PDDI-CDSサービス

機能仕様書

2024年1月30日

株式会社メタキューブ

目次

[1. はじめに 4](#_Toc157420595)

[2. 機能概要 4](#_Toc157420596)

[2.1. PDDI-CDSサービスの位置 4](#_Toc157420597)

[2.2. PDDI-CDSサービス機能概要 4](#_Toc157420598)

[3. 機能詳細 5](#_Toc157420599)

[3.1. サービスAPI 5](#_Toc157420600)

[3.1.1. CDSルール実行リクエスト 6](#_Toc157420601)

[3.1.2. CDSルール実行レスポンス 9](#_Toc157420602)

[3.2. ストレージ 12](#_Toc157420603)

[3.2.1. CQLルール、 13](#_Toc157420604)

[3.2.2. ELM 14](#_Toc157420605)

[3.2.3. CQLルールからELMへの変換 16](#_Toc157420606)

[3.2.4. PlanDefinition 16](#_Toc157420607)

[3.2.5. Library 17](#_Toc157420608)

[3.2.6. ActivityDefinition 18](#_Toc157420609)

[3.2.7. CodeSystem、ValueSet 19](#_Toc157420610)

[3.3. ルール実行エンジン 19](#_Toc157420611)

[3.3.1. CQL解析準備 19](#_Toc157420612)

[3.3.2. CQL解析 21](#_Toc157420613)

[3.3.3. CarePlan, RequestGroup作成 23](#_Toc157420614)

[3.3.4. Card作成 30](#_Toc157420615)

[4. 未実装及び未対応事項 31](#_Toc157420616)

[4.1 未実装・未対応事項 31](#_Toc157420617)

[4.2 必要な対策を講じた事項 32](#_Toc157420618)

[4.3 対策が必要な事項 35](#_Toc157420619)

[5. 参考文献 36](#_Toc157420620)

図目次

[図 1　システム全体のイメージ 4](#_Toc157420621)

[図 2　PDDI-CDSサービスのシステムイメージ 5](#_Toc157420622)

[図 3　CDSHooksリクエスト 7](#_Toc157420623)

[図 4　CDSHooksリクエストの例 8](#_Toc157420624)

[図 5　 CDSHooksリクエストのprefetch 9](#_Toc157420625)

[図 6　CDSHooksレスポンス（成功時のCard配列の構成） 10](#_Toc157420626)

[図 7　PDDI-CDSサービスのレスポンス例 12](#_Toc157420627)

[図 8　CQLルールの例（Warfarin+NSAIDs） 13](#_Toc157420628)

[図 9　ELMの例（Warfarin＋NSAIDｓ） 15](#_Toc157420629)

[図 10　PlanDefinitionの例（Warfasrin+NSAIDs、order-sign） 16](#_Toc157420630)

[図 11 Libraryの例(Warfarin+NSAIDs, order-sign) 17](#_Toc157420631)

[図 12　ActivityDefinitionの例 18](#_Toc157420632)

[図 13　ValueSetの例 19](#_Toc157420633)

[図 14　CDSHooksリクエストからFHIRリソースを抽出して一つのBundleリソースにまとめる 20](#_Toc157420634)

[図 15　フラグの付与 21](#_Toc157420635)

[図 16　CQL解析結果（Warfarin+NSAIDs、order-sign）の例 22](#_Toc157420636)

[図 17　CarePlan、RequestGroup作成の概略 23](#_Toc157420637)

[図 18　PlanDefinitionからCarePlan, RequestGroupの作成 24](#_Toc157420638)

[図 19　リソース間の対応関係 25](#_Toc157420639)

[図 20　condition、dynamicValueを含むPlanDefinitionリソースのイメージ図 25](#_Toc157420640)

[図 21 conditionの動作イメージ　その1 26](#_Toc157420641)

[図 22 conditionの動作イメージ　その2 27](#_Toc157420642)

[図 23 dynamicValueの動作イメージ 28](#_Toc157420643)

[図 24　CarePlan、RequestGroup作成機能での出力となるFHIR Bundleリソース 29](#_Toc157420644)

[図 25 CarePlan、RequestGroup作成機能での出力となるFHIR Bundleリソースの例 29](#_Toc157420645)

[図 26　CarePlan, RequestGroupリソースからCard作成のイメージ 30](#_Toc157420646)

[図 27 Card配列の例 31](#_Toc157420647)

[図 28　空のCard配列 31](#_Toc157420648)

[図 29 contextPrescriotionsフラグの付与 33](#_Toc157420649)

[図 30 CQLにおけるContextPrescriptionsの代替 34](#_Toc157420650)

[図 31 単位「mEq」を含むCQL 34](#_Toc157420651)

[図 32 ucum-essential.xml 35](#_Toc157420652)

[図 33 OperationOutcomeリソースによるPDDI-CDSサービスのエラーレスポンス例 36](#_Toc157420653)

表目次

[表 1　CDSHooksリクエストのスタイル 7](#_Toc157420654)

[表 2　レスポンスヘッダ 10](#_Toc157420655)

[表 3　CDSHooksレスポンス（Card）のスタイル 11](#_Toc157420656)

[表 4 path設定値に対応するCarePlan及びRequestGroupリソースの属性名 28](#_Toc157420657)

# はじめに

　本文書はNeXEHRS内公募「Implementation for PDDI CDSを元にしたCDS Hooksの実証開発」において、その中のPDDI-CDSサービスの機能仕様について説明したものである。

