

## 付録

ディスカッションペーパー用アーキテクチャ作成

コンセプトペーパー

2021年8月

## 目次

1	概要	3
1.1	背景	3
1.2	目的	3
1.3	概要	3
1.4	用語の定義	5
2	アーキテクチャ記述の基本要素	6
2.1	基本とするレイヤー構造	6
2.2	各エレメント、リレーションの定義及び使用例の整理	6
3	アーキテクチャの設計	14
3.1	アーキテクチャ設計の準備	14
3.2	アーキテクチャ設計の流れ	14
1)	戦略レイヤー	14
2)	組織レイヤー	15
3)	ルールレイヤー	16
4)	業務（サービス）レイヤー、利活用環境レイヤー、連携基盤（ツール）レイヤー	17
5)	データレイヤー、データ標準レイヤー	20
6)	インフラレイヤー	20
4	アーキテクチャの利用	22
4.1	全体の関係性の整理	22
4.2	戦略の整合性確保	23
4.3	プラットフォーム間の共通化や連携の検討	24
5	アーキテクチャの詳細化	25

## 1 概要

### 1.1 背景

昨今においては、社会全体が複雑化しており、分野横断でのプロジェクトというものが増えてきています。また、技術的な検討と並行して制度的な変更が行われるといったケースも増えていきます。このように複雑なサービス等の構築や運用では、全体像を可視化して関係者全体で共通の理解を図っていくことが重要となります。しかし、これまではアーキテクチャ的に設計が行われてこなかったために、プロジェクト開始までに時間がかかったり、運用間近になって検討項目の抜け漏れが発覚したりするなどの問題が発生していました。

アーキテクチャを使っていたとしても、独自のアーキテクチャ表現手法を使っているために、関係者の理解に時間がかかったり、誤解が生じてしまったりすることも散見されます。

このような課題を解決するために、アーキテクチャ推進団体のオープングループが、アーキテクチャ標準の TOGAF (R)<sup>1</sup>の開発を推進するとともに、アーキテクチャモデリング言語の ArchiMate (R)<sup>2</sup>を策定しています。他にアーキテクチャモデリング言語がないこともあり、国際的に利用が広がり始めています。

政府においても、アーキテクチャの活用が目指されており、Society5.0 参照アーキテクチャ<sup>3</sup>が整備され、スマートシティやデータ戦略の検討に展開されていますが、詳細な記述方法の定義がないため、アーキテクチャの導入が概念レベルで立ち止まっています。

### 1.2 目的

本ペーパーは、アーキテクチャ設計と活用の手法をディスカッションする作業を円滑にするために整備しているものとなります。

本ペーパーに沿って比較検証用モデルを設計することで、アーキテクチャ設計の構造や粒度の統一方法について検証を行うことができます。

### 1.3 概要

アーキテクチャの設計には、アーキテクチャ記述の統一が必要となります。アーキテクチャの標準としては、ISO/IEC/IEEE 42010 Systems and software

<sup>1</sup> <https://www.opengroup.org/togaf>

<sup>2</sup> <https://www.opengroup.org/archimate-home>

<sup>3</sup>

<https://www8.cao.go.jp/cstp/tyousakai/juyoukadai/14kai/siryo2-1.pdf>

engineering - Architecture description<sup>4</sup>があります。

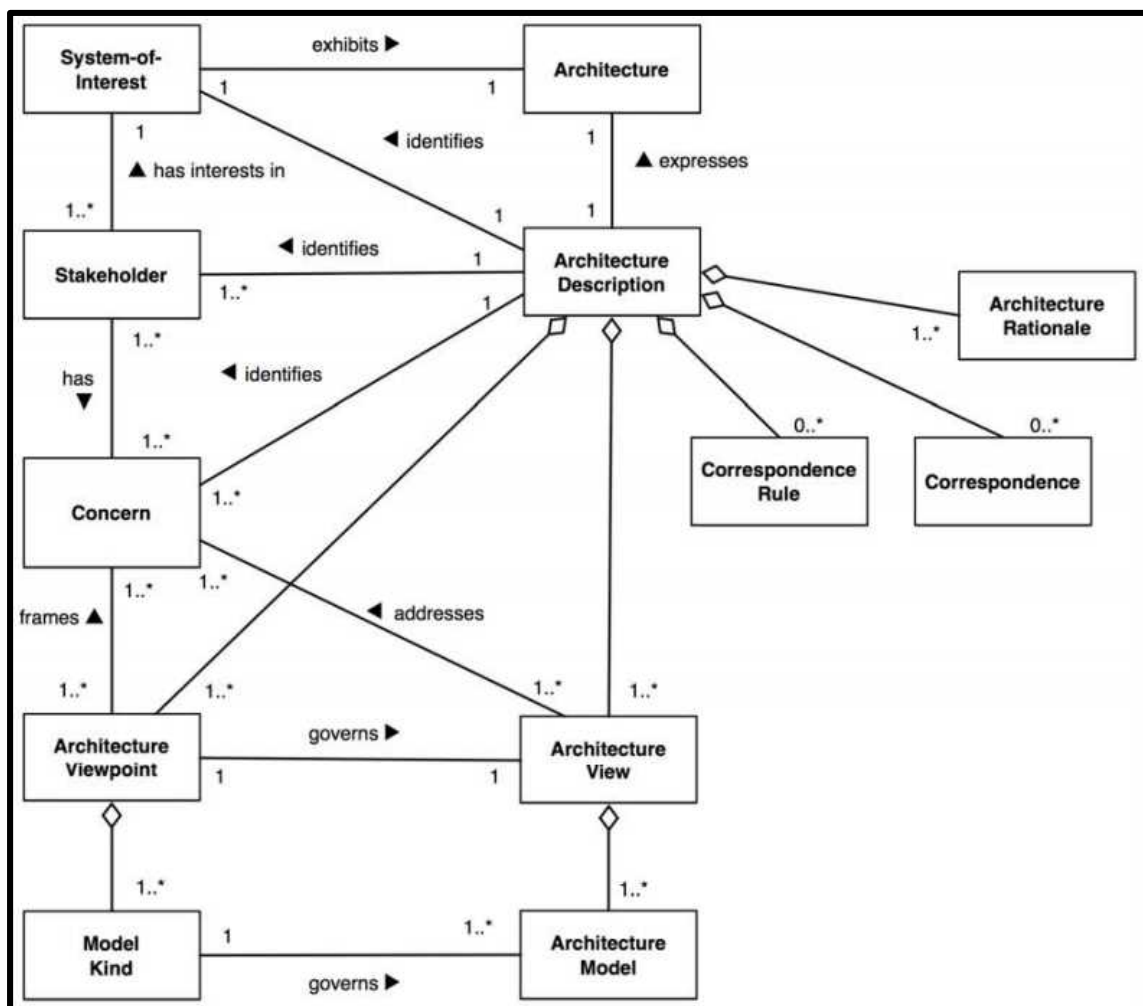


図 1 IS042010 のアーキテクチャ全体像<sup>5</sup>

本全体像を参考にしつつ、アーキテクチャの構造は、Society5.0 のアーキテクチャと、具体的なアーキテクチャ作成の取組が進んでいる我が国におけるデータ戦略のアーキテクチャを軸に整理していきます。

