4P-6

AWS ミドルウェアにおける電子帳票の互換性決定方法

安斎　太一朗†　　二宮　良太†　　大谷　真†

湘南工科大学†

1. はじめに

本研究の目的は自律型 Web サービス(AWS: Autonomous Web Services)[1]における、メッセージ形式(電子帳票様式)の互換性判定の実現である。AWS におけるメッセージ交換は、帳票様式に従ったメッセージデータの送受信によって行われる。本論文は帳票様式について、セマンティック Web 技術をベースとした互換性決定方法を提案する。

2. AWS ミドルウェアと**tMatch**()

AWS ミドルウェア[1]とは、自由な電子商取引を実現する次世代のソフトウェア基盤である。AWS は動的モデル協調(DMH:Dynamic Model Harmonization)を中心にその機能を実現している。この DMH の中で、オペレーションマッチング判定に互換性決定関数**tMatch**()を用いる。$\mathbf{tMatch}(\mathbf{f_{out}}, \boldsymbol{g_{in}})$は、以下の様に帳票様式$\mathbf{f_{out}}, \boldsymbol{g_{in}}$に対する 3 値関数である。

$$\mathbf{tMatch}(\mathbf{f_{out}}, \boldsymbol{g_{in}}) = \{\mathbf{true}, \mathbf{false}, \mathbf{undefined}\}$$

tMatch()の実装には様々なものが考えられるが、本研究ではこの**tMatch**()についての帳票様式に対するメタデータ定義及び、その情報収集と推論による決定方法を提案する。

3. 帳票様式の互換性決定方法

3.1 互換性について

帳票様式とは、システム間でやりとりするメッセージの作成規則であり、メッセージデータはこれに従って作成される。ここで、一方($\mathbf{f_{out}}$)が送信したメッセージを他方($\boldsymbol{g_{in}}$)が受信できる場合において互換性が有ると定義し、本論文では「適合している」と表現する。「適合している」と判定できた時に$\mathbf{tMatch}() = \mathbf{true}$となる。

3.2 帳票様式の定義

本研究において帳票様式を単項目様式と多項目様式の 2 つに分類し、それぞれに任意の名前を与え、これを帳票 ID と呼ぶ。単項目様式とは、単体のデータの様式であり、一つの帳票 ID から成る。多項目様式とは、複数の単項目様式または多項目様式で合成された帳票様式で、複数の帳票 ID(子 ID)をグループ化し、それらに親となる帳票 ID(親 ID)を与えることで多項目様式を表現する。子 ID 側の様式を項目と呼ぶ。

3.3 多項目様式の合成表現

多項目様式$\mathbf{f_{out}}, \boldsymbol{g_{in}}$を、それぞれ項目$\mathbf{A_1} \cdots \mathbf{A_n}$, $\mathbf{B_1} \cdots \mathbf{B_m}$によって以下のように表現する。

$$\mathbf{f_{out}} \Leftarrow \{(\boldsymbol{\tau_1}, \mathbf{A_1}) \oplus (\boldsymbol{\tau_2}, \mathbf{A_2}) \oplus \cdots \oplus (\boldsymbol{\tau_n}, \mathbf{A_n})\}$$
$$\boldsymbol{g_{in}} \Leftarrow \{(\boldsymbol{\varphi_1}, \mathbf{B_1}) \oplus (\boldsymbol{\varphi_2}, \mathbf{B_2}) \oplus \cdots \oplus (\boldsymbol{\varphi_m}, \mathbf{B_m})\}$$

$\oplus$:多項目様式の合成演算子　　$\boldsymbol{\tau}, \boldsymbol{\varphi}$:タグ

これを多項目様式の合成表現とし、合成時に、それぞれの親 ID から自らの帳票様式を識別するためのタグを付与する。$\boldsymbol{\tau_i}, \boldsymbol{\varphi_k}$はそれぞれ$\mathbf{f_{out}}, \boldsymbol{g_{in}}$に局所的な任意の ID とし、帳票 ID とは異なる ID を付与する。ここで$\boldsymbol{\tau_i} = \boldsymbol{\phi}$のとき$(\boldsymbol{\phi}, \mathbf{A_i})$は$\mathbf{A_i}$と表記できる。また、$\mathbf{A_i} = \mathbf{A_j}$なら$\boldsymbol{\tau_i} \neq \boldsymbol{\tau_j}$である。

