 審査の過程で生じた情報を行政機関が記載するための構成要素

以下に、今回のガイドラインで扱っている様式構成要素を、要素群ごとに示す。

(1) 「様式識別要素群」

- ① 文書名 …………… 申請・届出で提出される文書の名称を記載する項目
- ② 文書番号(様式番号) …… 様式等の文書を管理するための識別番号を記載する項目
- ③ 様式を規定している法令 …… 当該申請・届出に関する様式を規定している法令の法令名・条・項・号を記載する項目
- ④ あて先 …………… 様式等の文書の提出先を記載する項目
- ⑤ 申請届出根拠 …………… 該当する様式を規定している法令等に基づいて申請や届出をしていることを説明する項目
- ⑥ 法令名 …………… 申請手続が根拠としている法令の法令名・条・項・号を記載する項目
- ⑦ 手続名 …………… 法令で規定された申請手続名を記載する項目

※ ⑥、⑦については、実際には様式上には現れない場合が多いが、電子化する際に当該申請手続を特定するための重要な情報であるため、様式識別要素群の項目として列挙するものである。

(2) 「記載内容要素群」

- ⑧ 提出日 …………… 申請や届出をした日付を記載する項目
- ⑨ 提出者 …………… 文書を提出する（申請や届出をした）者に関する情報を記載する項目
- ⑩ 申請届出内容 …………… 法令ごとに決まる申請や届出の内容に関する情報を記載する項目

(3) 「記載支援要素群」

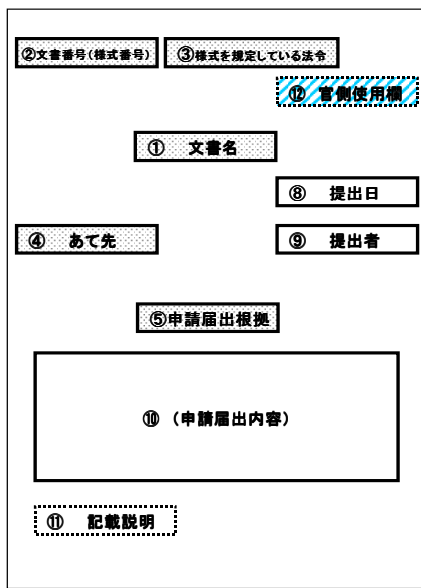
- ⑪ 記載説明 …………… 申請書の記載方法や注意事項等を説明するための情報を記載する項目
 実際の様式では、記載要領、記入要領、備考、注等の名称で呼ばれる

(4) 「行政使用要素群」

- ⑫ 官側使用欄 …………… 審査の過程で生じた情報を行政機関が記載する項目

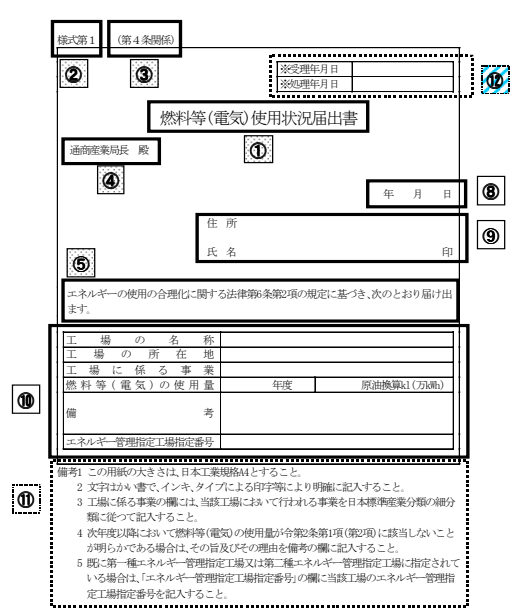
※ ⑫は紙の様式において行政機関側での管理等の利便性のために設けられている項目であり、申請者側が記入する項目ではない。従って電子化する場合には、他の項目と同様に一律にタグにする方法以外に、様式上のタグとしては扱わずに別な手段でシステム側で管理する方法も考えられる。この扱いはシステムの実装に依存するところである。

以下、様式構成要素を様式に見立てて配置した様式構成要素図を「図 7 様式構成要素の配置図」に、その様式構成要素を実際の様式に当てはめた様式構成要素図サンプルを「図 8 様式構成要素のサンプル図」に示す。



【凡例】
 : 様式識別要素群
 : 記載内容要素群

図 7 様式構成要素の配置図



⑪ 記載説明要素群
 : 行政使用要素群

図 8 様式構成要素のサンプル図

2.5. 申請書のスタイル

申請書のスタイルは、その用途に応じて要件が異なると考えられる。例えば、入力画面のスタイルと、印刷用のスタイルでは求められる要件が異なる。下記に一般的な要件を列挙する。

(1) 申請者入力画面用スタイル

申請者にとって入力しやすいことや、異なる様式間においても共通的な入力項目に関してスタイルに統一性があることが重要である。また、異なる行政機関の電子申請システムにおいて統一的な操作感を実現することが申請者の利便性という観点においては重要である。

(2) 行政機関審査画面用スタイル

申請者と審査官との間における問い合わせや確認作業の利便性のために、申請者入力画面との共通性を有することが重要である。また、例えば電子付箋機能のような電子申請システムにおける種々の審査支援機能の実装を踏まえたスタイルを設計することも必要である。

(3) プレビュー画面・印刷用スタイル

法令上の規定との関係や、当面は紙の申請と電子申請が並行して行われる状況では、既存の紙の様式のレイアウトを踏襲することが必要となったり、様々な制約条件に合わせたスタイル設計が求められることが多い。しかし、合理的なスタイル設計という考え方に立脚した場合には、例えば、申請者情報に関する部分（企業・団体名、氏名、住所等）の表示レイアウトを様式間で統一した方が良いという考え方もある。

その他、申請書のタグ設計との関係における留意事項として、申請書のタグとして記述するものと、スタイル上のみ記述して申請書のタグには記載しないものとの切り分けを明確にした上で申請書のタグ設計やスタイル設計を進めることが重要である。

例えば「文書番号（様式番号）」であるが、この部分は申請書のタグ側で記述する場合と、スタイル側で記述する場合とが考えられる。本ガイドラインにおいては、様式番号は単なる表示上の項目だけでなく、様式を特定する管理情報の一つとしてとらえ、タグとして記述することを推奨している。

2.6. 申請書のタグ設計規約および申請書共通ボキャブラリの主な内容

次章以降では、申請書様式の内容部分のタグ設計を支援するために、様式個別部分の設計を目的とした「3. 申請書のタグ設計規約」と、様式共通部分の設計を目的とした「4. 申請書共通ボキャブラリ」及び「4 申請書共通ボキャブラリ」を XML Schema にて対応した「5. 申請書共通ボキャブラリの XML Schema への適用」を規定する。以下、それぞれの内容について簡単に説明する。

「3. 申請書のタグ設計規約」の主な内容

- ・ 要素の構造化（どこまで細かく構造化するか等）
- ・ 典型的な申請書構造パターン
- ・ 要素内容と属性の使い分け、要素の命名規則
- ・ 申請書共通ボキャブラリの利用方法
- ・ 符号化文字集合、文字符号化方式、XML 宣言・文書型宣言等に関する規約

「4. 申請書共通ボキャブラリ」の主な内容

- ・ 申請書共通ボキャブラリの設計上の留意事項
- ・ 申請書共通ボキャブラリの具体的な解説

「5. 申請書共通ボキャブラリの XML Schema への適用」の主な内容

- ・ XML Schema 版申請書共通ボキャブラリの導入方針
- ・ XML Schema 版申請書共通ボキャブラリの具体的な解説

3. 申請書のタグ設計規約

本章では、様式個別部分の設計規約と符号化文字集合、文字符号化方式、XML 宣言・文書型宣言等に関する規約について説明する。

3.1. 様式個別部分の設計規約

申請書の様式個別部分のタグ設計では、法令が規定する様式における記載項目の構造を反映するように、スキーマ（DTD や XML Schema 等）を設計する。「図 5 申請書の論理構造モデルを反映したタグ設計方針」に示すように、スキーマの設計にあたっては、様式個別部分は様式分析により様式に従ったタグ設計を行い、様式共通部分は本ガイドラインで提供する申請書共通ボキャブラリを利用することを想定している。本節では、様式個別部分のスキーマの設計の進め方について、要素の構造化、要素内容と属性値の使い分け、要素の命名規則、申請書共通ボキャブラリの利用方法について説明する。

3.1.1. 要素の構造化

要素の構造化では、「どこまで細かく構造化するか」（粒度）や「どの程度の深さの階層構造をもたせるか」（深度）が問題となる。これらについては、電子申請支援システムにおける個別の機能要件を満たすように決めることが必要である。これらの機能要件は、申請書が対象とする法令、またその申請書を処理する業務システム機能の実装方式に依存する部分が大きいため、本ガイドラインでは具体的な規定を設けることはしない。以下に、一般的な留意事項や、配慮が求められる点について説明する。

- ・ 申請書の法令準拠性を満たすために、申請書のオンライン表示や印刷に必要なスタイルシートが作成しやすい構造であることが望ましい。また、スキーマを説明する仕様書（「様式説明書」）において、要素の名称や構造と法令が規定する記載項目との対応、記載項目間の関係等を説明することが望ましい。
- ・ 申請書の業務処理容易性を向上させるために、申請書の作成を支援する場合や、申請審査の支援、申請書の管理等を実現する各プログラムから参照しやすい構造であることが望ましい。特に、業務システムのデータベースで申請書内のデータを再利用する場合は、データベースのスキーマとのマッピングを取りやすい構造にすることが求められる場合もある。また、法令の改正に伴う様式変更に電子申請システムが容易に対応できるために、システムに携わる人間にとって読みやすい構造であることが望ましい。
- ・ 行政機関間の申請書データ交換容易性を向上させるために、できるだけ申請書共通ボキャブラリを利用することが望ましい。

3.1.2. 典型的な申請書構造パターンの提示

「3.1.1. 要素の構造化」に述べたように、本ガイドラインにおいて要素の構造化について規定は設けないが、経済産業省の実際の申請書を分析した結果によると、大半の申請書が「図9 申請書の典型的な構造」に示すような典型的な構造パターンを有していることがわかっている。よって本ガイドラインの実用性の観点から、この典型的な構造パターンを参考として提示する。これらは様式を電子化する際のひな形として利用可能なものであり、「付録I 電子申請用XML様式の作成手順 ―チュートリアル―」においてもこの典型的な構造パターンを考慮に入れて記述する。

多くの申請書様式がこの典型的な構造にあてはまるが、必ずしも全てではないため、あくまでも参考として提示するが、実用的な観点においては、この構造をひな形としてカスタマイズを行えば、大部分の様式が比較的容易に電子化可能である。

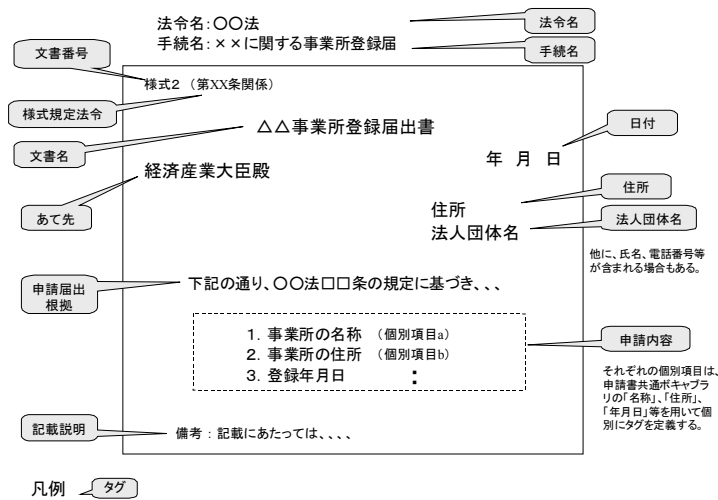


図9 申請書の典型的な構造

3.1.3. 要素内容と属性値の使い分け

申請者が入力した内容は要素の内容とし、属性値としないことを推奨する。ただし、属性値扱いとすることで電子申請システムにおけるデータ処理が容易になる場合は、その限りではない。また、この場合は、どのようなデータ処理が容易になるかについて、関連する仕様書に記載することが望ましい。

機械処理を行うための情報で、しかも構造を持たない情報は、属性値とすることを推奨する。この典型的な例として、他の情報を参照するためのハイパーリンクの記載や、参照する各種の定義ファイルのバージョン情報等が挙げられる。ただし、要素内容扱いとすることで電子申請システムにおけるデータ処理が容易になる場合は、その限りではない。また、この場合は、どのようなデータ処理が容易になるかについて、関連する仕様書に記載することが望ましい。

3.1.4. 要素の命名規則

システム開発や保守が容易になるように、要素や属性の名称は、電子申請システムの開発や申請書様式の保守に携わる人間にとって読みやすく、わかりやすいものとする。

要素の名称は、申請書様式を規定する法令で参照している名称と同一のものが望ましい。ただし、「名称が長すぎる」、「名称がわかりにくい」等の場合は、同一の名称を持つ要素が異なる意味を持たないように配慮すれば、短縮名称や別名称等を使用してもよい。ただし、要素の名称と法令用語の関係を把握できるようにするため、「短縮名称や別名称」と、「法令で参照している名称」との対応関係を、関連する仕様書に説明することが望ましい。

なお、以下の理由により、要素の名称には日本語を用いてもよいものとする。

- 現状の紙様式や、法令の記載項目名称との対応関係が取りやすく、スキーマの新規設計や法令改正時の保守が容易であること。
- システム開発に携わる人間にとって申請書の XML データが読みやすくなるため、システムの開発効率が向上すること。

3.1.5. 申請書共通ボキャブラリの利用方法

スキーマの設計を進める際は、様式を分析し、必要な記載項目を抽出して、各記載項目に対し申請書共通ボキャブラリ内の最適なタグを対応付けて行く。そのため、様式を設計する場合には、あらかじめ申請書共通ボキャブラリ内のタグの内容を把握しておくことが望ましい。

3.1.6. 設計にあたっての注意点

以上のタグの設計規約に基づき、4つの様式を電子化した結果を「付録A 申請書様式サンプル集」に示す。

このタグの設計作業中に、検討課題として「電子署名の扱い」と「行政機関側が記載する項目の扱い」の2つが挙げられた。

電子署名の扱いに関しては、行政機関への電子申請における電子署名情報の扱い方に関する動向を見ながら検討していく。従って、今回のサンプルではXML中に含めない方式としている。

紙の申請における行政機関側が記載する項目は、申請書を管理するための情報であり、申請書中に含まないという考え方と、紙の様式との見た目の互換性を保つため、XML中に記載すべきとの考え方がある。これらの考え方は、個々の電子申請システムの要件であることから、システム要件にあわせた実装にすることになる。

様式を電子化するにあたって今後検討が必要になると思われるその他の課題を「付録B 紙様式の電子化にあたっての課題」に示す。

3.2. 符号化文字集合、文字符号化方式、XML宣言・文書型宣言等に関する規約

本節では、符号化文字集合、文字符号化方式、XML宣言・文書型宣言等、XMLファイルとしての形式や宣言に関する規約について説明する。

3.2.1. 符号化文字集合・文字符号化方式等に関する規約

申請書に利用する文字は、システム間の相互運用性を意識した場合、システム間で共通して利用できるように考慮すべきである。

そのため、現段階では、日本語文字としてJIS X 0208(いわゆるJIS第1水準・第2水準)の漢字部分、およびJIS X 0201(アルファベット等)の符号化文字集合を利用することを推奨する。また、JIS X 0221(ISO 10646-1)の文字集合を利用してもよいが、日本語文字にJIS X 0212(補助漢字)が含まれているため、未対応のシステムで表示できないことを考慮する必要がある。

文字符号化方式としては、上記の符号化文字集合内の文字を十分に表現できること、他の文字符号化方式への変換が容易であることを確保できれば、本ガイドラインでは推奨や制約を設けない。他の文字符号化への変換を容易にするためには、規定した符号化文字集合内の文字に限定して利用することが重要である。

電子申請における外字の必要性についてここで言及すれば、電子申請システムのXML文書処理ソフトウェアは外字に対応していないため、外字は利用しないことを推奨する。ただし、今後は世の中の標準化動向を参考にしつつ、日本規格協会INSTAC等の検討委員会と情報交換を進めながら、外字への対応方式を検討していく。

3.2.2. XML宣言・文書型宣言等のXMLファイルとしての形式や宣言に関する規約

申請書のXMLファイルは、XML規約(JIS TR X 0008)に準拠し、妥当なXML文書であるものとする。すなわち、XMLファイル中にXML宣言と文書型宣言を持ち、文書型宣言で指示した文書型定義(DTD)の制約事項を満たすものとする。

4. 申請書共通ボキャブラリ

本章では、申請書共通ボキャブラリの設計上の留意事項について説明した後、申請書共通ボキャブラリを具体的に解説する。

本章で扱う申請書共通ボキャブラリは行政機関への申請書に特有または共通する記載項目であり、物理量に関する記載項目や金融分野に関する記載項目など、各分野・業界ごとの標準に合わせるべき範囲の記載項目は対象外とする（「2.4. 申請書共通ボキャブラリ作成の方針」参照）。しかし、各分野・業界に特化した記載項目の場合も、使用範囲の違いはあっても、データ交換の促進等、利用目的は同一であるため、本ガイドラインで定める規約に準拠して実装することを推奨する。

4.1. 申請書共通ボキャブラリとは

「4.2. 申請書共通ボキャブラリ」では、申請書において共通して出現する構造や要素を提供しており（「2.2. 申請書の論理構造モデル」参照）、様々な様式の申請書において共通の構造、要素を用いることで、「様式横断の申請書業務処理や管理の促進」や「行政機関間での申請書データ交換の促進」を目指している（「2.3.2.(2) 様式個別部分のタグ設計方針」参照）。

申請書共通ボキャブラリを設計・管理する際には、要素や構造を意味上の単位として扱い、この単位をモジュールと呼ぶこととする。例えば、様式は様式の内容を記載する部分である「記載内容要素群」からなり、その下に「氏名」、「日付」等が属し、「氏名」の下位に「氏」や「名」などが属すると考えられる。これらはそれぞれ意味をなすモジュールと考えることができ、「記載内容要素群モジュール」、「氏名モジュール」、「日付モジュール」、「氏モジュール」、「名モジュール」と呼ぶ。

申請書共通ボキャブラリの作成にあたっては、現状の省庁で扱われている紙の申請書様式を分析して、共通して出現する記載項目を収集した後、その記載項目中の構成要素を分析・整理することにより、一般に様式に共通して用いられる可能性が高いモジュールを選定した。選定した一連のモジュール群を定義した DTD を「ライブラリ」と呼び、利用者はこのライブラリを参照することにより、モジュールを利用することができる。

なお、申請書共通ボキャブラリの作成手順については「付録 D 申請書共通ボキャブラリの作成手順」を参照されたい。

モジュールの設計にあたっては、以下の点に留意した。

(1) タグ付けの粗さに対する配慮

申請書共通ボキャブラリを抽出する際に行った構成要素の分析・整理にあたっては、業務処理からの要求に応じてタグ付けの細かさを選べるように、内部構造を持たない「文字列」として扱うことを想定した「粗い」タグ付けを行う場合と、詳細なデータを扱う業務システムでの再利用を想定した「細かい」タグ付けを行う場合を併記することとした（※）。

※ 「粗い」タグ付け、「細かい」タグ付けとは、住所のタグ付けを例にとると、「住所全体を文字列として扱う場合」（「粗い」タグ付け）と「郵便番号、都道府県、市長村等と、住所の構成要素ごとにタグを付与する場合」（「細かい」タグ付け）を指す。

(2) 申請書共通ボキャブラリの設計を容易にするための「様式用語シソーラス」の提供

現状の紙の申請書様式を分析すると、例えば、「氏名」と抽象化できる用語だけでも、「報告者」、「事業者」、「申請者」、「使用者」、「被保険者」、「請求者」、「事業主」、「支払い者」、「あなたの氏名」、「届け出人」、「受給権者」、「請求人」等、法令や省令に応じて様々な表現が存在する。そこで、申請書共通ボキャブラリの設計者の便宜を図る目的で、「様式用語シソーラス」を付録Eとして提供した。「様式用語シソーラス」は「様式用語」を分類・モジュール化したもので、これに基づいて、例えば「氏名」や「住所」をモジュールとして利用できることになる。この情報は、申請書共通ボキャブラリを今後、充実させるための参考となる。