<sup>4</sup> <https://www.iso.org/standard/50508.html>

<sup>5</sup> <http://www.iso-architecture.org/ieee-1471/cm/>

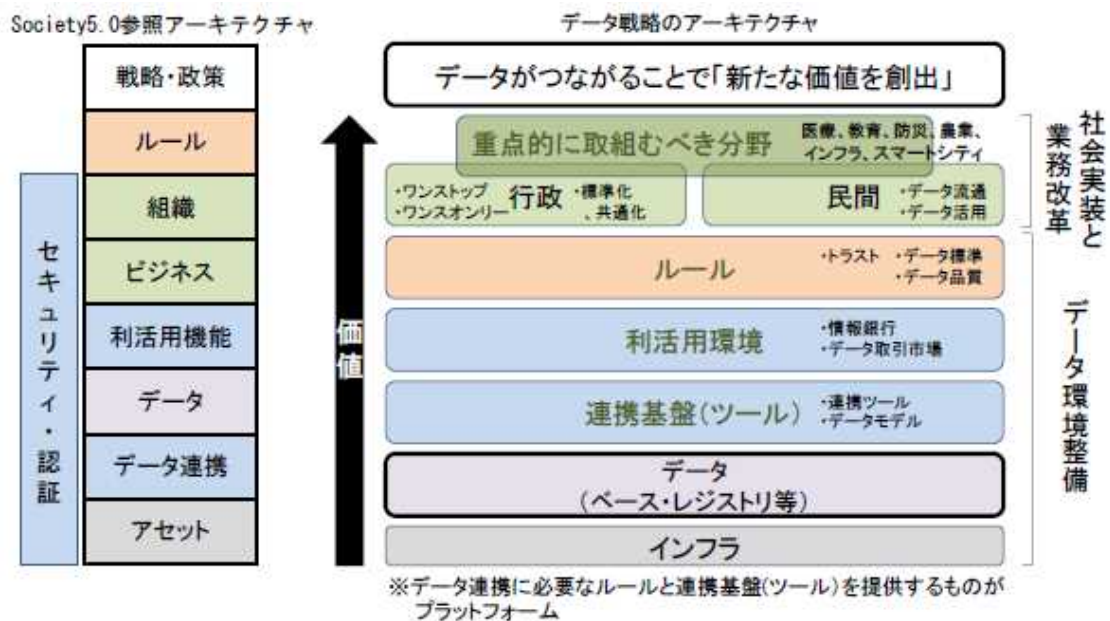


図 2 Society5.0 とデータ戦略のアーキテクチャ<sup>6</sup>

アーキテクチャモデリング言語は、ArchiMate(R)とし、そのモデリングツールとしてオープンソースのArchi<sup>7</sup>を使用します。

#### 1.4 用語の定義

- ・アーキテクチャ

ある目的を実現するための全体像を表す考え方であり、レイヤー構造を持ち、機能など目的に応じて分割したエレメント（部品）の組合せで表現します。

- ・レイヤー

アーキテクチャを構成する階層です。ルールやデータなど、検討が行いやすい単位で整理します。

- ・エレメント

アーキテクチャの個々の機能やデータなどのオブジェクトを定義する部品です。

- ・ArchiMate(R)

アーキテクチャ推進団体のオーピンググループが開発したアーキテクチャモデリング言語です。

<sup>6</sup> [https://www.kantei.go.jp/jp/singi/it2/dgov/dai10/siryou\\_a.pdf](https://www.kantei.go.jp/jp/singi/it2/dgov/dai10/siryou_a.pdf)

<sup>7</sup> <https://www.archimatetool.com/>

## 2 アーキテクチャ記述の基本要素

### 2.1 基本とするレイヤー構造

Society5.0 参照アーキテクチャをもとにした「データ戦略のアーキテクチャ」を参考に、レイヤー構造で記述します。

Society5.0 参照アーキテクチャにおける「利活用機能」は、データ取引市場のようなサービスと検索などの機能を明確に区別して記載するために「利活用環境」と「連携基盤」に分離しています。また、Society5.0 参照アーキテクチャの「データ連携」レイヤーは、エッジコンピューティングを意識したデバイスレベルでの連携用ツールや機能を記述するレイヤーとなりますが、データ戦略のように社会システムを記述する際にはエッジコンピューティングまで詳細化することが少ないため、このレイヤーは省略しています。

さらに、データ標準は「ルール」の一部となりますが、「データ」レイヤーと内容が近いため、「データ」レイヤーの下に「データ標準」レイヤーを設定しました。

以下のレイヤー構成で記述します。

- ・ 戦略
- ・ 組織
- ・ 業務（サービス）
- ・ ルール
- ・ 利活用環境
- ・ 連携基盤（ツール）
- ・ データ
- ・ データ標準
- ・ インフラ

### 2.2 各エレメント、リレーションの定義及び使用例の整理

ArchiMate(R)には多くのエレメントが定義されていますが、Stakeholder と Business Actor 等、意味の近いものが多く、すべてのエレメントを使用すると記述におけるバラツキが出て記述者にも読者にもわかりにくくなります。一方、ArchiMate(R)を導入している欧州の EIRA (European Interoperability Reference Architecture) <sup>8</sup>は、エレメントを絞りすぎておりアーキテクチャ記述として不便な面もあります。

---

<sup>8</sup> <https://joinup.ec.europa.eu/collection/european-interoperability-reference-architecture-eira>

そこで、本ペーパーは、主要なエレメント、リレーションの定義及び使用例を以下のように定めます。定義には使用可能なレイヤーを明示しました。

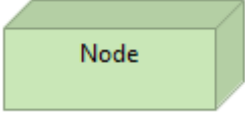

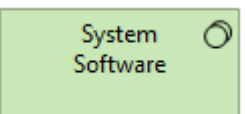
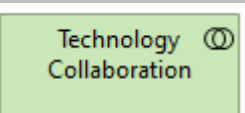
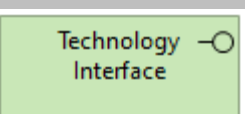
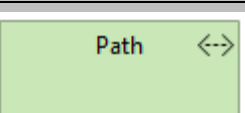
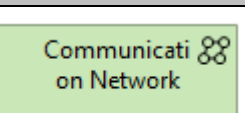
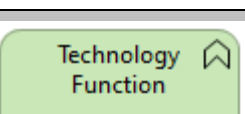
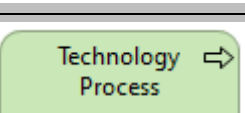
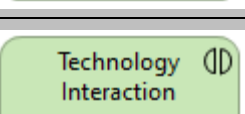
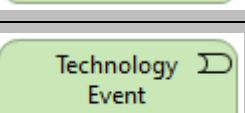
また、灰色のエレメントは、使用を推奨しないエレメントとなりますが、記述を明確にするうえで必要な場合には使用可能とします。