# 機能概要

## PDDI-CDSサービスの位置

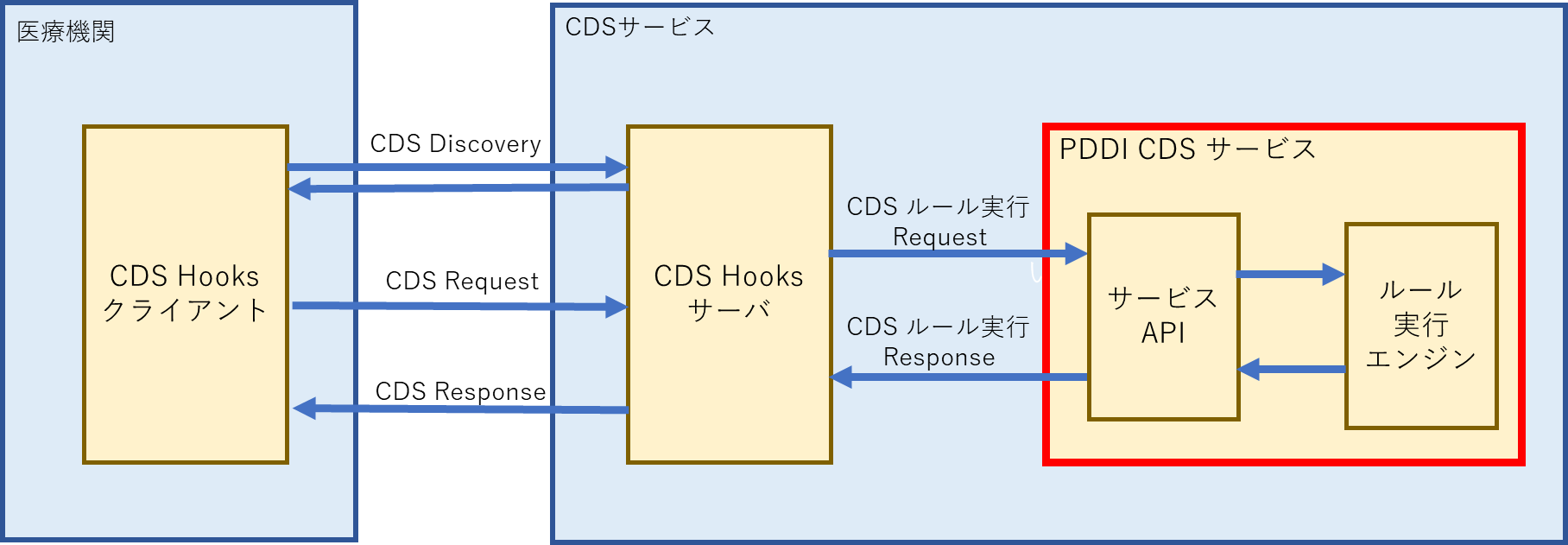


図 　システム全体のイメージ

図 1にシステム全体のイメージを示す。このうちPDDI-CDSサービスは図中の赤枠で示す部分となり、本文書での取り扱い範囲もこの部分となる。

## PDDI-CDSサービス機能概要

　図 1の赤枠部分をより詳しく表示したシステムイメージを、図 2に示す。

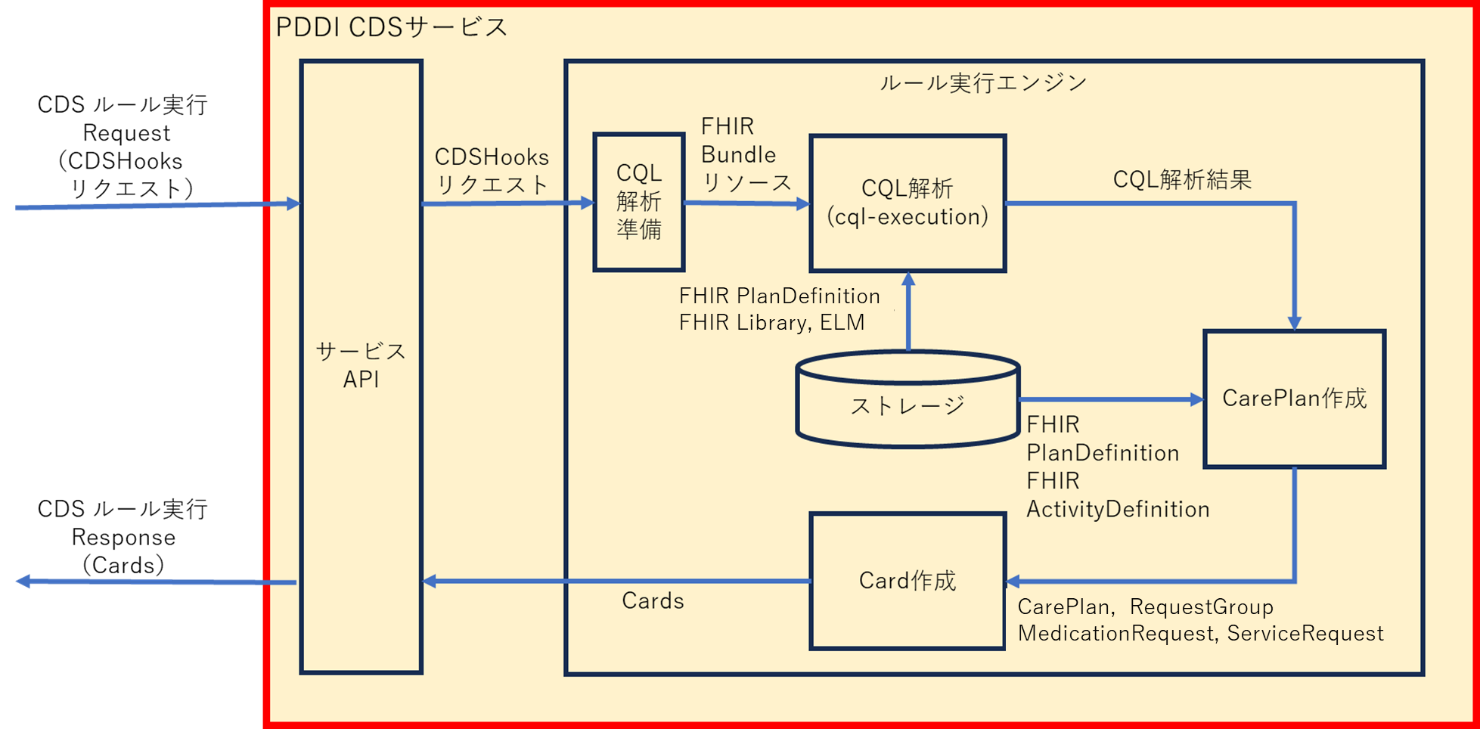


図 　PDDI-CDSサービスのシステムイメージ

PDDI-CDSサービスでは、以下の機能で構成される。各種機能に関する詳細は次章で述べる。

* サービスAPI

CDSHooksサーバから送られるCDSルール実行リクエストを受信し、CDSHooksリクエストを抽出する。またルール実行エンジンが生成するCardsを組み込んだCDSルール実行レスポンスを作成してCDSHooksサーバへ送信する。

* ストレージ

ルール実行エンジンで使用するCQLルール、ELM、各種FHIRリソースを格納する。

* ルール実行エンジン

CDSHooksリクエストに対してCQL解析を実行し、解析結果としてCDSHooksのCardsを作成する。ルール実行エンジンでは以下に示す機能を順に実行する。

* CQL解析準備
* CQL解析
* CarePlan、RequestGroup作成
* Card作成

# 機能詳細

## サービスAPI

サービスAPIでは、CDSHooksサーバから送られるCDSルール実行リクエストを受信し、CDSHooksリクエストを抽出する。またルール実行エンジンが生成するCardsを組み込んだCDSルール実行レスポンスを作成してCDSHooksサーバへ送信する。以下本機能で取り扱うCDSルール実行リクエスト、ならびにCDSルール実行レスポンスについて説明する。

### CDSルール実行リクエスト

PDDI-CDSサービスのリクエストを受け取るエンドポイントURLと、PDDI-CDSサービスで受け付けるCDSルール実行リクエストを以下に示す。

* エンドポイントURL

　POST　[base]/[serviceName]/[id]