(3) 想定するライブラリの提供モデル

設計した申請書共通ボキャブラリは、以下のようにライブラリとして提供することを想定している。

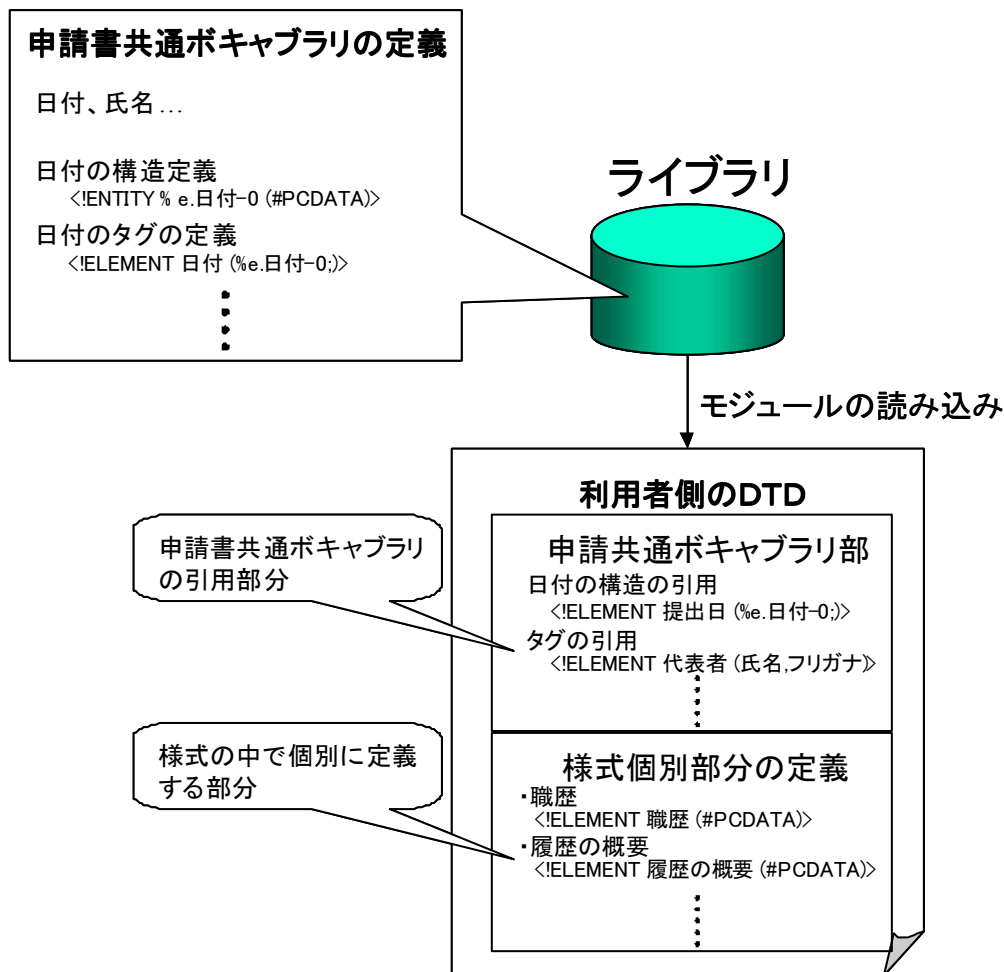


図 10 想定するライブラリの提供モデル

申請書共通ボキャブラリのモジュールは、タグ名や要素名として DTD に定義して様式設計者に提供する。様式設計者は提供された DTD を参照し、その中で定義されているタグ名や要素名を利用することにより、インターオペラビリティ（データ交換容易性）が保証されることになる。ただし、申請書共通ボキャブラリに依存しない形で新しくタグを定義する場合（タグの拡張）は、インターオペラビリティを阻害する要因になるので、やむを得ない場合を除き行わないことを推奨する。具体的な申請書共通ボキャブラリにおけるタグ名や要素名の定義の仕方については次節で詳述する。

(4) エンティティによる参照とエレメントによる参照

モジュールを参照する方法には、構造（構造に対するデータ型の定義を含む）による参照と要素による参照がある。

申請書共通ボキャブラリを構成するためには、法令の制約上、意味は同じであるが様式上の項目名が異なる場合を想定し、モジュールの構造を定義し、その構造を引用し、タグ名や要素名を定義できる方式が必要となる。本ガイドラインにおいては、構造による参照を DTD においては「エンティティによる参照」と呼び、後述する XML Schema においては「データ型による参照」と呼ぶ。DTD においては構造のみの定義が可能であり、XML Schema においては構造と構造に対するデータ型の定義が可能のため、この機能の違いに着目して便宜上、名称を分けて区別する。この構造を定義したモジュールを「e. XXX」（XXX は要素を示す任意の文字列）という記述で統一している。本章は DTD で説明を行うため、以下、エンティティについての説明を行う。

また、特に様式に依存せずタグの名称が決まっている場合には、エンティティで定義した構造に対し、そのままタグ名として利用できる方式が必要となる。このように要素の名称を参照する方式を「エレメントによる参照」と呼ぶ。

本申請書共通ボキャブラリは、双方の参照方式をもって構成する。

例えば「役職名」というモジュールを例にとると、

```
<ENTITY % e.役職名-0 "#PCDATA">… 「e.役職名-0」という構造の名称を定義
<ELEMENT 役職名 (%e.役職名-0;)> … 「役職名」という要素名を定義
```

というようにモジュールを提供する。この場合、役職名を「職名」のように別のタグ名として定義する場合には、エンティティによる参照を使用する。具体的には DTD で

```
<ELEMENT 職名 (%e.役職名-0;)>
```

と定義し、「職名」という要素に e. 役職名-0 という構造を当てはめる。そして XML で

```
<職名>代表取締役</職名>
```

と記述する。

逆に「役職名」というタグ名をそのまま使うには、エレメントによる参照を使用する。例えば代表者という要素の下に氏名と役職名という下位の要素を定義する場合である。この場合には DTD で

```
<!ELEMENT 代表者 (氏名,役職名)>
```

と定義し、そして XML で

```
<代表者>
  <氏名>山田太郎</氏名>
  <役職名>代表取締役</役職名>
</代表者>
```

と記述する。

どちらの参照方法を使うかは、項目の性質に応じて様式設計者が決定する。

(5) 既存の国際標準からのボキャブラリの流用

必要に応じて、既に確立している既存の国際標準のボキャブラリを参考にした。

(6) DTD から XML Schema への移行に対する配慮

将来的に、タグの内容のデータ型等を規定できる DTD の拡張であるスキーマの記述形式が標準化された場合を

想定して、申請書共通ボキャブラリが提供するモジュールをスキーマのデータ型と読みかえられるように配慮した。

なお XML Schema が W3C 勧告になったことを受け、本ガイドラインでは DTD での記述に加えて、XML Schema での表現を併記する。XML Schema に関する記載は「5. 申請書共通ボキャブラリの XML Schema への適用」を参照されたい。

(7) 申請書共通ボキャブラリの定義

申請書共通ボキャブラリの定義は、

<http://www.nmda.or.jp/sinsei/xml/vocabulary/vocabulary.dtd>

を参照されたい。

4.2. 申請書共通ボキャブラリ

本節では、選定したモジュールのうち、申請書様式の電子化にあたり、より利用度が高いと判断したものについて、以下のように「様式識別要素群」、「記載内容要素群」、「記載支援要素群」に分類し、個別に説明することとする。

- 「様式識別要素群」 …………… 文書名、文書番号（様式番号）、様式を規定している法令、あて先、申請届出根拠、法令名、手続名
- 「記載内容要素群」 …………… ■個人情報または組織情報に関する記載項目
 氏名、法人団体名、住所、電話番号、FAX 番号、電子メールアドレス、生年月日、性別、役職名、職業、国籍、本籍、続柄、業種
- 時間に関する記載項目
 日付、時刻
- 数量に関する記載項目
 金額、期間、年齢、従業員数、株式数
- 範囲に関する記載項目
 期間、区間
- その他の記載項目
 振り仮名、場所、理由、符号
- 「記載支援要素群」 …………… 記載説明

本章においては以下の順序に従って説明を行う。

- (1) 申請書共通ボキャブラリを構成するモジュールごとに「モジュール名」、「モジュールの申請書共通ボキャブラリにおける定義」、「説明」、「様式設計者の利用例」を提示する。
- (2) 「モジュールの申請書共通ボキャブラリにおける定義」はライブラリとして提供されるもの、「様式設計者の利用例」は様式設計者側で定義するものとして例示している。
- (3) 「説明」ではモジュールの構造と、データ形式について述べる。モジュールの構造では項目を1つの要素として扱うか、下位構造のある複数の要素に分解して扱うかについて説明し、データ形式では、要素のデータとしてどのような値（文字、数字等）を扱うべきかについて説明する。
- (4) 「様式設計者の利用例」では原則的に「モジュールの申請書共通ボキャブラリにおける定義」で定義されたモジュールを例にとり説明を行っている。ただし、構造が多岐にわたり個別に説明が必要な場合については、構造単位に適宜、説明を行っている。より詳細な定義については、「付録 H 申請書共通ボキャブラリ・リファレンス」を参照されたい。

4.2.1. 様式識別要素群

ここでは、「様式識別要素群」の様式構成要素である「文書名」、「文書番号（様式番号）」、「様式を規定している法令」、「あて先」、「申請届出根拠」、「法令名」、「手続名」について述べる。

4.2.1.1. 文書名に関する記載項目

文書名は様式上、「文書名」等で表現されている記載項目である。以下では文書名としての利用が想定される項目の例について述べる。

(1) モジュール名

モジュール名は文書名モジュールである。

文書名モジュールは、

- ・ エンティティ： e. 文書名-0
- ・ エレメント： 文書名

から構成される。

(2) モジュールの申請書共通ボキャブラリにおける定義

```
<!ENTITY % e.文書名-0 "#PCDATA">
<!ELEMENT 文書名 (%e.文書名-0;)>
```

(3) 説明

文書名は 1 つの要素として扱う。文書名のデータ形式は通常、全角の文字と考えられるが、申請によっては英語の文書名もあるため全角あるいは半角の英数字、どちらのデータにも対応可能にする必要がある。

(4) 様式設計者の利用例

文書名を表す要素は、「文書名」、「様式名」等の名称を用いる。

(i) DTD

- ・ 文書名モジュールのエンティティ（「e.文書名-0」）を参照し、別名（「様式名」）を宣言した例

```
<!ELEMENT 様式名 (%e.文書名-0;)>
```
- ・ 文書名モジュールのエレメント（「文書名」）を参照し、エレメント（「様式識別」）を宣言した例

```
<!ELEMENT 様式識別 (文書名,文書番号,××)>
```

(ii) XML 文書

- ・ 「様式名」のマークアップ例

```
<様式名>燃料等（電気）使用状況届出書</様式名>
```
- ・ 「様式識別」のマークアップ例

```
<様式識別>
<文書名>燃料等（電気）使用状況届出書</文書名>
<文書番号>様式第 10 号</文書番号>
<××>.....</××>
</様式識別>
```


4.2.1.2. 文書番号（様式番号）に関する記載項目

文書番号（様式番号）は様式上、「文書番号（様式番号）」等で表現されている記載項目である。以下では文書番号（様式番号）としての利用が想定される項目の例について述べる。

(1) モジュール名

モジュール名は文書番号モジュールである。

文書番号モジュールは、

- ・ エンティティ： e. 文書番号-0
- ・ エレメント： 文書番号

から構成される。

(2) モジュールの申請書共通ボキャブラリにおける定義

```
<!ENTITY % e.文書番号-0 "#PCDATA">
<!ELEMENT 文書番号 (%e.文書番号-0;)>
```

(3) 説明

文書番号は1つの要素として扱い、データ形式は全角または半角の文字である。

(4) 様式設計者の利用例

文書番号を表す要素は、「文書番号」、「様式番号」等の名称を用いる。

(i) DTD

- ・ 文書番号モジュールのエンティティ（「e. 文書番号-0」）を参照し、別名（「様式番号」）を宣言した例

```
<!ELEMENT 様式番号 (%e.文書番号-0;)>
```
- ・ 文書番号モジュールのエレメント（「文書番号」）を参照し、エレメント（「様式識別」）を宣言した例

```
<!ELEMENT 様式識別 (文書名, 文書番号, ××)>
```

(ii) XML 文書

- ・ 「様式番号」のマークアップ例

```
<様式番号>様式第1号</様式番号>
```
- ・ 「様式識別」のマークアップ例

```
<様式識別>
  <文書名>〇〇申請書</文書名>
  <文書番号>様式第1号</文書番号>
  <××>.....</××>
</様式識別>
```

4.2.1.3. 様式を規定している法令に関する記載項目

様式を規定している法令は様式上、「様式を規定している法令」等で表現されている記載項目である。以下では様式を規定している法令としての利用が想定される項目の例について述べる。

(1) モジュール名

モジュール名は様式規定法令モジュールである。

様式規定法令モジュールは、

- ・ エンティティ : e. 様式規定法令-0
- ・ エレメント : 様式規定法令

から構成される。

(2) モジュールの申請書共通ボキャブラリにおける定義

```
<!ENTITY % e.様式規定法令-0 "#PCDATA">
<!ELEMENT 様式規定法令 (%e.様式規定法令-0;)>
```

(3) 説明

様式規定法令は1つの要素として扱い、データ形式は全角または半角の文字である。

(4) 様式設計者の利用例

様式規定法令を表す要素は、「様式規定法令」、「根拠法令」等の名称を用いる。

(i) DTD

- ・ 様式規定法令モジュールのエンティティ（「e. 様式規定法令-0」）を参照し、別名（「根拠法令」）を宣言した例

```
<!ELEMENT 根拠法令 (%e.様式規定法令-0;)>
```

- ・ 様式規定法令モジュールのエレメント（「様式規定法令」）を参照し、エレメント（「様式識別」）を宣言した例

```
<!ELEMENT 様式識別 (文書名,様式規定法令,××)>
```

(ii) XML 文書

- ・ 「根拠法令」のマークアップ例

```
<根拠法令>第4条関係</根拠法令>
```

- ・ 「様式識別」のマークアップ例

```
<様式識別>
```

```
  <文書名>〇〇申請書</文書名>
```

```
  <様式規定法令>第4条関係</様式規定法令>
```

```
  <××>.....</××>
```

```
</様式識別>
```

4.2.1.4. あて先に関する記載項目

あて先は様式上、「あて先」等で表現されている記載項目である。以下ではあて先としての利用が想定される項目の例について述べる。

(1) モジュール名

モジュール名はあて先モジュールおよび経由モジュールである。

あて先モジュールは、

- ・ エンティティ : e.あて先-0
- ・ エlement : あて先

から構成される。

経由モジュールは、

- ・ エンティティ : e.経由-0
- ・ エlement : 経由

から構成される。

(2) モジュールの申請書共通ボキャブラリにおける定義

```
<!ENTITY % e.あて先-0 "#PCDATA">
<!ELEMENT あて先 (%e.あて先-0;)>
<!ENTITY % e.経由-0 "#PCDATA">
<!ELEMENT 経由 (%e.経由-0;)>
```

(3) 説明

あて先および経由はそれぞれ1つの要素として扱い、データ形式はともに全角または半角の文字である。

(4) 様式設計者の利用例

あて先を表す要素は、「あて先」、「提出先」等の名称を、経由を表す要素は「経由」等の名称を用いる。

(i) DTD

- ・ あて先モジュールのエンティティ（「e.あて先-0」）を参照し、別名（「提出先」）を宣言した例

```
<!ELEMENT 提出先 (%e.あて先-0;)>
```
- ・ あて先モジュールのエレメント（「あて先」）を参照し、エレメント（「様式識別」）を宣言した例

```
<!ELEMENT 様式識別 (文書名,あて先,経由,××)>
```

(ii) XML 文書

- ・ 「提出先」のマークアップ例

```
<提出先>国土交通大臣殿</提出先>
```
- ・ 「様式識別」のマークアップ例

```
<様式識別>
<文書名>〇〇に関する申請</文書名>
<あて先>国土交通大臣殿</あて先>
<経由>東京都知事殿</経由>
<××>.....</××>
</様式識別>
```

4.2.1.5. 申請届出根拠に関する記載項目

申請届出根拠は様式上、「申請届出根拠」等で表現されている記載項目である。以下では申請届出根拠としての利用が想定される項目の例について述べる。

(1) モジュール名

モジュール名は申請届出根拠モジュールである。

申請届出根拠モジュールは、

- ・ エンティティ : e. 申請届出根拠-0
- ・ エレメント : 申請届出根拠

から構成される。

(2) モジュールの申請書共通ボキャブラリにおける定義

```
<!ENTITY % e. 申請届出根拠-0 "#PCDATA">
<!ELEMENT 申請届出根拠 (%e. 申請届出根拠-0;)>
```

(3) 説明

申請届出根拠は1つの要素として扱い、データ形式は全角または半角の文字である。

(4) 様式設計者の利用例

申請届出根拠を表す要素は、「申請届出根拠」、「届出根拠」等の名称を用いる。

(i) DTD

- ・ 申請届出根拠モジュールのエンティティ（「e. 申請届出根拠-0」）を参照し、別名（「届出根拠」）を宣言した例

```
<!ELEMENT 届出根拠 (%e. 申請届出根拠-0;)>
```

- ・ 申請届出根拠モジュールのエレメント（「申請届出根拠」）を参照し、エレメント（「様式識別」）を宣言した例

```
<!ELEMENT 様式識別 (文書名, 申請届出根拠, ××)>
```

(ii) XML 文書

- ・ 「届出根拠」のマークアップ例

```
<届出根拠>エネルギー使用の合理化に関する法律第6条第2項の規定に基づき次のとおり届け出ます。</届出根拠>
```

- ・ 「様式識別」のマークアップ例

```
<様式識別>
```

```
<文書名>〇〇に関する申請</文書名>
```

```
<申請届出根拠>エネルギー使用の合理化に関する法律第6条第2項の規定に基づき次のとおり届け出ます。</申請届出根拠>
```

```
<××>・・・</××>
```

```
</様式識別>
```

4.2.1.6. 法令名に関する記載項目

法令名は電子化の際に当該手続を特定するための重要な情報である。以下では法令名としての利用が想定される項目の例について述べる。

(1) モジュール名

モジュール名は法令名モジュールである。

法令名モジュールは、

- ・ エンティティ： e. 法令名-0
- ・ エレメント： 法令名

から構成される。

(2) モジュールの申請書共通ボキャブラリにおける定義

```
<!ENTITY % e.法令名-0 "#PCDATA">
<!ELEMENT 法令名 (%e.法令名-0;)>
```

(3) 説明

法令名は1つの要素として扱い、データ形式は全角または半角の文字である。

(4) 様式設計者の利用例

法令名を表す要素は、「法令名」、「法令の名称」等の名称を用いる。

(i) DTD

- ・ 法令名モジュールのエンティティ（「e. 法令名-0」）を参照し、別名（「法令の名称」）を宣言した例

```
<!ELEMENT 法令の名称 (%e.法令名-0;)>
```
- ・ 法令名モジュールのエレメント（「法令名」）を参照し、エレメント（「様式識別」）を宣言した例

```
<!ELEMENT 様式識別 (法令名, 手続名, 文書名, ××)>
```

(ii) XML 文書

- ・ 「法令の名称」のマークアップ例

```
<法令の名称>輸入の促進および対内投資事業の円滑化に関する臨時措置法</法令の名称>
```
- ・ 「様式識別」のマークアップ例

```
<様式識別>
  <法令名>輸入の促進および対内投資事業の円滑化に関する臨時措置法</法令名>
  <手続名>特定製品輸入事業者認定申請</手続名>
  <文書名>〇〇に関する申請</文書名>
  <××>・・・</××>
</様式識別>
```

4.2.1.7. 手続名に関する記載項目

手続名は電子化の際に当該手続を特定するための重要な情報である。以下では手続名としての利用が想定される項目の例について述べる。

(1) モジュール名

モジュール名は手続名モジュールである。

手続名モジュールは、

- ・ エンティティ : e.手続名-0
- ・ エlement : 手続名

から構成される。

(2) モジュールの申請書共通ボキャブラリにおける定義

```
<!ENTITY % e.手続名-0 "#PCDATA">
<!ELEMENT 手続名 (%e.手続名-0;)>
```

(3) 説明

手続名は1つの要素として扱い、データ形式は全角または半角の文字である。

(4) 様式設計者の利用例

手続名を表す要素は、「手続名」、「手続の名称」等の名称を用いる。

(i) DTD

- ・ 手続名モジュールのエンティティ（「e.手続名-0」）を参照し、別名（「手続の名称」）を宣言した例

```
<!ELEMENT 手続の名称 (%e.手続名-0;)>
```
- ・ 手続名モジュールのエレメント（「手続名」）を参照し、エレメント（「様式識別」）を宣言した例

```
<!ELEMENT 様式識別 (法令名,手続名,文書名,××)>
```

(ii) XML 文書

- ・ 「手続の名称」のマークアップ例

```
<手続の名称>特定製品輸入事業者認定申請</手続の名称>
```
- ・ 「様式識別」のマークアップ例

```
<様式識別>
  <法令名>輸入の促進および対内投資事業の円滑化に関する臨時措置法</法令名>
  <手続名>特定製品輸入事業者認定申請</手続名>
  <文書名>〇〇に関する申請</文書名>
  <××>・・・</××>
</様式識別>
```

4.2.2. 記載内容要素群

ここでは様式上、記載内容要素群で使用される記入項目について述べる。

記載内容要素群は申請・届出内容に申請者が内容を記入する項目である。この中で分析済みの記載項目である、個人または組織情報に関する記載項目、時間に関する記載項目、数量に関する記載項目、範囲に関する記載項目、その他の記載項目について以下に説明する。

4.2.2.1. 個人情報または組織情報に関する記載項目

個人情報に属する主な項目としては、氏名、住所、郵便番号、電話番号などがある。

組織情報に属する主な項目としては、名称、氏名（代表者氏名等）、住所、郵便番号、電話番号などがある。

個人情報および組織情報は、様式上、複合的な項目として形成され、使用されるパターンが多い。例えば申請書の提出主体等を記載する項目として「氏名または名称」という複合的な項目があり、項目を1つとするか分割するかが実装上の課題となる。また、住所においては所在地という項目の下に郵便番号と住所を併記する場合があります。項目の論理的な構造に着目して実装を考える必要がある。

本節では個人情報、組織情報に関する記載項目を、氏名、法人団体名、住所に関するもの、連絡先に関するもの、その他のものに整理して記載する。

A 氏名に関する記載項目

氏名は様式上、「代表者氏名」等で表現されている記載項目である。以下では氏名としての利用が想定される項目の例について述べる。

氏名は単体の項目として存在する場合と、振り仮名の項目を含んでいる場合がある。本ガイドラインでは、氏名の中に振り仮名を包含している例で記述する。振り仮名については、「4.2.2.5.A 振り仮名に関する記載項目」を参照されたい。

(1) モジュール名

モジュール名は氏名モジュールである。

氏名モジュールは、

- ・ エンティティ：
 - e. 氏名-0（1つの要素として扱う場合）
 - e. 氏名-1（氏と名の要素に分ける場合）
 - e. 氏-0（氏と名の要素に分ける場合の「氏」）
 - e. 名-0（氏と名の要素に分ける場合の「名」）
- ・ エレメント：
 - 氏名（1つの要素として扱う場合）
 - 氏名-1（氏と名の要素に分ける場合）
 - 氏（氏と名の要素に分ける場合の「氏」）
 - 名（氏と名の要素に分ける場合の「名」）

から構成される。

(2) モジュールの申請書共通ボキャブラリにおける定義

- ・基本パターン（1つの要素として扱う場合）


```
<!ENTITY % e.氏名-0 "#PCDATA">
<!ELEMENT 氏名 (%e.氏名-0;)>
```
- ・その他のパターン（複数の要素に分解して扱う場合）


```
<!ENTITY % e.氏名-1 "氏, 名">
<!ENTITY % e.氏-0 "#PCDATA">
<!ENTITY % e.名-0 "#PCDATA">
<!ELEMENT 氏 (%e.氏-0;)>
<!ELEMENT 名 (%e.名-0;)>
<!ELEMENT 氏名-1 (%e.氏名-1;)>
```