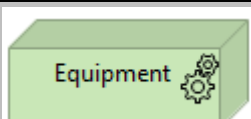
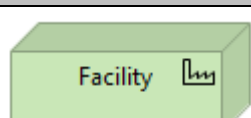
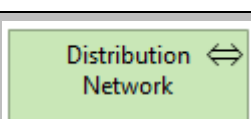
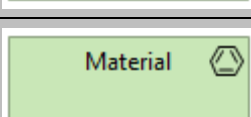
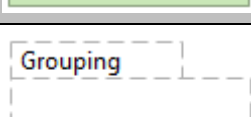
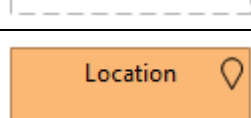

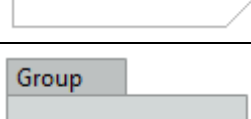
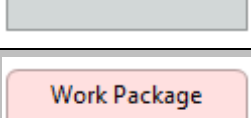
No.	エレメント	定義	使用例
1	 Stakeholder	関係者 戦略・ルールレイヤーで使用。	
2	 Driver	ゴールやアウトカム等を実現するためのドライバー 戦略・ルールレイヤーで使用。	
3	 Assessment	戦略を推進するために必要なアセスメント 戦略・ルールレイヤーで使用。	
4	 Goal	ゴール 戦略・ルールレイヤーで使用。	戦略等に明記されている目標
5	 Outcome	ゴールを実現するための具体的な成果 戦略・ルールレイヤーで使用。	
6	 Principle	原理・原則 戦略・ルールレイヤーで使用。	
7	 Requirement	要求 戦略・ルールレイヤーで使用。	法律や制度、提言等
8	 Constraint	制約 戦略・ルールレイヤーで使用。	法律や制度
9	 Meaning	意味 戦略・ルールレイヤーで使用。	
10	 Value	価値 戦略・ルールレイヤーで使用。	

No.	エレメント	定義	使用例
11		動作する人 業務（サービス）レイヤーで使用。	
12		業務ロール 業務（サービス）レイヤーで使用。	
13		※使用を推奨しないコンポーネント	
14		業務インターフェース 業務（サービス）レイヤーで使用。	端末、スマートフォン等
15		業務プロセス 業務（サービス）レイヤーで使用。	
16		業務機能 業務（サービス）レイヤーで使用。	
17		※使用を推奨しないコンポーネント	
18		※使用を推奨しないコンポーネント	
19		業務サービス 業務（サービス）レイヤーで使用。	
20		業務オブジェクト ルール・データ標準レイヤーで使用。	法律、制度で定められた標準、データモデル
21		契約 業務（サービス）レイヤーで使用。	



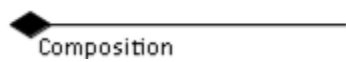
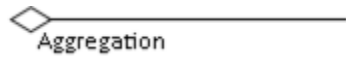
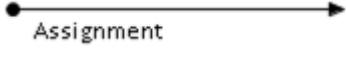
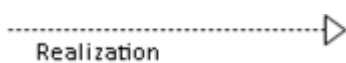

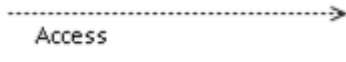




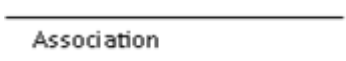
No.	エレメント	定義	使用例
22		表現 業務（サービス）レイヤーで使用。	申請、証明 データ
23		※使用を推奨しないコンポーネント	
24		内部サービス 連携基盤（ツール）レイヤーで使用。	
25		※使用を推奨しないコンポーネント	
26		インターフェース 連携基盤（ツール）レイヤーで使用。	API
27		※使用を推奨しないコンポーネント	
28		※使用を推奨しないコンポーネント	
29		※使用を推奨しないコンポーネント	
30		※使用を推奨しないコンポーネント	
31		外部サービス、サービス単独で 動作するもの 利活用環境レイヤーで使用。	認証、決済
32		データオブジェクト データレイヤーで使用。	交通データ、 登記データ

No.	エレメント	定義	使用例
33		ノード インフラレイヤーで使用。	端末やセン サー
34		※使用を推奨しないコンポーネント	
35		※使用を推奨しないコンポーネント	
36		※使用を推奨しないコンポーネント	
37		※使用を推奨しないコンポーネント	
38		※使用を推奨しないコンポーネント	
39		ネットワーク インフラレイヤーで使用。	5G
40		※使用を推奨しないコンポーネント	
41		※使用を推奨しないコンポーネント	
42		※使用を推奨しないコンポーネント	
43		※使用を推奨しないコンポーネント	

No.	エレメント	定義	使用例
44		※使用を推奨しないコンポーネント	
45		※使用を推奨しないコンポーネント	
46		※使用を推奨しないコンポーネント	
47		設備 インフラレイヤーで使用。	
48		※使用を推奨しないコンポーネント	
49		※使用を推奨しないコンポーネント	
50		パック化されたコンセプト アルなまとめ 全レイヤーで使用。	
51		場所 ルールレイヤーで使用。	地域
52		メモ 全レイヤーで使用。	
53		レイヤーの枠、類似の分類 全レイヤーで使用。	
54		※使用を推奨しないコンポーネント	

No.	エレメント	定義	使用例
55		※使用を推奨しないコンポーネント	
56		※使用を推奨しないコンポーネント	
57		※使用を推奨しないコンポーネント	
58		ギャップ（現状とのギャップを記入） 全レイヤーで使用	
59		リソース 組織レイヤーで使用	府省 人
60		※使用を推奨しないコンポーネント	
61		※使用を推奨しないコンポーネント	
62		行動指針 組織レイヤーで使用	ロードマップ、方針

また、リレーションは、以下を使用し、わからない場合には Association で関連付けます。

No.	リレーション	定義
1	 Composition	コンポジット (UML と同義) 「全体」と「その全体の部分」の関係 全体エレメントの中に部分エレメントを入れて表現することを推奨。
2	 Aggregation	データ等を収集するときに使用する。
3	 Assignment	機能から、リソースやアセットをアサインするときに使用する。
4	 Realization	実現 ルール・データ標準レイヤーのエレメントの実現を表現。(例えば、「データ標準」の実現は「ベースレジストリ」)
5	 Serving	別途整理 (Association を利用)
6	 Access	データ等にアクセスするときに使用する。
7	 Influence	別途整理 (Association を利用)
8	 Trigger	別途整理 (Association を利用)
9	 Flow	データの流れを示す。
10	 Specialization	汎用的な要素を目的などに特化することを示す。(施設→学校 等)
11	 Association	エレメント同士の関連を示す。