ただし

**[base]**：サービスベースURL（例： http://abc.org:8080）

**[serviceName]**：PDDI-CDSサービスのサーバ名。インストール時において変更可能。「PDDI-CDSサービス-機能仕様書\_附属書\_インストール手順書.docx」を参照のこと。

**[id]**：CDSHooksリクエストに含まれる、薬剤の組み合わせに応じて以下の値を用いる

- Warfarin＋NSAID, order-signの場合

　[id] = warfarin-nsaids-cds-sign

- Digoxin+Cyclosporine, order-signの場合

　　　　[id] = digoxin-cyclosporine-cds-sign

-もし上記以外の[id]が設定された場合は、ステータスコード200とし、空のCard配列を返すものとする

* リクエストヘッダ

なし

* リクエストボディ

JSON形式のCDSHooksリクエストを設定する。リクエストスタイルは以下の図 3および表 1を参照のこと。またリクエストの例を図 4に示す。

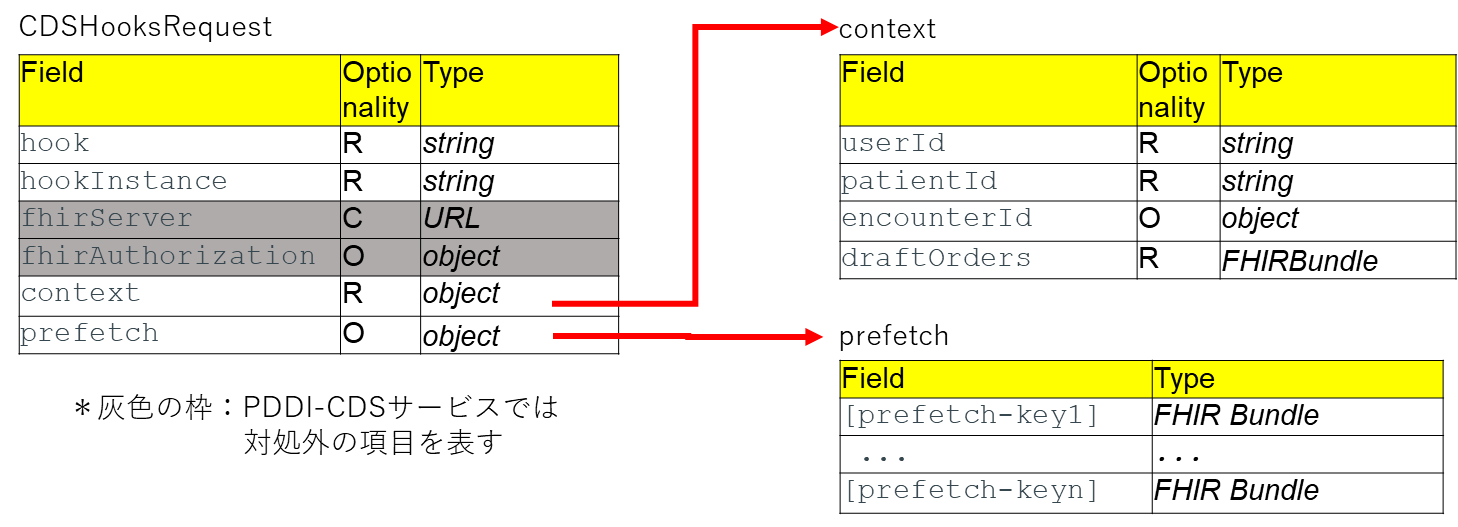


図 　CDSHooksリクエスト

表 　CDSHooksリクエストのスタイル



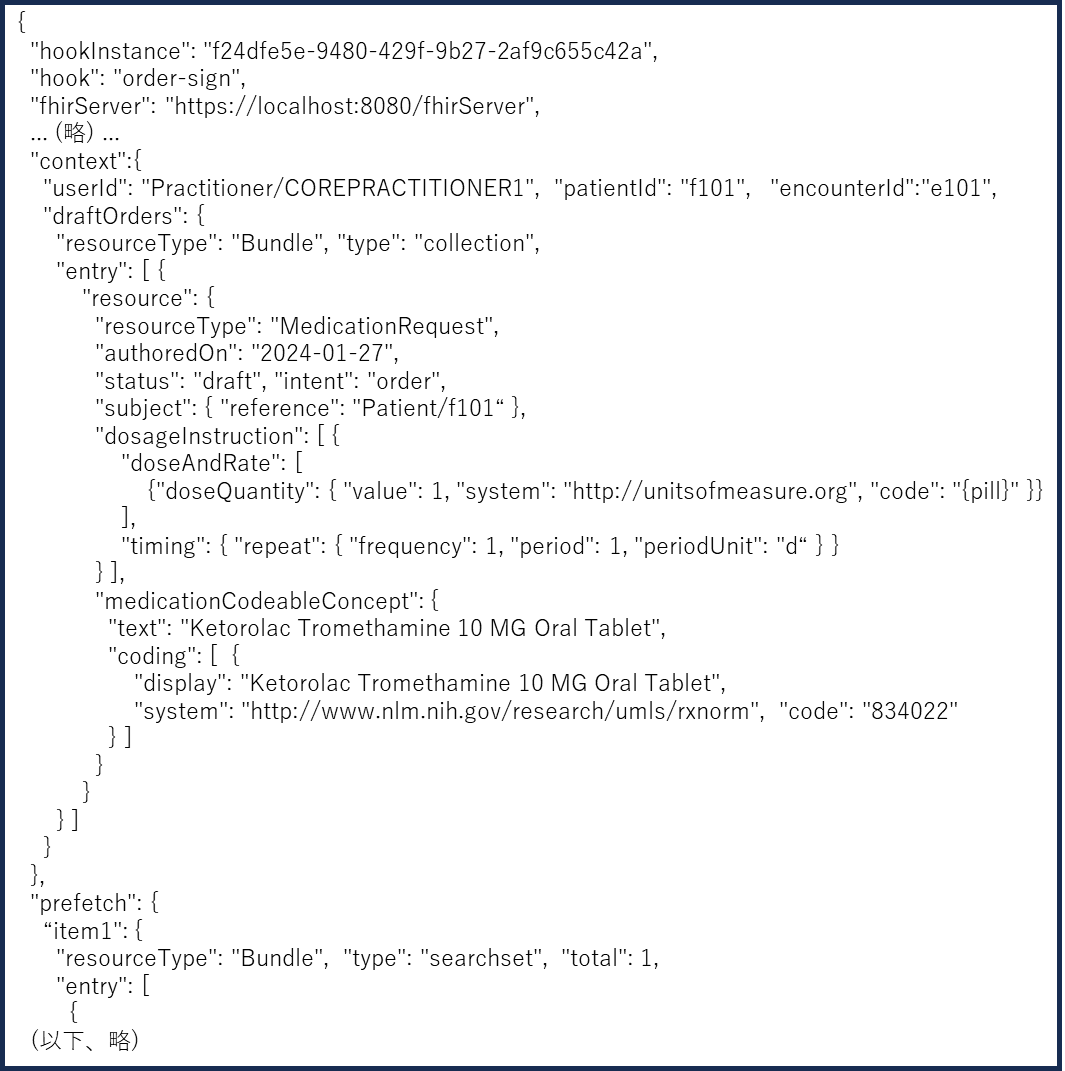


図 　CDSHooksリクエストの例

　図 4に示したリクエストの全文は「PDDI-CDSサービス-機能仕様書\_附属書\_各種メッセージ」のタブ「CDSHooksRequest」を参照のこと。

**CDSHooksリクエストのprefetchについて**

CDSHooksリクエストのprefetchにどのような情報を含めるのかについては、CDSHooksクライアント側がCDSHooksサーバから受け取ったCDSDiscoveryレスポンスに定義されている。具体的にはCDSDiscoveryレスポンスに含まれている各serviceのprefetchに、FHIRサーバに対するReadリクエスト及びSearchリクエストの形（絶対URLまたは相対URLとして）で、CDSHooksリクエストのprefetchへどのような種類のFHIRリソースが含まれるべきかが示されている。

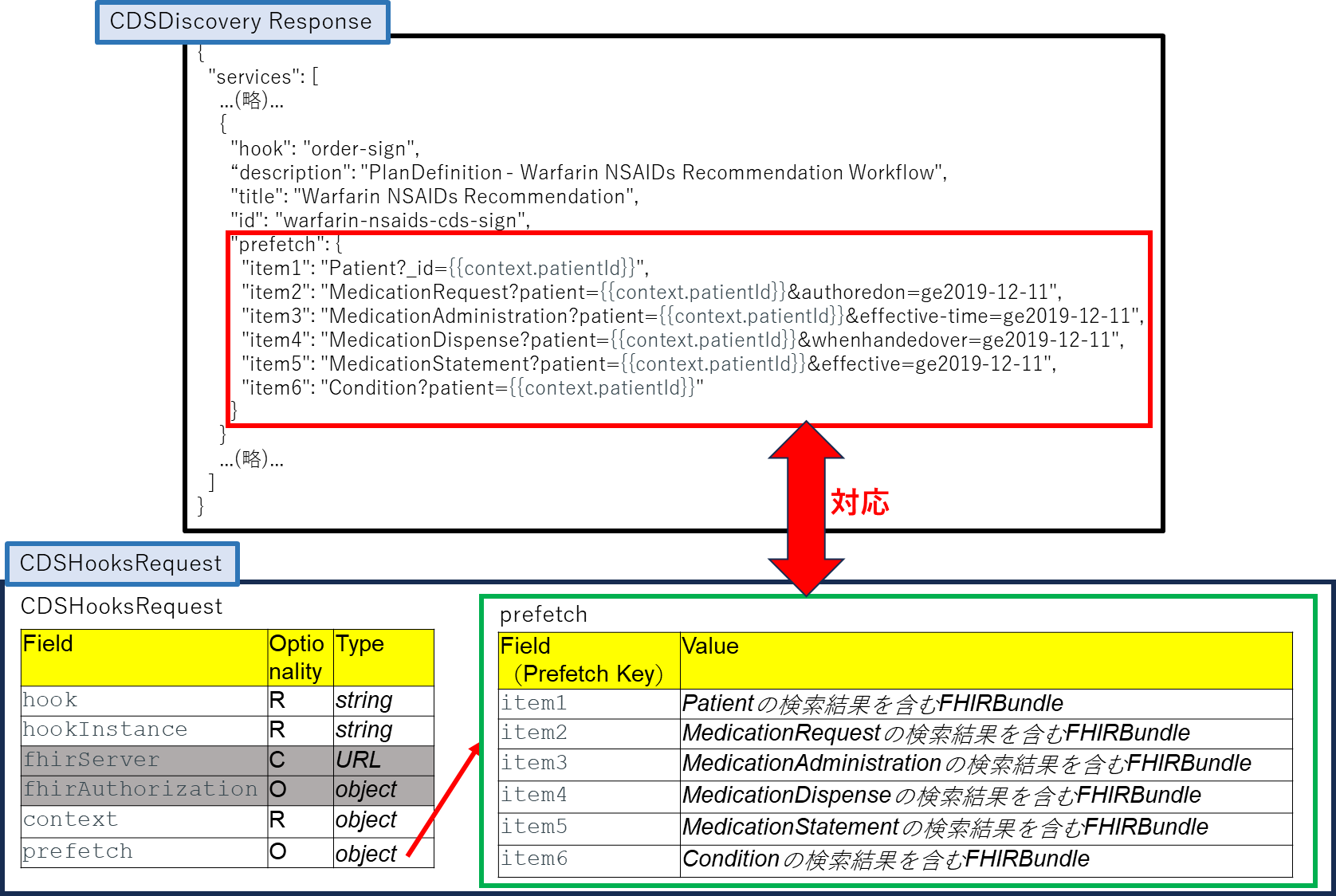


図 　 CDSHooksリクエストのprefetch

CDSHooksクライアントがCDSDiscoveryレスポンスからidがwarfarin-nsaids-cds-sign,でhookがorder-signであるserviceを参照して、特定の患者（Patient.idが{{context.patientId}}で表される）に対するCDSHooksリクエストを作成する例を図 5に示す。Prefetchに関して、CDSDiscoveryレスポンスの赤枠内とCDSHooksリクエストの緑枠内が対応し、赤枠内のitem1～item6に示すSearchリクエストの結果が緑枠内のitem1～item6にFHIRBundleリソースとしてそれぞれ設定される。