(3) 説明

氏名を1つの要素として扱うか、氏と名とに要素分割して扱うかは業務に依存する。

氏名の各要素のデータ形式は基本パターン、その他のパターンともに通常、全角の文字と考えられるが、申請によっては英語で併記するような場合もあり、全角あるいは半角の英数字、どちらにも対応可能にする必要がある。

(4) 様式設計者の利用例

氏名を表す要素は、「氏名」、「提出者」等の名称を用いる。

(i) DTD

- ・氏名モジュールのエンティティ（「e.氏名-0」）を参照し、別名（「提出者」）を宣言した例


```
<!ELEMENT 提出者 (%e.氏名-0;)>
```
- ・氏名モジュールのエレメント（「氏名」）を参照し、エレメント（「代表者氏名」）を宣言した例


```
<!ELEMENT 代表者氏名 (氏名,フリガナ)>
```

(ii) XML 文書

- ・「提出者」のマークアップ例


```
<提出者>山田太郎</提出者>
```
- ・「代表者氏名」のマークアップ例


```
<代表者氏名>
  <氏名>山田太郎</氏名>
  <フリガナ>ヤマダタロウ</フリガナ>
</代表者氏名>
```

※ 留意事項

「氏名または名称」という項目が存在する場合、現行の紙の申請書様式では、氏名、法人団体名のどちらが入ってくるかは特定できない。業務上、どちらなのかを判別する必要がある場合には、氏名、法人団体名と明示的に項目を分ける必要がある。

B 法人団体名に関する記載項目

法人団体名は様式上、「法人または団体の名称」等で表現されている記載項目である。以下では法人団体名としての利用が想定される項目の例について述べる。

法人団体名は単体の項目として存在する場合と、振り仮名の項目を含んでいる場合がある。本ガイドラインでは、法人団体名の中に振り仮名を包含している例で記述する。

(i) モジュール名

モジュール名は法人団体名モジュールである。

法人団体名モジュールは、

- ・ エンティティ： e. 法人団体名-0
- ・ エレメント： 法人団体名

から構成される。

(2) モジュールの申請書共通ボキャブラリにおける定義

```
<!ENTITY % e.法人団体名-0 "#PCDATA">
<!ELEMENT 法人団体名 (%e.法人団体名-0;)>
```

(3) 説明

法人団体名は 1 つの要素として扱う。法人団体名のデータ形式は通常、全角の文字であるが、申請によっては英語で併記するような場合もあり、全角あるいは半角の英数字、どちらにも対応可能にする必要がある。

(4) 様式設計者の利用例

法人団体名を表す要素は、「法人団体名」、「名称」等の名称を用いる。

(i) DTD

- ・ 法人団体名モジュールのエンティティ（「e. 法人団体名-0」）を参照し、別名（「名称」）を宣言した例

```
<!ELEMENT 名称 (%e.法人団体名-0;)>
```
- ・ 法人団体名モジュールのエレメント（「法人団体名」）を参照し、エレメント（「事業所名」）を宣言した例

```
<!ELEMENT 事業所名 (法人団体名,フリガナ)>
```

(ii) XML 文書

- ・ 「名称」のマークアップ例

```
<名称>〇〇株式会社</名称>
```
- ・ 「事業所名」のマークアップ例

```
<事業所名>
  <法人団体名>〇〇株式会社</法人団体名>
  <フリガナ>マルマルカブシキガイシャ</フリガナ>
</事業所名>
```

※ 留意事項

氏名の項で述べたように、現行の紙の申請書様式には、「氏名または名称」というように複合的な項目として存在する場合がある。業務上、分けて使う必要がある場合には項目を明示的に分ける必要がある。

C 住所に関する記載項目

住所は様式上、「代表者の住所」等で表現されている記載項目である。以下では住所としての利用が想定される項目の例について述べる。

住所は単一の場合と、郵便番号と併記される場合がある。本ガイドラインでは住所の中に郵便番号を包含している例で記述する。

(1) モジュール名

モジュール名は住所モジュールおよび郵便番号モジュールである。

住所モジュールは、

- ・ エンティティ :
 - e. 住所-0 (1つの要素として扱う場合)
 - e. 住所-1 (複数の要素に分解して扱う場合)
 - e. 都道府県-0 (複数の要素に分解して扱う場合の「都道府県」)
 - e. 市郡-0 (複数の要素に分解して扱う場合の「市郡」)
 - e. 区-0 (複数の要素に分解して扱う場合の「区」)
 - e. 町名等-0 (複数の要素に分解して扱う場合の「町名等」)
- ・ エレメント :
 - 住所 (1つの要素として扱う場合)
 - 住所-1 (複数の要素に分解して扱う場合)
 - 都道府県 (複数の要素に分解して扱う場合の「都道府県」)
 - 市郡 (複数の要素に分解して扱う場合の「市郡」)
 - 区 (複数の要素に分解して扱う場合の「区」)
 - 町名等 (複数の要素に分解して扱う場合の「町名等」)

から構成される。

郵便番号モジュールは

- ・ エンティティ : e. 郵便番号-0
- ・ エレメント : 郵便番号

から構成される。

(2) モジュールの申請書共通ボキャブラリにおける定義

```

・基本パターン（1つの要素として扱う場合）

<!ENTITY % e.住所-0 "#PCDATA">
<!ELEMENT 住所 (%e.住所-0;)>

・その他のパターン（複数の要素に分解して扱う場合）

<!ENTITY % e.住所-1 "都道府県, 市郡, 区, 町名等">
<!ENTITY % e.都道府県-0 "#PCDATA">
<!ENTITY % e.市郡-0 "#PCDATA">
<!ENTITY % e.区-0 "#PCDATA">
<!ENTITY % e.町名等-0 "#PCDATA">
<!ELEMENT 都道府県 (%e.都道府県-0;)>
<!ELEMENT 市郡 (%e.市郡-0;)>
<!ELEMENT 区 (%e.区-0;)>
<!ELEMENT 町名等 (%e.町名等-0;)>
<!ELEMENT 住所-1 (%e.住所-1;)>

・郵便モジュール（住所モジュールの基本パターンおよびその他のパターンで共用）

<!ENTITY % e.郵便番号 "#PCDATA">
<!ELEMENT 郵便番号 (%e.郵便番号-0;)>

```

(3) 説明

住所を1つの要素として扱うか、都道府県、市、郡、区、町名等の要素に分割して扱うかは業務に依存する。

住所を要素分割する場合、どこまで分割するかについて検討する必要がある。考え方によっては都道府県、市、区、町名、番地、施設名等のように分割することも可能であるが、住所の要素は規則性が認められず、また、様式によっては略記される可能性もあるため、本ガイドラインでは都道府県、市郡、区、町名等といった粗いレベルで要素分割することとする。

住所の各要素のデータ形式は基本パターン、その他のパターンともに通常、全角の文字と考えられるが、申請によっては英語で併記するような場合もあり、全角あるいは半角の英数字、どちらにも対応可能にする必要がある。

郵便番号は通常、論理的に分解せずに1つで使用する場合が多いと考えられるため、1つの要素として扱う。

郵便番号のデータ形式は半角数字3桁、“-”（ハイフン）、半角数字4桁である。

(4) 様式設計者の利用例

住所を表す要素は、「住所」、「申請者の住所」等の名称を用いる。

(i) DTD

```

・住所モジュールのエンティティ（「e.住所-0」）を参照し、別名（「申請者の住所」）を宣言した例
<!ELEMENT 申請者の住所 (%e.住所-0;)>

・住所モジュールのエレメント（「住所」）を参照し、エレメント（「代表者の住所」）を宣言した例
<!ELEMENT 代表者の住所 (郵便番号,住所)>

```

(ii) XML 文書

```

・「申請者の住所」のマークアップ例
<申請者の住所>東京都港区三田一丁目4番28号</申請者の住所>

・「代表者の住所」のマークアップ例
<代表者の住所>
  <郵便番号>100-1111</郵便番号>
  <住所>東京都港区三田一丁目4番28号</住所>
</代表者の住所>

```

D 連絡先に関する記載項目

連絡先に関する項目として電話番号およびFAX番号がある。これらは様式上、「電話番号」、「FAX番号」等で表現されている記載項目である。以下では電話番号およびFAX番号としての利用が想定される項目の例について述べる。

現在の様式中には存在していないが、「電子メールアドレス」という項目も今後普及が予想され、申請書共通ボキャブラリに含める必要がある。

(1) モジュール名

モジュール名は電話番号モジュール、FAX番号モジュール、電子メールアドレスモジュールである。

電話番号モジュールは、

- ・ エンティティ : e. 電話番号-0
- ・ エレメント : 電話番号

から構成される。

FAX番号モジュールは

- ・ エンティティ : e. FAX番号-0
- ・ エレメント : FAX番号

から構成される。

電子メールアドレスモジュールは

- ・ エンティティ : e. 電子メールアドレス-0
- ・ エレメント : 電子メールアドレス

から構成される。

(2) モジュールの申請書共通ボキャブラリにおける定義

```

<!ENTITY % e.電話番号-0 "#PCDATA">
<!ELEMENT 電話番号(%e.電話番号-0;)>
<!ENTITY % e.FAX番号-0 "#PCDATA">
<!ELEMENT FAX番号(%e.FAX番号-0;)>
<!ENTITY % e.電子メールアドレス-0 "#PCDATA">
<!ELEMENT 電子メールアドレス(%e.電子メールアドレス-0;)>

```

(3) 説明

電話番号および FAX 番号は市外局番等によって構成されており、通常、業務上、これらを論理的に分割して利用するパターンは少ないと思われる。ただし、業務によっては論理的に分割する必要もある場合も想定される。その際には業務で利用するのに最適な単位で分割する。本ガイドラインでは、電話番号および FAX 番号を 1 つの要素として扱う。

電話番号および FAX 番号のデータ形式は半角の英数字である。

電子メールアドレスについては通常、業務上、論理的に分割して利用するパターンは少ないと思われる。ただアカウントとドメイン等のように論理的に分割して業務で利用する場合には、業務で最適な単位に分割する。本ガイドラインでは、電子メールアドレスを 1 つの要素として扱う。

電子メールアドレスのデータ形式は全角または半角の文字である。

(4) 様式設計者の利用例

連絡先を表す要素は、「電話番号」、「電話」等の名称を用いる。

(i) DTD

<ul style="list-style-type: none"> ・ 電話番号、FAX 番号、電子メールアドレスモジュールのエンティティ（「e.電話番号-0」、「e.FAX 番号-0」、「e.電子メールアドレス-0」）を参照し、別名（「電話」、「FAX」、「メール」）を宣言した例 <pre><!ELEMENT 電話 (%e.電話番号-0;)> <!ELEMENT FAX (%e.FAX 番号-0;)> <!ELEMENT メール (%e.電子メールアドレス-0;)></pre> <ul style="list-style-type: none"> ・ 電話番号、FAX 番号、電子メールアドレスモジュールのエレメント（「電話番号」、「FAX 番号」、「電子メールアドレス」）を参照し、エレメント（「連絡先」）を宣言した例 <pre><!ELEMENT 連絡先(電話番号,FAX 番号,電子メールアドレス)></pre>
--

(ii) XML 文書

<ul style="list-style-type: none"> ・ 「電話」、「FAX」、「メール」のマークアップ例 <pre><電話>03-1234-5678</電話> <FAX>03-8765-4321</FAX> <メール>aaaa@bbb.co.jp</メール></pre> <ul style="list-style-type: none"> ・ 「連絡先」のマークアップ例 <pre><連絡先> <電話番号>03-1234-5678</電話番号> <FAX 番号>03-8765-4321</FAX 番号> <電子メールアドレス>aaaa@bbb.co.jp</電子メールアドレス> </連絡先></pre>

E その他の個人情報または組織情報に関する記載項目

その他に個人情報または組織情報として、生年月日、年齢、性別、役職名、職業、国籍、本籍、続柄、従業員数、株式数、業種がある。

生年月日は日付モジュールを、本籍は住所モジュールをそれぞれ様式設計者側で適用する。年齢、従業員数、株式数については「4.2.2.3. 数量に関する記載項目」を参照されたい。その他の項目は論理的に1つの要素を持つ構造としてDTDに定義しモジュールとして提供する。

これらのモジュールの利用例については「付録H 申請書共通ボキャブラリ・リファレンス」を参照されたい。

4.2.2.2. 時間に関する記載項目

時間に関する記載項目としては日付、時刻がある。日付と時刻について以下に説明する。

A 日付に関する記載項目

日付は様式上、「到達日」等の日付で表現されている記載項目である。以下では日付としての利用が想定される項目の例について述べる。

日付の表現として以下の4つが考えられる。

- ・ 文字列型日付形式
1つの文字列として扱う形式である。
- ・ 年号型日付形式
和暦もしくは西暦を表現する年号（西暦を表現する場合には年号に「西暦」が代入される）、年、月、日から構成される形式である。
この形式は様式上、和暦と西暦いずれで記入する場合も存在することから、どちらにも対応可能なものとして用意したものである。
- ・ 西暦型日付形式
西暦で年、月、日から構成される形式である。
- ・ ISO8601型日付形式
ISO8601で規定している日付の形式である。

(1) モジュール名

モジュール名は日付モジュールである。

日付モジュールは、

- ・ エンティティ：
 - e. 日付-0（文字列型日付形式）
 - e. 日付-1（年号型日付形式）
 - e. 日付-2（西暦型日付形式）
 - e. 日付-3（ISO8601型日付形式）
 - e. 年号-0（年号型日付形式における「年号」）
 - e. 年-0（年号型日付形式における「年」）
 - e. 年-1（年号型日付形式における和暦の「年」）
 - e. 年-2（年号型日付形式もしくは西暦型日付形式における西暦の「年」）
 - e. 月-0（年号型日付形式もしくは西暦型日付形式における「月」）
 - e. 日-0（年号型日付形式もしくは西暦型日付形式における「日」）

- ・ エレメント : 日付 (文字列型日付形式)
- 日付-1 (年号型日付形式)
- 日付-2 (西暦型日付形式)
- 日付-3 (ISO8601 型日付形式)
- 年号 (年号型日付形式における「年号」)
- 年 (年号型日付形式における「年」)
- 年-1 (年号型日付形式における和暦の「年」)
- 年-2 (年号型日付形式あるいは西暦型日付形式における西暦の「年」)
- 月 (年号型日付形式もしくは西暦型日付形式における「月」)
- 日 (年号型日付形式もしくは西暦型日付形式における「日」)

から構成される。

(2) モジュールの申請書共通ボキャブラリにおける定義

```

・ 文字列型日付形式
<!ENTITY % e.日付-0 "#PCDATA">
<!ELEMENT 日付 (%e.日付-0;)>

・ 年号型日付形式
<!ENTITY % e.日付-1 "年号 ,年, 月, 日">
<!ENTITY % e.年号-0 "#PCDATA">
<!ENTITY % e.年-0 "#PCDATA">
<!ENTITY % e.年-1 "#PCDATA">
<!ENTITY % e.年-2 "#PCDATA">
<!ENTITY % e.月-0 "#PCDATA">
<!ENTITY % e.日-0 "#PCDATA">
<!ELEMENT 年号 (%e.年号-0;)>
<!ELEMENT 年 (%e.年-0;)>
<!ELEMENT 年-1 (%e.年-1;)>
<!ELEMENT 年-2 (%e.年-2;)>
<!ELEMENT 月 (%e.月-0;)>
<!ELEMENT 日 (%e.日-0;)>
<!ELEMENT 日付-1 (%e.日付-1;)>

・ 西暦型日付形式
<!ENTITY % e.日付-2 "年-2,月,日">
<!ELEMENT 日付-2 (%e.日付-2;)>

・ ISO8601 型日付形式
<!ENTITY % e.日付-3 "#PCDATA">
<!ELEMENT 日付-3 (%e.日付-3;)>

```

(3) 説明

- ・ 文字列型日付形式
文字列型日付形式は1つの要素として扱い、データ形式は全角または半角の文字である。
- ・ 年号型日付形式
年号型日付形式は年号、年、月、日から構成される。
年号には和暦の場合、“明治”、“大正”、“昭和”、“平成”の値を用い、西暦の場合には“西暦”の値を用いる。
年、月および日のデータ形式は半角数字である。
- ・ 西暦型日付形式
西暦型日付形式は年-2（西暦年）、月、日から構成される。
年-2、月および日のデータ形式は半角数字である。
- ・ ISO8601型日付形式
ISO8601の5.2.1.1のComplete representationのBasic Formatに従い、（[ISO8601] ISO（International Organization for Standardization）. Representations of dates and times, 1988-06-15.
<http://www.iso.ch/iso/en/ISOonline.frontpage>）1つの要素として扱い、データ形式は半角数字である。

(4) 様式設計者の利用例

日付を記載するの要素は、「日付」、「到達日」等の名称を用いる。

(i) DTD

■文字列型日付形式

- ・ 文字列型日付形式モジュールのエンティティ（「e.日付-0」）を参照し、別名（「許認日」）を宣言した例

```
<!ELEMENT 許認日 (%e.日付-0;)>
```

- ・ 文字列型日付形式モジュールのエレメント（「日付」）を参照し、エレメント（「許認」）を宣言した例

```
<!ELEMENT 許認(氏名,日付)>
```

■年号型日付形式

- ・ 年号型日付形式モジュールのエンティティ（「e.日付-1」）を参照し、別名（「許認日」）を宣言した例

```
<!ELEMENT 許認日 (%e.日付-1;)>
```

- ・ 年号型日付形式モジュールのエレメント（「日付-1」）を参照し、エレメント（「許認」）を宣言した例

```
<!ELEMENT 許認(氏名,日付-1)>
```


■西暦型日付形式

- ・西暦型日付形式モジュールのエンティティ（「e.日付-2」）を参照し、別名（「許認日」）を宣言した例

```
<!ELEMENT 許認日 (%e.日付-2;)>
```

- ・西暦型日付形式モジュールのエレメント（「日付-2」）を参照し、エレメント（「許認」）を宣言した例

```
<!ELEMENT 許認(氏名,日付-2)>
```

■IS08601型日付形式

- ・IS08601型日付形式モジュールのエンティティ（「e.日付-3」）を参照し、別名（「許認日」）を宣言した例

```
<!ELEMENT 許認日 (%e.日付-3;)>
```

- ・IS08601型日付形式モジュールのエレメント（「日付-3」）を参照し、エレメント（「許認」）を宣言した例

```
<!ELEMENT 許認(氏名,日付-3)>
```

(ii) XML 文書

■文字列型日付形式

- ・「許認日」のマークアップ例

```
<許認日>13年1月30日</許認日>
```

- ・「許認」のマークアップ例

```
<許認>
```

```
  <氏名>山田太郎</氏名>
```

```
  <日付>13年1月30日</日付>
```

```
</許認>
```

■年号型日付形式

(1) 和暦の場合

- ・「許認」のマークアップ例

```
<許認日>
```

```
  <年号>平成</年号>
```

```
  <年>13</年>
```

```
  <月>1</月>
```

```
  <日>30</日>
```

```
</許認日>
```

- ・「許認」のマークアップ例

```
<許認>
```

```
  <氏名>山田太郎</氏名>
```

```
  <日付-1>
```

```
  <年号>平成</年号>
```

```
    <年>13</年>
```

```
    <月>1</月>
```

```
    <日>30</日>
```

```
  </日付-1>
```

```
</許認>
```

(2) 西暦の場合

- ・「許認日」のマークアップ例

```
<許認日>
```

<p><年号>西暦</年号> <年>2001</年> <月>1</月> <日>30</日> </許認日></p> <p>・「許認」のマークアップ例 <許認> <氏名>山田太郎</氏名> <日付-1> <年号>西暦</年号> <年>2001</年> <月>1</月> <日>30</日> </日付-1> </許認></p> <p>■西暦型日付形式</p> <p>・「許認日」のマークアップ例 <許認日> <西暦年>2001</西暦年> <月>1</月> <日>30</日> </許認日></p> <p>・「許認」のマークアップ例 <許認> <氏名>山田太郎</氏名> <日付-2> <西暦年>2001</西暦年> <月>1</月> <日>30</日> </日付-2> </許認></p> <p>■ISO8601型日付形式</p> <p>・「許認日」のマークアップ例 <許認日>20010131</許認日></p> <p>・「許認」のマークアップ例 <許認> <氏名>山田太郎</氏名> <日付-3>20010131</日付-3> </許認></p>
--

B 時刻に関する記載項目

時刻は様式上、「時刻」等で表現されている記載項目である。以下では時刻としての利用が想定される項目の例について述べる。

なお、様式の分析により、時、分から構成される項目が存在することが判明した。他に時や分で単独で存在する記載項目への対応の検討、あるいは秒の扱い等については必要に応じて検討する。

時刻を表現する形式には以下の3つが考えられる。

- ・ 文字列型時刻形式
1つの文字列として扱う形式である。
- ・ 時分型時刻形式
時と分から構成される形式である。
- ・ ISO8601 型時刻形式
ISO8601 で規定している時刻の形式である。

(1) モジュール名

モジュール名は時刻モジュールである。

時刻モジュールは、

- ・ エンティティ :
 - e. 時刻-0 (文字列型時刻形式)
 - e. 時刻-1 (時分型時刻形式)
 - e. 時刻-2 (ISO8601 型時刻形式)
 - e. 時-0 (時分型時刻形式における「時」)
 - e. 分-0 (時分型時刻形式における「分」)
- ・ エレメント :
 - 時刻 (文字列型時刻形式)
 - 時刻-1 (時分型時刻形式)
 - 時刻-2 (ISO8601 型時刻形式)
 - 時 (時分型時刻形式における「時」)
 - 分 (時分型時刻形式における「分」)

