

# **データ連携基盤に向けて(5)**

## **企業間情報交換を容易に実装するために**

2022年3月1日

# 目次

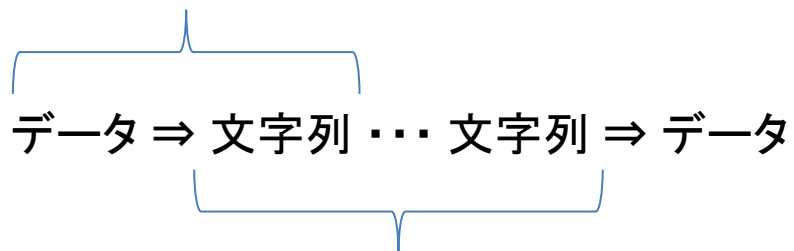
1. データ連携基盤とは
2. ビジネスコンポーネント(BC)ライブラリ  
プログラム内のデータを文字列に相互変換する仕組み
3. 相互変換ソフトウェア  
プログラム内のデータを文字列に相互変換する仕組み
4. BDAA変換ソフトウェア  
ビジネス文書プロファイル合意形成(BDAP/BDAA)

# データ連携基盤とは

再掲

連携元のプログラム内のデータを連携先のプログラムにコピーする仕組みを簡単に作るソフトウェア群

## 1) プログラム内のデータを文字列に相互変換する仕組み



## 2) 文字列をプログラム間で相互伝達する仕組み (通信等となるが、本資料では詳細は割愛)

# プログラム内のデータを文字列に相互変換する仕組み(1)

再掲

データとは

- ・ 取り敢えず、取引情報
  - 国連CEFACT CCL (Core Component Library)に基づくデータモデル
  - 約2万個の情報項目
  - (国連CEFACTおよびSIPSが定義しているコードリスト、IDリスト一覧も含む)
- ・ 機能: 生成、コピーなど
- ・ 言語: 取り敢えず、JAVA
- ・ 成果物:
  - ビジネスコンポーネント(BC)ライブラリ

# ビジネスコンポーネント(BC)ライブラリ(1)

- ・ 国連CEFACT CCL (Core Component Library)に基づくデータモデル  
約2万個の情報項目(22937個:CCL 21A)からデータモデル化
  - ・ ABIE: クラスに対応(1330個)
  - ・ BBIE: 基本型の配列型のインスタンス変数に対応(8434個)
  - ・ ASBIE: クラス型の配列型のインスタンス変数に対応(4457個)
  - ・ uDT: 基本型(20個)「コード型、ID型、単位付き数量型、文字列型など」その他に国連CEFACTおよびSIPSが定義しているコードリスト、IDリスト一覧も含む

## ・ ビジネスコンポーネント

ABIEをACC単位に集約したクラス(**352個**)で構成、(ACCは606個)

基本型(**19個**)を含む

- ・ uDT Picture と ABIE Picture が重複し、かつ uDT Picture は未使用のため削除

## ・ ABIEコンポーネント

BCのメソッドを限定したABIE単位のクラス(**1330個**)

- ・ 集約によりBCのメソッドが多くなったため、間違ったメソッド利用をコンパイル時チェックで検出可能とする(例えば、BCのDocumentは281組×6個のメソッド以上を持つ)

# ビジネスコンポーネント(BC)ライブラリ(2-1)

- 機能:

- 生成 ( new Xxxx() )

- ~~コピー ( cp( ) )~~

この後実装予定

- 変数代入 ( stXxxx( i, オブジェクト ) stXxxx( オブジェクト ) )

nは要素数

i: 0~n-1 i番目の要素を置き換える

i: n~ 最後に追加

i: -1 最後に追加

i: -2 最後の要素を置き換える

i: -n-1~-2 -i-1番目の要素を置き換える

i: ~ -n-1 最初に追加

# ビジネスコンポーネント(BC)ライブラリ(2-2)

- ・ 機能:

- ・ 変数参照 ( gtXxxx(i) -> オブジェクト gtXxxx() -> オブジェクト )

nは要素数

i: 0~n-1 i番目の要素を返す

i: n~ ヌル要素を返す

i: ~ -1ヌル要素を返す

- ・ 変数追加 ( ptXxxx(i) -> オブジェクト ptXxxx() -> オブジェクト )

