

デジタルツールのデータ連携における標準化に関する調査業務

国内外主要仕様の比較表

2026年3月16日

v1.0

全国旅館ホテル生活衛生同業組合連合会

目次

目次	2
0. 本章の位置づけ	7
1. エグゼクティブサマリー	7
1.1 調査対象規格の概要	7
1.2 主要な発見事項.....	8
(1) 料金体系の対応状況.....	8
(2) 技術世代の差異.....	8
(3) 包括性とシンプルさのトレードオフ	9
(4) 対象レイヤーの違い.....	9
1.3 相互運用性の実現可能性	10
2. OpenTravel Alliance (OTA)	10
2.1 世代別サマリー	10
2.2 規格の基本情報.....	11
2.2.1 OTA 2.0 (Object Suite)	11
2.2.2 OTA 1.0 (XML Message Suite 2024A)	11
2.3 技術仕様の詳細.....	12
2.3.1 OTA 2.0 (Object Suite)	12
2.3.2 OTA 1.0 (XML Message Suite)	12
2.4 データ構造の詳細分析.....	13
2.4.1 OTA 2.0 (モデル構造)	13
2.4.2 OTA 1.0 (メッセージ構造)	13
2.5 拡張性の仕組み.....	14
2.5.1 OTA 2.0 (Object Suite)	14
2.5.2 OTA 1.0 (XML Message Suite)	14
2.6 日本市場との適合性分析	15
2.7 技術的な長所と短所	16
OTA 2.0 (Object Suite)	16
OTA 1.0 (XML Message Suite)	16
2.8 OTA コードリストについて.....	17
コードリストの概要.....	17

継続的な更新と拡張性	17
コードリストの入手方法	18
日本市場への適用	18
2.9 客室タイプと ABS (Attribute-Based Selling)	19
ABS の概要.....	19
OTA 2.0 のデータモデルと ABS	20
ABS の業務シナリオ (HTNG / HEDNA)	20
日本市場への適用における留意点.....	21
まとめと次章への接続.....	21
3. TravelXML	22
3.1 規格の基本情報.....	22
3.2 策定背景と設計思想	22
策定の経緯	22
3.3 技術仕様の詳細.....	22
データ交換形式.....	22
スキーマ構成.....	23
3.4 日本固有の特徴.....	23
1. 3 者間取引構造.....	23
2. 旅館とホテルの明確な区別	23
3. 日本固有項目の完全対応	24
4. 部屋タイプの固定分類	25
3.5 技術的な長所と短所	25
長所	25
短所	25
4. GIF (政府相互運用性フレームワーク)	26
4.1 規格の基本情報.....	26
4.2 策定背景と設計思想	26
4.3 技術仕様の詳細.....	26
データ交換形式.....	26
フラットテーブル設計の特徴	27
4.4 主要エンティティの構成	27
主要エンティティ (13 種類)	27

データ型定義（4種類）	28
4.5 日本市場との適合性分析	28
GIF 特有の日本対応項目	28
OTA・国際標準との構造的差異	29
4.6 技術的な長所と短所	29
長所	29
短所	29
4.7 宿泊業での活用可能性.....	30
5. HTNG Express	30
5.1 規格の基本情報.....	30
5.2 策定背景と設計思想	31
5.3 技術仕様の詳細.....	31
データ交換形式.....	31
エンティティ構成	31
5.4 データ構造の特徴.....	32
Reservation エンティティの構造.....	32
列挙型の3分類.....	32
5.5 日本市場との適合性分析	33
適合する点	33
適合しない点・要調整項目	33
5.6 技術的な長所と短所	34
長所	34
短所	34
6. HTNG Kiosk	35
6.1 規格の基本情報.....	35
6.2 策定背景と設計思想	35
6.3 技術仕様の詳細.....	35
データ交換形式.....	35
仕様構成	35
6.4 データ構造の特徴.....	36
OTA 準拠のメッセージ構造.....	36
属性のプレフィックス表現.....	36

6.5 日本市場との適合性分析	36
主な課題	36
6.6 技術的な長所と短所	37
長所	37
短所	37
7. 包括的な比較分析	38
7.1 技術仕様の比較マトリクス	38
7.2 日本市場適合度の詳細分析	39
料金体系対応	39
食事プラン対応	40
日本固有項目の対応状況	41
子供料金の商習慣の違い（重要）	42
7.3 技術世代による分類	43
7.4 相互運用性の実現可能性	44
データ項目のマッピング可能性	44
7.5 主要データ項目のマッピング参考表	44
予約基本情報	44
ゲスト情報	47
施設基本情報	48
料金情報	51
8. まとめ	54
8.1 各規格の特徴整理（6規格をOTA統合で5分類）	54
OpenTravel Alliance（OTA）	54
TravelXML	55
GIF（政府相互運用性フレームワーク）	55
HTNG Express	55
HTNG Kiosk	56
8.2 技術的視点からの主要な知見	56
1. 料金体系の国際標準と日本市場の関係	56
2. 技術世代の移行	57
3. GIFの特殊な位置づけ	57
4. 対象レイヤーの違いの認識	57

8.3 次章への接続	58
参考資料	58
OpenTravel Alliance	58
TravelXML	59
GIF（政府相互運用性フレームワーク）	59
HTNG Express	59
HTNG Kiosk.....	59
その他参考資料.....	59

0. 本章の位置づけ

本章は、観光庁「デジタルツールのデータ連携における標準化に関する調査業務」における基礎調査編として、国内外の主要な宿泊業界関連データ規格を技術的観点から比較分析したものです。

本章の役割:

- 各規格の技術仕様と特徴の正確な把握
- 規格間の客観的な比較分析
- 後続章（日本の宿泊業界実態とのギャップ分析、標準項目定義に向けた設計指針）の判断基礎の提供

後続章との関係:

- 第II章「日本の宿泊業界実態とのギャップ分析」: 本章で整理した各規格の特徴を踏まえ、日本市場との適合性を詳細分析
- 第III章「標準項目定義に向けた設計指針」: 本章と第II章の分析結果を統合し、実用的な標準化の方向性を提示

1. エグゼクティブサマリー

1.1 調査対象規格の概要

本調査では、宿泊業界に関連する以下6つの主要規格を対象としました。

規格名	策定主体	対象地域	データ形式	主な用途
OpenTravel Alliance 2.0 (Object Suite)	OpenTravel Alliance	グローバル	JSON / XML	REST/WSDL による旅行データ連携
OpenTravel Alliance 1.0 (OTA XML Message Suite)	OpenTravel Alliance	グローバル	XML/SOAP	予約・在庫・通知メッセージ交換
TravelXML	日本旅行業協会	日本国内	XML	B2B パッケージ取引
GIF	デジタル庁	日本国内	CSV	行政システム連携
HTNG Express	HTNG	グローバル	JSON/REST	館内サービス連携
HTNG Kiosk	HTNG	グローバル	XML/SOAP	キオスク-PMS 連携

1.2 主要な発見事項

(1) 料金体系の対応状況

各規格は異なる料金体系をサポートしています。

【OTA 1.0 / OTA 2.0】

対応方式: Room Charge 方式と Personal Rate 方式の両方に対応

実装方法: ChargeType (CHG) コードリストで per person / per room 等の課金単位を選択可能

備考: - 海外では主に Room Charge が使われるが、オールインクルーシブ等で per person も使用される
 - 日本市場の 1 名料金にも標準で対応可能
 - AdditionalGuestAmount/@Percent でパーセンテージ表記も可能 (年齢情報も併せて指定可能)

【HTNG】

対応方式: Room Charge 方式が基本

実装方法: 拡張フィールドで Personal Rate 対応

【TravelXML】

対応方式: Personal Rate 方式 (1 人あたり料金)

実装方法: ネイティブサポート

備考: 日本の商習慣 (1 人あたり料金、食事付きプラン) に完全対応

重要: OTA は両方の料金体系に標準対応しているため、日本市場の 1 名料金方式にも拡張なしで対応可能です。

(2) 技術世代の差異

策定期間により、採用技術が大きく異なります。

第 1 世代 (2000 年代策定) : XML/SOAP

- OpenTravel Alliance 1.0 (XML Message Suite)
- TravelXML
- HTNG Kiosk

第 2 世代 (2020 年代策定) : JSON/REST、CSV

- HTNG Express
- GIF

世代横断 (モデル駆動設計) :

- OpenTravel Alliance 2.0 (Object Suite)

OTA 2.0 はモデル駆動設計を採用しており、REST/JSON および WSDL/XML の両方式に対応しています。新しい規格ほど実装が容易で、開発者の学習コストが低い傾向にあります。

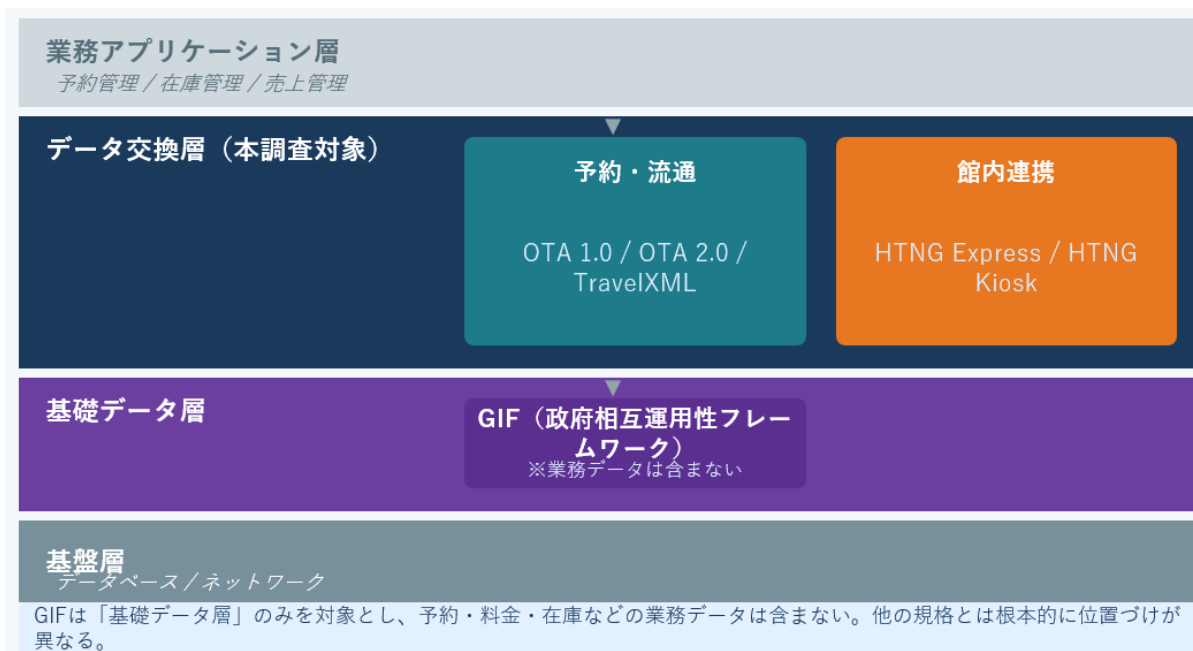
(3) 包括性とシンプルさのトレードオフ

規格	カバー範囲	複雑さ	特徴
OTA 2.0	旅行業界全体（モデル駆動）	高	Object Suite 基盤・REST/WSDL・JSON/XML 両対応
OTA 1.0	旅行業界全体（メッセージ）	高	SOAP ベースで成熟度高
TravelXML	日本 B2B 取引	中	日本商習慣に特化
GIF	基礎データのみ ※	低	シンプルだが業務データ不足
HTNG Express	館内サービス	低	軽量で実装容易
HTNG Kiosk	キオスク連携	高	特定用途に特化

