

**日立システム**

**鉄道情報システム株式会社殿における、  
COBOL - XML連携システム開発事例**

**株式会社 日立システムアンドサービス  
アプリケーションソリューション事業部  
東京AS第2部 岩城 祥一**



# *Contents*

---

- 1 顧客会社概要
- 2 マルスとは・・・
- 3 マルスにおけるCOBOL - XML連携システム
- 4 なぜ、COBOL？
- 5 マルスでの適用例
- 6 マルスにおけるCOBOL - XML連携適用評価

# *Contents*

---

- 1** 顧客会社概要
- 2 マルスとは…
- 3 マルスにおけるCOBOL - XML連携システム
- 4 なぜ、COBOL？
- 5 マルスでの適用例
- 6 マルスにおけるCOBOL - XML連携適用評価

# 1

## 顧客会社概要

社名 **鉄道情報システム株式会社**

資本金 **10億円**

社員数 **516名(H15.4現在)**

設立年月日 **1986年12月9日**

**JR分割民営化を機に「みどりの窓口」のマルスを承継**

**社会公共性の高いシステムの開発・運営**

# Contents

---

- 1 顧客会社概要
- 2 マルスとは…
- 3 マルスにおけるCOBOL - XML連携システム
- 4 なぜ、COBOL？
- 5 マルスでの適用例
- 6 マルスにおけるCOBOL - XML連携適用評価

# 2

## マルスとは・・・



**21世紀の社会ニーズに適応するサービス提供**

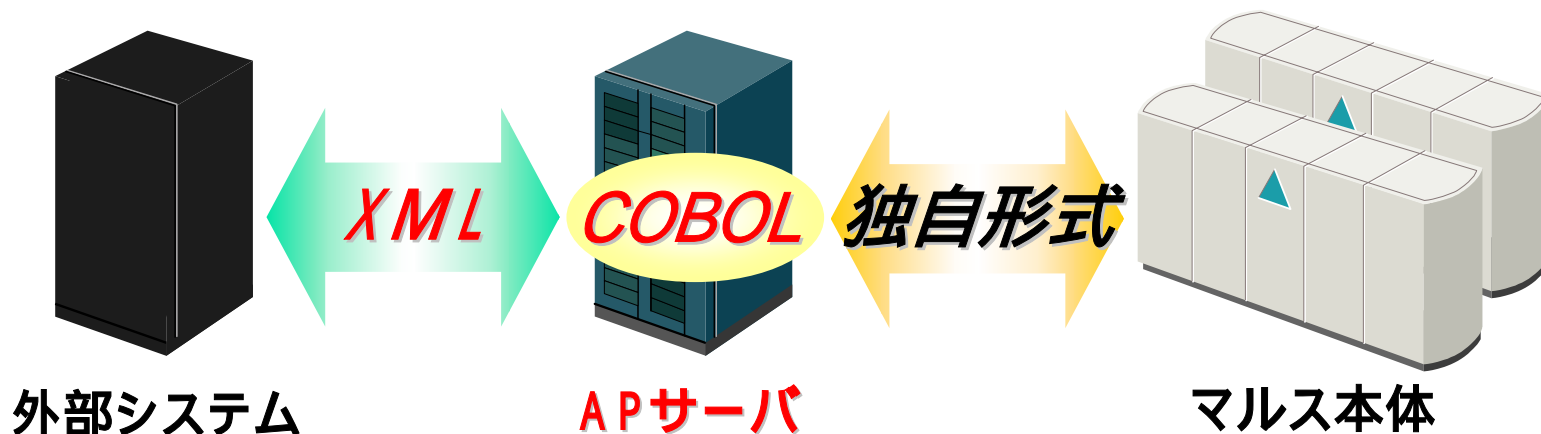
# Contents

---

- 1 顧客会社概要
- 2 マルスとは…
- 3 マルスにおけるCOBOL - XML連携システム
- 4 なぜ、COBOL？
- 5 マルスでの適用例
- 6 マルスにおけるCOBOL - XML連携適用評価

# 3

## マルスにおけるCOBOL - XML連携システム システム概要



- ・ マルス本体と連携し外部システムの要求に応えるオンラインシステム
- ・ オンラインインタフェースとしてXMLを採用
- ・ マルスのサブシステムとして、ミッションクリティカルなシステム構成



## XMLの特徴

XML (*Extensible Markup Language*) は、W3C<sup>1</sup>の提唱するドキュメント形式に関する規格で、拡張性に優れている