### 3 アーキテクチャの設計

#### 3.1 アーキテクチャ設計の準備

アーキテクチャを作成するために必要となる、対象分野の戦略やビジョン、システムやサービスなどの情報を収集します。

また、何を参照して資料を作成したかということを明確にするために、アーキテクチャのモデルの左上部分に主要な参照ドキュメントを明記します。

#### 3.2 アーキテクチャ設計の流れ

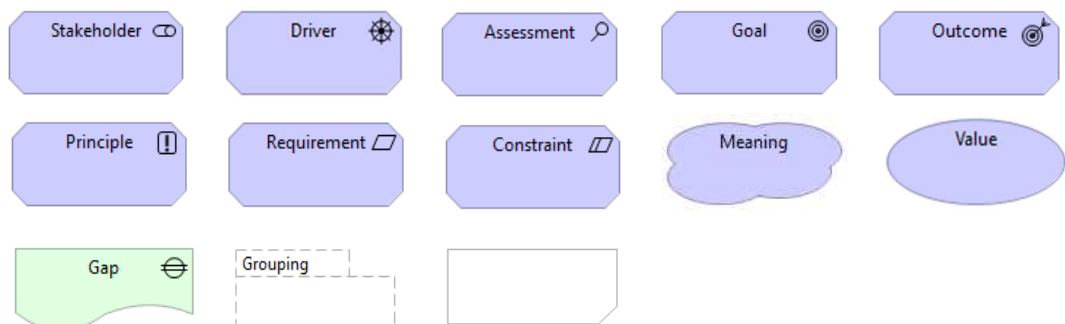
アーキテクチャの設計は、戦略目標からトップダウンで、以下の流れで実施します。

1. 戦略や目的、生み出す価値や、それを目指すうえでの原則を明確にします。
2. 組織、データ等、わかっているものからエレメントを各レイヤーに記述していきます。リレーションは同時に記述していくのではなく、各エレメントを列挙した後に記述した方が効率的です。一方、関連性が明確なものはエレメントと同時に記述しても構いません。また、必要に応じてグルーピング機能を使ってエレメントをまとめていきます。
3. 概要が記述できたら、各エレメント間の関係を整理しながらモデル上の配置を調整していきます。この際、必要に応じて業務改革の可能性も検討していきます。
4. 概要のアーキテクチャができたところで各エレメントの細分化を図り、ルールなどの不足した部分を追記していきます。また、現状とギャップがある場合には、そのギャップなどを注記していきます。

##### 1) 戦略レイヤー

アーキテクチャが何を目指しているかを明確にします。

##### a) 利用エレメント

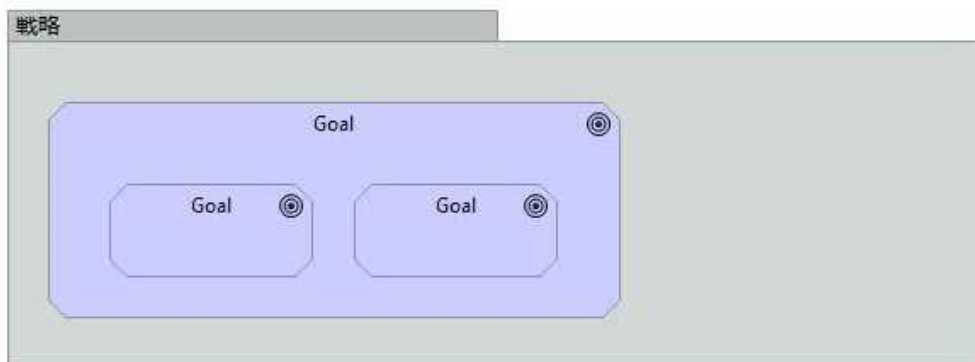


## b) 記述の流れ

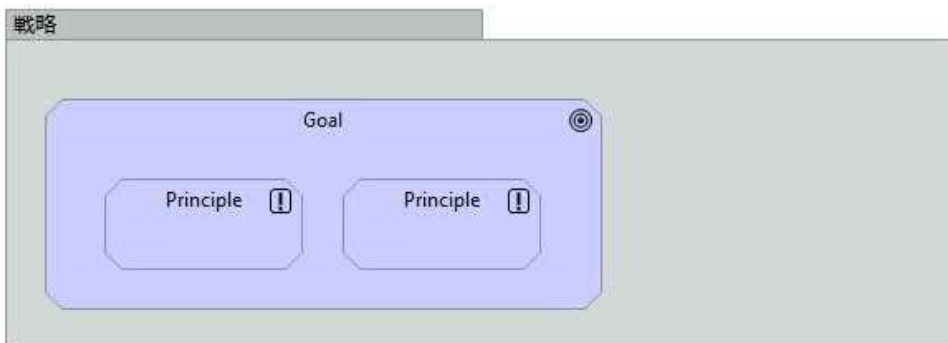
1. 戦略目標である Goal を確定します。その際、サブ Goal がある場合には、その内容も記述します。
2. Goal が生み出す価値である Value を明確にします。
3. Stakeholder を明確にします。
4. 原則である Principal がある場合にはそれを明確にします。
5. 制約条件があるときには Constrain を明確にします。
6. その他必要項目を明確にします。

## c) 記述例

- ・ 大きな政策（ゴール）の中に複数のゴールがある場合



- ・ 大きな政策（ゴール）の中に複数のプリンシプルがある場合



## 2) 組織レイヤー

アーキテクチャに関係する主要な組織を明確にします。組織レイヤーでは、関連性より「全体としての特徴」を意識します。（エレメントの名称重複可）  
ここで組織名が出ていない場合でも、機能などを説明するために Business Actor（業務（サービス）レイヤー）や Stakeholder（戦略・ルールレイヤー）