※ GIF は基礎データ（住所、個人情報、施設情報）のみを対象とし、予約・料金・在庫等の業務データは対象外

(4) 対象レイヤーの違い

各規格は異なるレイヤーを対象としています。



1.3 相互運用性の実現可能性

各規格間での概念的なマッピングは可能ですが、直接的なデータ変換には以下の課題があります。

技術的課題:

- データ形式の変換 (XML ⇔ JSON ⇔ CSV)
- 階層構造の違いの吸収

ビジネスロジック的課題:

- 料金体系の変換ロジック構築 (Room Charge ⇔ Personal Rate)
- 日本固有概念の拡張定義 (和室、2食付きなど)
- コード体系の標準化とマッピングテーブルの整備

2. OpenTravel Alliance (OTA)

2.1 世代別サマリー

OTA は XML メッセージスイートを中心とする「1.0 系」と、モデル駆動設計を採用した「2.0 系」が併存している。両者は設計思想が異なる独立した仕様体系であり、日本市場での技術検討では双方の特性を正確に理解する必要がある。

OTA 1.0 と OTA 2.0 の関係:

- OTA 2.0 と OTA 1.0 は、設計思想が異なる独立した仕様体系である
- 共通のコードリストや業界概念を共有しているが、データ表現方法やモデル構造は異なる
- OTA 1.0 は現在も独立した仕様として有効であり、2024A として継続的に更新されている
- 既存システムは 1.0 に対応しているケースが多いため、両仕様への対応が必要なケースもある
- 例：OTA 1.0 では客室タイプをコード表 (数値コード) で表現するが、OTA 2.0 ではテキスト string 内に客室タイプ情報を記述でき、コード表を参照せずに識別できる

項目	OTA 2.0 (Object Suite)	OTA 1.0 (XML Message Suite 2024A)
インタフェース	REST/WSDL による API・サービス構築 (JSON/XML 両対応)	SOAP/HTTP によるメッセージ交換 (XML のみ)
代表バージョン	Object Suite Libraries: Common 5.0.0 / CodeList 4.0.0 / Hospitality 2.0.0	2024A (2024 年 12 月リリース)
モデリング手法	OTM モデル駆動設計 (OpenAPI \$ref / XSD include/import)、ExtensionPoint	include/import、ComplexType 継承、TPA_Extensions
メッセージ構成	モデル→API 生成 (Order Library と連携)	RQ/RS/Notif を含む 87 種のメッセージ
想定活用	クラウド/マイクロサービス連携、REST/WSDL API (JSON/XML)	既存 GDS・PMS 連携、XML 資産の継続利用

2.2 規格の基本情報

2.2.1 OTA 2.0 (Object Suite)

****出典に関する注記**:** OTA 2.0 (Object Suite) の技術仕様に関する記述は、Object Suite スキーマ定義 (OpenAPI 3.0.3 形式) および OpenTravel Alliance 公式概要資料に基づく。本比較表の主たる突合対象である OpenTravel 2024A Message Users Guide は OTA 1.0 (XML Message Suite) のガイドであり、Object Suite 固有の仕様詳細 (通信方式の推奨事項、採用事例等) については別途 Object Suite 公式ドキュメントを参照されたい。

項目	内容
正式名称	OpenTravel 2.0 Object Suite
策定主体	OpenTravel Alliance (旅行業界の国際標準化団体、メンバーシップ制非営利組織)
初版策定	1999 年 (Object Suite 移行は 2020 年代)
対象バージョン	Object Suite Libraries: Common 5.0.0 / CodeList 4.0.0 / Hospitality 2.0.0
コンパイル情報	4.0-SNAPSHOT (2020 年 1 月 31 日コンパイル、FINAL)
対象地域	グローバル
策定プロセス	会員企業によるワーキンググループ合議制
主な採用事例	国際ホテルチェーン等での採用が報告されている

2.2.2 OTA 1.0 (XML Message Suite 2024A)

項目	内容
正式名称	OpenTravel Alliance 2024A XML Message Suite
策定主体	OpenTravel Alliance (旅行業界の国際標準化団体、メンバーシップ制非営利組織)
初版策定	1999 年 (現行版: 2024A / 2024 年 12 月公開)
対象バージョン	2024A (OTA2024A)
コンパイル情報	XSD 88 本、メッセージユーザズガイド、リリースノート付属
対象地域	グローバル
策定プロセス	会員主導のワーキンググループ+公開レビュー
主な採用事例	GDS、グローバル OTA、ホテルチェーンの XML 連携 (PMS/CRS)

2.3 技術仕様の詳細

2.3.1 OTA 2.0 (Object Suite)

プロトコル: HTTP/HTTPS
 通信方式: REST / WSDL
 データ形式: JSON / XML
 スキーマ定義: スキーマ定義: OpenAPI 3.0.3、OTM (OpenTravel Model)
 文字コード: UTF-8

- Common Library (v5.0.0): 258 スキーマ (Object 135 / Enum 96 / その他 27)
- CodeList Library (v4.0.0): 184 スキーマ (Object 1 / Enum 183)
- Hospitality Library (v2.0.0): 104 スキーマ (Object 56 / Enum 30 / その他 18)
- Order Library (参考): 143 スキーマ (Object 85 / Enum 38 / その他 20)
- **\$ref** で共通部品を参照し、モデル駆動で API 定義を生成。
- ExtensionPoint プロパティでベンダー拡張を許容しつつ、JSON Schema で検証可能。

2.3.2 OTA 1.0 (XML Message Suite)

プロトコル: HTTP/HTTPS
 通信方式: SOAP 1.1 / SOAP 1.2
 データ形式: XML (XSD 1.0)
 スキーマ定義: XSD (メッセージ/機能別/Industry Common/Common Types)
 名前空間: <http://www.opentravel.org/OTA/2003/05>
 文字コード: UTF-8

- メッセージレベル (87 件) ⇔機能別スキーマ⇔業界共通型⇔共通型の 4 層構造。
- Request/Response/Notification (RQ/RS/NotifRQ/NotifRS) のペア設計。
- WSDL・WS-Security・WS-I Basic Profile に関する実装ガイドラインが提供されている (WS-I Basic Profile 準拠は推奨)
- SOAP メッセージングガイドライン (2024A Message Users Guide 準拠) :
 - SOAP Body には OpenTravel メッセージを **1つだけ格納**する規約
 - **WS-I Basic Profile 準拠**を推奨 (相互運用性確保のため)
 - SOAP Action ヘッダによるオペレーション識別

2.4 データ構造の詳細分析

2.4.1 OTA 2.0 (モデル構造)

- `$ref` による正規化とライブラリ横断の再利用が特徴。
- `components/schemas/Common_5_0_0_Address` など明示的な命名で依存関係を管理。
- `ExtensionPoint` を通じて JSON オブジェクト内に日本固有情報をネスト可能。

```
{
  "RoomStay": {
    "StayDateRange": {"Start": "2025-11-15", "End": "2025-11-17"},
    "Total": {"AmountBeforeTax": 40000, "CurrencyCode": "JPY"},
    "ExtensionPoint": {
      "JPAddOn": {
        "MealPlan": "2 食付き",
        "HotSpringTax": 150
      }
    }
  }
}
```

2.4.2 OTA 1.0 (メッセージ構造)

- `OTA_HotelResRQ/RS` など深い階層構造で部屋・料金・ゲスト情報を管理。
- `OTA_HotelCommonTypes.xsd` に 70 種の ComplexType と 28 の AttributeGroup を収録。
- メッセージカテゴリ：Reservation 9、Availability 6、Inventory 14、RatePlan 7、Descriptive 6、Search 2、Event 7、RFP 9、Stats 4、その他 23。
- 包括的なコードリスト (Unique ID Type [UIT]、Age Qualifying Code [AQC]、Room View Type [RVT] 等) を提供し、国際的なコード体系を統一。

```
<OTA_HotelAvailRQ xmlns="http://www.opentravel.org/OTA/2003/05"
  EchoToken="12345"
  TimeStamp="2025-10-26T10:30:00Z"
  Version="4.001"> <!-- ※Version 属性は xs:decimal 型。2024A はリリース識別子 -->
  <AvailRequestSegments>
    <AvailRequestSegment>
      <HotelSearchCriteria>
        <Criterion>
          <HotelRef HotelCode="HOTEL001"/>
          <StayDateRange Start="2025-11-15" End="2025-11-17"/>
          <RoomStayCandidates>
            <RoomStayCandidate Quantity="1">
              <GuestCounts>
                <GuestCount AgeQualifyingCode="10" Count="2"/>
              </GuestCounts>
            </RoomStayCandidate>
          </RoomStayCandidates>
        </Criterion>
      </HotelSearchCriteria>
    </AvailRequestSegment>
  </AvailRequestSegments>
</OTA_HotelAvailRQ>
```

※ 上記 XML サンプルの AgeQualifyingCode 値 ("10"=大人) は OTA 公式サンプルに基づくが、OTA コードリスト (Age Qualifying Code / AQC) の最新版では年齢範囲ベースのコード体系 (AQC 1: Over 21、AQC 4: Under 12 等) が定義されている。OTA 2.0 ではテキスト Enum ("Adult"、"Child"等) を使用する。実装時は最新コードリストを参照のこと。

2.5 拡張性の仕組み

2.5.1 OTA 2.0 (Object Suite)

- **ExtensionPoint** プロパティで JSON/XML 双方に拡張フィールドを追加可能。
- Enum は **_Enum** と **_Enum_Base** を分離し、**extension** 属性で外部コードを許容。
- OTM モデルに拡張を定義し再コンパイルすることで、仕様書と実装の同期を維持。

```
{
  "HotelReservation": {
    "RoomStays": {...},
    "ExtensionPoint": {
      "JapaneseExtension": {
        "MealPlan": "2 食付き",
        "RoomStyle": "和室",
        "HotSpringTax": 150
      }
    }
  }
}
```

2.5.2 OTA 1.0 (XML Message Suite)

- **TPA_Extensions** 要素で標準外ノードを追加し、ベンダー独自項目を保持。
- **AttributeGroup/ComplexType** の **extension/restriction** により継承拡張が可能。
- CodeList は OpenTravel 独自コードを基本としつつ、**@OtherType** 等で外部体系とのマッピングを許容。