- ・ タグを自由に定義できる
- ・ 複雑な構造をもったデータも記述できる
- ・ プラットホームに依存しない自由なデータ交換

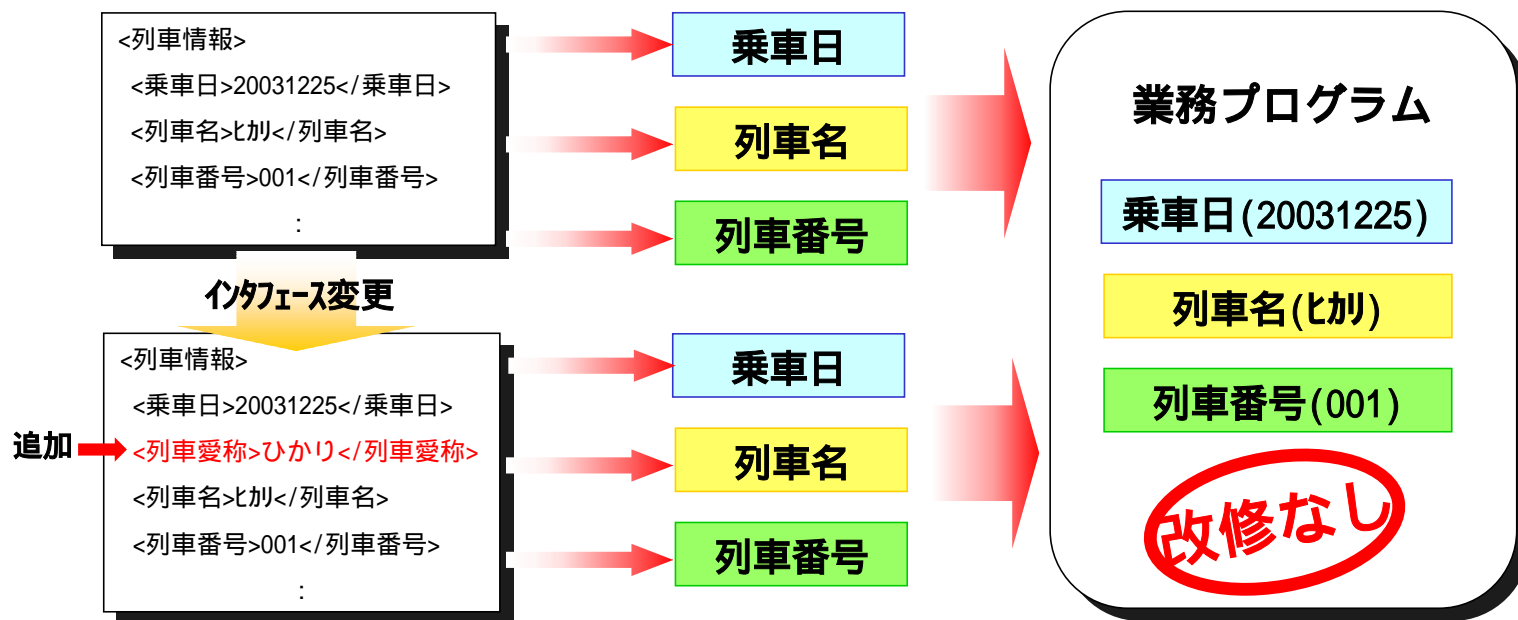
1: World Wide Web Consortium

- ・ インタフェース変更柔軟に対応できる
- ・ 一目でわかりやすいインタフェース
- ・ データ交換技術としての将来性
- ・ Webシステムとの親和性がある

# 3

## マルスにおけるCOBOL - XML連携システム XML採用ポイント(1) インタフェース変更柔軟に対応

### 必要な情報だけをピックアップできる



インタフェース項目追加（タグ追加）が発生しても、  
取得するデータ項目に変更がない場合、業務プログラムの改修は不要

## 一目でわかりやすいインタフェース

## 従来の独自形式と異なり一目でわかりやすい

## 独自形式

20031225比加 001...



列車番号は？

えーっと

先頭から13バイト目から3バイト...

## XMLインタフェース

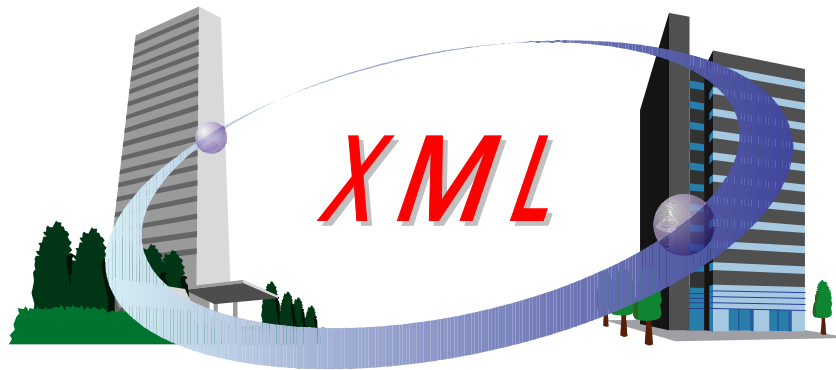
```
<列車情報>
<乗車日>20031225</乗車日>
<列車名>比加</列車名>
<列車番号>001</列車番号>
:
</列車情報>
```



列車番号は、“001”

## データ交換技術としての将来性

- ・ EDI(電子データ交換)やアプリケーション間連携のための標準データフォーマットとして適用範囲が拡大
- ・ e-ビジネスでの企業間相互接続(BtoB)のデータ交換技術として注目



## Webシステムとの親和性がある

- ・ 将来的にWebシステムとの連携を視野に・・・  
Webシステムとの親和性のあるデータ交換技術を・・・



# Contents

---

- 1 顧客会社概要
- 2 マルスとは…
- 3 マルスにおけるCOBOL - XML連携システム
- 4 なぜ、COBOL？
- 5 マルスでの適用例
- 6 マルスにおけるCOBOL - XML連携適用評価