として関連付けして図中に記述されることはあります。

### a) 利用エレメント

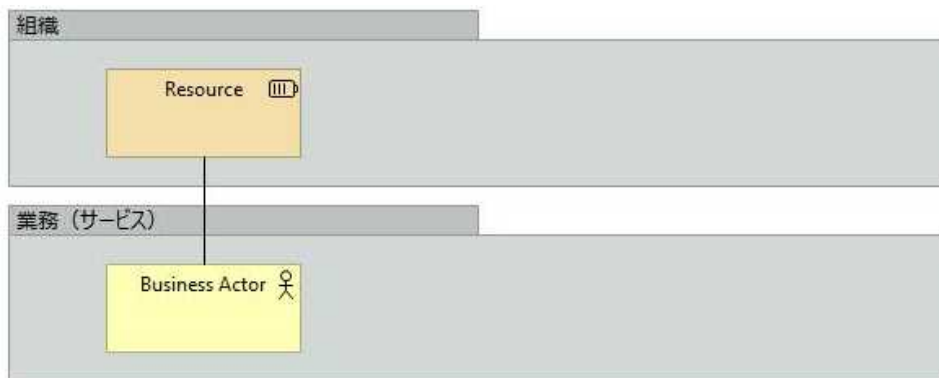


### b) 記述の流れ

1. アーキテクチャに関する組織を Resource として明記します。
2. 組織内に手順や活動指針がある場合には、Course of Action として明記します。
3. 必要に応じて業務（サービス）のレイヤーに記述される Business Actor と関連付けます。

### c) 記述例

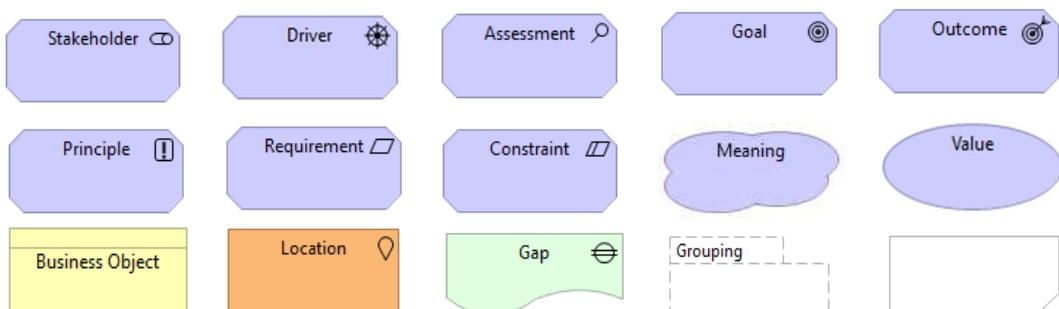
- ・組織レイヤーの Resource と業務（サービス）レイヤーの Business Actor の場合



## 3) ルールレイヤー

アーキテクチャに関連した法や制度に関する事項を記述します。主に要求事項の Requirement や制約条件の Constraint で記述されます。

### a) 利用エレメント





## b) 記述の流れ

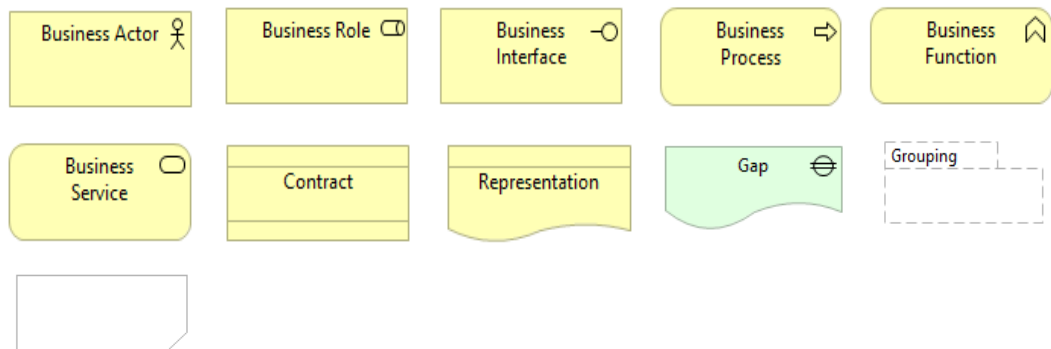
1. ルールは包括的に多くのエレメントに関連することが多いことから備忘録的にエレメントを記述していきます。
2. 業務やサービスを行ううえでクリティカルに関係がある場合には、関係エレメントとのリレーションを明記します。
3. 制度的な課題がある場合には、課題を Gap として明記します。

## 4) 業務（サービス）レイヤー、利活用環境レイヤー、連携基盤（ツール）レイヤー

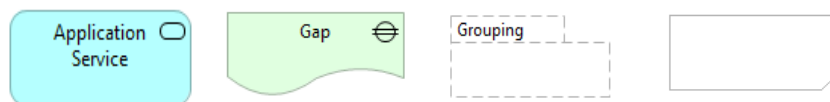
この3レイヤーは、業務からそれを実現するアプリケーションまでを表します。そのため一体で検討されることが多くなります。

### a) 利用エレメント

#### ・業務（サービス）レイヤー



#### ・利活用環境レイヤー



#### ・連携基盤（ツール）レイヤー



## b) 記述の流れ

1. 業務やサービスといった観点から、そこに関係する人である Business Actor や、何を介してアクセスするかという Business Interface と業務サービスである Business Service を明確にします。業務サービスの中には、機能や役割や業務プロセスを含むことがあります。
2. このサービスを実現するための Application Service (Web サービス