2.6 日本市場との適合性分析

共通して以下の強みがある。

- 多通貨・多言語・タイムゾーン対応など、グローバル要件を標準でカバー。
- Room Type/Rate Plan/Inventory の概念が整備され、在庫管理モデルが明確。

一方で、日本固有要件は拡張が前提となる。

観点	OTA 1.0	OTA 2.0
料金モデル	Room Charge と Personal Rate 両方に標準対応。ChargeType (CHG) コードリストで課金単位を選択可能。 AdditionalGuestAmount/@Percent でパーセンテージ表記も可能。 TPA_Extensions で日本固有の詳細を追加。	Room Charge と Personal Rate 両方に標準対応。ChargeType (CHG) コードリストで課金単位を選択可能。 AdditionalGuestAmount/@Percent でパーセンテージ表記も可能。ExtensionPoint で日本固有の詳細（入湯税等）を追加。
食事プラン	Meal Plan Type (MPT) コードリスト (Full board, Dinner bed and breakfast plan 等) および AdditionalDetailType Code #43 (Meal plan description) で食事プランを表現可能。日本の食事プラン (2食付き等) にも対応。	AdditionalDetailType Code #43 (Meal plan description) で食事プランの詳細を標準で表現可能。2食付き・3食付き等の日本の食事プランにも対応。
部屋タイプ	Bed Type (BED) #7 (Tatami mats) で和室を標準表現。必要なコードは OTA コードキーパーに追加依頼可能。	Bed Type (BED) #7 (Tatami mats) で和室を標準表現。メゾネット等も対応可能 (必要なコードは OTA コードキーパーに追加依頼可能)。
住所表記	自由形式の AddressLine を組み合わせる必要がある。	Common.Address で階層表現。町丁目・建物の精度は拡張で補完。
コード拡張	CodeList + OtherType / TPA_Extensions で独自コードを保持。コードリストは常にアップデート中。	_Enum + extension で外部コード拡張が容易。コードリストは常にアップデート中。
実装知見	XML ゲートウェイに豊富な実績があり、周辺ツールも整備。	REST/WSDL および JSON/XML の両方で API 構築が可能。

共通課題:

- 入湯税・温泉税など日本固有の税項目は、OTA の Fee Tax Type コードリストへの追加依頼、または TPA_Extensions/ExtensionPoint での拡張で対応可能。なお、サービス料は FTT 14 (Service charge) として既に OTA コードリストに存在する。ただし、国内 PMS では入湯税・宿泊税を他の税項目と独立して管理することが一般的なため、本標準仕様においても独立した項目 (JBTX、JATX) として定義している。

● 子供料金の商習慣の違い:

- OTA: `AdditionalGuestAmount` で `AgeQualifyingCode` (年齢区分) と `Percent` (パーセンテージ) を柔軟に組み合わせ
- 日本: 固定料金区分 (子供 A/B/C/D) でパーセンテージが決定
- OTA の方式で表現は可能だが、日本の「子供 B」といった固定区分名には直接対応しない

2.7 技術的な長所と短所

OTA 2.0 (Object Suite)

長所

- モデル駆動設計で API 定義を生成可能。REST/JSON および WSDL/XML の両方式に対応し、モダンな開発にも既存資産との連携にも適合。
- コンポーネントの再利用性が高く、保守と拡張が行いやすい。
- DevOps/クラウド環境への適合性が高い。WSDL/XML による既存資産との互換も確保。

短所

- モデルと生成成果物の差管理が必要で、ガバナンスが課題。
- OTA 1.0 とは設計思想が異なるため、既存 1.0 資産との相互変換には別途実装が必要。
- 日本語ドキュメントや導入事例が限定的。

OTA 1.0 (XML Message Suite)

長所

- 20 年以上の運用実績と豊富なベストプラクティスが蓄積。
- SOAP/WSDL ベースで既存 PMS・GDS との互換性が高い。
- 予約・在庫・通知を網羅する細粒度メッセージが揃う。

短所

- XML が冗長で通信量が大きく、処理負荷も高い。
- SOAP/WS-*スタックの実装コストが高く、モダン開発者には敷居が高い。
- バージョンアップ時の XSD 差管理が煩雑。

2.8 OTA コードリストについて

OTA では、業界標準のコードリストを提供しており、宿泊業界での相互運用性を確保するための重要な基盤となっています。

コードリストの概要

規模と範囲:

- ホテル関係で約 90 の項目が定義されている
- 予約、料金、客室、サービスなど幅広い業務領域をカバー
- グローバルに統一されたコード体系により、異なるシステム間でのデータ交換が容易

主なコード項目の例:

- 客室タイプコード (Bed Type [BED] 等)
- 年齢資格コード (AgeQualifyingCode)
- 料金プランタイプ (RatePlanType)
- 食事プランコード (Meal Plan Type [MPT] / MealPlanType_Enum)
- 予約ステータスコード
- 支払い方法コード

継続的な更新と拡張性

最新性の維持:

- コードリストは常に更新されている
- **最新更新:** 2025 年 10 月
- 業界のニーズに応じて新しいコードが追加される

日本固有コードの追加:

- AHLA (American Hotel & Lodging Association) と連携
- AHLA が OTA のコード管理を担当
- 日本固有のデータをコードリストに追加することが可能
- 例: 和室を表す **Bed Type (BED) #7 (Tatami mats)** は既に標準コードとして追加済み
- 必要なコードがあれば OTA コードキーパー (Sandy 氏) に追加依頼が可能

コードリストの入手方法

公式ダウンロードサイト:

- URL: <https://opentravel.org/download-code-list/>
- 最新版のコードリストを随時ダウンロード可能
- 形式: XML、JSON、Excel 等の複数フォーマットで提供

推奨事項:

- コードリストは常に更新されているため、**最新版をダウンロードすることを推奨**
- 実装時には最新のコード定義を参照することで、他システムとの**互換性を確保**

日本市場への適用

OTA コードリストの特徴により、日本市場でも以下のメリットがあります:

1. **グローバル互換性:** 国際標準コードを使用することで、海外 OTA とのシームレスな連携が可能
2. **日本固有対応:** 必要に応じて日本固有のコードを追加できる柔軟性
3. **長期的な保守性:** AHLA/OTA による継続的な管理により、長期的な安定性が確保

2.9 客室タイプと ABS (Attribute-Based Selling)

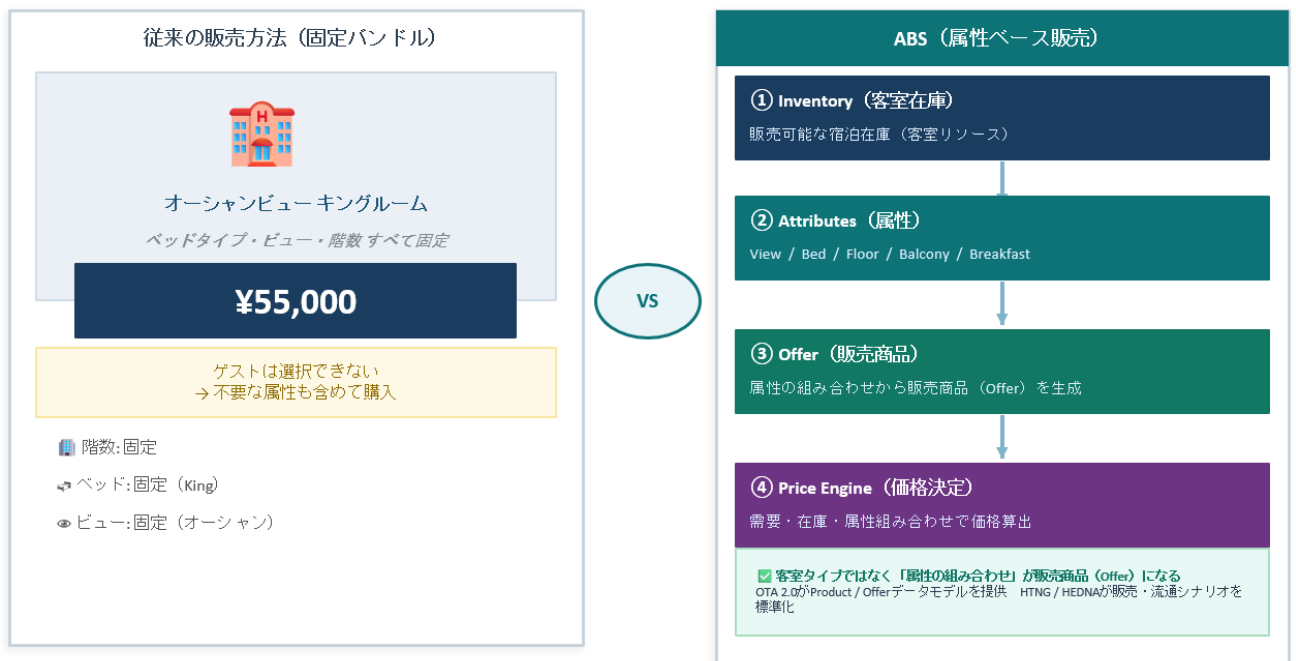
近年、ホテル業界では客室の販売方法に関する新しいアプローチとして ABS (Attribute-Based Selling: 属性ベース販売) が注目されています。

ABS の概要

定義:

- Attribute-Based Selling (ABS) とは、客室タイプを固定的な「部屋タイプ」として販売するのではなく、客室の属性 (Attribute) に分解して販売する手法
- ゲストが自分の好みに応じて属性を選択し、カスタマイズされた宿泊体験を購入できるようにする

従来の販売方法との違い:



主な客室属性の例:

- ベッドタイプ (King、Twin、Tatami mats 等)
- ビュー (オーシャン、シテイ、マウンテン等)
- 階層 (低層階、高層階等)
- 喫煙/禁煙
- 客室の広さ
- 設備 (バルコニー、バスタブ等)

OTA 2.0 のデータモデルと ABS

OTA 2.0 の役割（データモデル提供）：

- OTA 2.0 は、客室属性を構造化して表現するための**データモデル**を提供している
- OpenTravel 2.0 Object Model を使用した ABS のデータ表現サンプルが公開されている
- ABS に必要な属性情報を標準的なデータ構造で記述できる基盤が整備されている

****補足****: OTA 2.0 は ABS の「データモデル（表現手段）」を提供するものであり、ABS の業務フローや販売シナリオそのものを定義するものではない。

実装の進展:

- 既にアメリカでは実装が進んでいる
- グローバルホテルチェーンを中心に導入が拡大中

ABS の業務シナリオ（HTNG / HEDNA）

HTNG の取り組み（業務シナリオ定義）：

- HTNG では、OTA 2.0 のデータモデルを活用した PMS での ABS 業務フローのシナリオを整理している
- **4つのパターン**を定義しており、この4つを網羅することで ABS 対応が可能
- 具体的なシナリオドキュメント: <https://htng.stoplight.io/docs/attribute-modeling-for-the-distribution-ecosystem/3f6flhlsddxdc-attribute-modeling-for-the-distribution-ecosystem-v1-0>