# 4

## なぜ、COBOL？

### 生産性が高く保守性に優れたCOBOL

- ・ COBOL既存資産の再利用
- ・ COBOL技術者が多い
- ・ 業務ロジックを誰が書いても均一な品質と性能



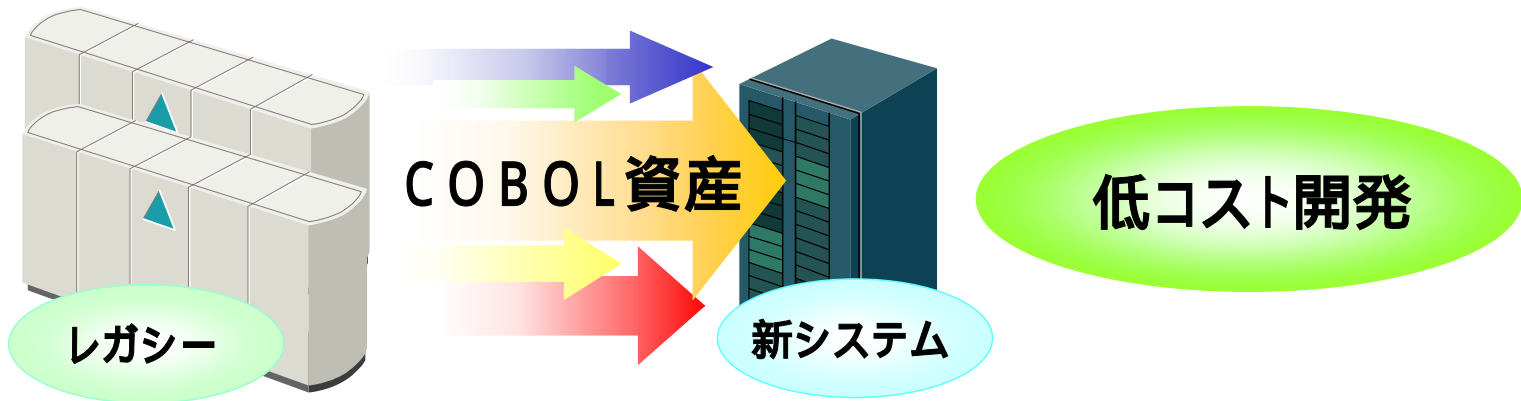
# 4

なぜ、COBOL? (1)

## COBOL既存資産の再利用

マルスにおける業務ロジックの多くがCOBOLで記述されている

既存の業務ロジックを再利用したい



# 4

なぜ、COBOL？(2)

## COBOL技術者が多い

マルス開発担当者の多くがCOBOL技術者である

大規模システム構築

システム構築の  
ノウハウが豊富



豊富な人材

# 4

なぜ、COBOL？(3)

業務ロジックを誰が書いても均一な品質と性能

他言語と比較し担当者間でのスキルの差が出にくい  
業務ロジックを記述する上でパターン化されている



均一な品質と性能



大規模システム構築

保守性の高い  
システム構築

# 4

なぜ、COBOL？

## XML連携における問題

COBOLプログラムでXMLデータを扱うにはどうすればよいか？  
一般的なXMLデータ処理方式は、DOM / SAXプログラミング



COBOLの言語特性上、  
DOM / SAXプログラミングは困難

**DOM**: XMLパーサでメモリ展開したXMLデータのツリー構造を扱う  
動的メモリをポインタで扱うプログラミングが必要

**SAX**: XMLデータを走査しながら特定の要素を見つけイベント化  
イベント起動型のプログラミングが必要

# 4

なぜ、COBOL?