3.4 様式間の関連情報

関連情報とは、異なる 2 つの帳票様式間の関連について、様式$\mathbf{X}$のインスタンスが様式$\mathbf{Y}$に適合するか否かを定義した情報のことである。ここでは、表 1 の 6 種類で関連情報を定義し、これを左辺と右辺に帳票 ID を持つ式で表現する。

表 1. 帳票様式間の関連情報

$\mathbf{X} \equiv \mathbf{Y}$	Xに「適合している」インスタンスは、全てYにも「適合している」かつ、Yに「適合している」インスタンスは、全てXにも「適合している」 すなわち、$(X \rightarrow Y)$and$(Y \rightarrow X)$
$\mathbf{X} \rightarrow \mathbf{Y}$	Xに「適合している」インスタンスは、全てYに「適合している」
$\mathbf{X} \nrightarrow \mathbf{Y}$	Xに「適合している」インスタンスは、全てYに「適合していない」
$\mathbf{not}(\mathbf{X} \rightarrow \mathbf{Y})$	Xに「適合している」あるインスタンスは、Yに「適合していない」
$\mathbf{not}(\mathbf{X} \equiv \mathbf{Y})$	Xに「適合している」あるインスタンスは、Yに「適合していない」かつ、Yに「適合している」あるインスタンスは、Xに「適合していない」
$\mathbf{X}?\mathbf{Y}$	Xに「適合している」インスタンスは、全てYへの適合が判定不能

関連情報は単項目様式、多項目様式のそれぞれの間で定義され、関連情報の一つひとつが web 上で公開されているものとする。この関連情報によって結ばれた帳票様式について、項目$\mathbf{A_i}$と関連情報を持つ項目$\mathbf{B_j}$が双方の多項目様式に対して一つだけ決まる場合に、$\mathbf{B_j}$を$\boldsymbol{\gamma}(\mathbf{A_i})$と表現する。この$\boldsymbol{\gamma}()$は上記のタグ付与に際して、XML タグ表現の一致や、メッセージ文書のシンタックスによって対応付けの実現が可能であるが、前もって対応先が決定していない場合もある。

A proposal on determination of format compatibility in AWS
† Taichiro Anzai , Ryota Ninomiya, Makoto Oya,
Shonan Institute of Technology

3.5 関連情報の収集

関連情報について、本研究では必要な情報を検索する為のシステムの存在を仮定する。検索システムに対して、帳票 ID と、その帳票 ID が左辺か右辺かの情報を入力することで、入力情報を満たす関連情報の一覧が出力されるとする。

3.6 $\mathbf{tMatch}(\mathbf{f_{out}}, \mathbf{g_{in}})$の実現

前述した 3 値関数**tMatch()**について関連情報を利用した**qMatch()**による判定条件を示す。ここで$\mathbf{f_{out}}$は出力側、$\mathbf{g_{in}}$ は入力側の帳票 ID である。

$$\mathbf{tMatch}(\mathbf{f_{out}}, \mathbf{g_{in}}) = \begin{cases} \mathbf{true} & \{\mathbf{if}(\mathbf{qMatch}(\mathbf{X},\mathbf{Y}) = \mathbf{X} \equiv \mathbf{Y}, \mathbf{X} \rightarrow \mathbf{Y}) \\ \mathbf{false} & \{\mathbf{if}(\mathbf{qMatch}(\mathbf{X},\mathbf{Y}) = \mathbf{X} \nrightarrow \mathbf{Y}, \mathbf{not}(\mathbf{X} \rightarrow \mathbf{Y})) \\ \mathbf{undefined} & \{\mathbf{if}(\mathbf{qMatch}(\mathbf{X},\mathbf{Y}) = \mathbf{not}(\mathbf{X} \equiv \mathbf{Y}), \mathbf{X}\,?\,\mathbf{Y}) \end{cases}$$