業界での推進（HEDNA）：

- HEDNA（Hotel Electronic Distribution Network Association）が中心となり ABS の業界標準化・推進を行っている
- 「HEDNA: SHAPING DISTRIBUTION」の枠組みの中で、販売方法や流通モデルの議論が活発化
- 参考サイト: <https://www.hedna.org/>

日本市場への適用における留意点

ホテルごとの考え方の違い:

- ホテルによって「何を属性として価格設定・販売するか」の考え方が異なる
- 日本の旅館とホテルでは重要視する属性が異なる可能性がある

日本固有の属性の必要性:

- 和室、洋室、和洋室を表す属性の定義が必要
- 現在、OTA コードリストには「Tatami mats (畳)」は存在するが、「和洋室」等の詳細な区分は未定義
- 必要に応じて新しいコードの追加を OTA に依頼することが可能

日本の客室タイプと ABS の関係:

日本の宿泊施設では、以下のような属性が重要になる可能性があります：

属性カテゴリ	日本での重要な属性例
部屋スタイル	和室、洋室、和洋室
床材	畳、フローリング、カーペット
ベッド/寝具	ベッド、布団
ビュー	庭園、海、山、街
温泉	客室露天風呂、温泉付き
食事場所	部屋食、個室、レストラン

まとめと次章への接続

ABS は、ゲストのパーソナライゼーション需要に応えるための重要なアプローチです。OTA 2.0 がデータモデル（属性の表現手段）を提供し、HTNG/HEDNA が業務シナリオ（販売フロー・流通モデル）の標準化を推進しています。

日本市場での課題:

- 日本固有の客室属性（和室、洋室、和洋室等）の標準定義
- 既存の固定的な「部屋タイプ」販売から ABS への移行の実現可能性
- PMS の対応状況とシステム改修の必要性

これらの詳細については、次章「II 日本の宿泊業界実態とのギャップ分析」で詳しく分析します。

3. TravelXML

3.1 規格の基本情報

項目	内容
正式名称	TravelXML 標準仕様
策定主体	一般社団法人 日本旅行業協会 (JATA)
初版策定	2000 年代初頭
対象バージョン	Ver 1.0 系 (1.400RC、1.402RC)
対象地域	日本国内
策定目的	日本固有のパッケージ旅行商品取引の標準化
主な採用事例	国内旅行会社、ランドオペレーター、宿泊施設

3.2 策定背景と設計思想

策定の経緯

TravelXML 仕様書には、以下の背景が記載されています。

「現在、欧米では OTA (Open Travel Alliance) において個人の個別予約を中心としたメッセージの標準化が進められています。しかしながら、わが国固有の商習慣である、**交通機関・宿泊施設・各種サービスを組み合わせたパッケージ旅行商品の大規模な取引**については、OTA では対応しておらず、OTA をそのまま利用することは難しい状況です。」

この背景から、TravelXML は以下の設計思想で構築されています。

1. 日本の商習慣を完全にサポート
2. OTA との将来的な融合を模索
3. 実務で使われているデータ項目の標準化

3.3 技術仕様の詳細

データ交換形式

プロトコル: HTTP/HTTPS、FTP、専用回線、メール
 通信方式: Web サービス方式 (SOAP 1.1 / WSDL 1.1)、メール送受信方式 (添付 ZIP 等)
 データ形式: XML
 文字コード: UTF-8、Shift-JIS (両対応)
 スキーマ定義 (整合性チェック手段): DTD (付録 A では XML Schema / Relax NG によるチェックにも言及)
 名前空間: txdi (国内宿泊)、txdc (国内取引)、txoc (海外取引)
 準拠規格 (Web サービス方式): SOAP 1.1、WSDL 1.1、UDDI 2.0

****送受信方式について**** (TravelXML 仕様書 付録 A 準拠) : TravelXML は複数の送受信方式を規定している。****Web サービス方式****では SOAP 1.1 をメッセージ層とし、WSDL 1.1 によるサービス記述、UDDI 2.0 によるサービス検索が前提となる。一方、****メール送受信方式**** (XML ファイルを ZIP 添付して送信) など、非リアルタイムの交換方式も規定されている。

スキーマ構成

TravelXML は 3 つの名前空間で構成されています。

国内宿泊施設情報 (txdi) - 10 スキーマ

- └ 宿泊施設基本情報
- └ 旅館/ホテル別の料金パターン
 - └ 日別パターン (DayByDay)
 - └ 季節別パターン (Seasonal)
- └ 写真・動画情報

国内取引メッセージ (txdc) - 13 スキーマ

- └ 在庫予約 (Allotment Booking)
- └ ルーミングリスト
- └ ブロック管理
- └ 増室・減室リクエスト

海外取引メッセージ (txoc) - 7 スキーマ

- └ フライトデータ
- └ 客室空室照会・回答
- └ 予約依頼・回答

3.4 日本固有の特徴

1.3 者間取引構造

日本の旅行業界の商流を反映した情報構造。

取引の流れ:

旅行会社 ⇄ ツアーオペレーター ⇄ ホテル

各者の情報を明示:

- 旅行会社情報 (発注元)
- ツアーオペレーター情報 (仲介者、該当する場合)
- ホテル情報 (受注先)

2. 旅館とホテルの明確な区別

スキーマレベルで旅館とホテルを分離。

```
<txdi:RyokanRackRateDayByDayPatternInJapan>
  <!-- 旅館専用の料金パターン -->
</txdi:RyokanRackRateDayByDayPatternInJapan>

<txdi:HotelRackRateDayByDayPatternInJapan>
  <!-- ホテル専用の料金パターン -->
</txdi:HotelRackRateDayByDayPatternInJapan>
```

3. 日本固有項目の完全対応

- 入湯税 (HotSpringTaxAdult、HotSpringTaxChild)
- 入湯税合計額 (TotalAccommodationHotSpringTax、TotalPerRoomHotSpringTax)
- 子供料金の固定区分: 日本の商習慣に基づく 4 区分 (A/B/C/D) + その他の固定料金区分
 - 子供 A (70%) : ChildA70Count、PerChildA70Rate、ChildAFee
 - 子供 B (50%) : ChildB50Count、PerChildB50Rate、ChildBFee
 - 子供 C (30%) : ChildC30Count、PerChildC30Rate、ChildCFee
 - 子供 D (無料) : ChildDNoneCount、PerChildDRate、ChildDFee
 - その他: ChildOtherCount、PerChildOtherRate
 - 各区分ごとに「人数 (予約全体/室別)」「単価」「合計料金」を定義
 - 重要: パーセンテージはフィールド名に含まれ固定。年齢基準は施設ごとに独自設定
 - 例: 「小学生は子供 B (50%)、幼児は子供 C (30%)」と施設が独自に決定
- カタカナ・漢字表記の多様性
 - ゲスト氏名: GuestSurName、GuestGivenName (基本)、GuestMiddleName、GuestNamePrefix
 - ゲスト氏名 (漢字) : GuestKanjiName
 - ゲスト氏名 (半角) : GuestNameSingleByte
 - グループ/ゲスト名 (漢字) : GuestOrGroupKanjiName
 - グループ/ゲスト名 (全角) : GuestOrGroupNameDoubleByte
 - グループ/ゲスト名 (半角) : GuestOrGroupNameSingleByte
 - 施設名 (全角カナ) : AccommodationKatakanaNameDoubleByte
 - 施設名 (半角カナ) : AccommodationKatakanaNameSingleByte
 - 施設名 (英語) : AccommodationEnglishName
 - 施設名 (パンフレット用) : AccommodationNameForPamphlet
- 和暦対応 (創業年など)
- 日本旅館協会加盟情報 (AlliedJapanRyokanAssociation)
- 2食付きプラン (標準的な食事パターン)

4. 部屋タイプの固定分類

日本の商習慣に基づく標準部屋タイプ。

コード	名称	説明
TB	Twin	ツインルーム（2名利用）
SB	Single	シングルルーム（1名利用）
SV	Single Use of Twin	ツインの1名利用（シングルユース）
TP	Triple	トリプルルーム（3名利用）
OT	Other	その他の部屋タイプ

3.5 技術的な長所と短所

長所

日本の商習慣への完全対応:

- Personal Rate 方式（1人あたり料金）
- 団体予約の部屋割り
- 3者間取引（旅行会社・ツアーオペレータ・ホテル）
- 日本語の項目名・説明

実務的なシンプルさ:

- 取引種別ごとの専用メッセージ
- フラットに近い構造（深い階層なし）
- 直感的な項目名（CheckInDate、HotelName 等）
- 自由記述欄（FreeNote）の活用

短所

技術的な古さ:

- DTD を中心とした記述だが、付録 A では XML Schema / Relax NG によるフォーマットチェックにも言及がある
- 名前空間の限定的使用
- Web サービス方式は SOAP 1.1 前提であり、現代的な RESTful API には非対応

グローバル展開の困難性:

- 日本固有の概念が多く、海外システムとの直接連携が難しい

- 国際標準との整合性が限定的

拡張性の制約:

- スキーマ変更には仕様改定が必要
- 新しい要件への迅速な対応が困難

4. GIF（政府相互運用性フレームワーク）

4.1 規格の基本情報

項目	内容
正式名称	Government Interoperability Framework - コアデータモデル
策定主体	日本政府（内閣官房、デジタル庁）
バージョン	4.3.8
対象地域	日本国内
参照元標準	IMI（共通語彙基盤）、schema.org
主な利用目的	政府・自治体システム間のデータ互換性確保

4.2 策定背景と設計思想

GIF は、政府・自治体の情報システム間でのデータ連携を効率化するために策定された基礎データ標準です。

策定目的:

1. 行政データの標準化: 政府・自治体システム間のデータ互換性確保
2. データ連携の効率化: システム連携時のカスタマイズコスト削減
3. データ品質の向上: 統一されたデータ定義による品質向上
4. 国際互換性: schema.org など国際標準との整合性確保

4.3 技術仕様の詳細

データ交換形式

プロトコル: 任意（実装依存）
 データ形式: CSV / Excel（フラット構造）
 文字コード: UTF-8
 スキーマ定義: Excel 形式のデータ定義書
 階層表現: 参照型による関連定義

フラットテーブル設計の特徴

GIF は XML や JSON のような階層構造ではなく、テーブル形式のフラットなデータ定義を採用しています。

プレフィックス方式:

pd: Public Data (公共データ) の項目
pcd: Public Code (公共コード) の項目

例:

pd:個人
pd:住所
pd:施設
pcd:性別コード一覧

参照型定義:

複雑な構造は「型」として定義し、参照する

個人エンティティ

- ├ pd:氏 (文字列)
- ├ pd:名 (文字列)
- ├ pd:住所 (pd:住所型を参照)
- └ pd:ID 群 (pd:ID 情報型を参照)

住所型

- ├ 都道府県 (文字列)
- ├ 市区町村 (文字列)
- ├ 郵便番号 (文字列・半角数字)
- └ ...