既にオブジェクトがあればその値、無ければ生成

nは要素数

i: 0~n-1 i番目の要素を返す

i: n~ 最後に追加し、返す

i: -1 最後に追加し、返す

i: -2 最後の要素を返す

i: -n-1~-2 -i-1番目の要素を返す

i: ~ -n-1 最初に追加し、返す

- ・ 言語: JAVA

# ビジネスコンポーネント(BC)ライブラリ(3)

再掲

## ・開発方針

コンパクトに書ける。

## Set(stXxxx)を使う場合

```
AccessContollist acl = new AccessContollist()
    .stAccessRightCode(new Code()
        .setValue("1")
        .setlistID("10001"))
    .stTypeCode(new Code()
        .setValue("99")
        .setlistID("20001"))
    .stIncludedLaboratoryObservationParty(new Party()
        .stName(new Text()
            .setValue("名前"))
        .stCustomerID(new ID()
            .setValue("A0001")));
```

## 通常の場合

```
AccessContollist acl = new AccessContollist();
    Code c1 = new Code();
    c1.setValue("1");
    c1.setlistID("10001");
    acl.setAccessRightCode(c);
    Code c2 = new Code();
    c2.setValue("99");
    c2.setlistID("20001");
    acl.setTypeCode(new Code());
    Party p = new Party();
    Text tx = new Text();
    tx.setValue("名前");
    p.setName(tx);
    ID id = new ID();
    id.setValue("A0001");
    p.setCustomerID(id);
    acl.setIncludedLaboratoryObservationParty(p);
```



# ビジネスコンポーネント(BC)ライブラリ(4)

再掲

## ・開発方針

コンパクトに書ける。

## Put(ptXxxx)を使う場合

```
AccessContollist acl = new AccessContollist()

acl.ptAccessRightCode()
    .setValue("1")
    .setlistID("10001")

acl.ptTypeCode()
    .setValue("99")
    .setlistID("20001"))

acl.ptIncludedLaboratoryObservationParty()
    .ptName()
    .setValue("名前")

acl.ptIncludedLaboratoryObservationParty()
    .ptCustomerID()
    .setValue("A0001"));
```

## 通常の場合

```
AccessContollist acl = new AccessContollist();
    Code c1 = new Code();
        c1.setValue("1");
        c1.setlistID("10001");
    acl.setAccessRightCode(c);
    Code c2 = new Code();
        c2.setValue("99");
        c2.setlistID("20001");
    acl.setTypeCode(new Code().
        Party p = new Party();
            Text tx = new Text();
                tx.setValue("名前");
            p.setName(tx);
            ID id = new ID();
                id.setValue("A0001");
            p.setCustomerID(id);
    acl.setIncludedLaboratoryObservationParty(p);
```

# プログラム内のデータを文字列に相互変換する仕組み(2)

再掲

文字列とは

- ・ 取り敢えず、XMLメッセージ（次にJSONメッセージ）
- ・ 固定長メッセージ、CSVメッセージは、今後の課題

取引情報的には

- ・ 国連CEFACTメッセージ（CCBDA、NDRに従ったXMLメッセージ）
- ・ 各種EDIのXMLメッセージ（選定中）

# プログラム内のデータを文字列に相互変換する仕組み(3)

再掲

相互変換とは

BC(データ) ⇔ メッセージ(文字列)

相互変換の課題

- ・ メッセージからBCへの変換:メッセージ内の要素に対応するBC内の要素が無い  
CCLに新規追加(CCLの更新は年2回、追加は原則受理されている)
- ・ BCからメッセージへの変換: BC内の要素に対応するメッセージ内の要素が無い  
メッセージに無い要素は当該取引では使用しないため問題では無いが、  
使えるBCの要素を確認する機能(Validation)が必要

# プログラム内のデータを文字列に相互変換する仕組み(4)

再掲

## 相互変換の成果物

- ・ 国連CEFACTメッセージ ⇔ BC  
汎用国連CEFACTメッセージマッピングソフトウェア  
各国連CEFACTメッセージに対応するValidationソフトウェア
- ・ 各種EDIのXMLメッセージ ⇔ BC  
メッセージ定義から生成したマッピングソフトウェア、Validationソフトウェア  
(メッセージ定義は、XMLスキーマ生成用を流用予定)

## 検証方法

- ・ BC、マッピングソフトを使ったプログラムを作成し、評価する。  
(ISO20022導入ガイドの商流情報処理に適応予定)