$\mathbf{qMatch}(\mathbf{X},\mathbf{Y})$は$\mathbf{tMatch}(\mathbf{f_{out}}, \mathbf{g_{in}})$で与えられた帳票様式$\mathbf{f_{out}}$を**X**に、$\mathbf{g_{in}}$ を**Y**に代入し、以下の操作によって**qMatch()**の判定を下す。

① $\mathbf{X},\mathbf{Y}$をそれぞれ分解する。

$$\mathbf{X} \Leftarrow \{(\boldsymbol{\tau_1}, \mathbf{A_1}) \oplus (\boldsymbol{\tau_2}, \mathbf{A_2}) \oplus \cdots \oplus (\boldsymbol{\tau_n}, \mathbf{A_n})\}$$
$$\mathbf{Y} \Leftarrow \{(\boldsymbol{\varphi_1}, \mathbf{B_1}) \oplus (\boldsymbol{\varphi_2}, \mathbf{B_2}) \oplus \cdots \oplus (\boldsymbol{\varphi_m}, \mathbf{B_m})\}$$

ここで**X**の項目数**n**と**Y**の項目数**m**を比較し、$\mathbf{n} < \mathbf{m}$だった場合、**X**は**Y**に対応する項目が常に不足するため表 3 の場合分け$\mathbf{n} < \mathbf{m}$によって関連情報$\mathbf{X} \nrightarrow \mathbf{Y}$が決定する。

② **X**の項目から**Y**の全ての項目の$\boldsymbol{\gamma}$()が既知の場合、**qMatch()**の判定に必要なのは、項目の関連情報のみである。

Xと**Y**が両方とも単項目の場合、二つの帳票 ID を検索システムに入力し、得られた関連情報と④の表 3 の規則とを適用し判定を行う。

Xと**Y**が両方とも多項目だった場合、または、**Y**が単項目で、**X**が多項目だった場合、上記と同様に、項目間の関連情報を検索システムから取得し、④の表 3 の規則を適用し判定する。

③ $\boldsymbol{\gamma}$()が利用できない場合、**X**の項目から**Y**の項目への対応を検索する。探索方法として前方決定を利用する。前方決定とは項目$\mathbf{A_i}$に対応する項目$\boldsymbol{\gamma}(\mathbf{A_i})$を検索し、その結果の中に$\mathbf{B_1}, \mathbf{B_2}, \cdots, \mathbf{B_m}$が存在するかを再帰的に判定する探索方法である。以下に、その手順を示す。

1. 項目$\mathbf{A_i}$を左辺に持つという入力で検索を行う。この結果一覧を$\mathbf{S(x)}$とする。
2. $\mathbf{S(x)}$から"→"または"≡"を持つ関連情報を抽出して$\mathbf{R(x)}$とし、それ以外の関連情報を$\mathbf{nR(x)}$に格納する。
3. $\mathbf{R(x)}, \mathbf{nR(x)}$に$\mathbf{B_1}, \mathbf{B_2}, \cdots, \mathbf{B_m}$が含まれるか確認する。含まれていた場合$\mathbf{A_i}, \mathbf{R(x)}, \mathbf{B_j}$または、$\mathbf{A_i}, \mathbf{nR(x)}, \mathbf{B_j}$の関連の推移が成り立ち、表 2 での対応が決まった後、探索を終了する。表 2 の$\mathbf{S(x)}$は$\mathbf{R(x)}, \mathbf{nR(x)}$の親集合であるため統一した$\mathbf{S(x)}$で表す。

表 2. 関連推移による対応規則

	$\mathbf{S(x) \rightarrow B_j}$	$\mathbf{S(x) \equiv B_j}$	$\mathbf{not(S(x) \rightarrow B_j)}$
$\mathbf{A_i \rightarrow S(x)}$	$\mathbf{A_i \rightarrow B_j}$	$\mathbf{A_i \rightarrow B_j}$	$\mathbf{not(A_i \rightarrow B_j)}$
$\mathbf{A_i \equiv S(x)}$	$\mathbf{A_i \equiv B_j}$	$\mathbf{A_i \equiv B_j}$	$\mathbf{not(A_i \rightarrow B_j)}$
	$\mathbf{S(x) \nrightarrow B_j}$	$\mathbf{S(x)\,?\,B_j}$	$\mathbf{not(S(x) \equiv B_j)}$
$\mathbf{A_i \rightarrow S(x)}$	$\mathbf{A_i \nrightarrow B_j}$	$\mathbf{A_i\,?\,B_j}$	$\mathbf{not(A_i \equiv B_j)}$
$\mathbf{A_i \equiv S(x)}$	$\mathbf{A_i \nrightarrow B_j}$	$\mathbf{A_i\,?\,B_j}$	$\mathbf{not(A_i \equiv B_j)}$