4.4 主要エンティティの構成

GIF コアデータモデルは 17 のエンティティ (データモデル) で構成されます。

主要エンティティ (13 種類)

No.	エンティティ名	プロパティ数	宿泊業での活用
1	個人 (Person)	19	ゲスト情報
2	法人 (Corporation)	20	法人契約管理
3	事業所 (Office)	6	事業所情報
4	連絡先 (Contact)	8	ゲスト連絡先
5	住所 (Address)	13	施設・ゲスト住所
6	施設 (Facility)	13	宿泊施設マスター
7	アクセシビリティ	19	バリアフリー情報

8	子供預かり情報	10	託児サービス
9	土地 (Land)	13	土地情報
10	建物 (Building)	11	建物情報
11	その他構造物	11	インフラ情報
12	設備 (Equipment)	10	施設設備
13	イベント (Event)	34	イベント管理

データ型定義 (4 種類)

No.	データ型名	プロパティ数	用途
1	ID 情報型	2	ID 種別・ID 値
2	コード情報型	2	コード種別・コード値
3	役割関与情報型	2	役割とエンティティの関連
4	関連組織情報型	2	組織の役割と法人番号

総プロパティ数: 195 個

4.5 日本市場との適合性分析

GIF 特有の日本対応項目

住所表現:

- 全国地方公共団体コード (6 桁)
- 町字 ID (デジタル庁アドレス・ベース・レジストリ準拠)
- 都道府県・市区町村・町名・丁目・番地・号を個別に定義
- 郵便番号 (7 桁、ハイフンなし)

氏名表現:

```
pd:氏: 山田
pd:名: 花子
pd:氏(カナ): ヤマダ
pd:名(カナ): ハナコ
pd:氏(英字): Yamada
pd:名(英字): Hanako
```

日本固有コード:

- 全国地方公共団体コード
- 法人番号 (13 桁)

- 性別コード (ISO 5218 準拠)

OTA・国際標準との構造的差異

比較項目	GIF	OTA 2.0	OTA 1.0
データ形式	フラット (CSV/Excel)	階層構造 (JSON/XML)	階層構造 (XML)
住所表現	都道府県から号まで個別項目	<code>Common.Address</code> (階層 + ExtensionPoint)	AddressLine1-3 形式 (自由記述)
氏名表現	氏・名・カナ・英語を個別定義	<code>PersonName</code> (多言語 + ExtensionPoint)	PersonName (英語基準)
コード体系	日本政府標準コード	OTA コード + <code>_Enum</code> 拡張	OTA 独自 CodeList + <code>@OtherType</code>
必須項目	複数パターン (推奨度別)	モデル定義 + required 属性	XSD の必須定義

4.6 技術的な長所と短所

長所

日本市場への最適化:

- 日本の住所体系、氏名表記に完全対応
- 日本の法規制 (法人番号、個人番号等) に準拠
- 日本の商習慣に適合

政府標準としての信頼性:

- 政府・自治体で既に採用
- 長期的な保守・更新が期待できる
- 法的根拠のある標準

実装の容易性:

- フラット構造で既存システムと統合しやすい
- CSV 形式で簡単にデータ交換可能
- データベース設計に直接適用可能

短所

階層構造の表現制約:

- 複雑なネスト構造は参照型で対応するため煩雑
- OTA や HTNG との直接変換には中間処理が必要

国際連携の限界:

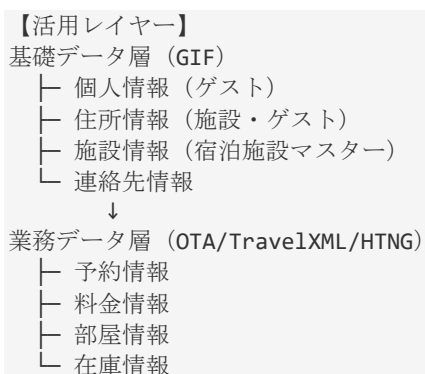
- 日本固有の項目が多く、海外システムとの直接連携には変換が必要
- schema.org 準拠だが、日本拡張が多い

宿泊業特化項目の不足:

- 部屋タイプ、料金プラン、予約情報など宿泊業固有項目は含まれていない
- 追加定義が必要

4.7 宿泊業での活用可能性

GIF は基礎データ層の標準として位置づけられます。



推奨される活用方法:

1. GIF をベース層として採用（個人、法人、住所、施設などの基本情報）
2. 業界固有項目の拡張（部屋、料金、予約は OTA/TravelXML を参考に追加）
3. 政府システムとの連携が容易に

5. HTNG Express

5.1 規格の基本情報

項目	内容
正式名称	HTNG Express
策定主体	Hospitality Technology Next Generation (HTNG)
バージョン	1.0
対象地域	グローバル
対象範囲	PMS と館内サービスシステム間の軽量データ連携
主な利用目的	予約情報と客室状態情報の効率的連携

5.2 策定背景と設計思想

HTNG Express は、HTNG の包括的な仕様群（フルスペック）の複雑さを解消し、**最小限かつ必須のデータ項目のみに**絞り込んだ軽量仕様です。

設計思想:

- **最小限主義**: HTNG フルスペック仕様から実用的な最小項目セットに絞り込み
- **現代的技術**: JSON/REST API による実装容易性
- **柔軟性**: ホテル・PMS 固有の値を許容

対象システム（館内サービス向け簡潔な情報連携）:

- ハウスキーピングシステム
- 室内エンターテイメント
- コンシェルジュシステム
- 建物管理・予防保守
- 無線/Wi-Fi 管理
- 音声アシスタント

対象外システム: CRS、レベニューマネジメント、バックオフィス会計、セールス&ケータリング、キオスク（HTNG Kiosk 使用）、スマートキーシステム、OTA・予約流通（OpenTravel Alliance 等を使用）等

5.3 技術仕様の詳細

データ交換形式

プロトコル: HTTPS
 通信方式: RESTful API
 データ形式: JSON
 文字コード: UTF-8
 スキーマ定義: OpenAPI 3.0.3 (YAML)
 認証: Token-based (実装依存)

エンティティ構成

HTNG Express は7つのシンプルなエンティティで構成されます。

エンティティ名	プロパティ数	必須フィールド数	説明
Reservation	24	12	予約の基本情報と宿泊詳細
Guest	13	3	ゲスト個人情報と連絡先
Property	2	1	ホテル施設識別情報

RoomRate	6	3	宿泊料金と料金プラン
Email	3	1	メールアドレスと種別
Loyalty	3	2	ロイヤリティプログラム 情報
Room	9	1	客室状態と清掃・占有情 報

総プロパティ数: 60 個

5.4 データ構造の特徴

Reservation エンティティの構造

```
{
  "property": {
    "property_code": "ORDEC",
    "chain_code": "IHG"
  },
  "primary_guest": {
    "first_name": "John",
    "last_name": "Smith",
    "email": [{
      "email_address": "john.smith@provider.com",
      "email_type": "Personal",
      "email_is_primary": true
    }]
  },
  "arrival_date": "2025-11-15T15:00:00+00:00",
  "departure_date": "2025-11-17T11:00:00+00:00",
  "number_of_adults": 2,
  "number_of_children": 0,
  "room_type_code": "KING",
  "room_number": "1205",
  "reservation_status_htng": "Reserved",
  "pms_reservation_id": "RES123456",
  "pms_confirmation_code": "CONF789"
}
```

列挙型の 3 分類

HTNG Express は、列挙型を 3 つに分類することで柔軟性を実現しています。

【HTNG 列挙】標準化された固定値セット
reservation_status_htng:

- Reserved (予約済み)
- NoShow (無断不泊)
- PreCheckedIn (事前チェックイン済み)
- InHouse (館内滞在中)
- CheckedOut (チェックアウト済み)
- Open (オープン状態)
- Canceled (キャンセル済み)
- PreConfirmed (事前確認済み)

【PMS 列挙】 PMS システムごとに異なる値

reservation_status_pms: "CONFIRMED" / "IN_HOUSE" / "DEPARTED" など

【ホテル列挙】 ホテルごとにカスタマイズ可能

room_type_code: "KING" / "和室 8 畳" / "DELUXE_TWIN" など

rate_plan_code: "BAR" / "早割 30" / "CORPORATE" など

5.5 日本市場との適合性分析

適合する点

国際標準準拠:

- ISO 8601 (日時形式)
- ISO 639-1 (言語コード)
- ISO 4217 (通貨コード)

軽量性:

- 実装が容易
- 学習コストが低い
- モダンな技術スタック (JSON/REST)

柔軟性:

- ホテル列挙により日本固有のコードを自由に定義可能
- 拡張フィールドの追加が容易

適合しない点・要調整項目

日本特有項目の不足:

標準に含まれない項目 (拡張が必要) :

- 性別
- 国籍
- パスポート番号
- 食事時間 (夕食・朝食)
- 送迎サービス
- 温泉利用

料金体系の違い:

- Room Charge 方式が基本
- Personal Rate 方式への対応には拡張が必要

名前表記:

- first_name / last_name の順序
- 日本では姓・名の順が一般的

5.6 技術的な長所と短所

長所

軽量性:

- 必要最小限の項目に絞り込み
- ネットワーク負荷が低い

実装の容易さ:

- JSON/REST で実装が簡単
- 開発者に馴染みのある技術
- OpenAPI 仕様で自動コード生成が可能

明確性:

- readOnly フラグで更新可能範囲が明確
- 必須項目の明示
- シンプルな構造

短所

カバー範囲の限定:

- 館内サービス連携に特化
- 予約流通、会計処理などは対象外

仕様の未成熟さ:

- 歴史が浅い (HTNG Kiosk は 10 年以上の実績)
- 実装事例が限定的

日本固有要件への対応:

- 拡張が必要な項目が多い
- 標準化されたマッピングが未整備

6. HTNG Kiosk

6.1 規格の基本情報

項目	内容
正式名称	HTNG Kiosk Integration
策定主体	Hospitality Technology Next Generation (HTNG)
バージョン	2.1
策定年	2013年 (2013B)
対象地域	グローバル
対象範囲	ホテルキオスク端末と PMS 間の API 連携
主な利用目的	セルフチェックイン/チェックアウト、予約検索、料金照会、キー発行

6.2 策定背景と設計思想

HTNG Kiosk は、ホテルのセルフサービスキオスク端末と PMS の間で標準化されたデータ交換を実現するために策定されました。8つの業務フロー（空室状況照会、複数名義登録、フォリオ取得、キー発行、予約変更、ゲストメッセージ表示、滞在ステータス更新、アップセル提案）をカバーしています。

****注記**:** HTNG Kiosk は、チェックイン/チェックアウト業務を成立させるための業務フロー統合仕様であり、宿泊商品の設計や料金体系の定義は対象外である。

6.3 技術仕様の詳細

データ交換形式

プロトコル: HTTP/HTTPS
 通信方式: SOAP (Simple Object Access Protocol)
 データ形式: XML
 文字コード: UTF-8
 スキーマ定義: XSD + WSDL
 ベースメッセージ: OpenTravel Alliance (OTA) 準拠