# 相互変換ソフトウェア(1)

- ・ 国連CEFACTメッセージ ⇔ BC
  - ・ 汎用国連CEFACTメッセージマッピングソフトウェア
  - ・ 各国連CEFACTメッセージに対応するValidationソフトウェア
- 国連CEFACTおよびSIPSに登録されているメッセージ

**Unmarshal()** XML -> オブジェクト(木構造型)

CCLに登録されていない要素は、XMLデータとしてBC内に格納

**Marshal()** オブジェクト(木構造型) -> XML

CCLに登録されている要素だけをXML化

この後実装予定

~~validate()~~

deep/shallow 配下のオブジェクトも検証/そのオブジェクトのみ検証

upper/lower/restricted 上限数制約/下限数制約/制限制約

MAはXML名前空間名の検証

# 相互変換ソフトウェア(2)

## gMAクラス

汎用国連CEFACTメッセージマッピング(BCへ)

- **Unmarshal**

XMLメッセージから gMA/BC/DT に変換  
XMLの断片からBC/DTへの変換も可能

- **Marshal**

gMA/BC/DT からXMLメッセージに変換  
BC/DT から XMLの断片への変換も可能

## MAクラス

汎用国連CEFACTメッセージマッピング(BIEへ)

- **Unmarshal**

XMLメッセージから MA/BIE/DT に変換  
XMLの断片からBIE/DTへの変換も可能

- **Marshal**

MA/BIE/DT からXMLメッセージに変換  
BIE/DT から XMLの断片への変換も可能

# 相互変換ソフトウェア(3)

この後実装予定

- ・ CCBDAメッセージクラス(メッセージのルートタグ名のクラス)  
国連CEFACTおよびSIPSに登録されているメッセージマッピング

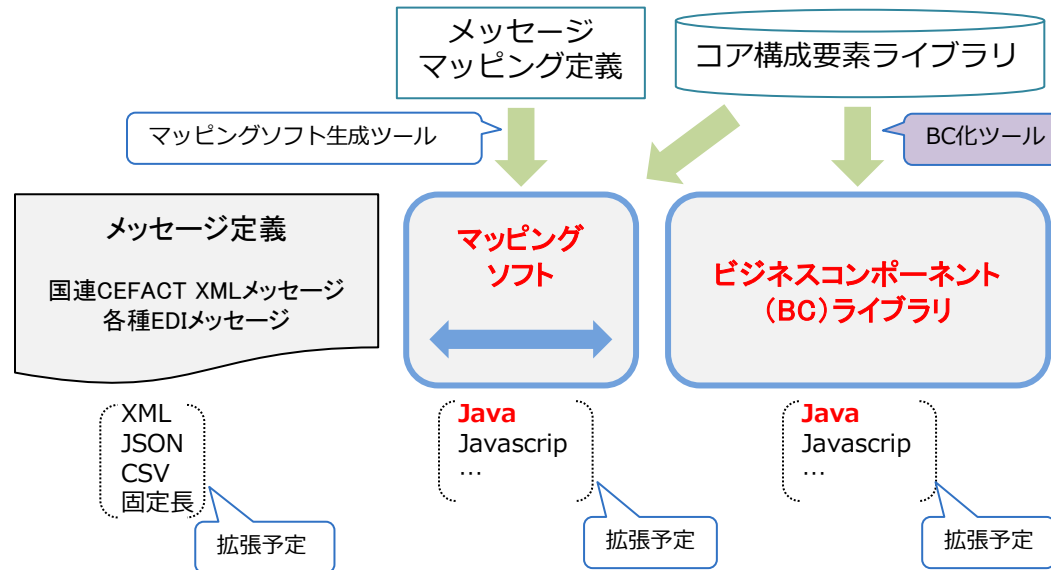
この後実装予定

- ・ 各種EDIのXMLメッセージ ⇔ BC  
メッセージ定義から生成したマッピングソフトウェア  
Validationソフトウェア  
(メッセージ定義は、XMLスキーマ生成用を流用予定)