### CDSルール実行レスポンス

PDDI-CDSサービスが返すCDSルール実行レスポンスを以下に示す

* HTTPステータスコード

成功時：　200

エラー発生時：　400番台または500番台

* レスポンスヘッダ

表 2を参照

表 　レスポンスヘッダ

|  |  |
| --- | --- |
| レスポンスヘッダ名 | レスポンスヘッダ設定値 |
| Date | レスポンス作成日時 |
| Content-Type | application/json |
| X-Request-Id | CDSHooksRequest.hookInstance |
| X-Correlation-Id | 当サービスで設定するUUID値 |

* レスポンスボディ

成功時：　Card配列

エラー発生時：　なし[[1]](#footnote-1)

成功時にレスポンスボディに設定するCard配列はCDSHooks [1]でのCardの定義に準ずる。図 6および表 3参照。

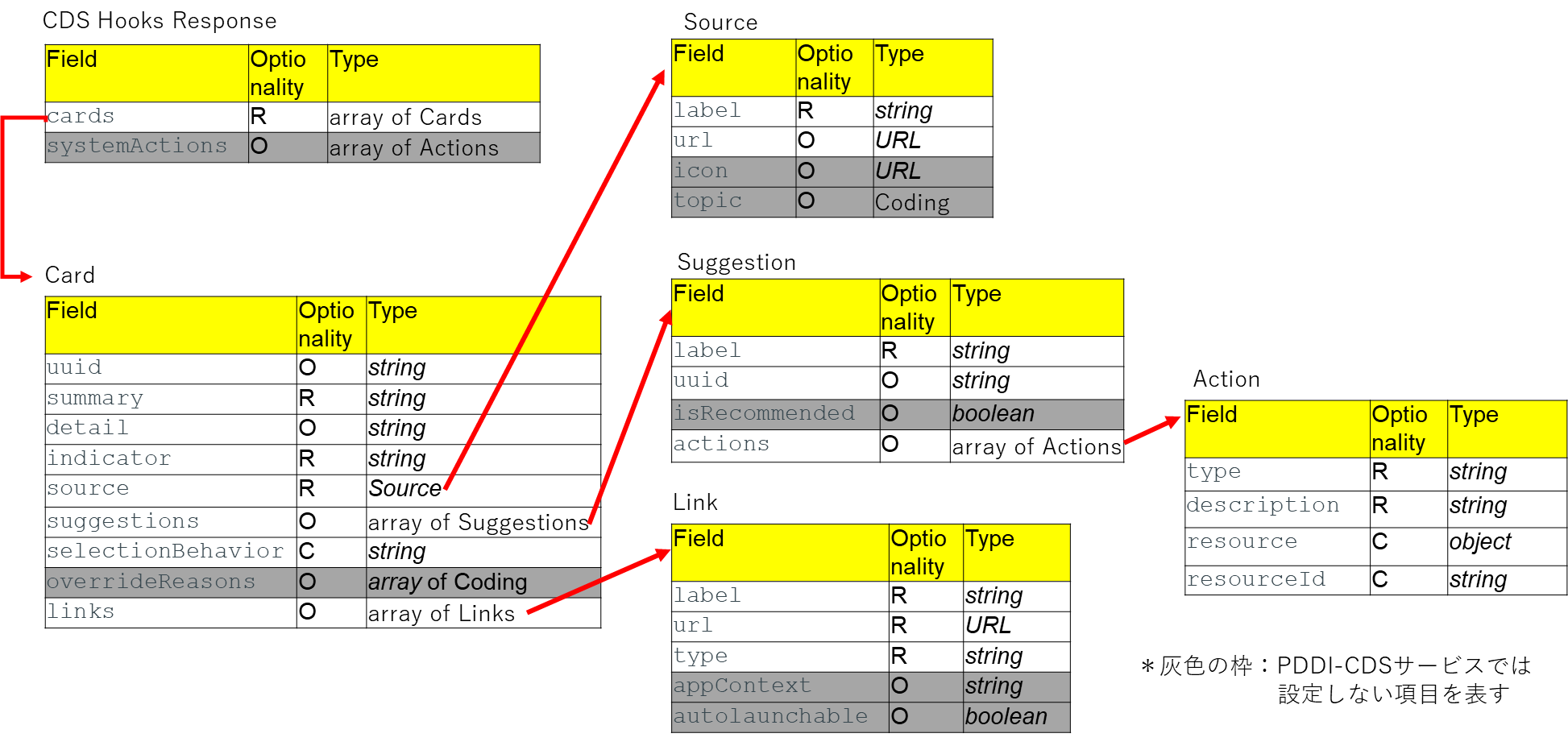


図 　CDSHooksレスポンス（成功時のCard配列の構成）

表 　CDSHooksレスポンス（Card）のスタイル



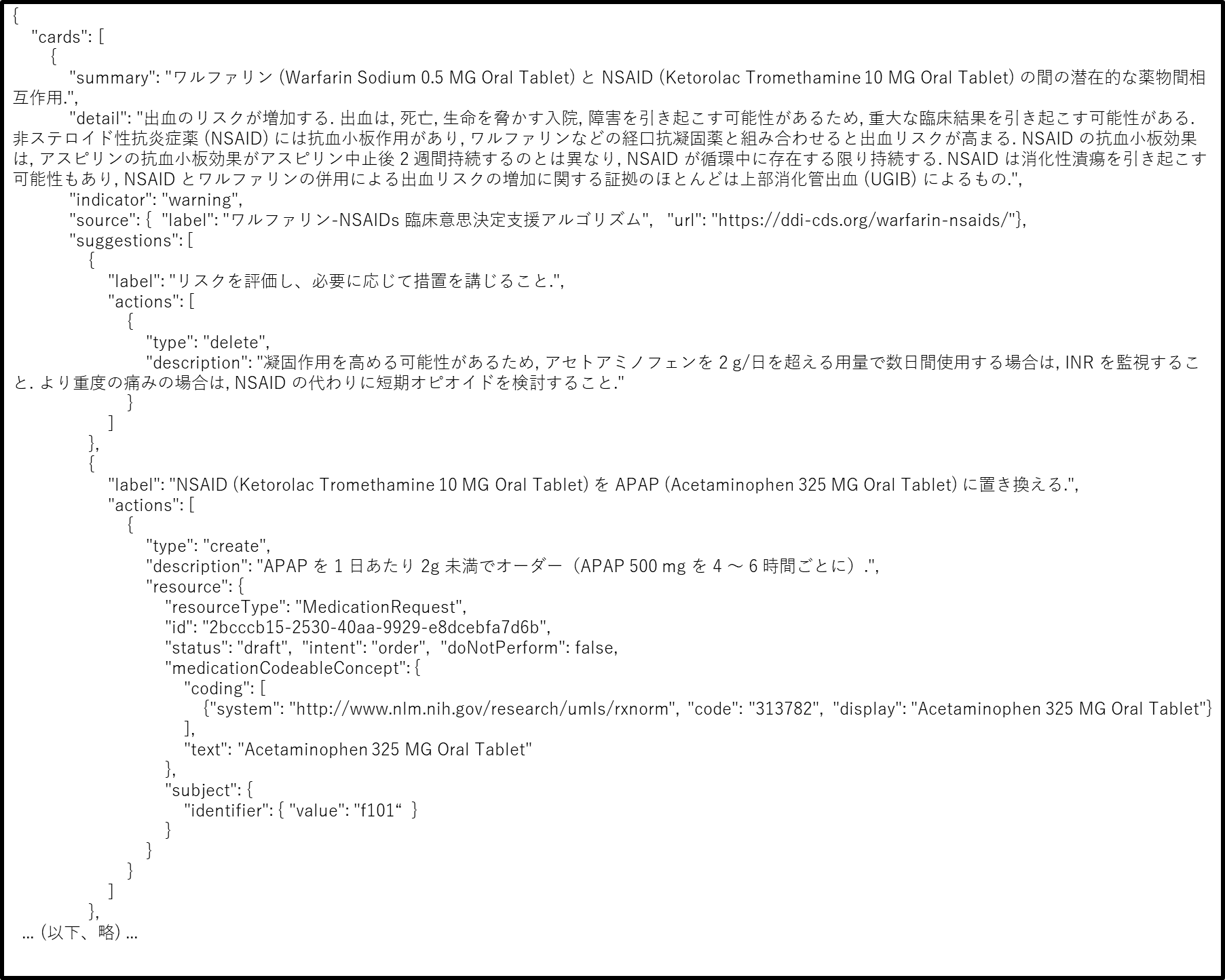


図 　PDDI-CDSサービスのレスポンス例

図 7にレスポンス例を示す。レスポンス例の全文は「PDDI-CDSサービス-機能仕様書\_附属書\_各種メッセージ」のタブ「Cards」を参照のこと。

## ストレージ

　PDDI-CDSサービスには「ストレージ」が用意されていて、これが「FHIRサーバの代役」を果たす。ストレージには以下に示すデータがファイルとして格納され、必要に応じて後述のルール実行エンジンから呼び出される。

* CQLルール
* ELM
* FHIR　PlanDefinitionリソース
* FHIR　Libraryリソース
* FHIＲ　ＡｃｔｉｖｉｔｙＤｅｆｉｎｉｔｉｏｎリソース
* FHIR CodeSystem　リソース
* FHIR ValueSetリソース

### CQLルール、

CQL（Clinical Quality Language：臨床品質言語） [2]は、臨床品質に焦点を当てた高レベルの医療ドメイン固有の言語であり、医療品質の測定と意思決定支援アーティファクトのオーサリングを対象とする。