仕様構成

HTNG Kiosk の仕様は以下で構成されています。

仕様ドキュメント:

- PDF 形式の技術仕様書
- 業務フロー定義書

実装サンプル:

- XSD スキーマ定義ファイル
- WSDL サービス定義ファイル

6.4 データ構造の特徴

OTA 準拠のメッセージ構造

HTNG Kiosk は、OpenTravel Alliance のメッセージ標準をベースとしています。

```
<OTA_HotelAvailRQ
  xmlns="http://www.opentravel.org/OTA/2003/05"
  EchoToken="12345"
  TimeStamp="2025-10-26T10:30:00Z"
  Version="2.1"
  HotelStayOnly="true"
  ExactMatchOnly="true">

  <HotelReservationIDs>
    <HotelReservationID ResID_Type="14" ResID_Value="RES123456"/>
  </HotelReservationIDs>
</OTA_HotelAvailRQ>
```

属性のプレフィックス表現

XML 属性は@プレフィックスで表現されます。

```
{
  "@EchoToken": "12345",
  "@TimeStamp": "2025-10-26T10:30:00Z",
  "@Version": "2.1",
  "@HotelStayOnly": "true"
}
```

6.5 日本市場との適合性分析

HTNG Kiosk は OTA ベースであるため、日本市場との適合性の課題は OTA と同様です。

主な課題

住所表記:

- 国際標準: 番地→市区町村→都道府県→国の順
- 日本: 都道府県→市区町村→番地の順

氏名表記:

- 国際標準: GivenName (名) + Surname (姓) の順
- 日本: 姓名の順が一般的

料金体系:

- OTA (国際標準) : Room Charge 方式 (1泊1室) が主流だが、Personal Rate 方式 (1泊1名) にも ChargeType コードリストで標準対応
- HTNG Kiosk: OTA 準拠のため同様の構造を持つが、料金体系の定義は担わず、PMS 側で確定した料金を表示・精算するのみ
- 日本: 1泊1名や人数別料金設定が一般的

6.6 技術的な長所と短所

長所

セルフサービス機器連携の標準:

- キオスク特有の業務フローを網羅
- 10年以上の実装実績

OTA 準拠による互換性:

- OTA メッセージをそのまま活用
- 既存の OTA 実装との親和性

短所

XML/SOAP の複雑性:

- 実装が複雑
- 学習コストが高い

メンテナンスモード:

- HTNG は新規開発を HTNG Express に移行
- HTNG Kiosk は維持管理のみ
- 将来的な縮小傾向

日本固有業務への対応不足:

- OTA と同様の課題
- 日本市場への適合には拡張が必要

7. 包括的な比較分析

7.1 技術仕様の比較マトリクス

比較項目	OTA 1.0	OTA 2.0	TravelXML	GIF	HTNG Express	HTNG Kiosk
基本情報						
策定期	1999年～（最新: 2024A）	1999年～（Object Suite: 2020年代）	2000年代初頭	2022年～	2021年	2013年
バージョン	2024A (XML) 0 CodeList 4.0.0 Hospitality 2.0.0	Ver 1.4-00RC （添付仕様書ベース）	4.3.8	2.1	1.0	
対象地域	グローバル	グローバル	日本国内	日本国内	グローバル	グローバル
技術仕様						
データ形式	XML	JSON / XML	XML	CSV	JSON	XML
通信方式	HTTP/HTTPS （HTTPメッセージング） / SOAP 1.1・1.2	REST / WSDL	Web サービス方式（S 送受信）	ファイル	REST	SOAP
スキーマ定義	XSD	OpenAPI / OTM	DTD	Excel	OpenAPI/YAML	XSD+WSDL
文字コード	UTF-8	UTF-8	UTF-8/SJIS	UTF-8	UTF-8	UTF-8
構造特性						
階層の深さ	深い（5層以上）	深い（モデル + \$ref）	浅い（2～3層）	最浅（1～2層）	浅い（2～3層）	深い（5層以上）
データ密度	低（冗長）	中（JSON/XML 両対応）	中	最高（簡潔）	高（簡潔）	低（冗長）
正規化度	高	高	中	中	高	高

拡張性	◎	◎	△	△	○	◎
実装特性						
学習コスト	非常に高	高（モデル理解が必要）	中	最低	低	高
実装難易度	高	中	中	最低	低	高
保守性	中	高	中	高	高	中
適用範囲						
カバー範囲	旅行業界全般	旅行業界全般	日本 B2B 取引	基礎データ	館内サービス	キオスク連携
グローバル対応	◎	◎	△	X（日本国内のみ）	◎	◎
日本商習慣対応	△	△	◎	◎（基礎のみ）	△	△
成熟度						
実装事例	非常に多い	増加中	多い	増加中	増加中	多い
将来性	維持	拡大	維持	拡大	拡大	縮小

◎優れている、○良い、△やや劣る、X不十分、— 対象範囲外・該当なし

7.2 日本市場適合度の詳細分析

料金体系対応

規格	Room Charge	Personal Rate	対応方法
OTA 1.0	○ ネイティブ	○ ネイティブ	ChargeType (CHG) コードリストで per person/per room 等の課金単位を選択可能。 AdditionalGuestAmount/@Percent でパーセンテージ表記も可能
OTA 2.0	○ ネイティブ	○ ネイティブ	ChargeType (CHG) コードリストで per person/per room 等の課金単位を選択可能。 AdditionalGuestAmount/@Percent

			でパーセンテージ表記も可能
TravelXML	△ 可能	○ ネイティブ	標準対応
GIF	-	-	対象外（基礎データのみ）
HTNG Express	○ ネイティブ	△ 拡張必要	カスタムフィールド
HTNG Kiosk	○ ネイティブ	△ 拡張必要	TPA_Extensions 使用

食事プラン対応

規格	素泊まり	朝食付き	2食付き	3食付き	備考
OTA 1.0	○	○	○	○	Meal Plan Type (MPT) コードリスト (Full board, Dinner bed and breakfast plan 等) および AdditionalDetailType Code #43 (Meal plan description) で食事プランを表現可能。日本の食事プラン (2食付き等) にも対応
OTA 2.0	○	○	○	○	AdditionalDetailType Code #43 (Meal plan description) で詳細を表現可能。既存フィールドの活用が推奨される
TravelXML	○	○	○	○	日本の食事プランにネイティブ対応
GIF	-	-	-	-	対象外（基礎データのみ）
HTNG Express	△	△	X	X	rate_plan_code による運用対応
HTNG Kiosk	△	△	△	△	予約情報に含まれる場合は参照可能 (Kiosk 仕様としての食事プラン設計・販売機能は対象外)

日本固有項目の対応状況

項目	OTA 1.0	OTA 2.0	TravelXML	GIF	HTNG Express	HTNG Kiosk	備考
入湯税	△	△	○	-	X	—	OTA: Fee Tax Type コードリストへの追加依頼、または TPA_Extensions/ExtensionPoint で対応可能。Kiosk: 料金設計・税計算を担わず、PMS 側で確定した結果を表示・精算するのみ
温泉税	△	△	○	-	X	—	OTA: Fee Tax Type コードリストへの追加依頼、または TPA_Extensions/ExtensionPoint で対応可能。Kiosk: 料金設計・税計算を担わず、PMS 側で確定した結果を表示・精算するのみ
子供料金区分 (A・B・C・D)	△	△	○	-	△	—	商習慣の根本的な違いあり (詳細は下記参照)。Kiosk: 料金区分の定義・計算を担わず、PMS 側で確定した結果を表示・精算するのみ
和室/洋室区分	○	○	○	-	△	○	OTA: Bed Type (BED) #7 (Tatami mats) で和室を標準表現。必要なコードは OTA コードキーパーに追加依頼可能
メゾネット	○	○	-	-	△	○	OTA: Guest Room Info (GRI) #89 (Multi-level suite) で表現可能
カタカナ表記	△	△	○	○	△	△	拡張で対応
郵便番号 (主要キー)	△	△	○	○	△	△	
法人番号	X	X	X	○	X	X	

子供料金の商習慣の違い（重要）

子供料金の定義方法において、OTAと日本（TravelXML）では根本的なアプローチが異なります。これは技術仕様の差異ではなく、ビジネスモデルと商習慣の違いを反映しています。

【日本の商習慣（TravelXML）】

方式：固定の料金区分でパーセンテージが事前に決定

料金区分：

- 子供 A（70%）： ChildA70Count, PerChildA70Rate, ChildAFee
- 子供 B（50%）： ChildB50Count, PerChildB50Rate, ChildBFee
- 子供 C（30%）： ChildC30Count, PerChildC30Rate, ChildCFee
- 子供 D（無料）： ChildDNoneCount, PerChildDRate, ChildDFee
- その他： ChildOtherCount, PerChildOtherRate

特徴：

- ✓ 料金区分（A/B/C/D）が主体
- ✓ パーセンテージは固定（70%/50%/30%/無料）
- ✓ フィールド名にパーセンテージが含まれる
- ✓ 年齢基準は宿泊施設ごとに独自設定（仕様では規定しない）

運用例：

「小学生は子供 B（50%）、幼児は子供 C（30%）」と施設が独自に決定
 同じ年齢でも施設によって異なる区分になることがある

【OTAの方式】

方式：年齢区分とパーセンテージを柔軟に組み合わせ

パス： AdditionalGuestAmounts/AdditionalGuestAmount

構造：

- AgeQualifyingCode： 年齢区分（Adult、Child、Infant、Teenager、Senior 等）
- Percent 属性： パーセンテージ（0-100 の範囲で柔軟に設定可能）

特徴：

- ✓ 年齢区分（AgeQualifyingCode）が主体
- ✓ パーセンテージは柔軟に設定可能（任意の値）
- ✓ 固定の A/B/C/D 区分は存在しない
- ✓ 年齢区分は仕様で標準定義されている

運用例：

「Child（子供区分）は大人料金の 50%」と年齢区分とパーセンテージで定義
 パーセンテージは施設が自由に設定可能

【商習慣の違いがもたらす影響】

観点	日本（TravelXML）	OTA
基準	料金区分（A/B/C/D）	年齢区分（Child、Infant 等）
パーセンテージ	固定（70%/50%/30%/無料）	柔軟に設定可能（0-100%）