から構成される。

(2) モジュールの申請書共通ボキャブラリにおける定義

```

・ 文字列型時刻形式

<!ENTITY % e.時刻-0 "#PCDATA">
<!ELEMENT 時刻 (%e.時刻-0;)>

・ 時分型時刻形式

<!ENTITY % e.時刻-1 "時, 分">
<!ENTITY % e.時-0 "#PCDATA">
<!ENTITY % e.分-0 "#PCDATA">
<!ELEMENT 時 (%e.時-0;)>
<!ELEMENT 分 (%e.分-0;)>
<!ELEMENT 時刻-1 (%e.時刻-1;)>

・ ISO8601 型時刻形式

<!ENTITY % e.時刻-2 "#PCDATA">
<!ELEMENT 時刻-2 (%e.時刻-2;)>

```

(3) 説明

- ・ 文字列型時刻形式
文字列型時刻形式は1つの要素として扱い、データ形式は全角または半角の文字である。
- ・ 時分型時刻形式
時分型時刻形式は24時間制の時、分から構成され、各々の要素のデータ形式は半角数字である。
- ・ ISO8601 型時刻形式
ISO8601 の 5.3.1.2 の Representations with reduced precision の Basic Format に従い ([ISO8601] ISO (International Organization for Standardization). Representations of dates and times, 1988-06-15. <http://www.iso.ch/iso/en/ISOonline.frontpage>)、1つの要素として扱い、データ形式は半角数字である。

(4) 様式設計者の利用例

時刻を表す要素は、「時刻」、「提出時刻」等の名称を用いる。

(i) DTD

```

■ 文字型時刻形式
・ 文字列型時刻形式モジュールのエンティティ（「e.時刻-0」）を参照し、別名（「提出時刻」）を宣言した例

<!ELEMENT 提出時刻 (%e.時刻-0;)>

・ 文字列型時刻形式モジュールのエレメント（「時刻」）を参照し、エレメント（「提出」）を宣言した例

<!ELEMENT 提出 (氏名,時刻)>

```

■時分型時刻形式

- ・時分型時刻形式モジュールのエンティティ（「e.時刻-1」）を参照し、別名（「提出時刻」）を宣言した例

```
<!ELEMENT 提出時刻 (%e.時刻-1;)>
```

- ・時分型時刻形式モジュールの元素（「時刻-1」）を参照し、元素（「提出」）を宣言した例

```
<!ELEMENT 提出 (氏名,時刻-1)>
```

■IS08601 型時刻形式

- ・IS08601 型時刻形式モジュールのエンティティ（「e.時刻-2」）を参照し、別名（「提出時刻」）を宣言した例

```
<!ELEMENT 提出時刻 (%e.時刻-2;)>
```

- ・IS08601 型時刻形式モジュールの元素（「時刻-2」）を参照し、元素（「提出」）を宣言した例

```
<!ELEMENT 提出 (氏名,時刻-2)>
```

(ii) XML 文書

■文字型時刻形式

- ・「提出時刻」のマークアップ例

```
<提出時刻>10 時 20 分</提出時刻>
```

- ・「提出」のマークアップ例

```
<提出>
```

```
  <氏名>山田太郎</氏名>
```

```
  <時刻>10 時 20 分</時刻>
```

```
</提出>
```

■時分型時刻形式

- ・「提出時刻」のマークアップ例

```
<提出時刻>
```

```
  <時>10</時>
```

```
  <分>20</分>
```

```
</提出時刻>
```

- ・「提出」のマークアップ例

```
<提出>
```

```
  <氏名>山田太郎</氏名>
```

```
  <時刻-1>
```

```
    <時>10</時>
```

```
    <分>20</分>
```

```
  </時刻-1>
```

```
</提出>
```

■IS08601 型時刻形式

- ・「提出時刻」のマークアップ例

```
<提出時刻>1020</提出時刻>
```

- ・「提出」のマークアップ例

<提出>
<氏名>山田太郎</氏名>
<時刻-2>1020</時刻-2>
</提出>

C 日付・時刻に関するその他のバリエーション

以上に述べた形式の他に、実際の様式の中には年だけもの、月だけのもの、あるいは年月等、いくつかのパターンの存在を確認した。これらについても日付型として定義しておくことも有効である。以下に実際に様式中に存在した日付型の候補を列記する。

- ① 年
- ② 月
- ③ 年+月
- ④ 年+月+日
- ⑤ 年号+年
- ⑥ 年号+年+月
- ⑦ 月+日
- ⑧ 日付と時刻の複合型

※ 留意事項

上記のうち、年号のない年については、現行の紙の申請書様式では和暦、西暦のどちらを記入するかの規定は特に定めていないため、和暦、西暦どちらの数値も入力する可能性がある。

4.2.2.3. 数量に関する記載項目

数量は数値および単位によって表現される項目である。

数量に関する項目は、数値とそれと対になる単位によって構成されており、単位をどのように扱うかについて以下に示す。

単位を扱うには以下の4方式が考えられる。

① 単位を明示しない方法

単位を項目に包含しているものとみなし数値のみをデータとして格納する。データの相互交換を行う際に単位の互換性を何らかの方法で担保する必要がある。

(例) <金額>1000</金額>

② 要素名に単位を表現する方法

単位をタグ名に含めて表現する方法である。

(例) <金額-円>1000</金額-円>

③ 要素で単位を表現する方法

要素で単位を定義する方法である。

(例) <金額>1000</金額>

<単位>円</単位>

④ 属性で単位を表現する方法

要素の属性で単位を表現する方法である。

(例) <金額 単位="円">1000</金額>

本ガイドラインにおいては想定される単位をモジュールとして提供することにより、単位の相互交換を担保する方式を採用する。ただし、単位の表現方法については①から④のいずれの方法を使っても良いこととする。

数量に関する記載項目を金額、期間、その他の数量それぞれについて説明する。

A 金額に関する記載項目

金額に関する事項として金額、資本金および売上高がある。これらは様式上、「給与」、「料金」（以上、金額）、「資本金」および「売上高」等で表現されている記載項目である。以下では金額、資本金および売上高としての利用が想定される項目の例について述べる。

資本金と売上高の組織の情報として様式中に多数存在したことから、これらを一般的な金額とは分けて扱うこととした。

なお、金額については、EC・EDI 分野で先行的に検討されており、それらとの整合性をどう取るかについては今後の検討課題とする。

(1) モジュール名

モジュール名は金額モジュール、資本金モジュール、売上高モジュールおよび金額の単位を表すモジュールである。

金額モジュールは、

- ・ エンティティ : e. 金額-0
- ・ エレメント : 金額

から構成される。

資本金モジュールは、

- ・ エンティティ : e. 資本金-0
- ・ エレメント : 資本金

から構成される。

売上高モジュールは、

- ・ エンティティ : e. 売上高-0
- ・ エレメント : 売上高

から構成される。

金額の単位を表すモジュールは、

- ・ エンティティ : e. 円-0
e. 千円-0
e. 百万円-0
e. ドル-0
e. 千ドル-0
e. 百万ドル-0
- ・ エレメント : 円
千円
百万円
ドル
千ドル
百万ドル

から構成される。

(2) モジュールの申請書共通ボキャブラリにおける定義

```
・金額
<!ENTITY % e.金額-0 "#PCDATA">
<!ELEMENT 金額 (%e.金額-0;)>

・資本金
<!ENTITY % e.資本金-0 "#PCDATA">
<!ELEMENT 資本金 (%e.資本金-0;)>

・売上高
<!ENTITY % e.売上高-0 "#PCDATA">
<!ELEMENT 売上高 (%e.売上高-0;)>

・金額の単位
<!ENTITY % e.円-0 "#PCDATA">
<!ENTITY % e.千円-0 "#PCDATA">
<!ENTITY % e.百万円-0 "#PCDATA">
<!ENTITY % e.ドル-0 "#PCDATA">
<!ENTITY % e.千ドル-0 "#PCDATA">
<!ENTITY % e.百万ドル-0 "#PCDATA">
<!ELEMENT 円 (%e.円-0;)>
<!ELEMENT 千円 (%e.千円-0;)>
<!ELEMENT 百万円 (%e.百万円-0;)>
<!ELEMENT ドル (%e.ドル-0;)>
<!ELEMENT 千ドル (%e.千ドル-0;)>
<!ELEMENT 百万ドル (%e.百万ドル-0;)>
```

(3) 説明

金額、資本金、売上高および金額の単位を表すモジュールはそれぞれ 1 つの要素として扱い、ともにデータ形式はカンマ区切りなしの半角数字である。

(4) 様式設計者の利用例

金額を表す要素は、「金額」、「料金」、「資本金」、「資本」、「売上高」、「売上金」等の名称を用いる。

(i) DTD

<ul style="list-style-type: none"> ・金額モジュールのエンティティ（「e.金額-0」）を参照し、別名（「支払額」）を宣言した例 <!ELEMENT 支払額 (%e.金額-0;)> ・金額モジュールのエレメント（「金額」）を参照し、エレメント（「支払」）を宣言した例 <!ELEMENT 支払 (氏名,金額)> ・資本金、売上金モジュールのエンティティ（「e.資本金-0」、「e.売上金-0」）を参照し、別名（「料金」、「資本」、「売上」）を宣言した例 <!ELEMENT 料金 (%e.金額-0;)> <!ELEMENT 資本 (%e.資本金-0;)> <!ELEMENT 売上 (%e.売上金-0;)> ・資本金、売上高モジュールのエレメント（「資本金」、「売上高」）を参照し、エレメント（「企業情報」）を宣言した例 <!ELEMENT 企業情報 (法人団体名,資本金,売上高)> ・金額の単位を表すモジュールのエンティティ（「e.円-0」）を参照し、別名（「交通費-円」）を宣言した例 <!ELEMENT 交通費-円 (%e.円-0;)>

(ii) XML 文書

<ul style="list-style-type: none"> ・「支払額」のマークアップ例 <支払額>300000</支払額> ・「支払」のマークアップ例 <支払> <氏名>山田太郎</氏名> <金額>300000</金額> </支払> ・「資本」、「売上」のマークアップ例 <資本>10000000</資本> <売上>50000000</売上> ・「企業情報」のマークアップ例 <企業情報> <法人団体名>A株式会社</法人団体名> <資本金>10000000</資本金> <売上高>50000000</売上高> </企業情報> ・「交通費-円」のマークアップ例 <交通費-円>300</交通費-円>
--

B 期間（数量）に関する記載項目

期間は年数、月数などの期間に関する記載項目である。以下では期間としての利用が想定される項目の例について述べる。

期間としては本節で説明する数量を表す期間の他に、「4.2.2.4.A 期間（範囲）に関する記載項目」で後述する開始日付から終了日付など範囲を意味する期間があり、数量を意味する期間を数量型期間モジュール、範囲を意味する期間を範囲型期間モジュールと呼んで区別する。

(1) モジュール名

モジュール名は数量型期間モジュールである。

数量型期間モジュールは、

- ・ エンティティ： e. 数量型期間-0
- ・ エレメント： 数量型期間

から構成される。

数量型期間の単位を表すモジュールは、

- ・ エンティティ：
 - e. 年数-0
 - e. 月数-0
 - e. 日数-0
 - e. 時間-0
- ・ エレメント：
 - 年数
 - 月数
 - 日数
 - 時間

から構成される。

(2) モジュールの申請書共通ボキャブラリにおける定義

```

・ 数量型期間
<!ENTITY % e.数量型期間-0 "#PCDATA">
<!ELEMENT 数量型期間 (%e.数量型期間-0;)>

・ 数量型期間の単位
<!ENTITY % e.年数-0 "#PCDATA">
<!ENTITY % e.月数-0 "#PCDATA">
<!ENTITY % e.日数-0 "#PCDATA">
<!ENTITY % e.時間-0 "#PCDATA">
<!ELEMENT 年数 (%e.年数-0;)>
<!ELEMENT 月数 (%e.月数-0;)>
<!ELEMENT 日数 (%e.日数-0;)>
<!ELEMENT 時間 (%e.時間-0;)>

```

(3) 説明

数量型期間および数量型期間の単位を表すモジュールは 1 つの要素として扱い、データ形式はカンマ区切りなしの半角数字である。

(4) 様式設計者の利用例

数量型期間を表す要素は、「期間」、「稼働期間」等の名称を用いる。

(i) DTD

・数量型期間モジュールのエンティティ（「e.数量型期間-0」）を参照し、別名（「稼働期間」）を宣言した例

```
<!ELEMENT 稼働期間 (%e.数量型期間-0;)>
```

・数量型期間モジュールのエレメント（「数量型期間」）を参照し、エレメント（「稼働実績」）を宣言した例

```
<!ELEMENT 稼働実績 (内容,数量型期間)>
```

・数量型期間の単位を表すモジュールのエンティティ（「e.日数-0」）を参照し、エレメント（「稼働実績-日」）を宣言した例

```
<!ELEMENT 数量型期間-日 (%e.日数-0;)>
```

(ii) XML 文書

・「稼働期間」のマークアップ例

```
<稼働期間>10日</稼働期間>
```

・「稼働実績」のマークアップ例

```
<稼働実績>
```

```
<内容>〇〇に関する作業</内容>
```

```
<数量型期間>10日</数量型期間>
```

```
</稼働実績>
```

・「稼働実績-日」のマークアップ例

```
<稼働実績-日>10</稼働実績-日>
```

C その他の数量に関する記載項目

上記の他に数量に関する記載項目として年齢、従業員数、株式数がある。それぞれ1つの要素として扱い、データ形式は半角の数字である。年齢は1歳単位、株式数は1株単位を基本とし単位を特にモジュールとして提供しない。従業員数は1人を単位として提供する。詳細は「付録H 申請書共通ボキャブラリ・リファレンス」を参照されたい。

また、その他に重さ、面積、発熱量等の物理的な分野に特化したものを基本物理量ボキャブラリとして示す。具体的な項目は「付録J 基本物理量ボキャブラリ」として記載したので、参照されたい。

4.2.2.4. 範囲に関する記載項目

時間あるいは物理的な起点と終点からなる範囲に関する記載項目について示す。

範囲に該当する記載項目として、期間、区間が存在する。以下、それぞれについて示す。

A 期間（範囲）に関する記載項目

期間は様式上、「（自）平成13年6月1日（至）平成13年6月30日」のように開始日付と終了日付によって表される記載項目である。以下では期間としての利用が想定される項目の例について述べる。

範囲型期間を表現する形式には以下の3つが考えられる。

- 文字列範囲型期間

1つの文字列として扱う形式である。

- 日付範囲型期間

開始日付、終了日付から構成される形式である。開始日付、終了日付はそれぞれ文字列型日付モジュールを適用する。

- 時刻範囲型期間

開始時刻、終了時刻から構成される形式である。開始時刻、終了時刻はそれぞれ文字列型時刻モジュールを適用する。

(1) モジュール名

モジュール名は範囲型期間モジュールである。

範囲型期間モジュールは、

- エンティティ：
 - e. 範囲型期間-0（文字列範囲型期間）
 - e. 範囲型期間-1（日付範囲型期間）
 - e. 範囲型期間-2（時刻範囲型期間）
- エレメント：
 - 範囲型期間（文字列範囲型期間）
 - 範囲型期間-1（日付範囲型期間）
 - 範囲型期間-2（時刻範囲型期間）
 - 開始日付（日付範囲型期間における「開始日付」）
 - 終了日付（日付範囲型期間における「終了日付」）
 - 開始時刻（時刻範囲型期間における「開始時刻」）
 - 終了時刻（時刻範囲型期間における「終了時刻」）

から構成される。

(2) モジュールの申請書共通ボキャブラリにおける定義

```

■文字列範囲型期間
<!ENTITY % e.範囲型期間-0 "#PCDATA">
<!ELEMENT 範囲型期間 (%e.範囲型期間-0;)>

■日付範囲型期間
<!ENTITY % e.範囲型期間-1 "開始日付,終了日付">
<!ELEMENT 開始日付 (%e.日付-0;)>
<!ELEMENT 終了日付 (%e.日付-0;)>
<!ELEMENT 範囲型期間-1 (%e.範囲型期間-1;)>

■時刻範囲型期間
<!ENTITY % e.範囲型期間-2 "開始時刻,終了時刻">
<!ELEMENT 開始時刻 (%e.時刻-0;)>
<!ELEMENT 終了時刻 (%e.時刻-0;)>
<!ELEMENT 範囲型期間-2 (%e.範囲型期間-2;)>

```

(3) 説明

以下、「文字列範囲型期間」、「日付範囲型期間」、「時刻範囲型期間」の形式別に説明する。

- ・ 文字列範囲型期間
文字列範囲型期間は1つの要素として扱い、データ形式は全角または半角の文字である。
- ・ 日付範囲型期間
前述した文字列型日付形式モジュールに従う。
- ・ 時刻範囲型期間
前述した文字列型時刻形式に従う。

(4) 様式設計者の利用例

範囲型期間を表す要素は、「期間」、「工事期間」等の名称を用いる。

(i) DTD

```

■文字列範囲型期間
・文字列範囲型期間モジュールのエンティティ（「e.範囲型期間-0」）を参照し、別名（「工事期間」）を宣言した例

<!ELEMENT 工事期間 (%e.範囲型期間-0;)>

・文字列範囲型期間モジュールのELEMENT（「範囲型期間」）を参照し、ELEMENT（「工事」）を宣言した例

<!ELEMENT 工事 (事業者,範囲型期間)>

■日付範囲型期間
・日付範囲型期間モジュールのエンティティ（「e.範囲型期間-1」）を参照し、別名（「工事期間」）を宣言した例

<!ELEMENT 工事期間 (%e.範囲型期間-1;)>

・日付範囲型期間モジュールのELEMENT（「範囲型期間-1」）を参照し、ELEMENT（「工事」）を宣言した例

<!ELEMENT 工事 (事業者,範囲型期間-1)>

■時刻範囲型期間
・時刻範囲型期間モジュールのエンティティ（「e.範囲型期間-2」）を参照し、別名（「工事期間」）を宣言した例

```

言した例

```
<!ELEMENT 工事期間 (%e.範囲型期間-2);>
```

・時刻範囲型期間モジュールのエレメント（「範囲型期間-2」）を参照し、エレメント（「工事」）を宣言した例

```
<!ELEMENT 工事 (事業者,範囲型期間-2)>
```

(ii) XML 文書

■文字列範囲型期間

・「工事期間」のマークアップ例

```
<工事期間>自平成13年10月12日～至平成13年10月23日</工事期間>
```

・「工事」のマークアップ例

```
<工事>
```

```
  <事業者>A株式会社</事業者>
```

```
  <範囲型期間>自平成13年10月12日～至平成13年10月23日</範囲型期間>
```

```
</工事>
```

■日付範囲型期間

・「工事期間」のマークアップ例

```
<工事期間>
```

```
  <開始日付>平成13年10月12日</開始日付>
```

```
  <終了日付>平成13年10月23日</終了日付>
```

```
</工事期間>
```

・「工事」のマークアップ例

```
<工事>
```

```
  <事業者>A株式会社</事業者>
```

```
  <範囲型期間-1>
```

```
    <開始日付>平成13年10月12日</開始日付>
```

```
    <終了日付>平成13年10月23日</終了日付>
```

```
  </範囲型期間-1>
```

```
</工事>
```

■時刻範囲型期間

・「工事期間」のマークアップ例

```
<工事期間>
```

```
  <開始時刻>11時30分</開始時刻>
```

```
  <終了時刻>16時00分</終了時刻>
```

```
</工事期間>
```

・「工事」のマークアップ例

```
<工事>
```

```
  <事業者>A株式会社</事業者>
```

```
  <範囲型期間-2>
```

```
    <開始時刻>11時30分</開始時刻>
```

```
    <終了時刻>16時00分</終了時刻>
```

```
  </範囲型期間-2>
```

```
</工事>
```


B 区間に関する記載項目

区間は様式上、「区間」、「乗車区間」などの区間に関する記載項目である。以下では区間としての利用が想定される項目の例について述べる。

(1) モジュール名

モジュール名は区間モジュールである。

区間モジュールは、

- ・ エンティティ :
 - e. 区間-0 (1つの要素として扱う場合)
 - e. 区間-1 (複数の要素に分解して扱う場合)
 - e. 始点-0 (複数の要素に分解して扱う場合の「始点」)
 - e. 終点-0 (複数の要素に分解して扱う場合の「終点」)
- ・ エレメント :
 - 区間 (1つの要素として扱う場合)
 - 区間-1 (複数の要素に分解して扱う場合)
 - 始点 (複数の要素に分解して扱う場合の「始点」)
 - 終点 (複数の要素に分解して扱う場合の「終点」)

から構成される。

(2) モジュールの申請書共通ボキャブラリにおける定義

- ・ 基本パターン (1つの要素として扱う場合)


```
<!ENTITY % e.区間-0 "#PCDATA">
<!ELEMENT 区間 (%e.区間-0;)>
```
- ・ その他のパターン (複数の要素に分解して扱う場合)


```
<!ENTITY % e.区間-1 "始点,終点">
<!ENTITY % e.始点-0 "#PCDATA">
<!ENTITY % e.終点-0 "#PCDATA">
<!ELEMENT 始点 (%e.始点-0;)>
<!ELEMENT 終点 (%e.終点-0;)>
<!ELEMENT 区間-1 (%e.区間-1;)>
```

(3) 説明

区間を1つの要素として扱うか下位構造を持つ要素として扱うかは業務に依存する。

区間の各要素のデータ形式は基本パターン、その他のパターンともに全角または半角の文字である。

(4) 様式設計者の利用例

区間を表す要素は、「区間」、「乗車区間」等の名称を用いる。

(i) DTD

- ・ 区間モジュールのエンティティ（「e.区間-0」）を参照し、別名（「乗車区間」）を宣言した例
`<!ELEMENT 乗車区間 (%e.区間-0;)>`
- ・ 区間モジュールのエレメント（「区間」）を参照し、エレメント（「乗車」）を宣言した例
`<!ELEMENT 乗車 (路線名,区間)>`

(ii) XML 文書

- ・ 「乗車区間」のマークアップ例
`<乗車区間>横浜駅から新橋駅まで</乗車区間>`
- ・ 「様式名」のマークアップ例
`<乗車>
 <路線名>東海道線</路線名>
 <区間>横浜駅から新橋駅まで</区間>
</乗車>`