## XML連携における問題

### 問題解決

## COBOL - XML連携製品の登場

COBOL - XML連携製品『*COBOL Adapter for XML*』により、  
XMLデータアクセスの複雑なプログラミングが不要



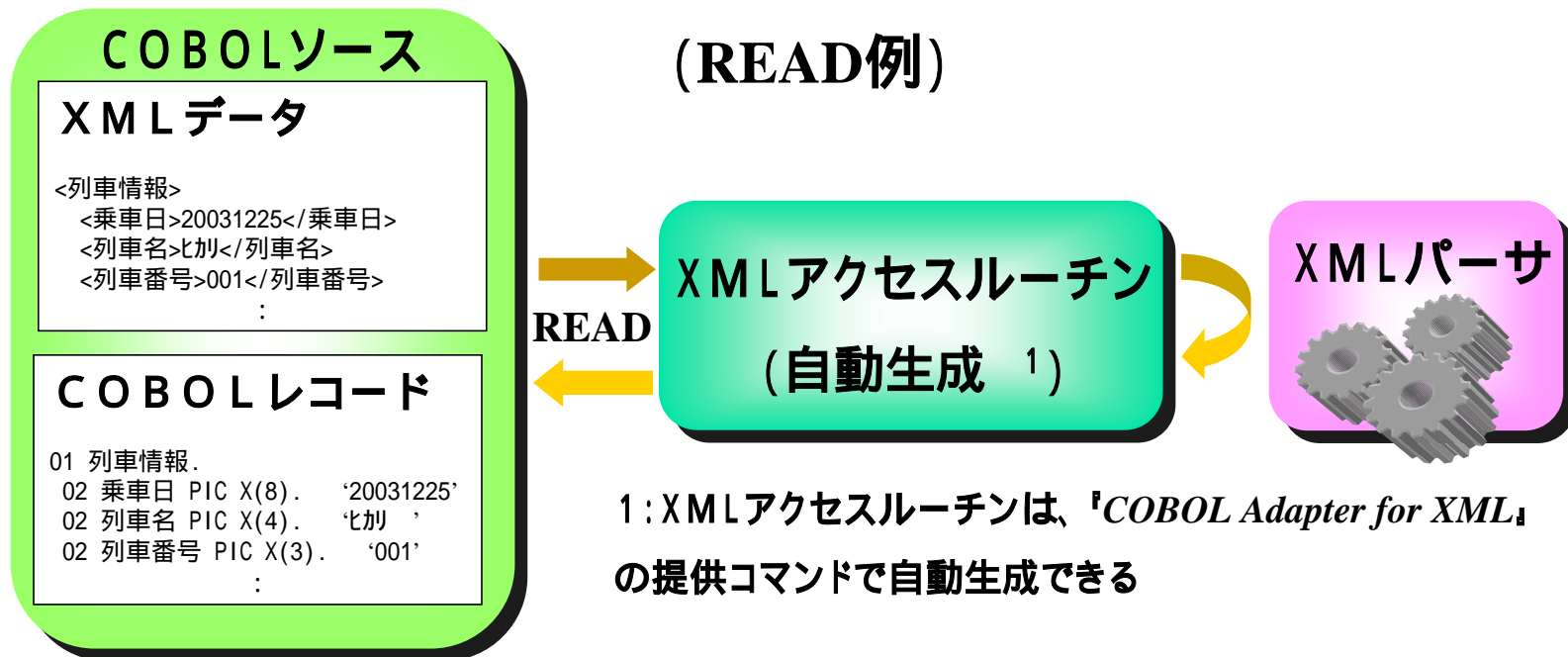
# 4

なぜ、COBOL?

## COBOL Adapter for XML 概要

### 基本機能

XMLデータをCOBOLのレコードとして、ファイルI/Oの要領でCOBOLプログラムから入出力できる



## 豊富なエラーチェック機能

- ・XMLの妥当性検証
- ・不当文字・オーバーフローの検出
- ・エラー発生位置の取得(パスを通知)



# Contents

---

- 1 顧客会社概要
- 2 マルスとは…
- 3 マルスにおけるCOBOL - XML連携システム
- 4 なぜ、COBOL？
- 5 マルスでの適用例
- 6 マルスにおけるCOBOL - XML連携適用評価