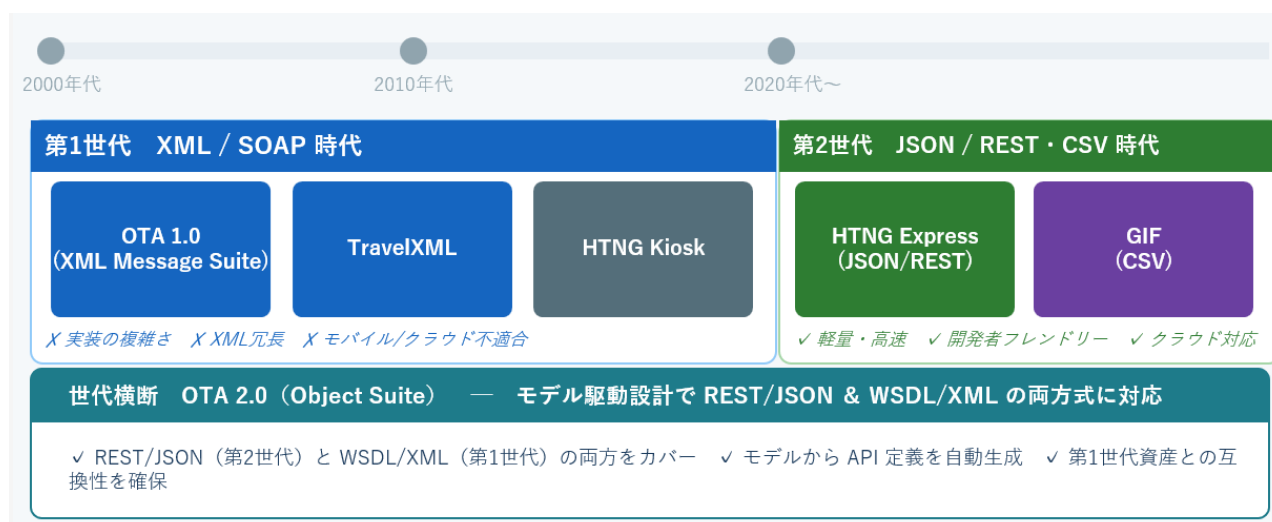
年齢定義	施設ごとに独自設定	年齢区分は仕様で標準定義
柔軟性	低（4区分（A/B/C/D）+その他の固定区分）	高（任意のパーセンテージ設定可能）
相互変換の難易度	OTA→TravelXML は困難 （どの区分に該当するか判定が必要）	TravelXML→OTA は比較的容易 （年齢区分を決めて変換）

【対応評価の理由】

OTA は **AgeQualifyingCode** と **Percent** の組み合わせで子供料金を表現できますが、日本の「固定料金区分（A/B/C/D）」という商習慣には直接対応していないため「△」としています。

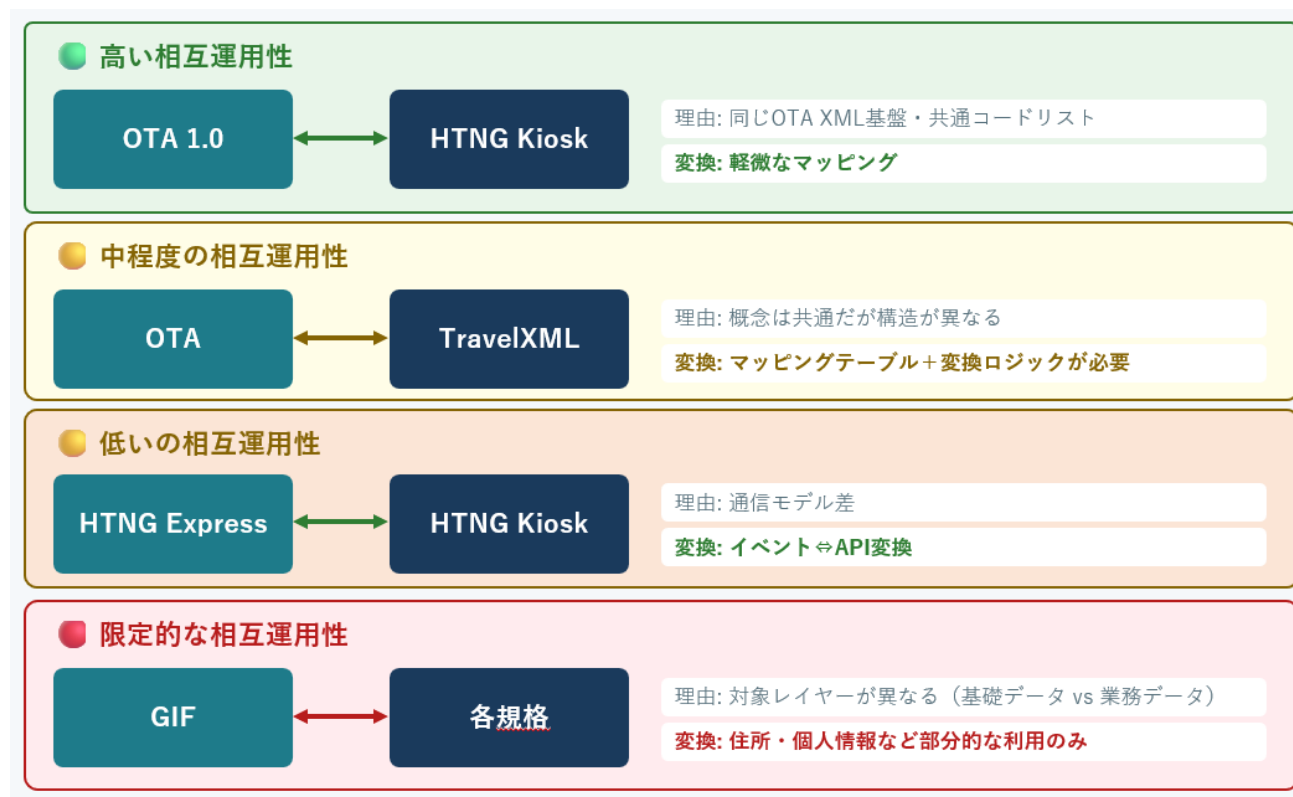
- OTA で「Child: 50%」と設定することは可能
- しかし、それが日本の「子供 B」に相当するのか、別の概念なのかは明確ではない
- 日本の施設は「子供 B」という区分名で運用しているため、OTA への変換時には判断が必要

7.3 技術世代による分類



7.4 相互運用性の実現可能性

データ項目のマッピング可能性



7.5 主要データ項目のマッピング参考表

予約基本情報

項目	OTA 2.0	OTA 1.0	TravelXML	GIF	HTNG Express	HTNG Kiosk
予約ID	HotelReservationIDs[].ID	UniqueID/@ID	BookingNumber	-	pms_reservation_id	UniqueID/@ID
確認番号	HotelReservationIDs[].ID (@type="14")	HotelReservationID	ConfirmationNumber	-	pms_confirmation_code	HotelReservationID
旅行会社子	-	-	TravelAgencyBookingNumber	-	-	-

約 番 号						
宿 泊 施 設 確 認 番 号	-	-	AccommodationConfirmationNumber	-	-	-
キ ャ ン セ ル 番 号	-	-	CancellationNumber	-	-	-
チ ェ ツ ク イ ン	RoomStay.TimeSpan.start	TimeSpan/@Start	CheckInDate	-	arrival_date	TimeSpan/@Start
チ ェ ツ ク ア ウ ト	RoomStay.TimeSpan.end	TimeSpan/@End	CheckOutDate	-	departure_date	TimeSpan/@End
宿 泊 人 数 （ 大 人 ）	GuestCounts[].count (ageQualifyingCode="Adult")	GuestCounts/GuestCount (@AgeQualifyingCode) ※1	-	-	number_of_adults	GuestCounts/GuestCount
宿 泊 人 数	GuestCounts[].count (ageQualifyingCode="Child")	GuestCounts/GuestCount (@AgeQualifyingCode) ※1	-	-	number_of_children	GuestCounts/GuestCount

(子供)						
子供 A 人数 (7 0%))	-	-	TotalChildA70Count RoomChildA70Count	-	-	-
子供 B 人数 (5 0%))	-	-	TotalChildB50Count RoomChildB50Count	-	-	-
子供 C 人数 (3 0%))	-	-	TotalChildC30Count RoomChildC30Count	-	-	-
子供 D 人数 (無 料))	-	-	TotalChildDNoneCount RoomChildDNoneCo unt	-	-	-
そ の 他 子 供 人	-	-	TotalChildOtherCount RoomChildOtherCo unt	-	-	-

数						
---	--	--	--	--	--	--

※1 OTA 2.0 では AgeQualifyingCode_Enum のテキスト値 ("Adult"、"Child"、"Infant"等) を使用。
 OTA 1.0 では Age Qualifying Code (AQC)の数値コード (AQC 1: Over 21、AQC 4: Under 12 等) を使用。
 実装時は最新の OTA コードリストを参照のこと。

ゲスト情報

項目	OTA 2.0	OTA 1.0	TravelXML	GI F	HTNG Express	HTNG Kiosk
姓	PersonName.surname	PersonName/Surname	GuestSurName	pd :氏	name_last	PersonName/Surname
名	PersonName.givenName	PersonName/GivenName	GuestGivenName	pd :名	name_first	PersonName/GivenName
ミドル ネーム	PersonName.middleName	PersonName/MiddleName	GuestMiddleName	-	name_middle	PersonName/MiddleName
敬称	PersonName.namePrefix	PersonName/NamePrefix	GuestNamePrefix	-	name_prefix	PersonName/NamePrefix
氏名 (漢字)	-	-	GuestKanjiName	pd :氏名	-	-
氏名 (半角)	-	-	GuestNameSingleByte	-	-	-
グループ名 (漢字)	-	-	GuestOrGroupKanjiName	-	-	-

グループ名 (全角)	-	-	GuestOrGroupNameDoubleByte	-	-	-
グループ名 (半角)	-	-	GuestOrGroupNameSingleByte	-	-	-
メール	Email.address	Email	GuestEmail	pd:連絡先/メールアドレス	email_address	Email
電話	Telephone.phoneNumber	Telephone/PhoneNumber	GuestPhoneNumber	pd:連絡先/電話番号	phone_mobile	Telephone/PhoneNumber
緊急連絡先	-	-	GuestEmergencyPhoneNumber	-	-	-
年齢	Profile.Customer.person.Age	Age	GuestAge	-	-	Age

施設基本情報

項	OTA 2.0	OTA 1.0	TravelXML	GI	HTNG	HTNG
---	---------	---------	-----------	----	------	------

目				F	Express	Kiosk
施設名	BasicPropertyInfo.hotelName	HotelName	AccommodationName	pd : 施設・名称	property_name	HotelName
施設名 (カナ全角)	-	-	AccommodationKatakanaNameDoubleByte	pd : 施設・名称カナ	-	-
施設名 (カナ半角)	-	-	AccommodationKatakanaNameSingleByte	-	-	-
施設名 (英語)	BasicPropertyInfo.hotelName.@language="en"	-	AccommodationEnglishName	-	-	-
施設名 (パンフレット用)	-	-	AccommodationNameForPamphlet	-	-	-