4.2.2.5. その他の記載内容要素群

その他の申請書共通ボキャブラリについて示す。

A 振り仮名に関する記載項目

氏名、住所等に付与される振り仮名を記載する項目である。以下では振り仮名としての利用が想定される項目の例について述べる。

振り仮名はカタカナで記述する場合と平仮名で記述する場合があり、本ガイドラインではそれぞれをモジュールとして提供する。

(1) モジュール名

モジュール名はフリガナモジュールおよびふりがなモジュールである。

フリガナモジュールおよびふりがなモジュールは、

- ・ エンティティ : e.フリガナ-0
e.ふりがな-0
- ・ エレメント : フリガナ
ふりがな

から構成される。

(2) モジュールの申請書共通ボキャブラリにおける定義

```
<!ENTITY % e.フリガナ-0 "#PCDATA">
<!ENTITY % e.ふりがな-0 "#PCDATA">
<!ELEMENT フリガナ (%e.フリガナ-0;)>
<!ELEMENT ふりがな (%e.ふりがな-0;)>
```

(3) 説明

フリガナモジュール、ふりがなモジュールはそれぞれ 1 つの要素として扱う。申請者の利便性を考慮して、フリガナモジュール、ふりがなモジュールともにデータ形式は、カタカナの全角文字と平仮名の全角文字のどちらにも対応可能にする。

(4) 様式設計者の利用例

振り仮名を表す要素は、「フリガナ」、「ふりがな」等の名称を用いる。

(i) DTD

■フリガナモジュール

・フリガナモジュールのエンティティ（「e.フリガナ-0」）を参照し、別名（「氏名フリガナ」）を宣言した例

```
<!ELEMENT 氏名フリガナ (%e.フリガナ-0;)>
```

・フリガナモジュールのエレメント（「フリガナ」）を参照し、エレメント（「代表者氏名」）を宣言した例

```
<!ELEMENT 代表者氏名 (氏名,フリガナ)>
```

■ふりがなモジュール

・ふりがなモジュールのエンティティ（「e.ふりがな-0」）を参照し、別名（「氏名ふりがな」）を宣言した例

```
<!ELEMENT 氏名ふりがな (%e.ふりがな-0;)>
```

・ふりがなモジュールのエレメント（「ふりがな」）を参照し、エレメント（「代表者氏名」）を宣言した例

```
<!ELEMENT 代表者氏名 (氏名,ふりがな)>
```

(ii) XML 文書

■フリガナモジュール

・「氏名フリガナ」のマークアップ例

```
<氏名フリガナ>ヤマダタロウ</氏名フリガナ>
```

・「代表者氏名」のマークアップ例

```
<代表者氏名>
  <氏名>山田太郎</氏名>
  <フリガナ>ヤマダタロウ</フリガナ>
</代表者氏名>
```

■ふりがなモジュール

・「氏名ふりがな」のマークアップ例

```
<氏名ふりがな>やまだたろう</氏名ふりがな>
```

・「代表者氏名」のマークアップ例

```
<代表者氏名>
  <氏名>山田太郎</氏名>
  <ふりがな>やまだたろう</ふりがな>
</代表者氏名>
```

B 場所に関する記載項目

場所は「仕入れ先」、「所在地」、「原産地」、「出荷先」等、場所を示す項目である。前述した宛先としての住所とは異なり、物理的な特定の位置を示す項目である。以下では場所としての利用が想定される項目の例について述べる。

(1) モジュール名

モジュール名は場所モジュールである。

場所モジュールは、

- ・ エンティティ : e. 場所-0
- ・ エレメント : 場所

から構成される。

(2) モジュールの申請書共通ボキャブラリにおける定義

```
<!ENTITY % e.場所-0 "#PCDATA">
<!ELEMENT 場所 (%e.場所-0;)>
```

(3) 説明

場所は1つの要素として扱い、データ形式は全角または半角の文字である。

(4) 様式設計者の利用例

場所を表す要素は「場所」、「納入場所」等の名称を用いる。

(i) DTD

- ・ 場所モジュールのエンティティ（「e. 場所-0」）を参照し、別名（「納入場所」）を宣言した例

```
<!ELEMENT 納入場所 (%e.場所-0;)>
```
- ・ 場所モジュールのエレメント（「場所」）を参照し、エレメント（「納入」）を宣言した例

```
<!ELEMENT 納入 (業者名, 場所)>
```

(ii) XML 文書

- ・ 「納入場所」のマークアップ例

```
<納入場所>××地区</納入場所>
```
- ・ 「納入」のマークアップ例

```
<納入>
  <業者名>〇〇株式会社</業者名>
  <場所>××地区</場所>
</納入>
```

C 理由に関する記載項目

理由は様式上、「変更の理由」、「〇〇に関する変更事由」等の理由を示す記載項目である。以下では理由としての利用が想定される項目の例について述べる。

(1) モジュール名

モジュール名は理由モジュールである。

理由モジュールは、

- ・ エンティティ： e.理由-0
- ・ エレメント： 理由

から構成される。

(2) モジュールの申請書共通ボキャブラリにおける定義

```
<!ENTITY % e.理由-0 "#PCDATA">
<!ELEMENT 理由 (%e.理由-0;)>
```

(3) 説明

理由は1つの要素として扱い、データ形式は全角または半角の文字である。

(4) 様式設計者の利用例

理由を表す要素は、「理由」、「変更の理由」等の名称を用いる。

(i) DTD

- ・ 理由モジュールのエンティティ（「e.理由-0」）を参照し、別名（「変更の理由」）を宣言した例

```
<!ELEMENT 変更の理由 (%e.理由-0;)>
```
- ・ 理由モジュールのエレメント（「理由」）を参照し、エレメント（「変更」）を宣言した例

```
<!ELEMENT 変更 (日付,理由)>
```

(ii) XML 文書

- ・ 「変更の理由」のマークアップ例

```
<変更の理由>住所の移転による</変更の理由>
```
- ・ 「変更」のマークアップ例

```
<変更>
  <日付>13年10月24日</日付>
  <理由>住所の移転による</理由>
</変更>
```

D 符号に関する記載項目

符号は様式上、「輸入者符号」、「貿易形態別符号」等の符号を示す記載項目である。以下では符号としての利用が想定される項目の例について述べる。

(1) モジュール名

モジュール名は符号モジュールである。

符号モジュールは、

- ・ エンティティ： e.符号-0
- ・ エレメント： 符号

から構成される。

(2) モジュールの申請書共通ボキャブラリにおける定義

```
<!ENTITY % e.符号-0 "#PCDATA">
<!ELEMENT 符号 (%e.符号-0;)>
```

(3) 説明

符号は1つの要素として扱い、データ形式は半角英数字である。

(4) 様式設計者の利用例

符号を表す要素は、「輸入者符号」、「貿易形態別符号」等の名称を用いる。

(i) DTD

- ・ 符号モジュールのエンティティ（「e.符号-0」）を参照し、別名（「輸入者符号」）を宣言した例

```
<!ELEMENT 輸入者符号 (%e.符号-0;)>
```
- ・ 符号モジュールのエレメント（「符号」）を参照し、エレメント（「輸入者」）を宣言した例

```
<!ELEMENT 輸入者 (符号,氏名)>
```

(ii) XML 文書

- ・ 「輸入者符号」のマークアップ例

```
<輸入者符号>A12345</輸入者符号>
```
- ・ 「輸入者」のマークアップ例

```
<輸入者>
  <符号>A12345</符号>
  <氏名>山田太郎</氏名>
</輸入者>
```

4.2.3. 記載支援要素群

ここでは、「記載支援要素群」の様式構成要素である「記載説明」について述べる。

4.2.3.1. 記載説明に関する記載項目

記載説明は様式上、「記載説明」等で表現されている記載項目である。以下では記載説明としての利用が想定される項目の例について述べる。

(1) モジュール名

モジュール名は記載説明モジュールである。

記載説明モジュールは、

- ・ エンティティ： e. 記載説明-0
- ・ エLEMENT： 記載説明

から構成される。

(2) モジュールの申請書共通ボキャブラリにおける定義

```
<!ENTITY % e. 記載説明-0 "#PCDATA">
<!ELEMENT 記載説明 (%e. 記載説明-0;)>
```

(3) 説明

記載説明は1つの要素として扱い、データ形式は全角または半角の文字である。

(4) 様式設計者の利用例

記載説明を表す要素は、「記載説明」、「記載要領」等の名称を用いる。

(i) DTD

- ・ 記載説明モジュールのエンティティ（「e. 記載説明-0」）を参照し、別名（「記載要領」）を宣言した例

```
<!ELEMENT 記載要領 (%e. 記載説明-0;)>
```
- ・ 記載説明モジュールのエLEMENT（「記載説明」）を参照し、ELEMENT（「備考欄」）を宣言した例

```
<!ELEMENT 備考欄 (記載説明, 注釈)>
```

(ii) XML 文書

- ・ 「記載要領」のマークアップ例

```
<記載要領>次年度以降において燃料等(電気)の使用量が令第2条第1項(第2項)に該当しないことが明らかである場合は、その旨およびその理由を備考の欄に記入すること。</記載要領>
```
- ・ 「備考欄」のマークアップ例

```
<備考欄>
  <記載説明>次年度以降において燃料等(電気)の使用量が令第2条第1項(第2項)に該当しないことが明らかである場合は、その旨およびその理由を備考の欄に記入すること。</記載説明>
  <注釈>ただし、〇〇の場合は除く</注釈>
</備考欄>
```


5. 申請書共通ボキャブラリの XML Schema への適用

5.1. XML Schema の導入方針

2001年5月にW3C勧告となったXMLのためのスキーマ言語「XML Schema」は、DTDと比べて要素のデータ型や値の範囲等を細かく定義することができるため、申請様式の電子化にあたっては、記載項目に対するより厳密な定義やチェックを行うことが可能となる。

一方で、「XML Schema」以外のスキーマ言語の標準化においては、RELAXやTREXなどの新しいスキーマ言語の発表や、標準化団体OASISによる「RELAX NG」の標準化の動きなど、その動向は目まぐるしい状況にあり、実際に世の中に標準として利用されるスキーマ言語の方向性は、未だ明確になっていないという状況にもある。

そこで本ガイドラインにおいては、これらの新しいスキーマ言語の動向も踏まえながら、XML Schemaが有する基本的な機能を用いたXML Schema版申請書共通ボキャブラリの検討を行う。すなわち、XML Schemaが有する特有な仕様の部分については現段階では用いずに、他のスキーマ言語においても共通すると考えられる仕様だけを用いた内容としている。

なお、今回、以下の方針に基づいてXML Schema版申請書共通ボキャブラリを記述している。

- ① ガイドラインとしての一貫性を考慮し、4章に示した構造や型を継承した。
- ② DTDにおいては明確に宣言することのできなかつた型を、XML Schemaにおいては明示的に宣言した。数値型、文字型等の型宣言を行っている。
- ③ 要素として存在し得る値が決まっているもの（日付の月や日など）には、値を検証する制限を行うものとする。例えば、月は「1～12までの整数値とする」といった制限を行っている。

5.2. XML Schema 版申請書共通ボキャブラリ

以下の説明では、本ガイドラインの 4 章と同様、申請書共通ボキャブラリを構成するモジュールごとに「モジュール名」、「モジュールの申請書共通ボキャブラリにおける定義」、「説明」、「様式設計者の利用例」を提示する。また、「モジュールの申請書共通ボキャブラリにおける定義」はライブラリとして提供されるもの、「様式設計者の利用例」は様式設計者が定義するものとして例示している。

なお、4 章では構造の定義をエンティティと呼んでいたものを、本章においては XML Schema における「データ型」の宣言と呼ぶこととする。

XML Schema 版申請書共通ボキャブラリの定義は、

<http://www.nmda.or.jp/sinsei/xml/vocabulary/vocabulary.xsd>

を参照されたい。

5.2.1. 様式識別要素群

5.2.1.1. 文書名に関する記載項目

文書名は様式上、「文書名」等で表現されている記載項目である。

(1) モジュール名

モジュール名は文書名モジュールである。

文書名モジュールは、

- ・ データ型 : e. 文書名-0
- ・ エレメント : 文書名

から構成される。

(2) モジュールの申請書共通ボキャブラリにおける定義

```
<xsd:simpleType name='e.文書名-0'>
  <xsd:restriction base="xsd:string">
    </xsd:restriction>
  </xsd:simpleType>
<xsd:element name='文書名' type='e.文書名-0'/>
```

(3) 説明

文書名は 1 つの要素として扱う。文書名のデータ形式は通常、全角の文字であるが、申請によっては英語の文書名もあるため全角あるいは半角の英数字、どちらのデータにも対応可能にする必要がある。

(4) 様式設計者の利用例

文書名を表す要素は、「文書名」、「様式名」等の名称を用いる。

(i) XML Schema

・ 文書名モジュールのデータ型（「e.文書名-0」）を参照し、別名（「様式名」）を宣言した例

```
<xsd:element name='様式名' type='e.文書名-0' />
```

・ 文書名モジュールのエレメント（「文書名」）を参照し、エレメント（「様式識別」）を宣言した例

```
<xsd:complexType name='様式識別'>
  <xsd:sequence>
    <xsd:element ref="文書名"/>
    <xsd:element ref="文書番号"/>
  </xsd:sequence>
</xsd:complexType>
<xsd:element name='様式識別' type='様式識別' />
```

(ii) XML 文書

・ 「様式名」のマークアップ例

```
<様式名>燃料等（電気）使用状況届出書</様式名>
```

・ 「様式識別」のマークアップ例

```
<様式識別>
  <文書名>燃料等（電気）使用状況届出書</文書名>
  <文書番号>様式第 10 号</文書番号>
</様式識別>
```

5.2.1.2. 文書番号（様式番号）に関する記載項目

文書番号（様式番号）は様式上、「文書番号（様式番号）」等で表現されている記載項目である。

(1) モジュール名

モジュール名は文書番号モジュールである。

文書番号モジュールは、

- ・ データ型 : e. 文書番号-0
- ・ エレメント : 文書番号

から構成される。

(2) モジュールの申請書共通ボキャブラリにおける定義

```
<xsd:simpleType name='e.文書番号-0'>
  <xsd:restriction base="xsd:string">
    </xsd:restriction>
  </xsd:simpleType>
<xsd:element name='文書番号' type='e.文書番号-0' />
```

(3) 説明

文書番号は1つの要素として扱い、データ形式は全角または半角の文字である。

(4) 様式設計者の利用例

文書番号を表す要素は、「文書番号」、「様式番号」等の名称を用いる。

(i) XML Schema

```
・ 文書番号モジュールのデータ型（「e. 文書番号-0」）を参照し、別名（「様式番号」）を宣言した例
<xsd:element name='様式番号' type='e.文書番号-0' />

・ 文書番号モジュールのエレメント（「文書番号」）を参照し、エレメント（「様式識別」）を宣言した例
<xsd:complexType name='様式識別'>
  <xsd:sequence>
    <xsd:element ref="文書名"/>
    <xsd:element ref="文書番号"/>
  </xsd:sequence>
</xsd:complexType>
<xsd:element name='様式識別' type='様式識別' />
```

(ii) XML 文書

・「様式番号」のマークアップ例
<様式番号>様式第 1 号</様式番号>

・「様式識別」のマークアップ例
<様式識別>
 <文書名>〇〇申請書</文書名>
 <文書番号>様式第 1 号</文書番号>
</様式識別>

5.2.1.3. 様式を規定している法令に関する記載項目

様式を規定している法令は様式上、「様式を規定している法令」等で表現されている記載項目である。

(1) モジュール名

モジュール名は様式規定法令モジュールである。

様式規定法令モジュールは、

- ・ データ型 : e. 様式規定法令-0
- ・ エレメント : 様式規定法令

から構成される。

(2) モジュールの申請書共通ボキャブラリにおける定義

```
<xsd:simpleType name='e.様式規定法令-0'>
  <xsd:restriction base="xsd:string">
    </xsd:restriction>
  </xsd:simpleType>
<xsd:element name='様式規定法令' type='e.様式規定法令-0'/>
```

(3) 説明

様式規定法令は1つの要素として扱い、データ形式は全角または半角の文字である。

(4) 様式設計者の利用例

様式規定法令を表す要素は、「様式規定法令」、「根拠法令」等の名称を用いる。

(i) XML Schema

- ・ 様式規定法令モジュールのデータ型（「e. 様式規定法令-0」）を参照し、別名（「根拠法令」）を宣言した例

```
<xsd:element name='根拠法令' type='e.様式規定法令-0'/>
```

- ・ 様式規定法令モジュールのエレメント（「様式規定法令」）を参照し、エレメント（「様式識別」）を宣言した例

```
<xsd:complexType name='様式識別'>
  <xsd:sequence>
    <xsd:element ref="文書名"/>
    <xsd:element ref="様式規定法令"/>
  </xsd:sequence>
</xsd:complexType>
<xsd:element name='様式識別' type='様式識別'/>
```

(ii) XML 文書

- 「根拠法令」のマークアップ例
<根拠法令>第 4 条関係</根拠法令>
- 「様式識別」のマークアップ例
<様式識別>
 <文書名>〇〇申請書</文書名>
 <様式規定法令>第 4 条関係</様式規定法令>
</様式識別>

5.2.1.4. あて先に関する記載項目

あて先は様式上、「あて先」等で表現されている記載項目である。

(1) モジュール名

モジュール名はあて先モジュールおよび経由モジュールである。

あて先モジュールは、

- ・ データ型 : e.あて先-0
- ・ エレメント : あて先

から構成される。

経由モジュールは、

- ・ データ型 : e.経由-0
- ・ エレメント : 経由

から構成される。

(2) モジュールの申請書共通ボキャブラリにおける定義

```
<xsd:simpleType name='e.あて先-0'>
  <xsd:restriction base="xsd:string">
    </xsd:restriction>
  </xsd:simpleType>
<xsd:element name='あて先' type='e.あて先-0' />
<xsd:simpleType name='e.経由-0'>
  <xsd:restriction base="xsd:string">
    </xsd:restriction>
  </xsd:simpleType>
<xsd:element name='経由' type='e.経由-0' />
```

(3) 説明

あて先および経由はそれぞれ1つの要素として扱い、データ形式は全角または半角の文字である。

(4) 様式設計者の利用例

あて先を表す要素は、「あて先」、「提出先」等の名称を用いる。また、経由を表す要素は「経由」等の名称を用いる。

(i) XML Schema

```

・あて先モジュールのデータ型（「e.あて先-0」）を参照し、別名（「提出先」）を宣言した例
<xsd:element name="提出先" type="e.あて先-0"/>

・あて先モジュールのエレメント（「あて先」）を参照し、エレメント（「様式識別」）を宣言した例
<xsd:complexType name='様式識別'>
  <xsd:sequence>
    <xsd:element ref="文書名"/>
    <xsd:element ref="あて先"/>
    <xsd:element ref="経由"/>
  </xsd:sequence>
</xsd:complexType>
<xsd:element name="様式識別" type="様式識別"/>

```

(ii) XML 文書

```

・「提出先」のマークアップ例
<提出先>国土交通大臣殿</提出先>

・「様式識別」のマークアップ例
<様式識別>
  <文書名>〇〇に関する申請</文書名>
  <あて先>国土交通大臣殿</あて先>
  <経由>東京都知事殿</経由>
</様式識別>

```

5.2.1.5. 申請届出根拠に関する記載項目

申請届出根拠は様式上、「申請届出根拠」等で表現されている記載項目である。

(1) モジュール名

モジュール名は申請届出根拠モジュールである。

申請届出根拠モジュールは、

- ・ データ型 : e. 申請届出根拠-0
- ・ エレメント : 申請届出根拠

から構成される。

(2) モジュールの申請書共通ボキャブラリにおける定義

```
<xsd:simpleType name='e.申請届出根拠-0'>
  <xsd:restriction base="xsd:string">
    </xsd:restriction>
  </xsd:simpleType>
<xsd:element name='申請届出根拠' type='e.申請届出根拠-0'/>
```

(3) 説明

申請届出根拠は1つの要素として扱い、データ形式は全角または半角の文字である。

(4) 様式設計者の利用例

申請届出根拠を表す要素は、「申請届出根拠」、「届出根拠」等の名称を用いる。

(i) XML Schema

- ・ 申請届出根拠モジュールのデータ型（「e. 申請届出根拠-0」）を参照し、別名（「届出根拠」）を宣言した例

```
<xsd:element name='届出根拠' type='e.申請届出根拠-0'/>
```

- ・ 申請届出根拠モジュールのエレメント（「申請届出根拠」）を参照し、エレメント（「様式識別」）を宣言した例

```
<xsd:complexType name='様式識別'>
  <xsd:sequence>
    <xsd:element ref="文書名"/>
    <xsd:element ref="申請届出根拠"/>
  </xsd:sequence>
</xsd:complexType>
<xsd:element name='様式識別' type='様式識別'/>
```

(ii) XML 文書

- ・「届出根拠」のマークアップ例

<届出根拠>エネルギー使用の合理化に関する法律第6条第2項の規定に基づき次のとおり届け出ます。</届出根拠>

- ・「様式識別」のマークアップ例

<様式識別>

<文書名>〇〇に関する申請</文書名>

<申請届出根拠>エネルギー使用の合理化に関する法律第6条第2項の規定に基づき次のとおり届け出ます。</申請届出根拠>

</様式識別>

5.2.1.6. 法令名に関する記載項目

法令名は電子化の際に当該手続を特定するための重要な情報である。以下では法令名としての利用が想定される項目の例について述べる。

(1) モジュール名

モジュール名は法令名モジュールである。

法令名モジュールは、

- ・ データ型 : e.法令名-0
- ・ エレメント : 法令名

から構成される。

(2) モジュールの申請書共通ボキャブラリにおける定義

```
<xsd:simpleType name='e.法令名-0'>
  <xsd:restriction base="xsd:string">
    </xsd:restriction>
  </xsd:simpleType>
<xsd:element name='法令名' type='e.法令名-0' />
```

(3) 説明

法令名は1つの要素として扱い、データ形式は全角または半角の文字である。

(4) 様式設計者の利用例

法令名を表す要素は、「法令名」、「法令の名称」等の名称を用いる。

(i) XML Schema

```
・ 法令名モジュールのデータ型（「e.法令名-0」）を参照し、別名（「法令の名称」）を宣言した例
<xsd:element name='法令の名称' type='e.法令名-0' />

・ 法令名モジュールのエレメント（「法令名」）を参照し、エレメント（「様式識別」）を宣言した例
<xsd:complexType name='様式識別'>
  <xsd:sequence>
    <xsd:element ref="法令名" />
    <xsd:element ref="手続名" />
    <xsd:element ref="文書名" />
  </xsd:sequence>
</xsd:complexType>
<xsd:element name='様式識別' type='様式識別' />
```