TP モニタ

*OpenTP1*

開発言語

*COBOL85*

XML パーサ

*COBOL Adapter for XML*

DBMS

*HiRDB*

CASE ツール

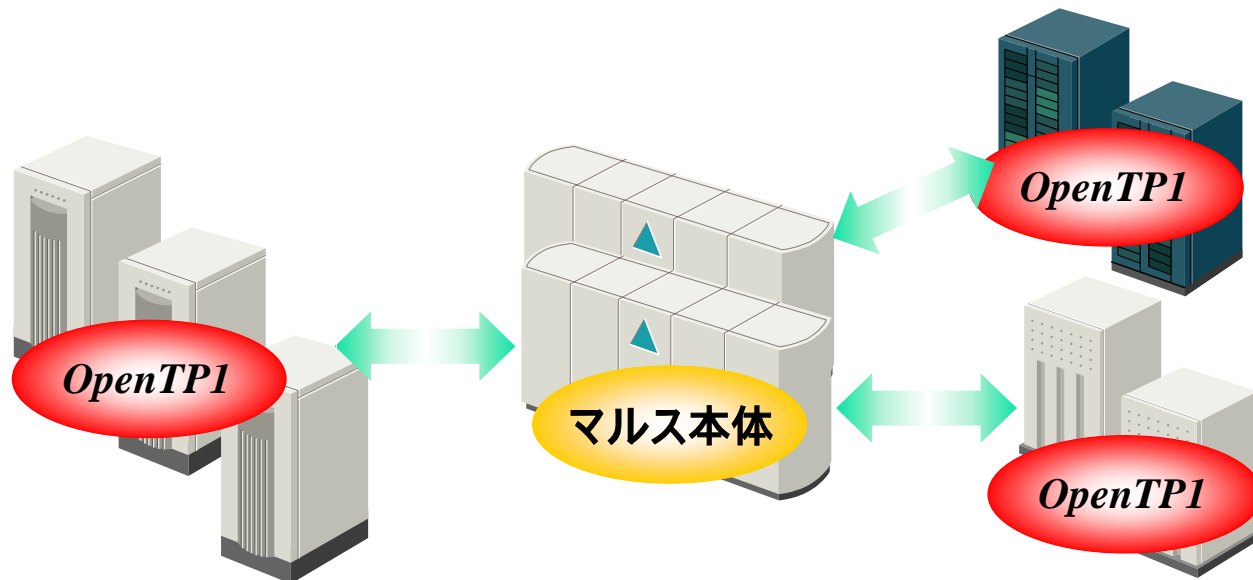
*SEWB+*

統合運用管理

*JP1*

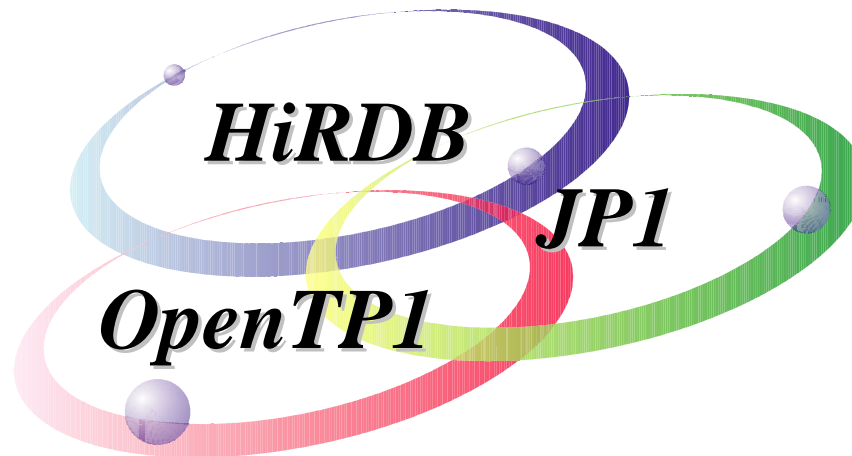
## OpenTP1 概要

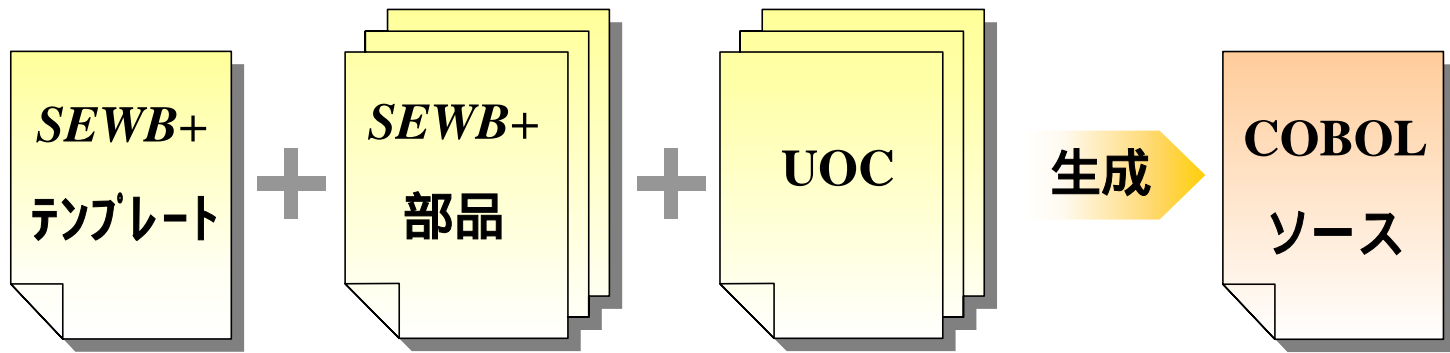
- ・ メインフレームと同等の信頼性
- ・ **COBOLフルサポートのAPI**