PDDI-CDSサービスにおいて、CQLは後述するルール実行エンジンでCQL解析を行うときのルールを記述するために使用する構文である。以下CQLで記したこのようなルールを「CQLルール」と記す。CQLルールの例を図 8　に示す。なお図 8に例として取り上げたCQLルールの全文は「PDDI-CDSサービス-機能仕様書\_附属書\_各種メッセージ」のタブ「CQL(Warfasrin+NSAIDs)」を参照のこと。

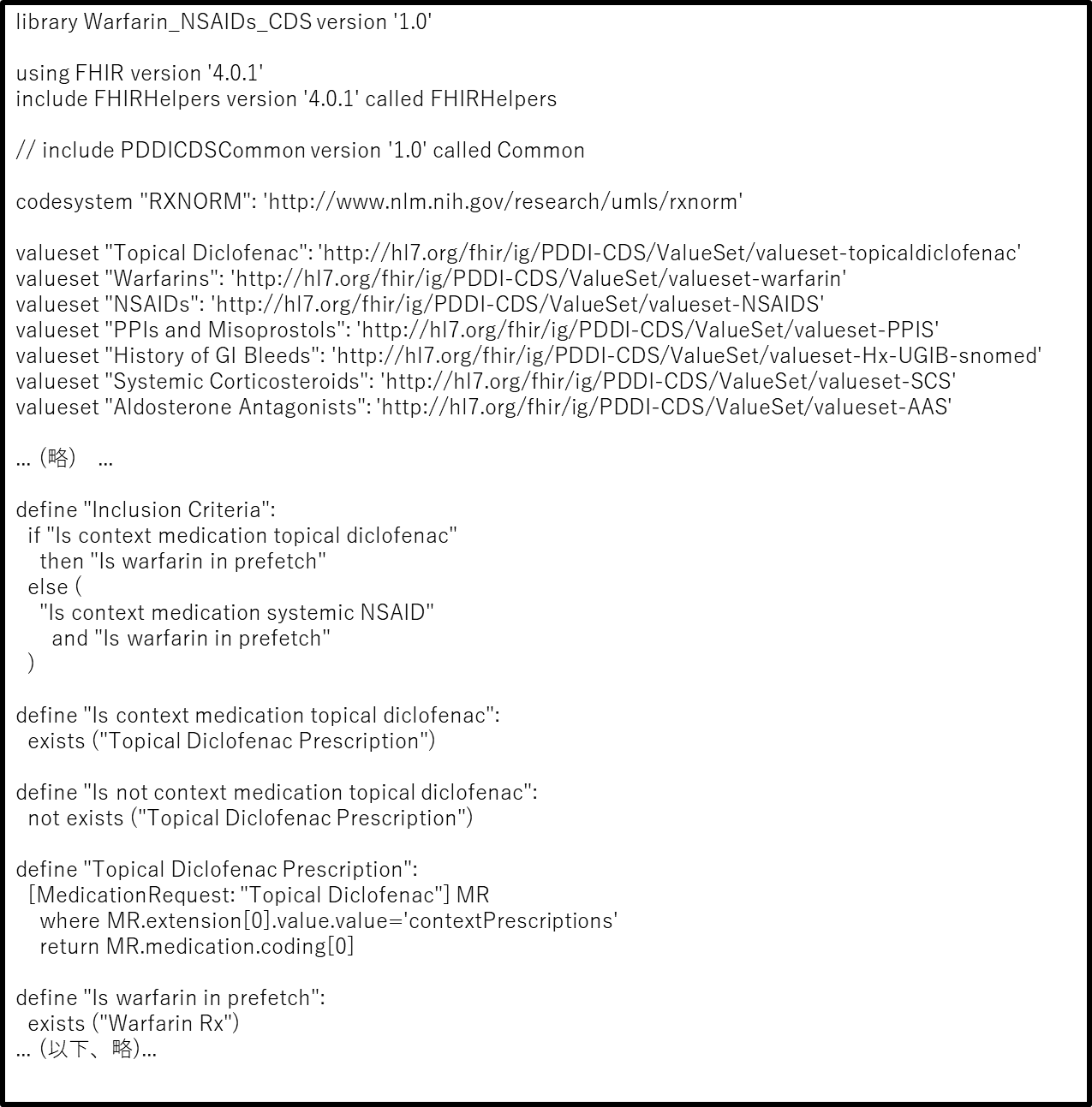


図 　CQLルールの例（Warfarin+NSAIDs）

本PDDI-CDSサービスでは、では「Warfarin＋NSAIDs」及び「Digoxin+Cyclosporine」に対するCQLルールを用いるが、これらはPDDI-CDS実装ガイドのKnowledge-artifacts [3]から「5.3 Knowlwdge Artifacts」から取得したCQLルールに対して、Cardのlabel,summary,destinationにあたる部分の日本語化、FHIRR4対応などの修正を行ったもので、以下のファイル名でストレージに格納する。

* warfarin-nsaids-cds-logic-r4-jp.cql

（「PDDI-CDSサービス-機能仕様書\_附属書\_各種メッセージ.xlsx」のタブ「CQL(Warfarin＋NSAIDs)」を参照のこと。）

* digoxin-cyclosporine-cds-logic-r4-jp.cql

（「PDDI-CDSサービス-機能仕様書\_附属書\_各種メッセージ.xlsx」のタブ「CQL(Digoxin＋Cyclosporine)」を参照のこと。）

また、CQL解析対象となるデータはCDSHooksリクエストに含まれるFHIRリソース（R4）となるため、下記に示すCQLルールも合わせて使用する。これは解析時に補助的に使用するもので、ストレージではなくPDDI-CDSサービス内部で保持する。

* FHIRHelper.cql

（「PDDI-CDSサービス-機能仕様書\_附属書\_各種メッセージ.xlsx」のタブ「CQL(FHIRHelpers)」を参照のこと。）

　なお、本PDDI-CDSサービスにおけるCQL解析ではCQLルールをそのまま使用せず、CQLルールをELMの形式に変換した上で使用する。ELMは次節で記す。

　.