(ii) XML 文書

- ・「法令の名称」のマークアップ例

<法令の名称>輸入の促進および対内投資事業の円滑化に関する臨時措置法</法令の名称>

- ・「様式識別」のマークアップ例

<様式識別>

<法令名>輸入の促進および対内投資事業の円滑化に関する臨時措置法</法令名>

<手続名>特定製品輸入事業者認定申請</手続名>

<文書名>〇〇に関する申請</文書名>

</様式識別>

5.2.1.7. 手続名に関する記載項目

手続名は電子化の際に当該手続を特定するための重要な情報である。以下では手続名としての利用が想定される項目の例について述べる。

(1) モジュール名

モジュール名は手続名モジュールである。

手続名モジュールは、

- ・ データ型 : e.手続名-0
- ・ エレメント : 手続名

から構成される。

(2) モジュールの申請書共通ボキャブラリにおける定義

```
<xsd:simpleType name='e.手続名-0'>
  <xsd:restriction base="xsd:string">
    </xsd:restriction>
  </xsd:simpleType>
<xsd:element name='手続名' type='e.手続名-0'/>
```

(3) 説明

手続名は1つの要素として扱い、データ形式は全角または半角の文字である。

(4) 様式設計者の利用例

手続名を表す要素は、「手続名」、「手続の名称」等の名称を用いる。

(i) XML Schema

```
・ 手続名モジュールのデータ型（「e.手続名-0」）を参照し、別名（「手続の名称」）を宣言した例
<xsd:element name='手続の名称' type='e.手続名-0'/>

・ 手続名モジュールのエレメント（「手続名」）を参照し、エレメント（「様式識別」）を宣言した例
<xsd:complexType name='様式識別'>
  <xsd:sequence>
    <xsd:element ref="法令名"/>
    <xsd:element ref="手続名"/>
    <xsd:element ref="文書名"/>
  </xsd:sequence>
</xsd:complexType>
<xsd:element name='様式識別' type='様式識別'/>
```

(ii) XML 文書

- ・「手続の名称」のマークアップ例

<手続の名称>特定製品輸入事業者認定申請</手続の名称>

- ・「様式識別」のマークアップ例

<様式識別>

<法令名>輸入の促進および対内投資事業の円滑化に関する臨時措置法</法令名>

<手続名>特定製品輸入事業者認定申請</手続名>

<文書名>〇〇に関する申請</文書名>

</様式識別>

5.2.2. 記載内容要素群

ここでは様式上、記載内容要素群で使用される記入項目について述べる。

5.2.2.1. 個人情報または組織情報に関する記載項目

個人情報に属する主な項目としては、氏名、住所、郵便番号、電話番号などがある。

組織情報に属する主な項目としては、名称、氏名（代表者氏名等）、住所、郵便番号、電話番号などがある。

A 氏名に関する記載項目

氏名は様式上、「代表者氏名」等で表現されている記載項目である。以下では氏名としての利用が想定される項目の例について述べる。

(1) モジュール名

モジュール名は氏名モジュールである。

氏名モジュールは、

- ・ データ型 :
 - e. 氏名-0 (1つの要素として扱う場合)
 - e. 氏名-1 (氏と名の要素に分ける場合)
 - e. 氏-0 (氏と名の要素に分ける場合の「氏」)
 - e. 名-0 (氏と名の要素に分ける場合の「名」)
- ・ エレメント :
 - 氏名 (1つの要素として扱う場合)
 - 氏名-1 (氏と名の要素に分ける場合)
 - 氏 (氏と名の要素に分ける場合の「氏」)
 - 名 (氏と名の要素に分ける場合の「名」)

から構成される。

(2) モジュールの申請書共通ボキャブラリにおける定義

```

・ 氏、名の定義と宣言
<xsd:simpleType name='e.氏-0'>
  <xsd:restriction base="xsd:string">
  </xsd:restriction>
</xsd:simpleType>
<xsd:simpleType name='e.名-0'>
  <xsd:restriction base="xsd:string">
  </xsd:restriction>
</xsd:simpleType>
<xsd:element name='氏' type='e.氏-0'/>
<xsd:element name='名' type='e.名-0'/>

・ 基本パターン (1つの要素として扱う場合)
<xsd:simpleType name='e.氏名-0'>
  <xsd:restriction base="xsd:string">
  </xsd:restriction>
</xsd:simpleType>
<xsd:element name='氏名' type='e.氏名-0'/>

・ その他のパターン (複数の要素に分解して扱う場合)
<xsd:complexType name='e.氏名-1'>
  <xsd:sequence>

```



```

    <xsd:element ref="氏"/>
    <xsd:element ref="名"/>
  </xsd:sequence>
</xsd:complexType>
<xsd:element name='氏名-1' type='e.氏名-1'/>

```

(3) 説明

氏名を1つの要素として扱うか、氏と名とに要素分割して扱うかは業務に依存する。

氏名の各要素のデータ形式は基本パターン、その他のパターンともに通常、全角の文字と考えられるが、申請によっては英語で併記するような場合もあり、全角あるいは半角の英数字、どちらにも対応可能にする必要がある。

(4) 様式設計者の利用例

氏名を表す要素は、「氏名」、「提出者」等の名称を用いる。

(i) XML Schema

- ・氏名モジュールのデータ型（「e.氏名-0」）を参照し、別名（「提出者」）を宣言した例

```
<xsd:element name='提出者' type='e.氏名-0'/>
```

- ・氏名モジュールのエレメント（「氏名」）を参照し、エレメント（「代表者氏名」）を宣言した例

```

<xsd:complexType name='代表者氏名'>
  <xsd:sequence>
    <xsd:element ref="氏名"/>
    <xsd:element ref="フリガナ"/>
  </xsd:sequence>
</xsd:complexType>
<xsd:element name='代表者氏名' type='代表者氏名'/>

```

(ii) XML 文書

- ・「提出者」のマークアップ例

```
<提出者>山田 太郎</提出者>
```

- ・「代表者氏名」のマークアップ例

```

<代表者氏名>
  <氏名>山田 太郎</氏名>
  <フリガナ>ヤマダ タロウ</フリガナ>
</代表者氏名>

```

B 法人団体名に関する記載項目

法人団体名は様式上、「法人または団体の名称」等で表現されている記載項目である。以下では法人団体名としての利用が想定される項目の例について述べる。

(1) モジュール名

モジュール名は法人団体名モジュールである。

法人団体名モジュールは、

- ・ データ型 : e. 法人団体名-0
- ・ エレメント : 法人団体名

から構成される。

(2) モジュールの申請書共通ボキャブラリにおける定義

```
<xsd:simpleType name='e.法人団体名-0'>
  <xsd:restriction base="xsd:string">
    </xsd:restriction>
  </xsd:simpleType>
<xsd:element name='法人団体名' type='e.法人団体名-0' />
```

(3) 説明

法人団体名は 1 つの要素として扱う。法人団体名のデータ形式は全角の文字であるが、申請によっては英語で併記するような場合もあり、全角あるいは半角の英数字、どちらにも対応可能にする必要がある。

(4) 様式設計者の利用例

法人団体名を表す要素は、「法人団体名」、「名称」等の名称を用いる。

(i) XML Schema

```
・ 法人団体名モジュールのデータ型（「e. 法人団体名-0」）を参照し、別名（「名称」）を宣言した例
<xsd:element name='名称' type='e.法人団体名-0' />

・ 法人団体名モジュールのエレメント（「法人団体名」）を参照し、エレメント（「事業所名」）を宣言した例
<xsd:complexType name='事業所名'>
  <xsd:sequence>
    <xsd:element ref="法人団体名" />
    <xsd:element ref="フリガナ" />
  </xsd:sequence>
</xsd:complexType>
<xsd:element name='事業所名' type='事業所名' />
```

(ii) XML 文書

・「名称」のマークアップ例
<名称>〇〇株式会社</名称>

・「事業所名」のマークアップ例
<事業所名>
 <法人団体名>〇〇株式会社</法人団体名>
 <フリガナ>マルマルカブシキガイシャ</フリガナ>
</事業所名>

C 住所に関する記載項目

住所は様式上、「代表者の住所」等で表現されている記載項目である。以下では住所としての利用が想定される項目の例について述べる。

(1) モジュール名

モジュール名は住所モジュールおよび郵便番号モジュールである。

住所モジュールは、

- ・ データ型 :
 - e. 住所-0 (1つの要素として扱う場合)
 - e. 住所-1 (複数の要素に分解して扱う場合)
 - e. 都道府県-0 (複数の要素に分解して扱う場合の「都道府県」)
 - e. 市郡-0 (複数の要素に分解して扱う場合の「市郡」)
 - e. 区-0 (複数の要素に分解して扱う場合の「区」)
 - e. 町名等-0 (複数の要素に分解して扱う場合の「町名等」)
- ・ エレメント :
 - 住所 (1つの要素として扱う場合)
 - 住所-1 (複数の要素に分解して扱う場合)
 - 都道府県 (複数の要素に分解して扱う場合の「都道府県」)
 - 市郡 (複数の要素に分解して扱う場合の「市郡」)
 - 区 (複数の要素に分解して扱う場合の「区」)
 - 町名等 (複数の要素に分解して扱う場合の「町名等」)

から構成される。

郵便番号モジュールは

- ・ データ型 :
 - e. 郵便番号-0
- ・ エレメント :
 - 郵便番号

から構成される。

(2) モジュールの申請書共通ボキャブラリにおける定義

```

・都道府県、市郡、区、町名等の定義と宣言
<xsd:simpleType name='e.都道府県-0'>
  <xsd:restriction base="xsd:string">
    <xsd:enumeration value="北海道"/>
    <xsd:enumeration value="青森県"/>
    (中略)
    <xsd:enumeration value="沖縄県"/>
  </xsd:restriction>
</xsd:simpleType>
<xsd:simpleType name='e.市郡-0'>
  <xsd:restriction base="xsd:string">
  </xsd:restriction>
</xsd:simpleType>
<xsd:simpleType name='e.区-0'>
  <xsd:restriction base="xsd:string">
  </xsd:restriction>
</xsd:simpleType>
<xsd:simpleType name='e.町名等-0'>
  <xsd:restriction base="xsd:string">
  </xsd:restriction>
</xsd:simpleType>
<xsd:element name='都道府県' type='e.都道府県-0'/>
<xsd:element name='市郡' type='e.市郡-0'/>
<xsd:element name='区' type='e.区-0'/>
<xsd:element name='町名等' type='e.町名等-0'/>

・基本パターン (1つの要素として扱う場合)
<xsd:simpleType name='e.住所-0'>
  <xsd:restriction base="xsd:string">
  </xsd:restriction>
</xsd:simpleType>
<xsd:element name='住所' type='e.住所-0'/>

・その他のパターン (複数の要素に分解して扱う場合)
<xsd:complexType name='e.住所-1'>
  <xsd:sequence>
    <xsd:element ref="都道府県"/>
    <xsd:element ref="市郡"/>
    <xsd:element ref="区"/>
    <xsd:element ref="町名等"/>
  </xsd:sequence>
</xsd:complexType>
<xsd:element name='住所-1' type='e.住所-1'/>

```

- ・郵便モジュール（住所モジュールの基本パターンおよびその他のパターンで共用）

```
<xsd:simpleType name='e.郵便番号-0'>
  <xsd:restriction base="xsd:string">
    <xsd:pattern value="¥d{3}-¥d{4}"/>
  </xsd:restriction>
</xsd:simpleType>
<xsd:element name='郵便番号' type='e.郵便番号-0' />
```

(3) 説明

住所の各要素のデータ形式は基本パターン、その他のパターンともに通常、全角の文字と考えられるが、申請によっては英語で併記するような場合もあり、全角あるいは半角の英数字、どちらにも対応可能にする必要がある。都道府県のデータ形式は“北海道”、“青森県”、その他都道府県として存在する値のみである。

郵便番号は1つの要素として扱い、データ形式は半角数字3桁、“-”（ハイフン）、半角数字4桁である。

(4) 様式設計者の利用例

住所を表す要素は、「住所」、「申請者の住所」等の名称を用いる。

(i) XML Schema

- ・住所モジュールのデータ型（「e.住所-0」）を参照し、別名（「申請者の住所」）を宣言した例

```
<xsd:element name='申請者の住所' type='e.住所-0' />
```

- ・住所モジュールのエレメント（「住所」）を参照し、エレメント（「代表者の住所」）を宣言した例

```
<xsd:complexType name='代表者の住所'>
  <xsd:sequence>
    <xsd:element ref="郵便番号"/>
    <xsd:element ref="住所"/>
  </xsd:sequence>
</xsd:complexType>
<xsd:element name='代表者の住所' type='代表者の住所' />
```

(ii) XML 文書

- ・「申請者の住所」のマークアップ例

```
<申請者の住所>東京都港区三田一丁目4番28号</申請者の住所>
```

- ・「代表者の住所」のマークアップ例

```
<代表者の住所>
  <郵便番号>100-1111</郵便番号>
  <住所>東京都港区三田一丁目4番28号</住所>
</代表者の住所>
```

D 連絡先に関する記載項目

連絡先に関する事項として電話番号および FAX 番号がある。これらは様式上、「電話番号」、「FAX 番号」等で表現されている記載項目である。以下では電話番号および FAX 番号としての利用が想定される項目の例について述べる。

現在の様式中には存在していないが、「電子メールアドレス」という項目も今後普及が予想され、申請書共通ボキャブラリに含める必要がある。

(1) モジュール名

モジュール名は電話番号モジュール、FAX 番号モジュール、電子メールアドレスモジュールである。

電話番号モジュールは、

- ・ データ型 : e.電話番号-0
- ・ エレメント : 電話番号

から構成される。

FAX 番号モジュールは

- ・ データ型 : e.FAX 番号-0
- ・ エレメント : FAX 番号

から構成される。

電子メールアドレスモジュールは

- ・ データ型 : e.電子メールアドレス-0
- ・ エレメント : 電子メールアドレス

から構成される。

(2) モジュールの申請書共通ボキャブラリにおける定義

```
<xsd:simpleType name='e.電話番号-0'>
  <xsd:restriction base="xsd:string">
    <xsd:pattern value="[0-9#+¥¥-¥¥(¥¥)]*"/>
  </xsd:restriction>
</xsd:simpleType>

<xsd:simpleType name='e.FAX 番号-0'>
  <xsd:restriction base="xsd:string">
    <xsd:pattern value="[0-9#+¥¥-¥¥(¥¥)]*"/>
  </xsd:restriction>
</xsd:simpleType>

<xsd:simpleType name='e.電子メールアドレス-0'>
  <xsd:restriction base="xsd:string">
  </xsd:restriction>
</xsd:simpleType>

<xsd:element name='電話番号' type='e.電話番号-0'/>
<xsd:element name='FAX 番号' type='e.FAX 番号-0'/>
<xsd:element name='電子メールアドレス' type='e.電子メールアドレス-0'/>
```

(3) 説明

電話番号および FAX 番号はそれぞれ 1 つの要素として扱い、ともにデータ形式は 0 から 9 までの半角数値および電話番号、FAX 番号で使用される特殊文字である “#”、“*”、“+”、“-”、“(”、“)” から構成される文字である。

電子メールアドレスは 1 つの要素として扱い、データ形式は全角または半角の文字である。

(4) 様式設計者の利用例

連絡先を表す要素は、「電話番号」、「電話」等の名称を用いる。

(i) XML Schema

```

・電話番号、FAX 番号、電子メールアドレスモジュールのデータ型（「e.電話番号-0」、「e.FAX 番号-0」、「e.電子メールアドレス-0」）を参照し、別名（「電話」、「FAX」、「メール」）を宣言した例
<xsd:element name='電話' type='e.電話番号-0' />
<xsd:element name='FAX ' type='e.FAX 番号-0' />
<xsd:element name='メール' type='e.電子メールアドレス-0' />

・電話番号、FAX 番号、電子メールアドレスモジュールのエレメント（「電話番号」、「FAX 番号」、「電子メールアドレス」）を参照し、エレメント（「連絡先」）を宣言した例
<xsd:complexType name='連絡先'>
  <xsd:sequence>
    <xsd:element ref="電話番号"/>
    <xsd:element ref="FAX 番号"/>
    <xsd:element ref="電子メールアドレス"/>
  </xsd:sequence>
</xsd:complexType>
<xsd:element name='連絡先' type='連絡先' />

```

(ii) XML 文書

```

・「電話」、「FAX」、「メール」のマークアップ例
<電話>03-1234-5678</電話>
<FAX>03-8765-4321</FAX>
<メール>aaaa@bbb.co.jp</メール>

・「連絡先」のマークアップ例
<連絡先>
  <電話番号>03-1234-5678</電話番号>
  <FAX 番号>03-8765-4321</FAX 番号>
  <電子メールアドレス>aaaa@bbb.co.jp</電子メールアドレス>
</連絡先>

```


E その他の個人情報または組織情報に関する記載項目

その他の個人情報または組織情報として、生年月日、年齢、性別、役職名、職業、国籍、本籍、続柄、従業員数、株式数、業種がある。

生年月日は日付モジュールを、本籍は住所モジュールをそれぞれ様式設計者側で適用する。年齢、従業員数、株式数については「5.2.2.3 数量に関する記載項目」を参照されたい。その他の項目は以下の通りである。

- 生年月日
日付モジュールを適用する。
- 性別
“男”、“女”のどちらかの値のみ記入が可能である。
- 役職名、職業、国籍、本籍、続柄、業種
全角または半角の文字を記入する。

5.2.2.2. 時間に関する記載項目

A 日付に関する記載項目

日付は様式上、「到達日」等の日付で表現されている記載項目である。以下では日付としての利用が想定される項目の例について述べる。

日付の表現として以下の4つが考えられる。

- 文字列型日付形式
1つの文字列として扱う形式である。
- 年号型日付形式
和暦もしくは西暦を表現する年号（西暦を表現する場合には年号に「西暦」が代入される）、年、月、日から構成される形式である。
この形式は様式上、和暦と西暦いずれで記入する場合も存在することから、どちらにも対応可能なものとして用意したものである。
- 西暦型日付形式
西暦で年、月、日から構成される形式である。
- ISO8601型日付形式
ISOで規定している日付の形式である。

(1) モジュール名

モジュール名は日付モジュールである。

日付モジュールは、

- データ型 :
 - e. 日付-0（文字列型日付形式）
 - e. 日付-1（年号型日付形式）
 - e. 日付-2（西暦型日付形式）
 - e. 日付-3（ISO8601型日付形式）
 - e. 年-0（年号型日付形式における「年」）
 - e. 年-1（年号型日付形式における和暦の「年」）
 - e. 年-2（年号型日付形式もしくは西暦型日付形式における西暦の「年」）
 - e. 月-0（年号型日付形式もしくは西暦型日付形式における「月」）
 - e. 日-0（年号型日付形式もしくは西暦型日付形式における「日」）
- エレメント :
 - 日付（文字列型日付形式）
 - 日付-1（年号型日付形式）
 - 日付-2（西暦型日付形式）
 - 日付-3（ISO8601型日付形式）
 - 年（年号型日付形式における「年」）
 - 年-1（年号型日付形式における和暦の「年」）
 - 年-2（年号型日付形式あるいは西暦型日付形式における西暦の「年」）
 - 月（年号型日付形式もしくは西暦型日付形式における「月」）
 - 日（年号型日付形式もしくは西暦型日付形式における「日」）

から構成される。

(2) モジュールの申請書共通ボキャブラリにおける定義

```

・年号、年、月、日の定義と宣言
<xsd:simpleType name='e.年号-0'>
  <xsd:restriction base="xsd:string">
    <xsd:enumeration value="西暦"/>
    <xsd:enumeration value="明治"/>
    <xsd:enumeration value="大正"/>
    <xsd:enumeration value="昭和"/>
    <xsd:enumeration value="平成"/>
  </xsd:restriction>
</xsd:simpleType>
<xsd:simpleType name='e.年-0'>
<xsd:restriction base="xsd:integer">
  <xsd:minInclusive value="1"/>
  <xsd:maxInclusive value="9999"/>
</xsd:restriction>
</xsd:simpleType>
<xsd:simpleType name='e.年-1'>
<xsd:restriction base="xsd:integer">
  <xsd:minInclusive value="1"/>
  <xsd:maxInclusive value="999"/>
</xsd:restriction>
</xsd:simpleType>
<xsd:simpleType name='e.年-2'>
<xsd:restriction base="xsd:integer">
  <xsd:minInclusive value="1"/>
  <xsd:maxInclusive value="9999"/>
</xsd:restriction>
</xsd:simpleType>
<xsd:simpleType name='e.月-0'>
<xsd:restriction base="xsd:integer">
  <xsd:minInclusive value="1"/>
  <xsd:maxInclusive value="12"/>
</xsd:restriction>
</xsd:simpleType>
<xsd:simpleType name='e.日-0'>
<xsd:restriction base="xsd:integer">
  <xsd:minInclusive value="1"/>
  <xsd:maxInclusive value="31"/>
</xsd:restriction>
</xsd:simpleType>
<xsd:element name='年号' type='e.年号-0'/>
<xsd:element name='年' type='e.年-0'/>
<xsd:element name='年-1' type='e.年-1'/>
<xsd:element name='年-2' type='e.年-2'/>
<xsd:element name='月' type='e.月-0'/>
<xsd:element name='日' type='e.日-0'/>

・文字列型日付形式
<xsd:simpleType name='e.日付-0'>