## *HiRDB/JP1* 概要

- ・ *HiRDB* ミッションクリティカル対応された RDBMS
- ・ *JP1* システム運用と管理を一元的に実現



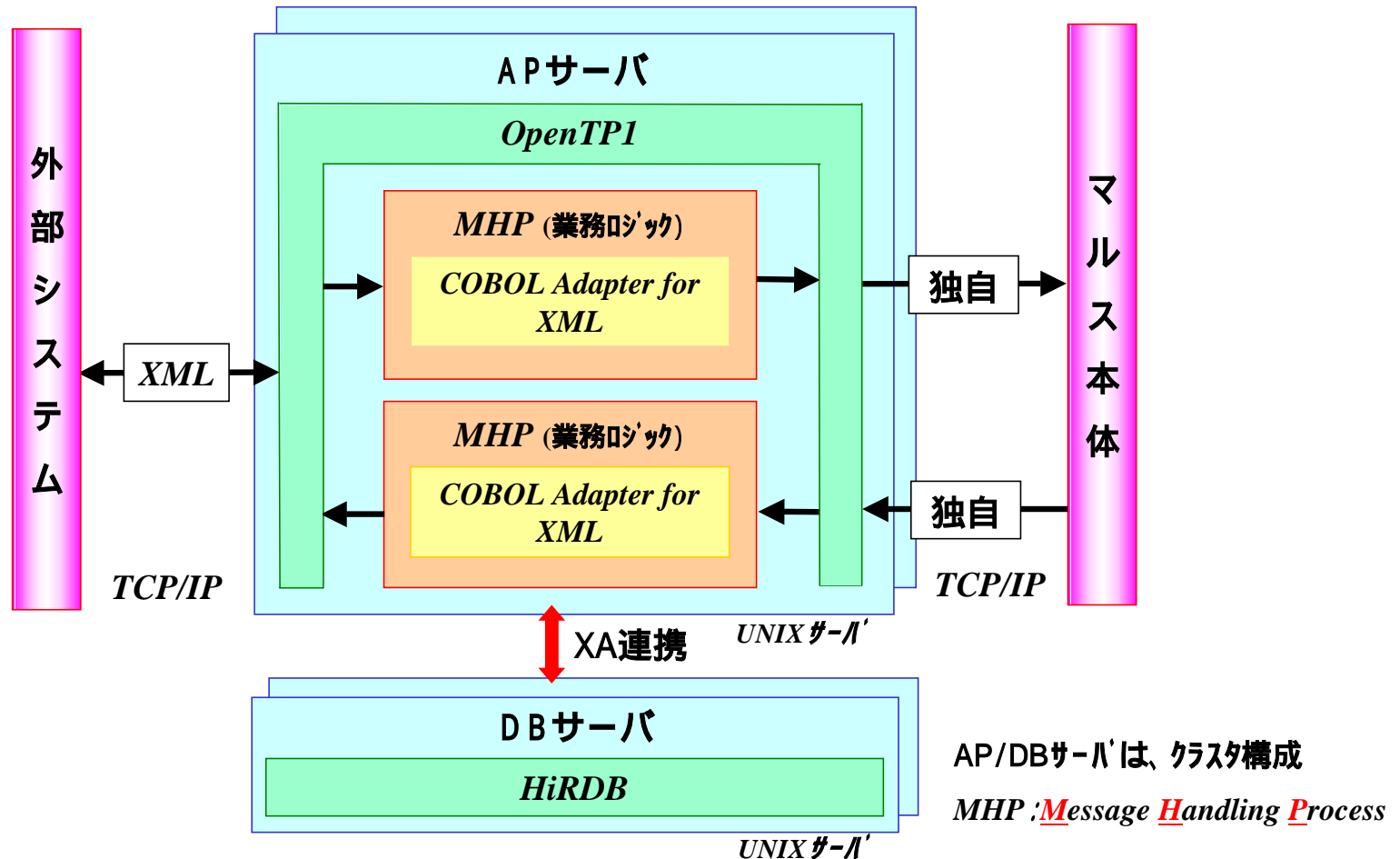
**・ プログラム自動生成機能を適用**

ひとつのテンプレートに複数の部品を取込みCOBOLソースを生成する  
プログラム固有の処理は、UOC(ユーザ追加コーディング)で補う

# 5

マルスでの適用例 システム構成(4)

## システム実装方式



# 5

マルスでの適用例

## アプリケーション構築手順

### 第1ステップ

#### DTD/DDF作成

DTD(文書型定義)

DDF(データ定義)

### 第2ステップ

#### XMLアクセスルーチン /データ定義 自動生成

「*COBOL Adapter for XML*」のコマンドを使用

データ定義とは、XMLアクセスルーチンの入出パラメタ

### 第3ステップ

#### XMLパース 処理作成

XMLのパース手順を簡易にするための共通モジュールの作成

「*SEWB+*」のプログラム自動生成機能を使用

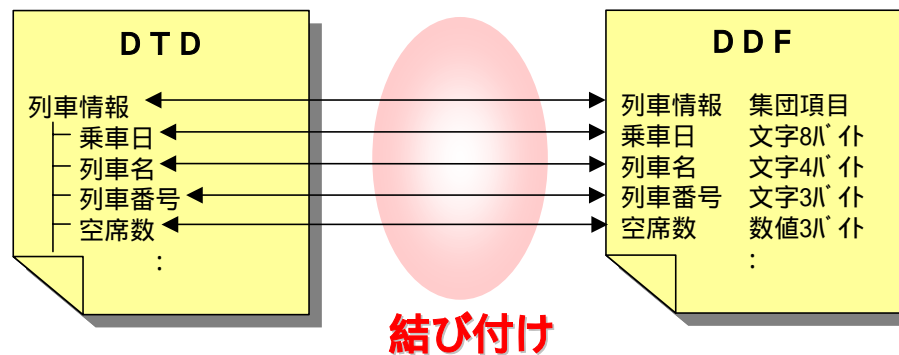
### 第4ステップ

#### ロードモジュール 作成

## DTD / DDF作成

DTD (文書型定義) : XMLデータの構造を定義する

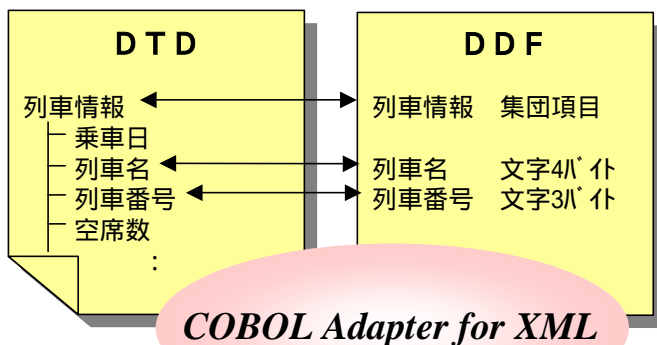
DDF (データ定義) : XML要素とCOBOLデータ項目と  
を結び付ける対応情報を定義