施設コード	BasicPropertyInfo.hotelCode	HotelCode	AccommodationCode	pd : 施設 .ID	property_code	HotelCode
チェーンコード	BasicPropertyInfo.chainCode	ChainCode	ChainCode	-	chain_code	ChainCode
チェーン名	BasicPropertyInfo.chainName	-	ChainName	-	-	-
郵便番号	Address.postalCode	Address/PostalCode	PostalCode	pd : 住所・郵便番号	-	Address/PostalCode
都道府県	Address.stateProv	Address/StateProv	StateProvidence	pd : 住所・都道府県	-	Address/StateProv
市区町村	Address.cityName	Address/CityName	CityName	pd : 住所・市区町村	-	Address/CityName
建物名	Address.addressLine[]	Address/AddressLine	BuildingName	-	-	Address/AddressLine
電話番号	ContactNumbers[].phoneNumber	Telephone/PhoneNumber	AccommodationPhoneNumber	pd : 連絡先 / 電	-	Telephone/PhoneNumber

				話 番 号		
F A X 番 号	ContactNumbers[].phoneNumber (phoneTechType="Fax")	Telephone/PhoneNumber >(@PhoneTechType="3")	AccommodationFacsi mileNumber	pd : 連 絡 先 / F A X	-	Telephone/P honeNumber
緯 度	Latitude	-	-	(空 間 ID 経 由)	-	Latitude
経 度	Longitude	-	-	(空 間 ID 経 由)	-	Longitude

料金情報

項 目	OTA 2.0	OTA 1.0	TravelXML	G I F	HTN G Expre ss	HTNG Kiosk
通 貨 コ ー ド	RoomRates[].cur rencyCode	RoomRate/@Curren cyCode	CurrencyCode	-	curre ncy_c ode	RoomRate/@Curren cyCode
室 料 金	RoomRates[].rat es[].amountBefo reTax	RoomRate/Rates/R ate/@AmountBefor eTax	-	-	room_ rate	RoomRate/Rates/R ate/@AmountBefor eTax
子 供 A 単 価 (7	-	-	PerChildA70Rate	-	-	-

0%)						
子供B単価(50%)	-	-	PerChildB50Rate	-	-	-
子供C単価(30%)	-	-	PerChildC30Rate	-	-	-
子供D単価(無料)	-	-	PerChildDRate	-	-	-
その他子供単価	-	-	PerChildOtherRate	-	-	-
子供A合計料金	-	-	ChildAFee	-	-	-
子供	-	-	ChildBFee	-	-	-



B 合計 料金						
子供 C 合計 料金	-	-	ChildCFee	-	-	-
子供 D 合計 料金	-	-	ChildDFee	-	-	-
入湯 税 (大人)	-	-	HotSpringTaxAdult	-	-	-
入湯 税 (子供)	-	-	HotSpringTaxChild	-	-	-
消費 税	taxes[].amount	Taxes/Tax/@Amount	TotalAccommodationConsumptionTax >TotalAccommodationCommissionConsumptionTax	-	-	Taxes/Tax/@Amount
総額 (税込 前)	total.amountBeforeTax	Total/@AmountBeforeTax	TotalAmountBeforeTax	-	-	Total/@AmountBeforeTax

総額 (税込後)	total.amountAfterTax	Total/@AmountAfterTax	TotalAmountAfterTax	-	-	Total/@AmountAfterTax
クレジットカード番号	paymentCard.cardNumber	PaymentCard/@CardNumber	CreditCardNumber	-	-	PaymentCard/@CardNumber

8. まとめ

8.1 各規格の特徴整理 (6 規格を OTA 統合で 5 分類)

OpenTravel Alliance (OTA)

+ 強み	× 弱み	▶ 適用シーン
グローバルスタンダードとしての認知度と実績	学習・実装コストの高さ	 国際OTAとの連携
旅行業界全般をカバーする包括性	日本固有の商習慣への対応不足	 大規模ホテルチェーン
25年以上の歴史による成熟度		 グローバル展開を見据えたシステム
豊富なコードリストと拡張性		

TravelXML

+ 強み	× 弱み	▶ 適用シーン
日本の商習慣への完全対応	技術的な古さ（DTD、XML）	 国内旅行会社との取引
Personal Rate方式（1人あたり料金）のネイティブサポート	拡張性の制約	 ランドオペレータ業務
団体旅行・パッケージ商品への対応	グローバル展開の困難性	 団体予約管理
詳細な子供料金区分（料金率を含む）		
日本語表記の多様性（漢字・カナ・英語）		
実務的なシンプルさと実装実績		

GIF（政府相互運用性フレームワーク）

+ 強み	× 弱み	▶ 適用シーン
公的データとの連携基盤	宿泊業務データは対象外	 施設マスターデータ管理
住所・位置情報の高精度標準化	基礎データ層のみ（単独では業務システム構築不可）	 観光庁統計連携
シンプルなCSV形式		 地域DMO連携
政府標準としての信頼性		 行政システムとのデータ交換

HTNG Express

+ 強み	× 弱み	▶ 適用シーン
現代的な技術（JSON/REST）	仕様の未成熟さ（歴史が浅い）	 ハウスキーピングシステム
実装の容易さと軽量性	日本固有要件は拡張が必要	 室内エンターテイメント
開発者フレンドリー	実装事例が限定的	 コンシェルジュシステム
館内サービス連携に特化	カバー範囲が限定的	 建物管理・無線/Wi-Fi管理
		 音声アシスタント

HTNG Kiosk

+ 強み	× 弱み	▶ 適用シーン
セルフサービス機器連携の標準	XML/SOAPの複雑性	 既存キオスクシステムの維持
8つの業務フローを包括	メンテナンスモード（縮小傾向）	 国際ホテルチェーンとの連携
OTA準拠による互換性	日本固有業務への対応不足	 HTNG Expressへの移行期の併用
多数の実装実績		

8.2 技術的視点からの主要な知見

1. 料金体系の国際標準と日本市場の関係

重要な発見: OTA（国際標準）は、Room Charge 方式と Personal Rate 方式の両方に標準対応しています。

【OTA の料金体系サポート】

対応方式:

- Room Charge 方式（1 室あたり料金）
 - Personal Rate 方式（1 人あたり料金）
- 両方に対応

実装方法:

- ChargeType (CHG) コードリストで per person / per room 等の課金単位を選択可能
- AdditionalGuestAmount/@Percent でパーセンテージ表記も可能
- 年齢情報も併せて指定可能

地域的な傾向:

- 海外: 主に Room Charge が使われる
- 日本: 主に Personal Rate が使われる
- オールインクルーシブ等: 海外でも per person を使用

結論:

OTA は両方の料金体系に対応しているため、日本市場の 1 名料金方式にも拡張なしで対応可能

TravelXML との違い:

【TravelXML】

対応方式: Personal Rate 方式に特化

強み: 日本の商習慣（2 食付き、子供料金細分化等）に完全対応

2. 技術世代の移行

第1世代（XML/SOAP）の課題：

- 実装の複雑さ
- パフォーマンスオーバーヘッド
- モバイル/クラウド時代への不適合

第2世代（JSON/REST、CSV）の優位性：

- 実装の容易さ
- 軽量・高速
- 現代的な開発環境への適合

OTA 2.0（Object Suite）の位置づけ：

- モデル駆動設計による API 定義の自動生成
- REST/JSON および WSDL/XML の両方式に対応

結論：

OTA 2.0 は REST/WSDL および JSON/XML の両方式に対応しており、新規開発にも既存資産との連携にも柔軟に活用可能

3. GIF の特殊な位置づけ

GIF は業務データではなく基礎データを対象：

- ✓ 住所データの標準化（CSV 形式で提供）
- ✓ アドレス・ベース・レジストリとの連携
- ✓ 地理情報（空間 ID、緯度経度）

- 宿泊業務システムの「土台」として活用
- 業務データ層（OTA/HTNG 等）と組み合わせて使用

4. 対象レイヤーの違いの認識

各規格は異なるレイヤーを対象としており、単純に比較できません。

【レイヤー構造】

業務アプリケーション層

- ├ 予約管理
- ├ 在庫管理
- └ 売上管理

データ交換層（本調査の対象）

- ├ 予約・流通：OTA、TravelXML
- ├ 館内連携：HTNG Express、HTNG Kiosk
- └ 基礎データ：GIF

基盤層

- ├ データベース
- └ ネットワーク

8.3 次章への接続

本章では、国内外の主要な宿泊業界関連データ規格について、技術仕様と特徴を客観的に比較分析しました。

本章で明らかになった主要なポイント:

1. **料金体系の商習慣の違い:** OTA は Room Charge ・ Personal Rate 両方式に対応するが、実運用上の主流が海外 (Room Charge) と日本 (Personal Rate) で異なる
2. **技術世代の差:** XML/SOAP 世代と JSON/REST 世代
3. **対象レイヤーの違い:** 業務データ層と基礎データ層
4. **日本固有項目への対応:** 各規格で対応状況が大きく異なる

次章「日本の宿泊業界実態とのギャップ分析」では:

- 本章で整理した各規格の特徴を踏まえ、日本の宿泊業界の実態 (PMS 各社の実装状況、商習慣、法的要件) との具体的なギャップを詳細に分析します。
- 実際の PMS ベンダーや OTA 事業者へのヒアリング結果を基に、標準化における課題を明確化します。

第Ⅲ章「標準項目定義に向けた設計指針」では:

- 本章と第Ⅱ章の分析結果を統合し、日本の宿泊業界に適した実用的なデータ標準化の方向性を提示します。

参考資料

本比較分析の作成にあたり、以下の資料を参照しました。

OpenTravel Alliance

- OpenTravel Alliance XML Schema (Common 5.0.0, CodeList 4.0.0, Hospitality 2.0.0)
- OpenTravel Alliance 2024A XML Message Suite (Message Users Guide, Release Notes, Schema Versioning)
- OpenTravel Alliance 公式概要資料 (OTA 1.0 XML Message Suite / OTA 2.0 Object Suite の製品概要)
- OpenAPI 3.0.3 Specification
- OTA Code List Specification
- OpenTravel Alliance Implementation Guide

TravelXML

- TravelXML Ver 1.0 系仕様書 (1.400RC、1.402RC) (付録 A: 送受信方式を含む)
- TravelXML 国内宿泊施設情報仕様
- TravelXML 国内取引メッセージ仕様
- TravelXML 海外取引メッセージ仕様
- 一般社団法人 日本旅行業協会 (JATA) 公開資料

GIF (政府相互運用性フレームワーク)

- 政府相互運用性フレームワーク コアデータモデル Ver 4.3.8
- コアデータパーツ: 住所・所在地データ
- コアデータパーツ: 地理情報データ
- アドレス・ベース・レジストリ データ解説書
- デジタル庁 アドレス・ベース・レジストリ

HTNG Express

- HTNG Express Specification Ver 1.0 (OpenAPI/YAML)
- HTNG Express Reservation Data Model
- HTNG Express Room Data Model
- HTNG Express Implementation Guide

HTNG Kiosk

- HTNG Kiosk Integration Specification v2.1 (2013B) - PDF 技術仕様書
- HTNG Kiosk 業務フロー定義書
- XSD スキーマ定義ファイル
- WSDL サービス定義ファイル

その他参考資料

- ISO 8601: 日付・時刻フォーマット
- ISO 3166: 国コード
- ISO 4217: 通貨コード
- ISO 639: 言語コード
- schema.org: 構造化データ語彙