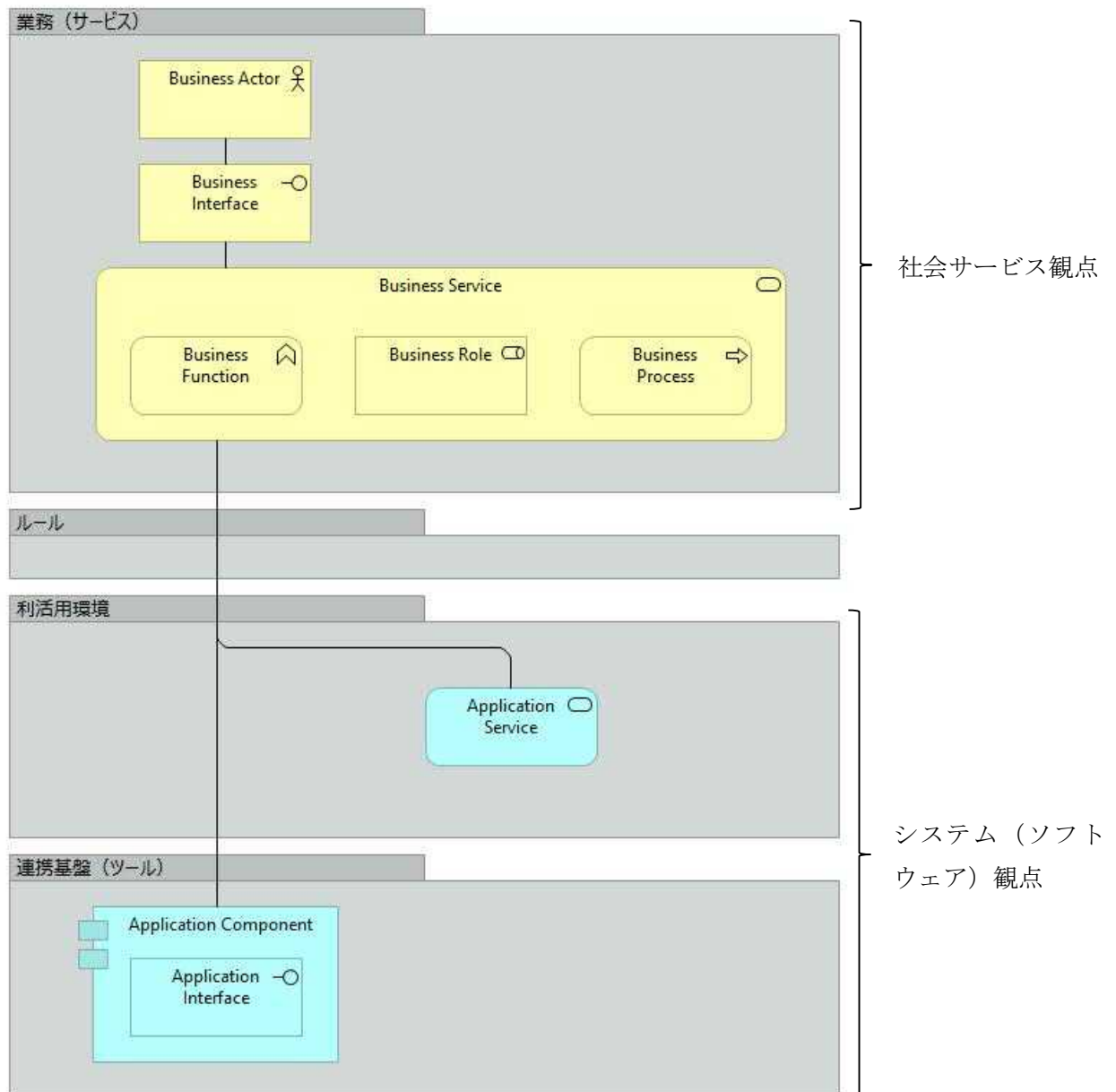
を含む) を明確化します。

3. そのアプリケーションを実現するための機能である Application Component を明確にします。

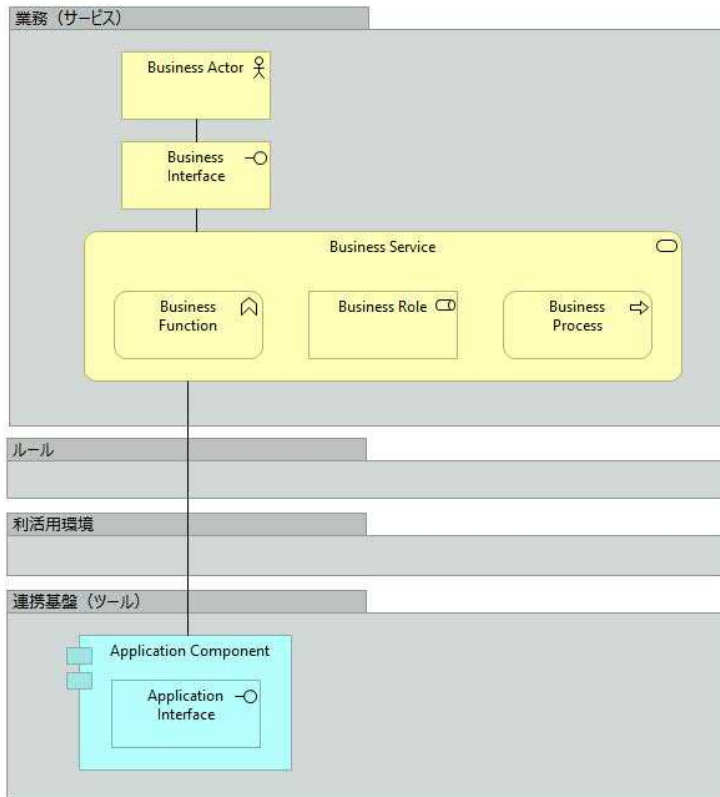
※上記の 1 で社会のサービスという観点で整理を行い、2、3 でシステム (ソフトウェア) の観点で整理を行います。

### c) 記述例

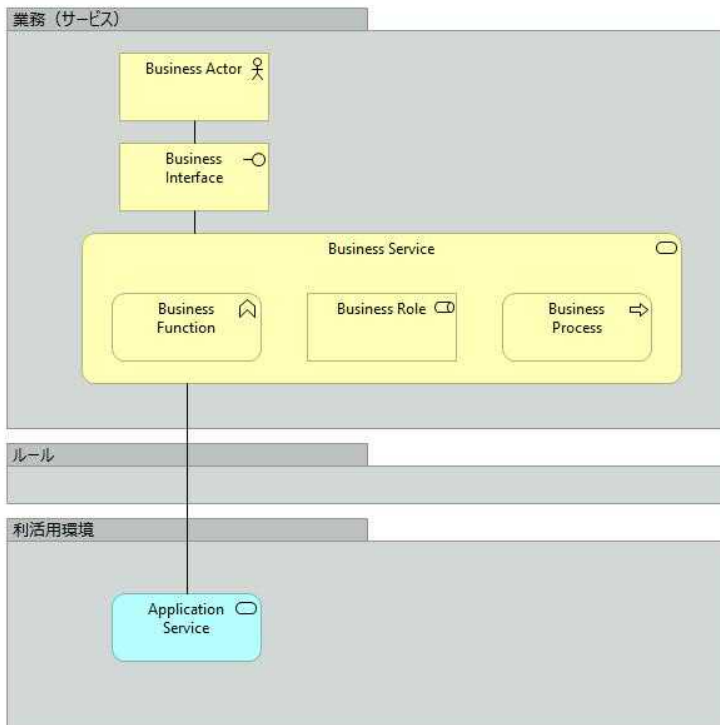
- 一般的な記述を下記に示します。  
(設計対象により途中の要素を省略可)



- Business Service の実現システムが Application Component の場合



- Business Service の実現システムが Application Service の場合



## 5) データレイヤー、データ標準レイヤー

データ間の関係性やデータ標準との関係性を明確にします。

### a) 利用エレメント

- ・データレイヤー



- ・データ標準レイヤー



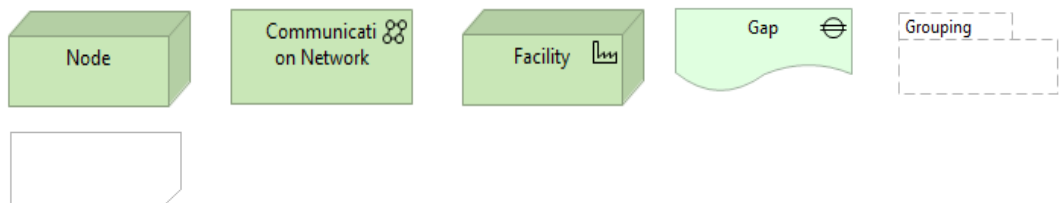
### b) 記述の流れ

1. 業務で使っているデータを明確にします。
2. 関係しそうなデータ標準を明確にし、必要に応じてデータと関連付けます。
3. ベースレジストリは、エレメントの文字を赤色（赤文字）で表現します。