# プログラム内のデータを文字列に相互変換する仕組み(5)

再掲

## 成果物(予定)の構成



## 成果物の配布方法

- ・ 使用制限の無いオープンソース (**Apache 2.0**)
- ・ 当面はSIPS内で配布
- ・ Zipファイル、**Github**

配布方法は検討中のため今後変更となる可能性があります。

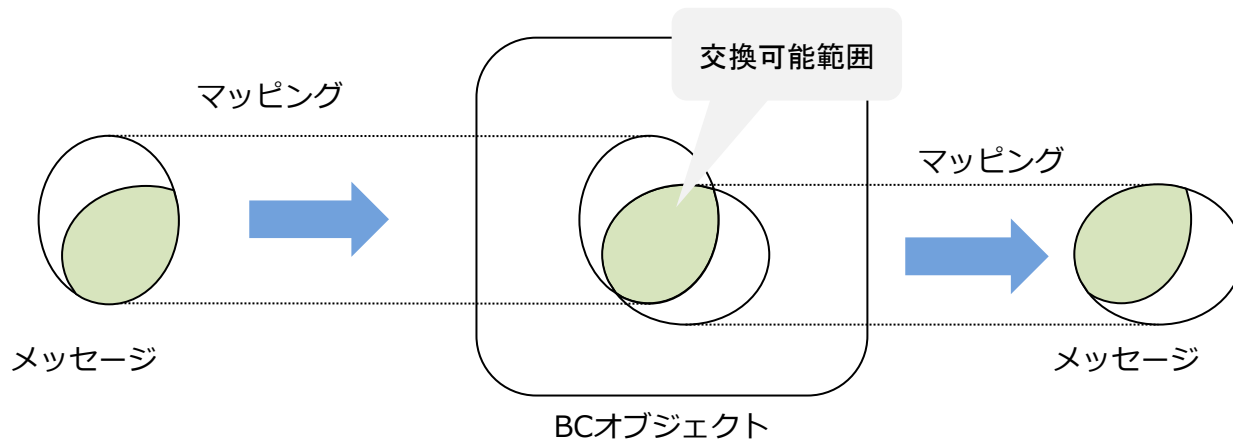


# プログラム内のデータを文字列に相互変換する仕組み(6)

再掲

## メッセージ間の相互変換の課題

国連CEFACTメッセージ ⇔ BC … BC ⇔ 各種EDIメッセージ

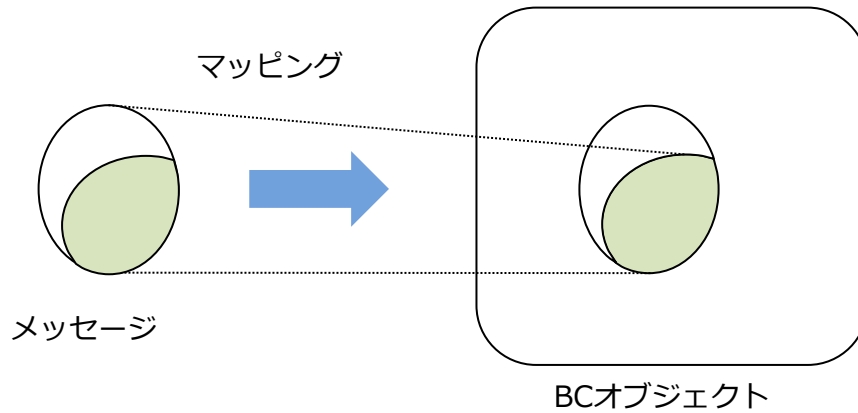


ビジネス文書プロファイル合意形成( BDAP/BDAA )で交換可能範囲を拡張

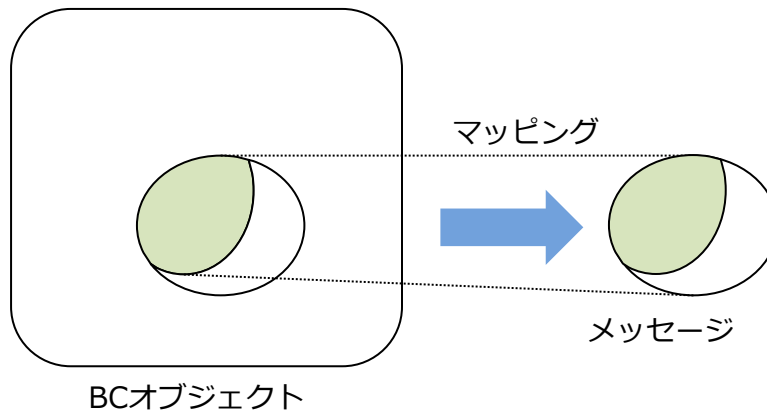
- ・ BDAAマッピングソフトウェアを相互変換の成果物とする予定

# BDAA変換ソフトウェア

再掲



メッセージから交換可能範囲のBCへのマッピング



交換可能範囲のBCからメッセージへのマッピング

# 活動スケジュール