## アプリケーション構築手順(第1ステップ)

## DDFは、各要求インタフェース単位に作成

- ・ DDFをひとつにするとデータ定義サイズが膨大に・・・
- ・ XMLの柔軟性を活かすため
- ・ XMLパース処理性能を考慮



生成

## データ定義 (COPY句)

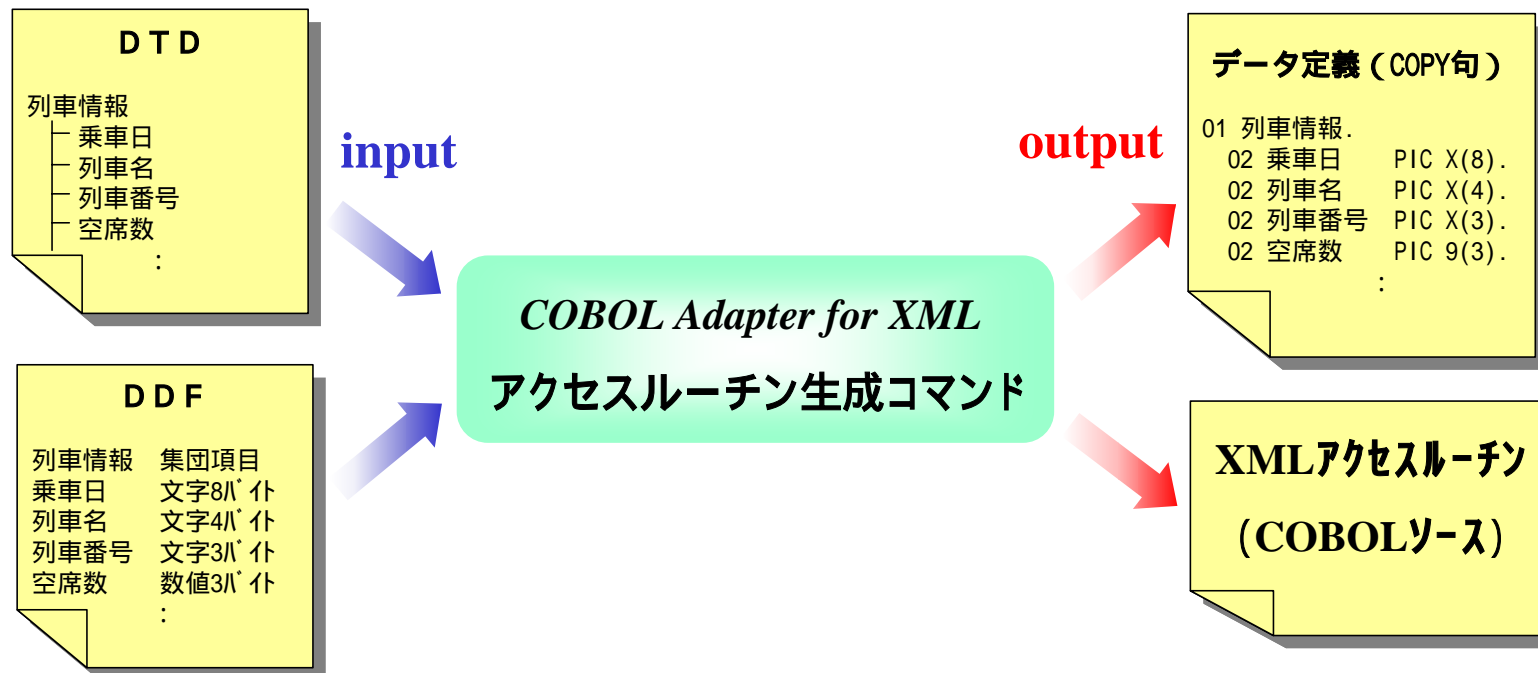
```
01 列車情報.
02 列車名    PIC X(4).
02 列車番号  PIC X(3).
:
```

データ定義は、XMLアクセスルーチンの入出力パラメタ

DDFに定義されていない要素は、データ定義されない、パースされない



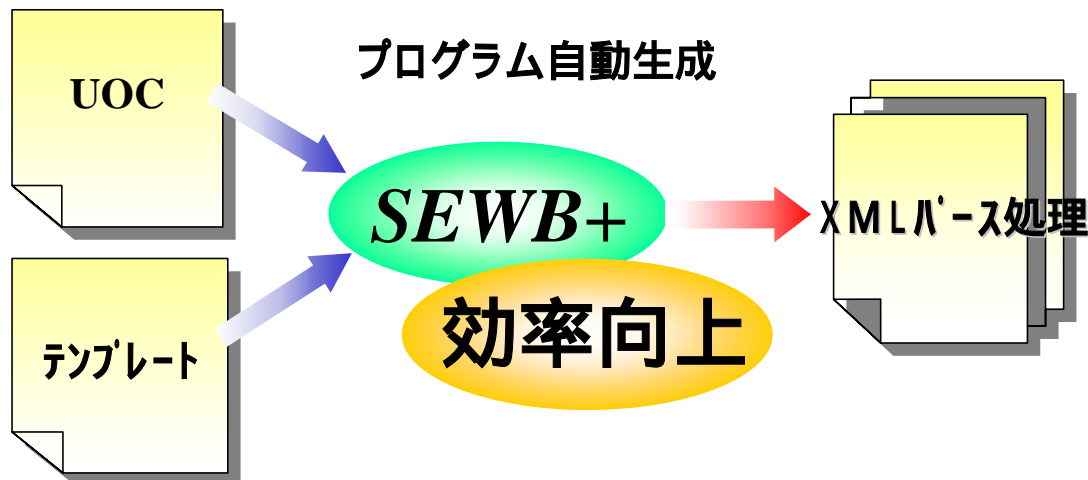
## XMLアクセスルーチン / データ定義自動生成

『*COBOL Adapter for XML*』のコマンドにより自動生成

## アプリケーション構築手順(第3ステップ)

XMLパース処理手順を簡易にするため、共通モジュールを作成

XMLパース処理は、要求インタフェースごと(DDF単位)に作成  
コールするXMLアクセスルーチンが異なるだけでワンパターン  
『SEWB+』のプログラム自動生成機能を使用し効率向上