## 6) インフラレイヤー

インフラレイヤーは、多くの機能やサービスの基盤となります。そのためリレーションを記述しないで各要素のみ記述しても構いません。

### a) 利用エレメント

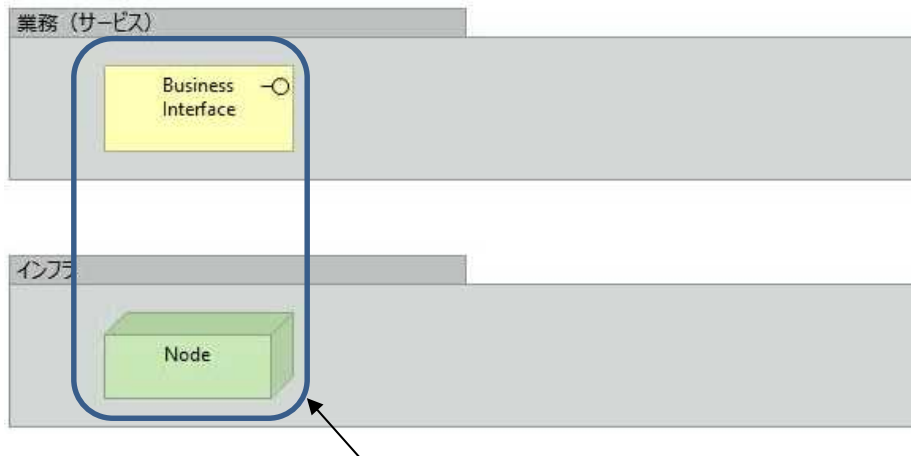


### b) 記述の流れ

1. アセットやインフラを列挙します。
2. アセットやインフラに関する Gap を明確にします。  
※インフラは全般的に使用されることが多いため、無理にリレーションを張る必要はありません。

c) 記述例

- ・インフラレイヤーのノードと業務インターフェースの場合



エレメント同士の関係性をリレーションにより示さない場合でも、縦のラインを合わせることで全体の関係性が分かりやすくなります。

## 4 アーキテクチャの利用

### 4.1 全体の関係性の整理

アーキテクチャ作成にモデリングツールを使うことで全体の整合性を確認することができます。本ガイド作成には Archi をモデリングツールとして使用しているため、下図において画面下部のリレーションのように、各要素間の関係性を確認することが可能になります。