```

```
<xsd:restriction base="xsd:string">
  </xsd:restriction>
</xsd:simpleType>
<xsd:element name='日付' type='e.日付-0' />

• 年号型日付形式
<xsd:complexType name='e.日付-1'>
  <xsd:sequence>
    <xsd:element ref="年号"/>
    <xsd:element ref="年"/>
    <xsd:element ref="月"/>
    <xsd:element ref="日"/>
  </xsd:sequence>
</xsd:complexType>
<xsd:element name='日付-1' type='e.日付-1' />

• 西暦型日付形式
<xsd:complexType name='e.日付-2'>
  <xsd:sequence>
    <xsd:element ref="年-2"/>
    <xsd:element ref="月"/>
    <xsd:element ref="日"/>
  </xsd:sequence>
</xsd:complexType>
<xsd:element name='日付-2' type='e.日付-2' />

• IS08601 型日付形式
<xsd:simpleType name='e.日付-3'>
  <xsd:restriction base="xsd:string">
    <xsd:pattern value="¥d{8}" />
  </xsd:restriction>
</xsd:simpleType>
<xsd:element name='日付-3' type='e.日付-3' />
```

(3) 説明

- 文字列型日付形式
文字列型日付形式は1つの要素として扱い、データ形式は全角または半角の文字である。
- 年号型日付形式
年号、年、月、日から構成される。
年号には和暦の場合、“明治”、“大正”、“昭和”、“平成”の値を用い、西暦の場合には“西暦”の値を用いる。
年のデータ形式は1から999の半角数字、月のデータ形式は1から12の半角数字、日のデータ形式は1から31の半角数字である。
- 西暦型日付形式
西暦型日付形式は年-2（西暦年）、月、日から構成される。
年-2のデータ形式は1から9999の半角数字、月のデータ形式は1から12の半角数字、日のデータ形式は1から31の半角数字である。
- ISO8601型日付形式
ISO8601の5.2.1.1のComplete representationのBasic Formatに従い、（[ISO8601] ISO（International Organization for Standardization）. Representations of dates and times, 1988-06-15.
<http://www.iso.ch/iso/en/ISOonline.frontpage>）1つの要素として扱う。データ形式は8桁の半角数字である。

(4) 様式設計者の利用例

日付を記載するの要素は、「日付」、「到達日」等の名称を用いる。

(i) XML Schema

■文字列型日付形式

- 文字列型日付形式モジュールのデータ型（「e.日付-0」）を参照し、別名（「許認日」）を宣言した例

```
<xsd:element name='許認日' type='e.日付-0' />
```

- 文字列型日付形式モジュールのエレメント（「日付」）を参照し、エレメント（「許認」）を宣言した例

```
<xsd:complexType name='許認'>
```

```
  <xsd:sequence>
```

```
    <xsd:element ref="氏名"/>
```

```
    <xsd:element ref="日付"/>
```

```
  </xsd:sequence>
```

```
</xsd:complexType>
```

```
<xsd:element name='許認' type='許認' />
```

■年号型日付形式

- ・年号型日付形式モジュールのデータ型（「e.日付-1」）を参照し、別名（「許認日」）を宣言した例

```
<xsd:element name='許認日' type='e.日付-1' />
```

- ・年号型日付形式モジュールのエレメント（「日付-1」）を参照し、エレメント（「許認」）を宣言した例

```
<xsd:complexType name='許認'>
  <xsd:sequence>
    <xsd:element ref="氏名"/>
    <xsd:element ref="日付-1"/>
  </xsd:sequence>
</xsd:complexType>
<xsd:element name='許認' type='許認' />
```

■西暦型日付形式

- ・西暦型日付形式モジュールのデータ型（「e.日付-2」）を参照し、別名（「許認日」）を宣言した例

```
<xsd:element name='許認日' type='e.日付-2' />
```

- ・西暦型日付形式モジュールのエレメント（「日付-2」）を参照し、エレメント（「許認」）を宣言した例

```
<xsd:complexType name='許認'>
  <xsd:sequence>
    <xsd:element ref="氏名"/>
    <xsd:element ref="日付-2"/>
  </xsd:sequence>
</xsd:complexType>
<xsd:element name='許認' type='許認' />
```

■ISO8601 型日付形式

- ・ISO8601 型日付形式モジュールのデータ型（「e.日付-3」）を参照し、別名（「許認日」）を宣言した例

```
<xsd:element name='許認日' type='e.日付-3' />
```

- ・ISO8601 型日付形式モジュールのエレメント（「日付-3」）を参照し、エレメント（「許認」）を宣言した例

```
<xsd:complexType name='許認'>
  <xsd:sequence>
    <xsd:element ref="氏名"/>
    <xsd:element ref="日付-3"/>
  </xsd:sequence>
</xsd:complexType>
<xsd:element name='許認' type='許認' />
```

(ii) XML 文書

■文字列型日付形式

- ・「許認日」のマークアップ例

```
<許認日>13年1月30日</許認日>
```

- ・「許認」のマークアップ例

```
<許認>
```

```
<氏名>山田太郎</氏名>
```

```
<日付>13年1月30日</日付>
```

```
</許認>
```

■年号型日付形式

(1) 和暦の場合

- ・「許認」のマークアップ例

```
<許認日>
```

```
<年号>平成</年号>
```

```
<年>13</年>
```

```
<月>1</月>
```

```
<日>30</日>
```

```
</許認日>
```

- ・「許認」のマークアップ例

```
<許認>
```

```
<氏名>山田太郎</氏名>
```

```
<日付-1>
```

```
<年号>平成</年号>
```

```
<年>13</年>
```

```
<月>1</月>
```

```
<日>30</日>
```

```
</日付-1>
```

```
</許認>
```

(2) 西暦の場合

- ・「許認日」のマークアップ例

```
<許認日>
```

```
<年号>西暦</年号>
```

```
<年>2001</年>
```

```
<月>1</月>
```

```
<日>30</日>
```

```
</許認日>
```

- ・「許認」のマークアップ例

```
<許認>
```

```
<氏名>山田太郎</氏名>
```

```
<日付-1>
```

```
<年号>西暦</年号>
```

```
<年>2001</年>
```

```
<月>1</月>
```

```
<日>30</日>
```

```
</日付-1>
```

```
</許認>
```

■西暦型日付形式

- ・「許認日」のマークアップ例

<許認日>

<年-2>2001</年-2>

<月>1</月>

<日>30</日>

</許認日>

- ・「許認」のマークアップ例

<許認>

<氏名>山田太郎</氏名>

<日付-2>

<年-2>2001</年-2>

<月>1</月>

<日>30</日>

</日付-2>

</許認>

■ISO8601 型日付形式

- ・「許認日」のマークアップ例

<許認日>20010131</許認日>

- ・「許認」のマークアップ例

<許認>

<氏名>山田太郎</氏名>

<日付-3>20010131</日付-3>

</許認>

B 時刻に関する記載項目

時刻は様式上、「時刻」等で表現されている記載項目である。以下では時刻としての利用が想定される項目の例について述べる。

なお、様式の分析により、時、分から構成される項目が存在することが判明した。他に時や分が単独で存在する記載項目への対応の検討、あるいは秒の扱い等については必要に応じて検討する。

時刻を表現する形式には以下の3つが考えられる。

- ・ 文字列型時刻形式
1つの文字列として扱う形式である。
- ・ 時分型時刻形式
時と分から構成される形式である。
- ・ ISO8601 型時刻形式
ISO8601 で規定している時刻の形式である。

(1) モジュール名

モジュール名は時刻モジュールである。

時刻モジュールは、

- ・ データ型 :
 - e. 時刻-0 (文字列型時刻形式)
 - e. 時刻-1 (時分型時刻形式)
 - e. 時刻-2 (ISO8601 型時刻形式)
 - e. 時-0 (時分型時刻形式における「時」)
 - e. 分-0 (時分型時刻形式における「分」)
- ・ エレメント :
 - 時刻 (文字列型時刻形式)
 - 時刻-1 (時分型時刻形式)
 - 時刻-2 (ISO8601 型時刻形式)
 - 時 (時分型時刻形式における「時」)
 - 分 (時分型時刻形式における「分」)

から構成される。

(2) モジュールの申請書共通ボキャブラリにおける定義

```

・時、分の定義の宣言
<xsd:simpleType name='e.時-0'>
  <xsd:restriction base="xsd:integer">
    <xsd:minInclusive value="0"/>
    <xsd:maxInclusive value="23"/>
  </xsd:restriction>
</xsd:simpleType>
<xsd:simpleType name='e.分-0'>
  <xsd:restriction base="xsd:integer">
    <xsd:minInclusive value="0"/>
    <xsd:maxInclusive value="59"/>
  </xsd:restriction>
</xsd:simpleType>
<xsd:element name='時' type='e.時-0'/>
<xsd:element name='分' type='e.分-0'/>

・文字列型時刻形式
<xsd:simpleType name='e.時刻-0'>
  <xsd:restriction base="xsd:string">
  </xsd:restriction>
</xsd:simpleType>
<xsd:element name='時刻' type='e.時刻-0'/>

・時分型時刻形式
<xsd:complexType name='e.時刻-1'>
  <xsd:sequence>
    <xsd:element ref="時"/>
    <xsd:element ref="分"/>
  </xsd:sequence>
</xsd:complexType>
<xsd:element name='時刻-1' type='e.時刻-1'/>

・IS08601 型時刻形式
<xsd:simpleType name='e.時刻-2'>
  <xsd:restriction base="xsd:string">
    <xsd:pattern value="¥d{4}"/>
  </xsd:restriction>
</xsd:simpleType>
<xsd:element name='時刻-2' type='e.時刻-2'/>

```

(3) 説明

- 文字列型時刻形式
文字列型時刻形式は1つの要素として扱い、データ形式は全角または半角の文字である。
- 時分型時刻形式
時分型時刻形式は24時間制の時、分から構成され、時のデータ形式は0から23の半角数字、分のデータ形式は0から59の半角数字である。
- IS08601型時刻形式
IS08601の5.3.1.2のRepresentations with reduced precisionのBasic Formatに従い([IS08601] ISO (International Organization for Standardization). Representations of dates and times, 1988-06-15. <http://www.iso.ch/iso/en/ISOonline.frontpage>)1つの要素として扱う。データ形式は4桁の半角数字で、そのうち時間を表す上二桁は01から24までの半角数字、分を表す下二桁は01から60までの半角数字である。

(4) 様式設計者の利用例

時刻を表す要素は、「時刻」、「提出時刻」等の名称を用いる。

(i) XML Schema

■文字列型時刻形式

- 文字列型時刻形式モジュールのデータ型（「e.時刻-0」）を参照し、別名（「提出時刻」）を宣言した例

```
<xsd:element name='提出時刻' type='e.時刻-0' />
```

- 文字列型時刻形式モジュールのエレメント（「時刻」）を参照し、エレメント（「提出」）を宣言した例

```
<xsd:complexType name='提出'>
  <xsd:sequence>
    <xsd:element ref="氏名"/>
    <xsd:element ref="時刻"/>
  </xsd:sequence>
</xsd:complexType>
<xsd:element name='提出' type='提出' />
```

■時分型時刻形式

- 時分型時刻形式モジュールのデータ型（「e.時刻-1」）を参照し、別名（「提出時刻」）を宣言した例

```
<xsd:element name='提出時刻' type='e.時刻-1' />
```

- 時分型時刻形式モジュールのエレメント（「時刻-1」）を参照し、エレメント（「提出」）を宣言した例

```
<xsd:complexType name='提出'>
  <xsd:sequence>
    <xsd:element ref="氏名"/>
    <xsd:element ref="時刻-1"/>
  </xsd:sequence>
</xsd:complexType>
<xsd:element name='提出' type='提出' />
```

■IS08601 型時刻形式

・ IS08601 型時刻形式モジュールのデータ型（「e.時刻-2」）を参照し、別名（「提出時刻」）を宣言した例

```
<xsd:element name='提出時刻' type='e.時刻-2' />
```

・ IS08601 型時刻形式モジュールのエレメント（「時刻-2」）を参照し、エレメント（「提出」）を宣言した例

```
<xsd:complexType name='提出'>
  <xsd:sequence>
    <xsd:element ref="氏名"/>
    <xsd:element ref="時刻-2"/>
  </xsd:sequence>
</xsd:complexType>
<xsd:element name='提出' type='提出' />
```

(ii) XML 文書

■文字型時刻形式

・ 「提出時刻」のマークアップ例

```
<提出時刻>10 時 20 分</提出時刻>
```

・ 「提出」のマークアップ例

```
<提出>
  <氏名>山田太郎</氏名>
  <時刻>10 時 20 分</時刻>
</提出>
```

■時分型時刻形式

・ 「提出時刻」のマークアップ例

```
<提出時刻>
  <時>10</時>
  <分>20</分>
</提出時刻>
```

・ 「提出」のマークアップ例

```
<提出>
  <氏名>山田太郎</氏名>
  <時刻-1>
    <時>10</時>
    <分>20</分>
  </時刻-1>
</提出>
```

■IS08601 型時刻形式

・ 「提出時刻」のマークアップ例

```
<提出時刻>1020</提出時刻>
```

・ 「提出」のマークアップ例

```
<提出>
  <氏名>山田太郎</氏名>
```

```
<時刻-2>1020</時刻-2>  
</提出>
```

C 日付・時刻に関するその他のバリエーション

以上に述べた形式の他に、実際の様式の中には年だけもの、月だけのもの、あるいは年月等、いくつかのパターンの存在を確認した。これらについても日付型として定義しておくことも有効である。以下に実際に様式中に存在したものの日付型の候補を列記する。

- ① 年
- ② 月
- ③ 年+月
- ④ 年+月+日
- ⑤ 年号+年
- ⑥ 年号+年+月
- ⑦ 月+日
- ⑧ 日付と時刻の複合型

※ 留意事項

上記のうち、年号のない年については、現行の紙の申請書様式では和暦、西暦のどちらを記入するかの規定は特に定めていないため、和暦、西暦どちらの数値も入力する可能性がある。

5.2.2.3. 数量に関する記載項目

数量は数値および単位によって表現される項目である。

数量に関する項目は、数値とそれと対になる単位によって構成されており、単位をどのように扱うかについては4章に示したとおりである。

数量に関する記載項目を金額、期間、その他の数量それぞれについて説明する。

A 金額に関する記載項目

金額に関する事項として資本金および売上高がある。これらは様式上、「資本金」、「売上高」等で表現されている記載項目である。

(1) モジュール名

モジュール名は金額モジュール、資本金モジュール、売上高モジュールおよび金額の単位を表すモジュールである。

金額モジュールは、

- ・ データ型 : e. 金額-0
- ・ エレメント : 金額

資本金モジュールは、

- ・ データ型 : e. 資本金-0
- ・ エレメント : 資本金

から構成される。

売上高モジュールは、

- ・ データ型 : e. 売上高-0
- ・ エレメント : 売上高

から構成される。

金額の単位を表すモジュールは、

- ・ データ型 : e. 円-0
e. 千円-0
e. 百万円-0
e. ドル-0
e. 千ドル-0
e. 百万ドル-0
- ・ エレメント : 円
千円
百万円
ドル
千ドル
百万ドル

から構成される。

(2) モジュールの申請書共通ボキャブラリにおける定義

```

・金額、資本金、売上高、単位付きの金額の定義と宣言
<xsd:simpleType name='e.金額-0'>
  <xsd:restriction base="xsd:integer">
  </xsd:restriction>
</xsd:simpleType>
<xsd:simpleType name='e.資本金-0'>
  <xsd:restriction base="xsd:integer">
  </xsd:restriction>
</xsd:simpleType>
<xsd:simpleType name='e.売上高-0'>
  <xsd:restriction base="xsd:integer">
  </xsd:restriction>
</xsd:simpleType>
<xsd:simpleType name='e.円-0'>
  <xsd:restriction base="xsd:integer">
  </xsd:restriction>
</xsd:simpleType>
<xsd:simpleType name='e.千円-0'>
  <xsd:restriction base="xsd:integer">
  </xsd:restriction>
</xsd:simpleType>
<xsd:simpleType name='e.百万円-0'>
  <xsd:restriction base="xsd:integer">
  </xsd:restriction>
</xsd:simpleType>
<xsd:simpleType name='e.ドル-0'>
  <xsd:restriction base="xsd:integer">
  </xsd:restriction>
</xsd:simpleType>
<xsd:simpleType name='e.千ドル-0'>
  <xsd:restriction base="xsd:integer">
  </xsd:restriction>
</xsd:simpleType>
<xsd:simpleType name='e.百万ドル-0'>
  <xsd:restriction base="xsd:integer">
  </xsd:restriction>
</xsd:simpleType>
<xsd:element name='金額' type='e.金額-0'/>
<xsd:element name='資本金' type='e.資本金-0'/>
<xsd:element name='売上高' type='e.売上高-0'/>
<xsd:element name='円' type='e.円-0'/>
<xsd:element name='千円' type='e.千円-0'/>
<xsd:element name='百万円' type='e.百万円-0'/>
<xsd:element name='ドル' type='e.ドル-0'/>
<xsd:element name='千ドル' type='e.千ドル-0'/>
<xsd:element name='百万ドル' type='e.百万ドル-0'/>

```

(3) 説明

金額、資本金、売上高および金額の単位を表すモジュールはそれぞれ 1 つの要素として扱い、ともにデータ形式はカンマ区切りなしの半角数字である。

(4) 様式設計者の利用例

金額を表す要素は、「金額」、「料金」、「資本金」、「資本」、「売上高」、「売上金」等の名称を用いる。

(i) XML Schema

```

・金額モジュールのデータ型（「e.金額-0」）を参照し、別名（「支払額」）を宣言した例
<xsd:element name='支払額' type='e.金額-0' />

・金額モジュールのエレメント（「金額」）を参照し、エレメント（「支払」）を宣言した例
<xsd:complexType name='支払'>
  <xsd:sequence>
    <xsd:element ref="氏名"/>
    <xsd:element ref="金額"/>
  </xsd:sequence>
</xsd:complexType>
<xsd:element name='支払' type='支払' />

・資本金、売上金モジュールのデータ型（「e.資本金-0」、「e.売上金-0」）を参照し、別名（「資本」、「売上」）を宣言した例
<xsd:element name='資本' type='e.資本金-0' />
<xsd:element name='売上' type='e.売上高-0' />

・資本金、売上高モジュールのエレメント（「資本金」、「売上高」）を参照し、エレメント（「企業情報」）を宣言した例
<xsd:complexType name='企業情報'>
  <xsd:sequence>
    <xsd:element ref="法人団体名"/>
    <xsd:element ref="資本金"/>
    <xsd:element ref="売上高"/>
  </xsd:sequence>
</xsd:complexType>

・金額の単位を表すモジュールのデータ型（「e.円-0」）を参照し、別名（「交通費-円」）を宣言した例
<xsd:element name='交通費-円' type='e.円-0' />

```


(ii) XML 文書

・「支払額」のマークアップ例
<支払額>300000</支払額>

・「支払」のマークアップ例
<支払>
 <氏名>山田太郎</氏名>
 <金額>300000</金額>
</支払>

・「資本」、「売上」のマークアップ例
<資本>10000000</資本>
<売上>50000000</売上>

・「企業情報」のマークアップ例
<企業情報>
 <法人団体名>〇〇株式会社</法人団体名>
 <資本金>10000000</資本金>
 <売上高>50000000</売上高>
</企業情報>

・「交通費-円」のマークアップ例
<交通費-円>400</交通費-円>

B 期間（数量）に関する記載項目

期間は年数、月数などの期間に関する記載項目である。以下では期間として利用される項目の例について述べる。

(1) モジュール名

モジュール名は数量型期間モジュールおよび数量型期間の単位を表すモジュールである。

数量型期間モジュールは、

- データ型 : e. 数量型期間-0
- エレメント : 数量型期間

から構成される。

数量型期間の単位を表すモジュールは、

- データ型 :
 - e. 年数-0
 - e. 月数-0
 - e. 日数-0
 - e. 時間-0
- エレメント :
 - 年数
 - 月数
 - 日数
 - 時間

から構成される。

(2) モジュールの申請書共通ボキャブラリにおける定義

- 数量型期間、単位付きの数量型期間の定義と宣言

```
<xsd:simpleType name='e.数量型期間-0'>
  <xsd:restriction base="xsd:string">
    </xsd:restriction>
  </xsd:simpleType>
<xsd:simpleType name='e.年数-0'>
  <xsd:restriction base="xsd:integer">
    </xsd:restriction>
  </xsd:simpleType>
<xsd:simpleType name='e.月数-0'>
  <xsd:restriction base="xsd:integer">
    </xsd:restriction>
  </xsd:simpleType>
<xsd:simpleType name='e.日数-0'>
  <xsd:restriction base="xsd:integer">
    </xsd:restriction>
  </xsd:simpleType>
<xsd:simpleType name='e.時間-0'>
  <xsd:restriction base="xsd:integer">
    </xsd:restriction>
  </xsd:simpleType>
```

```

</xsd:simpleType>
<xsd:element name='数量型期間' type='e.数量型期間-0' />
<xsd:element name='年数' type='e.年数-0' />
<xsd:element name='月数' type='e.月数-0' />
<xsd:element name='日数' type='e.日数-0' />
<xsd:element name='時間' type='e.時間-0' />

```

(3) 説明

数量型期間および数量型期間の単位を表すモジュールはそれぞれ 1 つの要素として扱い、ともにデータ形式はカンマ区切りなしの半角数字である。

(4) 様式設計者の利用例

数量型期間を表す要素は、「期間」、「稼働期間」等の名称を用いる。

(i) XML Schema

- ・数量型期間モジュールのデータ型（「e.数量型期間-0」）を参照し、別名（「稼働期間」）を宣言した例

```
<xsd:element name='稼働期間' type='e.数量型期間-0' />
```

- ・数量型期間モジュールのエレメント（「数量型期間」）を参照し、エレメント（「稼働実績」）を宣言した例

```

<xsd:complexType name='稼働実績'>
  <xsd:sequence>
    <xsd:element ref="内容"/>
    <xsd:element ref="数量型期間"/>
  </xsd:sequence>
</xsd:complexType>
<xsd:element name='稼働実績' type='稼働実績' />

```

- ・数量型期間の単位を表すモジュールのデータ型（「e.日数-0」）を参照し、エレメント（「稼働実績-日」）を宣言した例

```
<xsd:element name='稼働期間-日' type='e.日数-0' />
```

(ii) XML 文書

- ・「稼働期間」のマークアップ例

```
<稼働期間>10</稼働期間>
```

- ・「稼働実績」のマークアップ例

```

<稼働実績>
  <内容>〇〇に関する作業</内容>
  <数量型期間>10</数量型期間>
</稼働実績>

```

- ・「稼働実績-日」のマークアップ例

```
<稼働実績-日>10</稼働実績-日>
```

C その他の数量に関する項目

上記の他に数量に関する記載項目として年齢、従業員数、株式数がある。それぞれ 1 つの要素として扱い、データ形式は半角の数字である。年齢は 1 歳単位、株式数は 1 株単位を基本とし、単位を特にモジュールとして提供しない。従業員数は 1 人を単位として提供する。