### ELM

　ELM（Expression Logical Model：式論理モデル）は電子臨床品質測定ロジックの機械可読表現であり、電子医療記録からデータを自動的に取得するために必要な情報を提供する。ELMはXMLまたはJSONで記述できる。ELMの例を図 9　に示す。このELMは図 8に示したCQLルールを後述のCQL to ELM Translatorで変換して得られたものである。なお　図 9に例として示したELMの全文は「PDDI-CDSサービス-機能仕様書\_附属書\_各種メッセージ」のタブ「ELM(Warfasrin+NSAIDs)」を参照のこと。

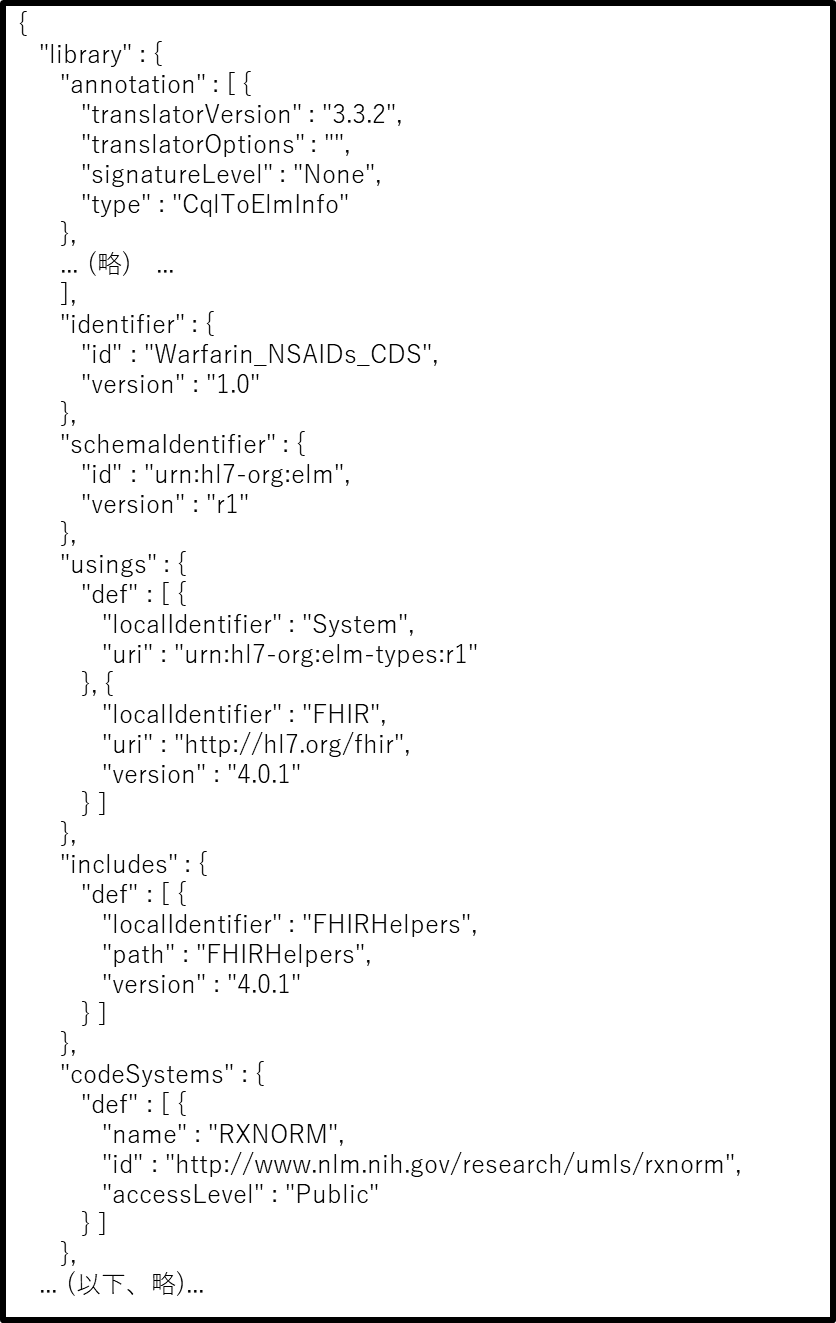


図 　ELMの例（Warfarin＋NSAIDｓ）

　本PDDI-CDSサービスでは、あらかじめ前節でのCQLルールをJSON形式のELMに変換したうえで利用する。ELMは以下のファイル名でストレージに格納する。

* warfarin-nsaids-cds-logic-r4-jp.json　（warfarin-nsaids-cds-logic-r4-jp.cqlを変換）

（「PDDI-CDSサービス-機能仕様書\_附属書\_各種メッセージ.xlsx」のタブ「ELM(Warfarin＋NSAIDs)」を参照のこと。）

* digoxin-cyclosporine-cds-logic-r4-jp.json (digoxin-cyclosporine-cds-logic-r4-jp.cqlを変換)

（「PDDI-CDSサービス-機能仕様書\_附属書\_各種メッセージ.xlsx」のタブ「ELM(Digoxin＋Cyclosporine)」を参照のこと。）

また、PDDI-CDSサービス内部で保持する「FHIRHelpers.cql」についても、JSON形式のELMに変換の上利用する（FHIRHeplers.json）。　FHIRHelper.jsonlの全文は「PDDI-CDSサービス-機能仕様書\_附属書\_各種メッセージ.xlsx」のタブ「ELM(FHIRHelpers)」を参照のこと。

### CQLルールからELMへの変換

本PDDI-CDSサービスでは、あらかじめCQLルールをJSON形式のELMに変換する必要があるが、この変換にはGitHub [4]で公開されているオープンソース、CQL to ELM Translatorを利用する。CQL to ELM TranslatorはPDDI-CDSサービスの内部で動かすのではなく、それとは別に動作させてCQLルールからELMへの変換を実行する。

CQL to ELM TranslatorによるCQLルールからELMへの変換方法については、別途「PDDI-CDSサービス-機能仕様書\_附属書\_インストール手順書.docx」にて説明する。

なお、CQLルールをELMに変換するためのツールはCQL to ELM Translator以外にも公開されているので、それを利用してもよい。

### PlanDefinition

　PlanDefinition [5]はFHIRリソースの一種で、患者に対して発生する可能性のあるアクションを定義する（PlanDefinitionの例を図 10に示す）。PlanDefinitionにはPDDI-CDSインスタンスにおいて利用されるTriggerDefinition、Condition、DynamicValue及びActionの４つの要素がある。これら４つの要素についての詳細は [3]を参照のこと。