4. $\mathbf{R(x)}, \mathbf{nR(x)}$に$\mathbf{B_1}, \mathbf{B_2}, \cdots, \mathbf{B_m}$が含まれていない場合、$\mathbf{R(x)}$ から次の検索に入力する項目を選択し、操作 1 に戻る。探索の終了ついて、ここでは任意のループ数の制限によって探索を終了し、判定不能であるとして$\mathbf{A_i\,?\,B_j}$を探索結果とする。
5. 表 2 の関連情報$\mathbf{not}(\mathbf{A_i \equiv B_j}), \mathbf{A_i\,?\,B_j}$のどちらかの決定で探索終了した場合、入力項目$\mathbf{A_i}$が多項目様式か調べる。単項目だった場合は④へ、多項目だった場合は、$\mathbf{A_i}$が多項目様式であると記録し次の検索項目を決定する。

④ 得られた項目間の関連情報を総合して、**X**と**Y**の関連情報を表 3 で決定する。表 3 の"場合分け"は、**X**と**Y**の項目数の対応によって最初の分岐を行い、"条件"によって帳票様式**X**と**Y**の関連情報を決定する。

表 3. 様式の対応による関連式の決定

場合分け	条件	関連式
n < m		X ↛ Y
n = m	全てのB_jについて$\gamma(B_j) \equiv B_j$ならば	X ≡ Y
	全てのB_jについて$\gamma(B_j) \equiv B_j$ または$\gamma(B_j) \rightarrow B_j$ ならば	X → Y
	全てのB_jについて$\gamma(B_j) \rightarrow B_j$ ならば	X → Y
	elseif , あるB_jについて$\gamma(B_j) \nrightarrow B_j$ ならば	X ↛ Y
	elseif , あるB_jについてnot$(\gamma(B_j) \rightarrow B_j)$ ならば	not(X→Y)
	elseif , あるB_jについてnot$(\gamma(B_j) \equiv B_j)$ ならば	not(X≡Y)
	elseif , あるB_jについて$\gamma(B_j)$? B_j ならば	X ? Y
n > m	全てのB_jについて$\gamma(B_j) \equiv B_j$ならば	X → Y
	全てのB_jについて$\gamma(B_j) \equiv B_j$ または$\gamma(B_j) \rightarrow B_j$ ならば	X → Y
	全てのB_jについて$\gamma(B_j) \rightarrow B_j$ ならば	X → Y
	elseif , あるB_jについて$\gamma(B_j) \nrightarrow B_j$ ならば	X ↛ Y
	elseif , あるB_jについてnot$(\gamma(B_j) \rightarrow B_j)$ ならば	not(X→Y)
	elseif , あるB_jについてnot$(\gamma(B_j) \equiv B_j)$ ならば	not(X≡Y)
	elseif , あるB_jについて$\gamma(B_j)$? B_j ならば	X ? Y

ここで決定した関連情報を$\mathbf{qMatch}(\mathbf{X},\mathbf{Y})$の判定結果として**tMatch()**に渡し、再帰的に計算を行う。

4. まとめ

$\mathbf{tMatch}(\mathbf{f_{out}}, \mathbf{g_{in}})$実現のために、帳票様式の分類定義と、メタデータ概念の適用を行った。また、6 種類の関連情報式を定義し、推論を行う為の規則と様式の対応決定一覧を作成し、帳票間の互換性決定方法を定義した。

5. 参考文献

[1] 大谷真, 自律型 Web サービス：原理と実装, 情報処理学会論文誌, 54 巻 2 号, 2013 年