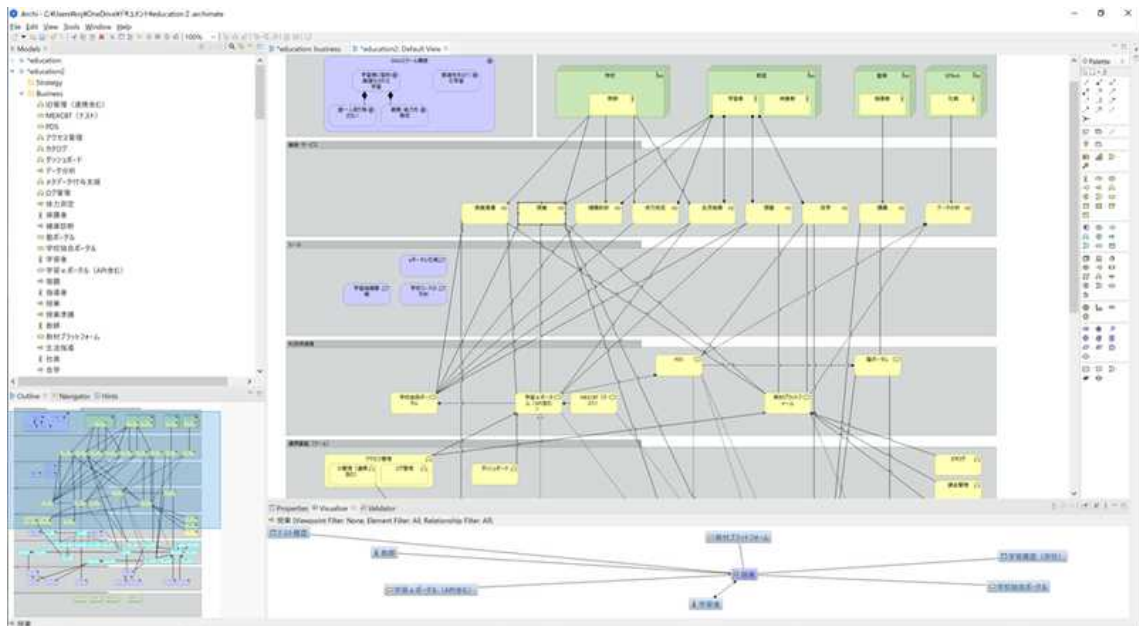


図 3 モデリングツール画面での全体の関係性の確認

## 4.2 戦略の整合性確保

戦略の文書等をアーキテクチャとして記述することで、戦略間の補完関係や連携するポイントが明確になります。

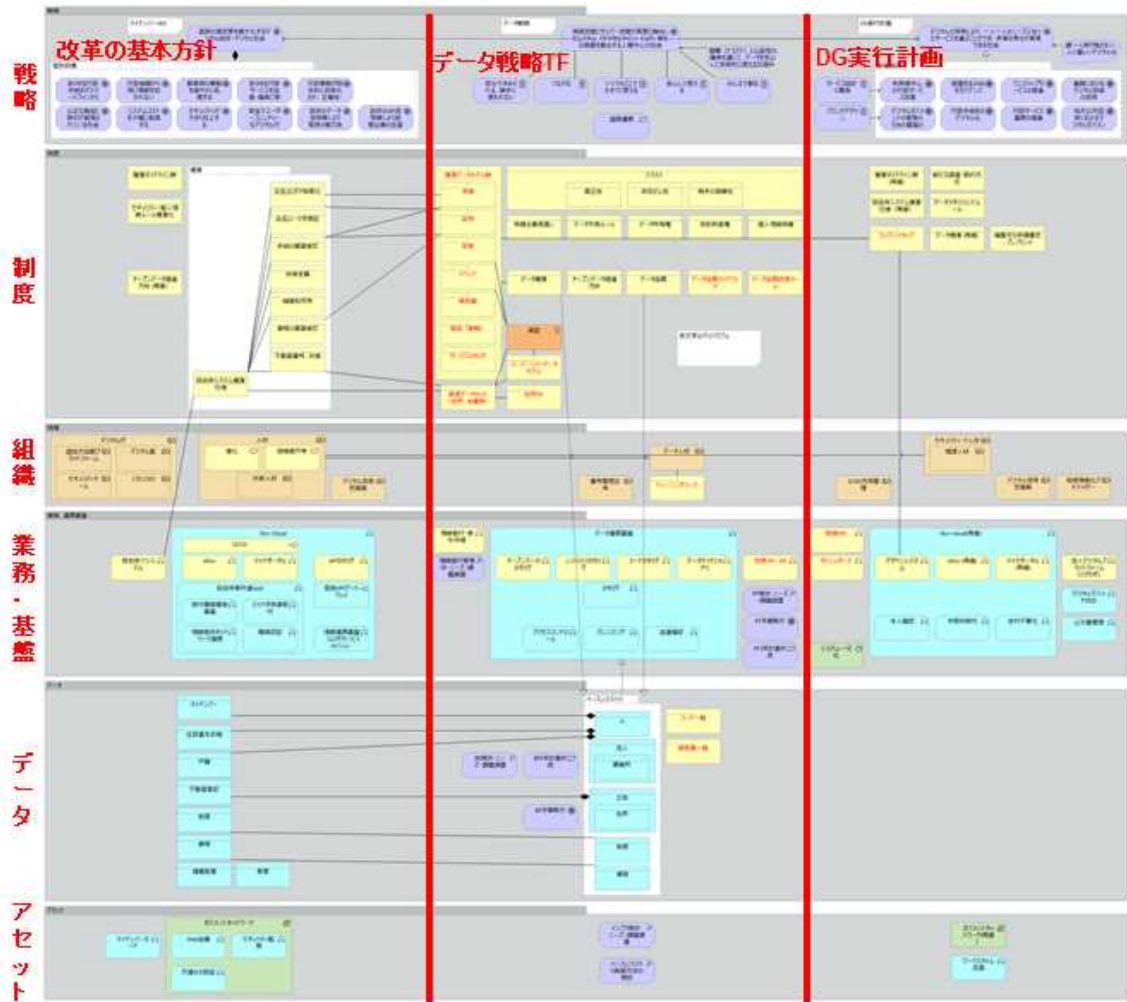


図 4 アーキテクチャを活用した複数の戦略の整合性の確保

### 4.3 プラットフォーム間の共通化や連携の検討

各分野のアーキテクチャを並べることで共通部品化が可能な部分や連携すべきポイントが明確になります。

	行政	スマートシティ	防災	教育	医療	農業	インフラ
戦略							
業務							
ルール							
利活用環境	<ul style="list-style-type: none"> <li>共通の検討課題を提示</li> <li>差分からのフィードバック</li> <li>共通モデルの提示</li> </ul>						
連携基盤、ツール			情報交換PF	教材交換PF		WAGRI	
データ				学校モデル 教材モデル			
データモデル							
インフラ							

図 5 アーキテクチャを活用したプラットフォーム間の共通化や連携の検討



## 5 アーキテクチャの詳細化

アーキテクチャの各レイヤーは、検討を深めたり実装を行ったりするために詳細化が行われます。

ArchiMate(R)で詳細化することも可能ですが、より専門的に記述するためには専門のモデリング手法を活用した方が正確かつ効率的にモデル化することができます。

機能モデリングは ArchiMate(R)で行い、業務プロセスモデリングに BPMN<sup>9</sup>、データモデルは UML のクラス図<sup>10</sup>で展開していくこととします。

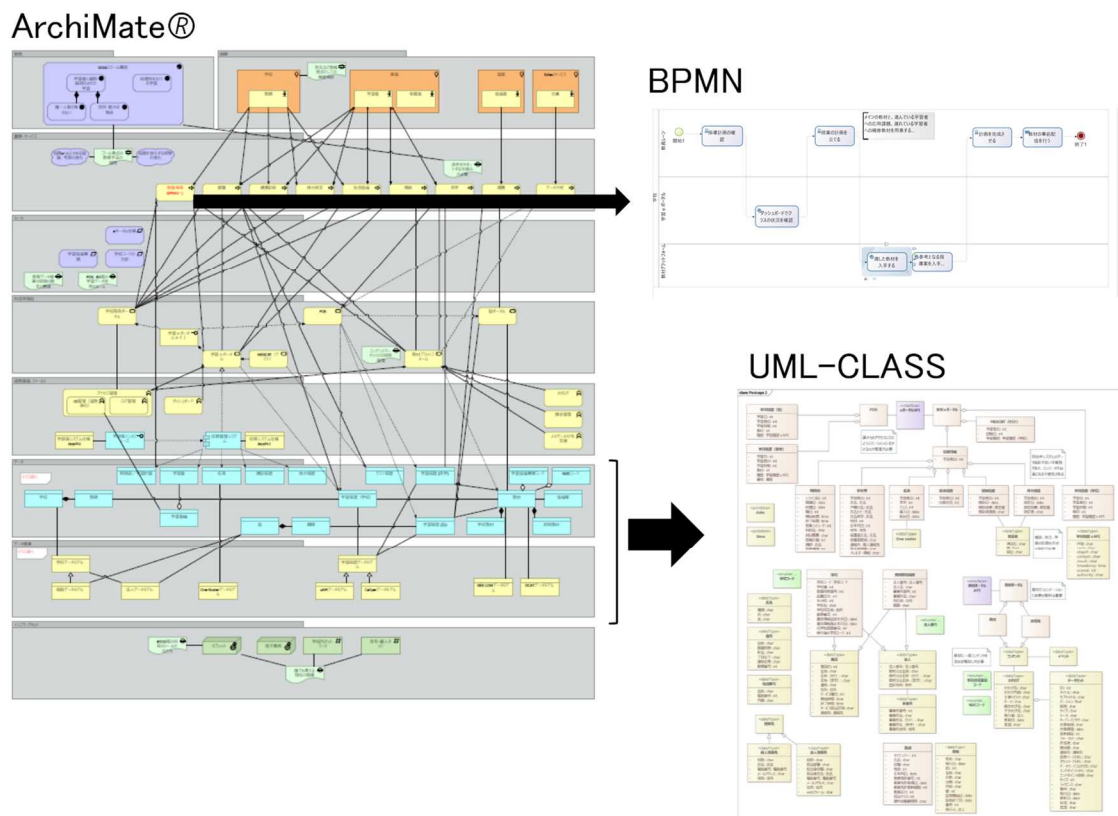


図 6 アーキテクチャの詳細化

<sup>9</sup> Business Process Model and Notation (ビジネスプロセス・モデルと表記法) の略で、世界で最も普及している業務プロセス表記標準。

<sup>10</sup> 世界で最も普及しているデータモデルの表記標準であり、本アーキテクチャのデータモデルの記述にクラス図を用いる。