また、その他に重さ、面積、発熱量等の物理的な分野に特化したものを基本物理量ボキャブラリとして示す。具体的な項目は「付録 J 基本物理量ボキャブラリ」として記載したので、参照されたい。

5.2.2.4. 範囲に関する記載項目

時間あるいは物理的な起点と終点からなる範囲に関する記載項目について示す。

A 期間（範囲）に関する記載項目

期間は様式上、「(自) 平成 13 年 6 月 1 日 (至) 平成 13 年 6 月 30 日」のように開始日付と終了日付によって表される記載項目である。以下では期間としての利用が想定される例について述べる。

(1) モジュール名

モジュール名は範囲型期間モジュールである。

範囲型期間モジュールは、

- ・ データ型 :
 - e. 範囲型期間-0 (文字列範囲型期間)
 - e. 範囲型期間-1 (日付範囲型期間)
 - e. 範囲型期間-2 (時刻範囲型期間)
- ・ エレメント :
 - 範囲型期間 (文字列範囲型期間)
 - 範囲型期間-1 (日付範囲型期間)
 - 範囲型期間-2 (時刻範囲型期間)
 - 開始日付 (日付範囲型期間における「開始日付」)
 - 終了日付 (日付範囲型期間における「終了日付」)
 - 開始時刻 (時刻範囲型期間における「開始時刻」)
 - 終了時刻 (時刻範囲型期間における「終了時刻」)

から構成される。

開始日付および終了日付は「5.2.2.2. 時間に関する記載項目」で示した文字列型日付形式のデータ型を参照して宣言する。

開始時刻および終了時刻は「5.2.2.2. 時間に関する記載項目」で示した文字列型時刻形式のデータ型を参照して宣言する。

(2) モジュールの申請書共通ボキャブラリにおける定義

```

・ 開始日付、終了日付、開始時刻、終了時刻の宣言
<xsd:element name='開始日付' type='e.日付-0' />
<xsd:element name='終了日付' type='e.日付-0' />
<xsd:element name='開始時刻' type='e.時刻-0' />
<xsd:element name='終了時刻' type='e.時刻-0' />

・ 文字列範囲型期間
<xsd:simpleType name='e.範囲型期間-0'>
  <xsd:restriction base="xsd:string">
  </xsd:restriction>
</xsd:simpleType>
<xsd:element name='範囲型期間' type='e.範囲型期間-0' />

```

```

・日付範囲型期間
<xsd:complexType name='e.範囲型期間-1'>
  <xsd:sequence>
    <xsd:element ref="開始日付"/>
    <xsd:element ref="終了日付"/>
  </xsd:sequence>
</xsd:complexType>
<xsd:element name='範囲型期間-1' type='e.範囲型期間-1'/>

・時刻範囲型期間
<xsd:complexType name='e.範囲型期間-2'>
  <xsd:sequence>
    <xsd:element ref="開始時刻"/>
    <xsd:element ref="終了時刻"/>
  </xsd:sequence>
</xsd:complexType>
<xsd:element name='範囲型期間-2' type='e.範囲型期間-2'/>

```

(3) 説明

以下、「文字列範囲型期間」、「日付範囲型期間」、「時刻範囲型期間」の形式別に説明する。

- ・ 文字列範囲型期間
文字列範囲型期間は1つの要素として扱い、データ形式は全角または半角の文字である。
- ・ 日付範囲型期間
前述した文字列型日付形式モジュールに従う。
- ・ 時刻範囲型期間
前述した文字列型時刻形式モジュールに従う。

(4) 様式設計者の利用例

範囲型期間を表す要素は、「期間」、「工事期間」等の名称を用いる。

(i) XML Schema

```

■文字列範囲型期間
・文字列範囲型期間モジュールのデータ型（「e.範囲型期間-0」）を参照し、別名（「工事期間」）を宣言した例（文字列型日付形式を適用した場合）

<xsd:element name='工事期間' type='e.範囲型期間-0'/>

・文字列範囲型期間モジュールのエレメント（「範囲型期間」）を参照し、エレメント（「工事」）を宣言した例（文字列型日付形式を適用した場合）

<xsd:complexType name='工事'>
  <xsd:sequence>
    <xsd:element ref="事業者"/>
    <xsd:element ref="範囲型期間"/>
  </xsd:sequence>
</xsd:complexType>
<xsd:element name='工事' type='工事'/>

```

■日付範囲型期間

・日付範囲型期間モジュールのデータ型（「e.範囲型期間-1」）を参照し、別名（「工事期間」）を宣言した例（文字列型日付形式を適用した場合）

```
<xsd:element name='工事期間' type='e.範囲型期間-1' />
```

・日付範囲型期間モジュールの要素（「範囲型期間-1」）を参照し、要素（「工事」）を宣言した例（文字列型日付形式を適用した場合）

```
<xsd:complexType name='工事'>
  <xsd:sequence>
    <xsd:element ref="事業者"/>
    <xsd:element ref="範囲型期間-1"/>
  </xsd:sequence>
</xsd:complexType>
<xsd:element name='工事' type='工事' />
```

■時刻範囲型期間

・時刻範囲型期間モジュールのデータ型（「e.範囲型期間-2」）を参照し、別名（「工事期間」）を宣言した例（文字列型日付形式を適用した場合）

```
<xsd:element name='工事期間' type='e.範囲型期間-2' />
```

・時刻範囲型期間モジュールの要素（「範囲型期間-2」）を参照し、要素（「工事」）を宣言した例（文字列型日付形式を適用した場合）

```
<xsd:complexType name='工事'>
  <xsd:sequence>
    <xsd:element ref="事業者"/>
    <xsd:element ref="範囲型期間-2"/>
  </xsd:sequence>
</xsd:complexType>
<xsd:element name='工事' type='工事' />
```

(ii) XML 文書

■文字列範囲型期間

- ・「工事期間」のマークアップ例

```
<工事期間>自平成 13 年 10 月 12 日～至平成 13 年 10 月 23 日</工事期間>
```

- ・「工事」のマークアップ例

```
<工事>
```

```
  <事業者>A株式会社</事業者>
```

```
  <範囲型期間>自平成 13 年 10 月 12 日～至平成 13 年 10 月 23 日</範囲型期間>
```

```
</工事>
```

■日付範囲型期間

- ・「工事期間」のマークアップ例

```
<工事期間>
```

```
  <開始日付>平成 13 年 10 月 12 日</開始日付>
```

```
  <終了日付>平成 13 年 10 月 23 日</終了日付>
```

```
</工事期間>
```

- ・「工事」のマークアップ例

```
<工事>
```

```
  <事業者>A株式会社</事業者>
```

```
  <範囲型期間-1>
```

```
    <開始日付>平成 13 年 10 月 12 日</開始日付>
```

```
    <終了日付>平成 13 年 10 月 23 日</終了日付>
```

```
  </範囲型期間-1>
```

```
</工事>
```

■時刻範囲型期間

- ・「工事期間」のマークアップ例

```
<工事期間>
```

```
  <開始時刻>11 時 30 分</開始時刻>
```

```
  <終了時刻>16 時 00 分</終了時刻>
```

```
</工事期間>
```

- ・「工事」のマークアップ例

```
<工事>
```

```
  <事業者>A株式会社</事業者>
```

```
  <範囲型期間-2>
```

```
    <開始時刻>11 時 30 分</開始時刻>
```

```
    <終了時刻>16 時 00 分</終了時刻>
```

```
  </範囲型期間-2>
```

```
</工事>
```


B 区間に関する記載項目

区間は様式上、「区間」、「乗車区間」などの区間に関する記載項目である。以下では区間としての利用が想定される項目の例について述べる。

(1) モジュール名

モジュール名は区間モジュールである。

区間モジュールは、

- ・ データ型 :
 - e. 区間-0 (1つの要素として扱う場合)
 - e. 区間-1 (複数の要素に分解して扱う場合)
 - e. 始点-0 (複数の要素に分解して扱う場合の「始点」)
 - e. 終点-0 (複数の要素に分解して扱う場合の「終点」)
- ・ エレメント :
 - 区間 (1つの要素として扱う場合)
 - 区間-1 (複数の要素に分解して扱う場合)
 - 始点 (複数の要素に分解して扱う場合の「始点」)
 - 終点 (複数の要素に分解して扱う場合の「終点」)

から構成される。

(2) モジュールの申請書共通ボキャブラリにおける定義

```

・ 始点、終点の定義と宣言
<xsd:simpleType name='e.始点-0'>
  <xsd:restriction base="xsd:string">
  </xsd:restriction>
</xsd:simpleType>
<xsd:simpleType name='e.終点-0'>
  <xsd:restriction base="xsd:string">
  </xsd:restriction>
</xsd:simpleType>
<xsd:element name='始点' type='e.始点-0' />
<xsd:element name='終点' type='e.終点-0' />

・ 基本パターン (1つの要素として扱う場合)
<xsd:simpleType name='e.区間-0'>
  <xsd:restriction base="xsd:string">
  </xsd:restriction>
</xsd:simpleType>
<xsd:element name='区間' type='e.区間-0' />

・ その他のパターン (複数の要素に分解して扱う場合)
<xsd:complexType name='e.区間-1'>
  <xsd:sequence>
    <xsd:element ref="始点" />
    <xsd:element ref="終点" />
  </xsd:sequence>
</xsd:complexType>
<xsd:element name='区間-1' type='e.区間-1' />

```

(3) 説明

区間を1つの要素として扱うか、始点と終点とに要素分割して扱うかは業務に依存する。

区間の各要素のデータ形式は基本パターン、その他のパターンともに全角または半角の文字である。

(4) 様式設計者の利用例

区間を表す要素は、「区間」、「乗車区間」等の名称を用いる。

(i) XML Schema

```

・ 区間モジュールのデータ型（「e.区間-0」）を参照し、別名（「乗車区間」）を宣言した例
<xsd:element name='乗車区間' type='e.区間-0' />

・ 区間モジュールのエレメント（「区間」）を参照し、エレメント（「乗車」）を宣言した例
<xsd:complexType name='乗車'>
  <xsd:sequence>
    <xsd:element ref="路線名" />
    <xsd:element ref="区間" />
  </xsd:sequence>
</xsd:complexType>
<xsd:element name='乗車' type='乗車' />

```

(ii) XML 文書

```

・ 「乗車区間」のマークアップ例
<乗車区間>横浜駅から新橋駅まで</乗車区間>

・ 「様式名」のマークアップ例
<乗車>
  <路線名>東海道線</路線名>
  <区間>横浜駅から新橋駅まで</区間>
</乗車>

```

5.2.2.5. その他の記載内容要素群

A 振り仮名に関する記載項目

氏名、住所等に付与される振り仮名を記載する項目である。以下では振り仮名としての利用が想定される項目の例について述べる。

(1) モジュール名

モジュール名はフリガナモジュールおよびふりがなモジュールである。

フリガナモジュールおよびふりがなモジュールは、

- ・ データ型 : e.フリガナ-0
e.ふりがな-0
- ・ エレメント : フリガナ
ふりがな

から構成される。

(2) モジュールの申請書共通ボキャブラリにおける定義

```
<xsd:simpleType name='e.フリガナ-0'>
  <xsd:restriction base="xsd:string">
    </xsd:restriction>
  </xsd:simpleType>
<xsd:simpleType name='e.ふりがな-0'>
  <xsd:restriction base="xsd:string">
    </xsd:restriction>
  </xsd:simpleType>
<xsd:element name='フリガナ' type='e.フリガナ-0'/>
<xsd:element name='ふりがな' type='e.ふりがな-0'/>
```

(3) 説明

フリガナモジュール、ふりがなモジュールはそれぞれ 1 つの要素として扱う。申請者の利便性を考慮して、フリガナモジュール、ふりがなモジュールともにデータ形式は、カタカナの全角文字と平仮名の全角文字のどちらにも対応可能にする。

(4) 様式設計者の利用例

振り仮名を表す要素は、「フリガナ」、「ふりがな」等の名称を用いる。

(i) XML Schema

■フリガナモジュール

- ・フリガナモジュールのデータ型（「e.フリガナ-0」）を参照し、別名（「氏名フリガナ」）を宣言した例

```
<xsd:element name='氏名フリガナ' type='e.フリガナ-0' />
```

- ・フリガナモジュールのエレメント（「フリガナ」）を参照し、エレメント（「代表者氏名」）を宣言した例

```
<xsd:complexType name='代表者氏名'>
  <xsd:sequence>
    <xsd:element ref="氏名"/>
    <xsd:element ref="フリガナ"/>
  </xsd:sequence>
</xsd:complexType>
<xsd:element name='代表者氏名' type='代表者氏名' />
```

■ふりがなモジュール

- ・ふりがなモジュールのデータ型（「e.ふりがな-0」）を参照し、別名（「氏名ふりがな」）を宣言した例

```
<xsd:element name='氏名ふりがな' type='e.ふりがな-0' />
```

- ・ふりがなモジュールのエレメント（「ふりがな」）を参照し、エレメント（「代表者氏名」）を宣言した例

```
<xsd:complexType name='代表者氏名'>
  <xsd:sequence>
    <xsd:element ref="氏名"/>
    <xsd:element ref="ふりがな"/>
  </xsd:sequence>
</xsd:complexType>
<xsd:element name='代表者氏名' type='代表者氏名' />
```

(ii) XML 文書

■フリガナモジュール

- ・「氏名フリガナ」のマークアップ例

```
<氏名フリガナ>ヤマダタロウ</氏名フリガナ>
```

- ・「代表者氏名」のマークアップ例

```
<代表者氏名>
  <氏名>山田太郎</氏名>
  <フリガナ>ヤマダタロウ</フリガナ>
</代表者氏名>
```

■ふりがなモジュール

- ・「氏名ふりがな」のマークアップ例

```
<氏名ふりがな>やまだたろう</氏名ふりがな>
```

- ・「代表者氏名」のマークアップ例

```
<代表者氏名>
  <氏名>山田太郎</氏名>
  <ふりがな>やまだたろう</ふりがな>
</代表者氏名>
```

B 場所に関する記載項目

場所は「仕入れ先」、「所在地」、「原産地」、「出荷先」等、場所を示す項目である。前述した宛先としての住所とは異なり、物理的な特定の位置を示す項目である。以下では場所としての利用が想定される項目の例について述べる。

(1) モジュール名

モジュール名は場所モジュールである。

場所モジュールは、

- ・ データ型 : e. 場所-0
- ・ エレメント : 場所

から構成される。

(2) モジュールの申請書共通ボキャブラリにおける定義

```
<xsd:simpleType name='e.場所-0'>
  <xsd:restriction base="xsd:string">
    </xsd:restriction>
  </xsd:simpleType>
<xsd:element name='場所' type='e.場所-0' />
```

(3) 説明

場所は1つの要素として扱い、データ形式は全角および半角の文字である。

(4) 様式設計者の利用例

場所を表す要素は「場所」、「納入場所」等の名称を用いる。

(i) XML Schema

```
・ 場所モジュールのデータ型（「e. 場所-0」）を参照し、別名（「納入場所」）を宣言した例
<xsd:element name='納入場所' type='e.場所-0' />

・ 場所モジュールのエレメント（「場所」）を参照し、エレメント（「納入」）を宣言した例
<xsd:complexType name='納入'>
  <xsd:sequence>
    <xsd:element ref="業者名" />
    <xsd:element ref="場所" />
  </xsd:sequence>
</xsd:complexType>
<xsd:element name='納入' type='納入' />
```

(ii) XML 文書

・「納入場所」のマークアップ例
<納入場所>××地区</納入場所>

・「納入」のマークアップ例
<納入>
 <業者名>〇〇株式会社</業者名>
 <場所>××地区</場所>
</納入>

C 理由に関する記載項目

理由は様式上、「変更の理由」、「〇〇に関する変更事由」等の理由を示す記載項目である。以下では理由としての利用が想定される項目の例について述べる。

(1) モジュール名

モジュール名は理由モジュールである。

理由モジュールは、

- ・ データ型 : e.理由-0
- ・ エレメント : 理由

から構成される。

(2) モジュールの申請書共通ボキャブラリにおける定義

```
<xsd:simpleType name='e.理由-0'>
  <xsd:restriction base="xsd:string">
    </xsd:restriction>
  </xsd:simpleType>
<xsd:element name='理由' type='e.理由-0' />
```

(3) 説明

理由は1つの要素として扱い、データ形式は全角および半角の文字である。

(4) 様式設計者の利用例

理由を表す要素は、「理由」、「変更の理由」等の名称を用いる。

(i) XML Schema

```
・ 理由モジュールのデータ型（「e.理由-0」）を参照し、別名（「変更の理由」）を宣言した例
<xsd:element name='変更の理由' type='e.理由-0' />

・ 理由モジュールのエレメント（「理由」）を参照し、エレメント（「変更」）を宣言した例
<xsd:complexType name='変更'>
  <xsd:sequence>
    <xsd:element ref="日付"/>
    <xsd:element ref="理由"/>
  </xsd:sequence>
</xsd:complexType>
<xsd:element name='変更' type='変更' />
```

(ii) XML 文書

```
• 「変更の理由」のマークアップ例
<変更の理由>住所の移転による</変更の理由>

• 「変更」のマークアップ例
<変更>
  <日付>13年10月24日</日付>
  <理由>住所の移転による</理由>
</変更>
```


D 符号に関する記載項目

符号は様式上、「輸入者符号」、「貿易形態別符号」等の符号を示す記載項目である。以下では符号としての利用が想定される項目の例について述べる。

(1) モジュール名

モジュール名は符号モジュールである。

符号モジュールは、

- ・ データ型 : e.符号-0
- ・ エレメント : 符号

から構成される。

(2) モジュールの申請書共通ボキャブラリにおける定義

```
<xsd:simpleType name='e.符号-0'>
  <xsd:restriction base="xsd:string">
    </xsd:restriction>
  </xsd:simpleType>
<xsd:element name='符号' type='e.符号-0' />
```

(3) 説明

符号は1つの要素として扱い、データ形式は半角英数字である。

(4) 様式設計者の利用例

符号を表す要素は、「輸入者符号」、「貿易形態別符号」等の名称を用いる。

(i) XML Schema

```
・ 符号モジュールのデータ型（「e.符号-0」）を参照し、別名（「輸入者符号」）を宣言した例
<xsd:element name='輸入者符号' type='e.符号-0' />

・ 符号モジュールのエレメント（「符号」）を参照し、エレメント（「企業情報」）を宣言した例
<xsd:complexType name='企業情報'>
  <xsd:sequence>
    <xsd:element ref="符号" />
    <xsd:element ref="法人団体名" />
  </xsd:sequence>
</xsd:complexType>
<xsd:element name='企業情報' type='企業情報' />
```

(ii) XML 文書

・「輸入者符号」のマークアップ例
<輸入者符号>A12345</輸入者符号>

・「輸入者」のマークアップ例
<企業情報>
 <符号>A12345</符号>
 <法人団体名>A株式会社</法人団体名>
</企業情報>

5.2.3. 記載支援要素群

ここでは、「記載支援要素群」の様式構成要素である「記載説明」について述べる。

5.2.3.1. 記載説明に関する記載項目

記載説明は様式上、「記載説明」等で表現されている記載項目である。以下では記載説明としての利用が想定される項目の例について述べる。

(1) モジュール名

モジュール名は記載説明モジュールである。

記載説明モジュールは、

- ・ データ型 : e. 記載説明-0
- ・ エレメント : 記載説明

から構成される。

(2) モジュールの申請書共通ボキャブラリにおける定義

```
<xsd:simpleType name='e.記載説明-0'>
  <xsd:restriction base="xsd:string">
  </xsd:restriction>
</xsd:simpleType>
<xsd:element name='記載説明' type='e.記載説明-0'/>
```

(3) 説明

記載説明は1つの要素として扱い、データ形式は全角または半角の文字である。

(4) 様式設計者の利用例

記載説明を表す要素は、「記載説明」、「記載要領」等の名称を用いる。

(i) XML Schema

```
・ 記載説明モジュールのデータ型（「e. 記載説明-0」）を参照し、別名（「記載要領」）を宣言した例
<xsd:element name='記載要領' type='e.記載説明-0'/>
・ 記載説明モジュールのエレメント（「記載説明」）を参照し、エレメント（「備考欄」）を宣言した例
<xsd:complexType name='備考欄'>
  <xsd:sequence>
    <xsd:element ref="記載説明"/>
    <xsd:element ref="記載説明"/>
  </xsd:sequence>
</xsd:complexType>
<xsd:element name='備考欄' type='備考欄'/>
```

(ii) XML 文書

・「記載要領」のマークアップ例

<記載要領>次年度以降において燃料等(電気)の使用量が令第2条第1項(第2項)に該当しないことが明らかである場合は、その旨およびその理由を備考の欄に記入すること。</記載要領>

・「備考欄」のマークアップ例

<備考欄>

<記載説明>次年度以降において燃料等(電気)の使用量が令第2条第1項(第2項)に該当しないことが明らかである場合は、その旨およびその理由を備考の欄に記入すること。</記載説明>

<記載説明>注釈。ただし〇〇の場合は除く</記載説明>

</備考欄>

NMDA 電子申請フレームワーク

電子申請用 XML 様式的设计ガイドライン — タグ設計方針と申請書共通ボキャブラリ —

2001 年 12 月 26 日 第 1 版発行

監修 財団法人ニューメディア開発協会

編集 インターネット汎用電子申請システム
タグ標準化タスクフォース

発行者 国 分 明 男

発行所 財団法人ニューメディア開発協会

108-0073 東京都港区三田 1-4-28 三田国際ビル

電話 03-3457-0672

FAX 03-3451-9604

URL: <http://www.nmda.or.jp>

sinsei-info@nmda.or.jp (問合せ先)

Copyright © 2001 Information-technology Promotion Agency, Japan (IPA)

New Media Development Association (NMDA) All Rights Reserved.

※本書の全部または一部を無断で複写、複製、転記載することを禁じます