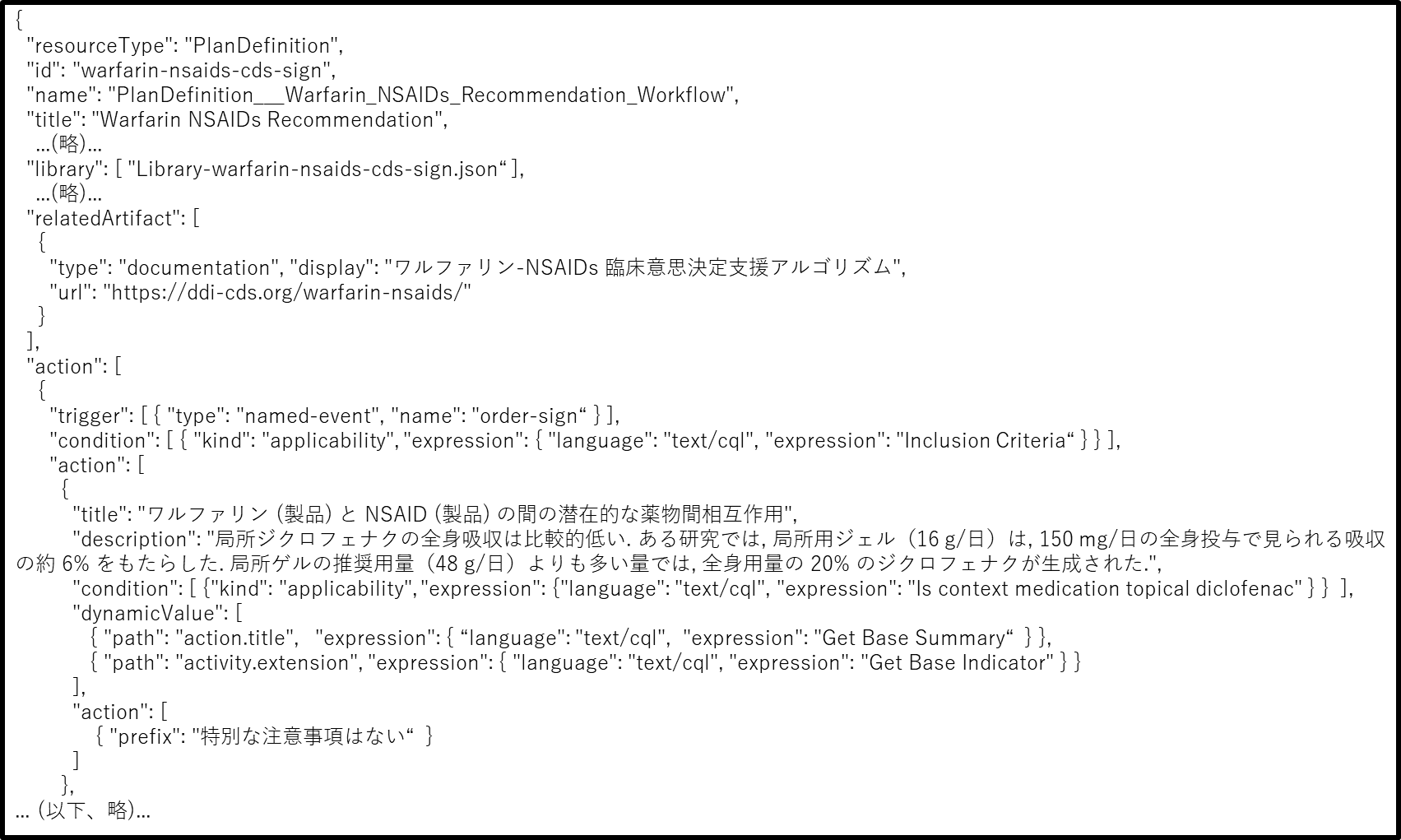


図 　PlanDefinitionの例（Warfasrin+NSAIDs、order-sign）

PDDI-CDSサービスでは「Warfarin＋NSAIDs, order-sign」及び「Digoxin+Cyclosporine, order-sign」に対するPlanDefinitionリソースを用いるが、これらはPDDI-CDS実装ガイドのKnowledge-artifacts [3]から「5.3 Knowlwdge Artifacts」から取得したJSON形式のPlanDefinitionリソースに対して、Cardのlabel,　summary,　destinationにあたる部分の日本語化、FHIRR4対応などの修正を行ったもので、以下のファイル名でストレージに格納する。

* PlanDefinition-warfarin-nsaids-cds-sign.json　（Warfarin+NSAIDs、order-sign）

（「PDDI-CDSサービス-機能仕様書\_附属書\_各種メッセージ.xlsx」のタブ「PlanDefinition(Warfarin＋NSAIDs)」を参照のこと。）

* PlanDefinition-digoxin-cyclosporine-cds-sign.json　（Digoxin＋Cyclosporine、order-sign）

（「PDDI-CDSサービス-機能仕様書\_附属書\_各種メッセージ.xlsx」のタブ「PlanDefinition(Digixin＋Cyclosporine)」を参照のこと。）

上記PlanDefinitionリソースには、後述するLibraryリソースへの参照を、Libraryリソースのファイル名として含める。本PDDI-CDSサービスにおいてLibraryリソースにはCQLルールやELMへの参照情報を記載するため、これによりPlanDefinitionリソース、対応するCQLルールとELMとの関連付けが可能となる。

### Library

Library [6]もまたPlanDefinitionと同じくFHIRリソースの一種であり、これを使用して、ロジック ライブラリや情報モデルの記述などの既存の知識資産を記述および公開すること、および知識資産のコレクションを記述することが可能である。Libraryリソースの例を図 11に示す。

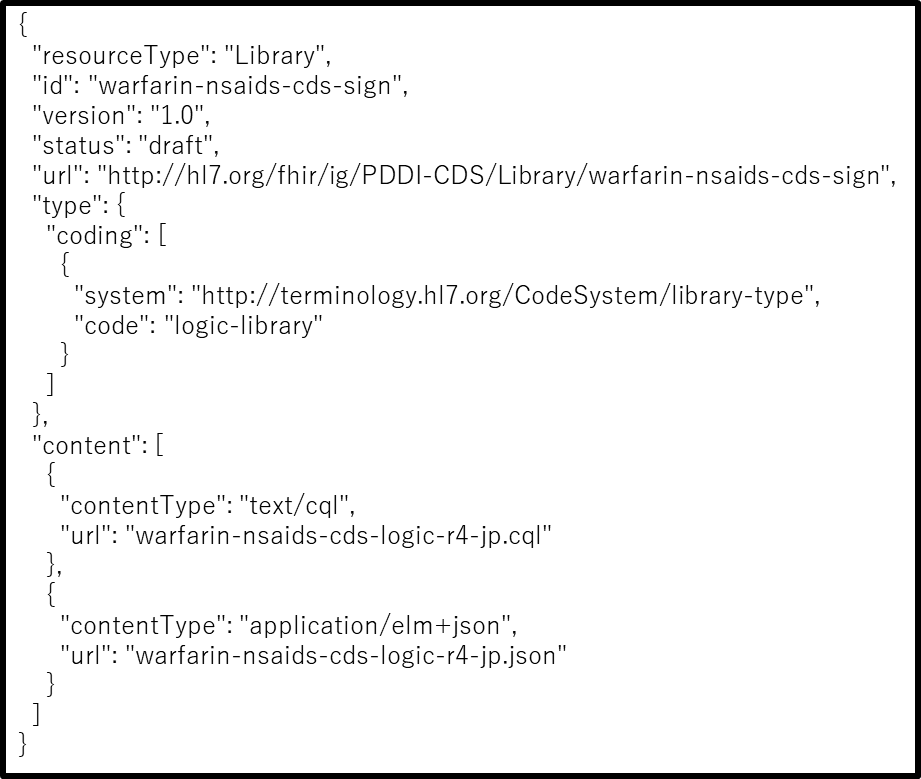


図 Libraryの例(Warfarin+NSAIDs, order-sign)

PDDI-CDSサービスにおいて、Libraryリソースにはストレージに含まれるCQLルールファイルやELMファイルの名前が定義されている。本サービスではCQLルールならびにELMは「Warfarin＋NSAIDs, order-sign」及び「Digoxin+Cyclosporine, order-sign」の2種類を扱うのでLibraryリソースもこの2種類に対してJSON形式のファイルとして作成し、ストレージに格納する。

* Library-warfarin-nsaids-cds-sign.json　（Warfarin+NSAIDs、order-sign）
* Library-digoxin-cyclosporine-cds-sign.json　（Digoxin＋Cyclosporine、order-sign）

（どちらも「PDDI-CDSサービス-機能仕様書\_附属書\_各種メッセージ.xlsx」のタブ「Library」を参照のこと。）

これらのLibraryファイルは、対応するPlanDefinitionリソースに参照情報としてファイル名を付与することでPlanDefinitionリソースからLibraryリソースを介してCQLルールやELMを参照することが可能となる。

なお実際のFHIRサーバへPDDI-CDSサービスで利用するLibraryリソースを格納する場合、LibraryリソースにCQLルールファイルやELMファイルの名前を設定するのではなく、これらのファイルの内容をBase64でエンコードしてできた文字列を張り付ける。

### ActivityDefinition

　ActivityDefinitionリソースは実行されるアクティビティの共有可能で利用可能な記述を定義できる。これは、ワークフロー、注文セット、またはプロトコルの一部として実行されるアクションを指定するために使用することも、注文可能なものなどのアクティビティのカタログの一部として独立して使用することもできる。例を図 12　に示す

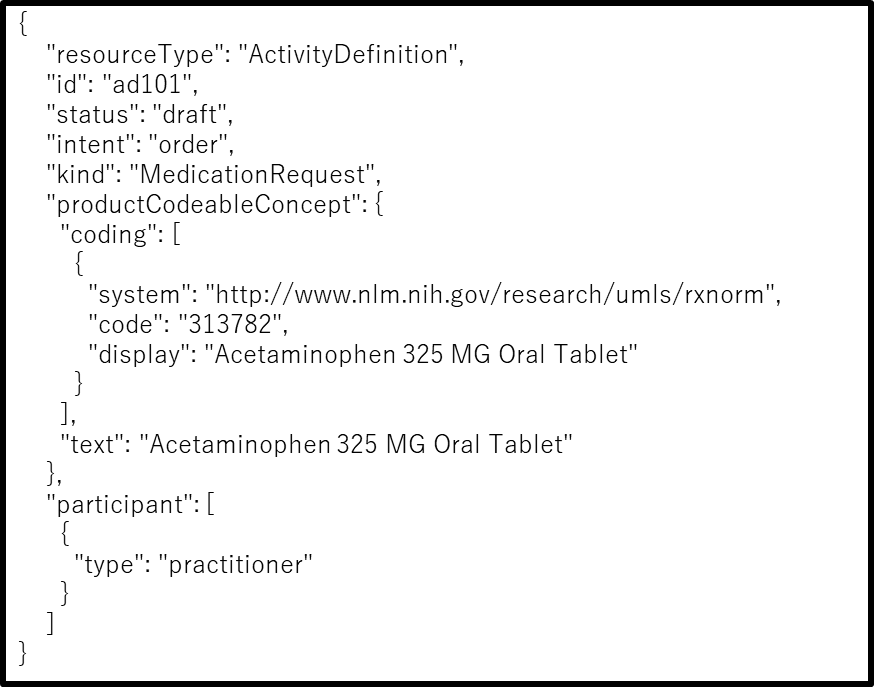


図 　ActivityDefinitionの例

PDDI-CDSサービスにおいては、ActivityDefinitionリソースはJSONファイルとしてストレージに格納されていて、PlanDefinitionリソース内で参照される。ActivityDefinitionリソースは、「Warfarin＋NSAIDs, order-sign」用に2種類、及び「Digoxin+Cyclosporine, order-sign」用に6種類、ストレージに用意してある。