# 5

マルスでの適用例

## アプリケーション構築手順(第4ステップ)

### ロードモジュール作成

業務プログラム

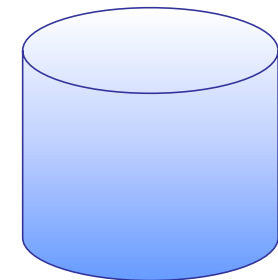
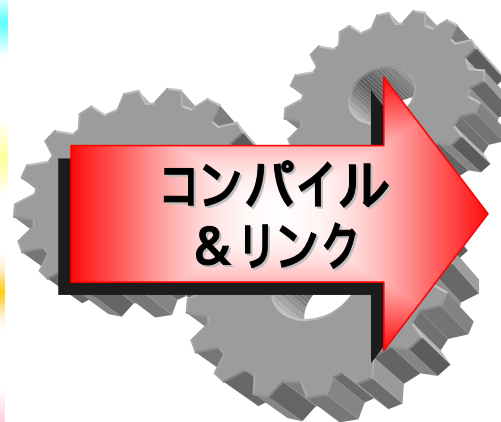
XMLアクセスルーチン

データ定義(COPY句)

XMLパース処理

各種ライブラリ

OpenTP1ライブラリなど



ロードモジュール

# *Contents*

---

- 1 顧客会社概要
- 2 マルスとは…
- 3 マルスにおけるCOBOL - XML連携システム
- 4 なぜ、COBOL？
- 5 マルスでの適用例
- 6 マルスにおけるCOBOL - XML連携適用評価

## 親和性の評価

### マルスにおける、COBOL - XML連携適用評価

COBOL - XML連携は、親和性が悪いが・・・

『*COBOL Adapter for XML*』採用により、**親和性が良くなる**

パース後のCOBOLレコードは、英数字、日本語、数値項目のデータ属性をサポート  
繰り返し要素は、OCCURS句で対応など・・・ 従来の開発と同様、違和感がない

**XMLを意識することなく業務ロジック開発に専念できる**

## 性能の評価

### マルスにおける、COBOL - XML連携適用評価

COBOL - XML連携は、

独自形式インタフェース連携より**性能が劣る**

XMLパースには、CPUパワーが必要。パースする要素は最小限にする必要がある  
独自形式と比較しメッセージ長が長い。ネットワーク負荷も考慮にいれる必要がある

**パースする要素は、必要最小限にすること**

## 生産性の評価

### マルスにおける、COBOL - XML連携適用評価

COBOL - XML連携は、

『*COBOL Adapter for XML*』のエラーチェック機能により、  
独自形式インタフェース連携より入力データの検定処理を  
簡易にすることができる

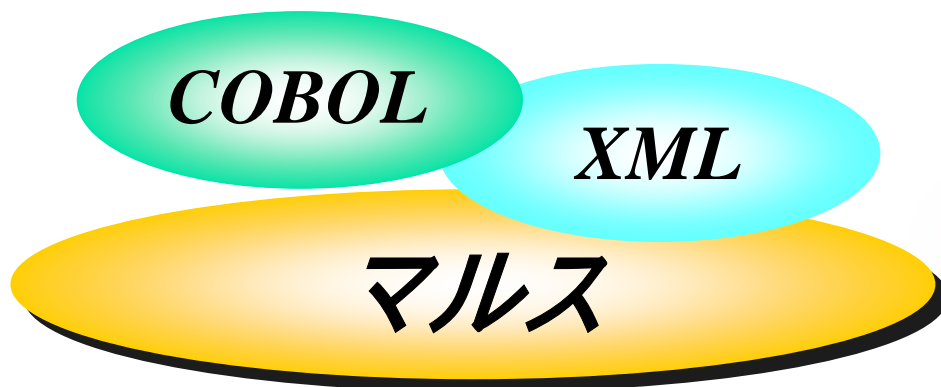
『*COBOL Adapter for XML*』が面倒な入力データ検定処理を行う

入力データ検定処理にかかる工数を低減

## 総合評価

### マルスにおける、COBOL - XML連携適用評価

COBOL - XML連携は、性能面での留意点はあるが、システム要件に柔軟に対応できる実用性のあるシステムを構築することができる



**新たな選択肢  
の誕生**

21世紀の社会ニーズに適應するサービス提供の可能性が広がる



# 日立システム

## 鉄道情報システム株式会社殿における、 COBOL - XML連携システム開発事例

株式会社 日立システムアンドサービス  
アプリケーションソリューション事業部  
東京AS第2部 岩城 祥一