### CodeSystem、ValueSet

　PDDI-CDSサービスで利用するコード系を定義するFHIRリソース。ルール実行エンジンでCQL解析を行う際に参照される。

　本サービスで使用するCodeSystem、ValueSetはPDDI-CDSの実装ガイドに添付されているJSONファイルをストレージに格納している。図 13にValueSetの例を示す

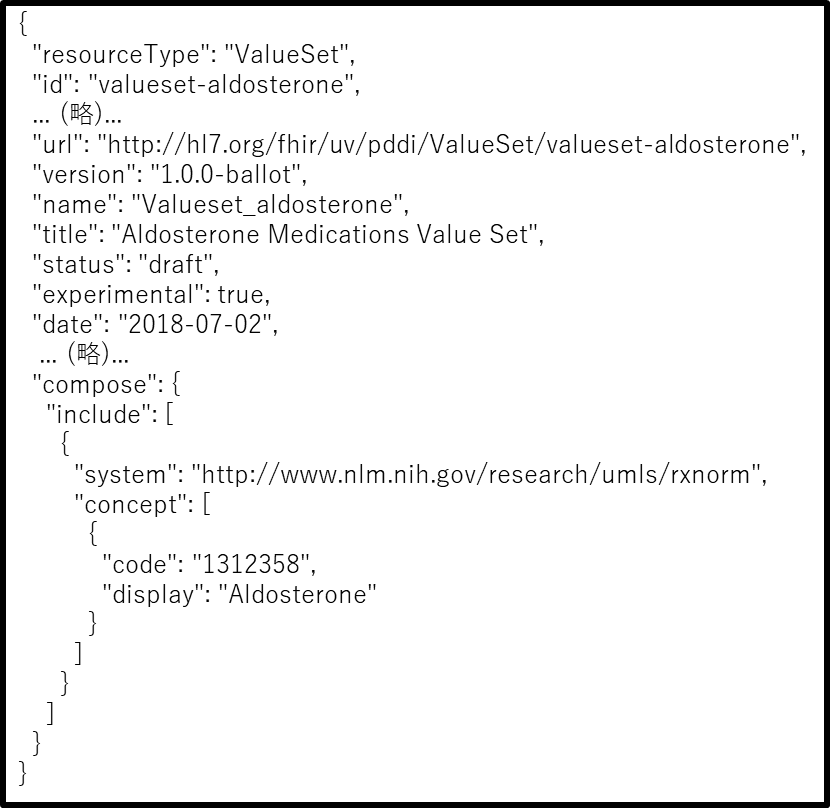


図 　ValueSetの例

## ルール実行エンジン

CDSHooksリクエストに対してCQL解析を実行し、解析結果としてCDSHooksのCardsを作成する。ルール実行エンジンでは以下に示す機能を順に実行する。

* CQL解析準備
* CQL解析
* CarePlan作成
* Card作成

### CQL解析準備

CDSHooksリクエストのcontext.draftOrdersやprefetchに関連付けられたFHIR Bundleリソースから、その中に含まれているFHIRリソースを抽出し、これらを一つのFHIR Bundleリソースにまとめる。そのイメージ図を図 14に示す。

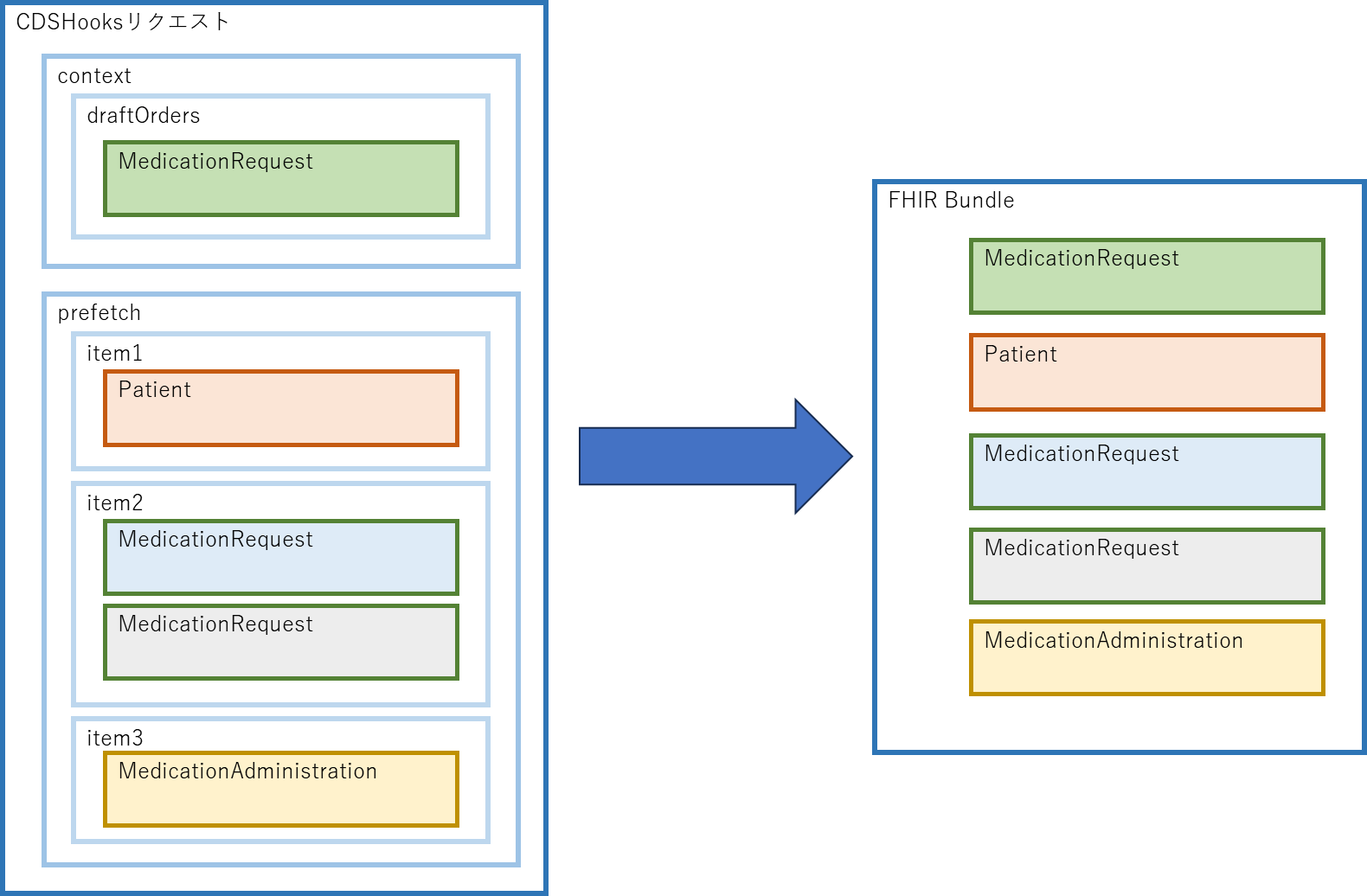


図 　CDSHooksリクエストからFHIRリソースを抽出して一つのBundleリソースにまとめる

ただし、作成されたFHIR Bundleリソースに含まれる各FHIRリソースはもともとCDSHooksリクエストのcontext.draftOrders以下にあったのか、prefetch以下にあったのか区別がつかなくなるため、context.draftOrders以下にあったFHIRリソースにはextensionを利用して以下に示すフラグ（図 15で赤字の部分）を付与する。

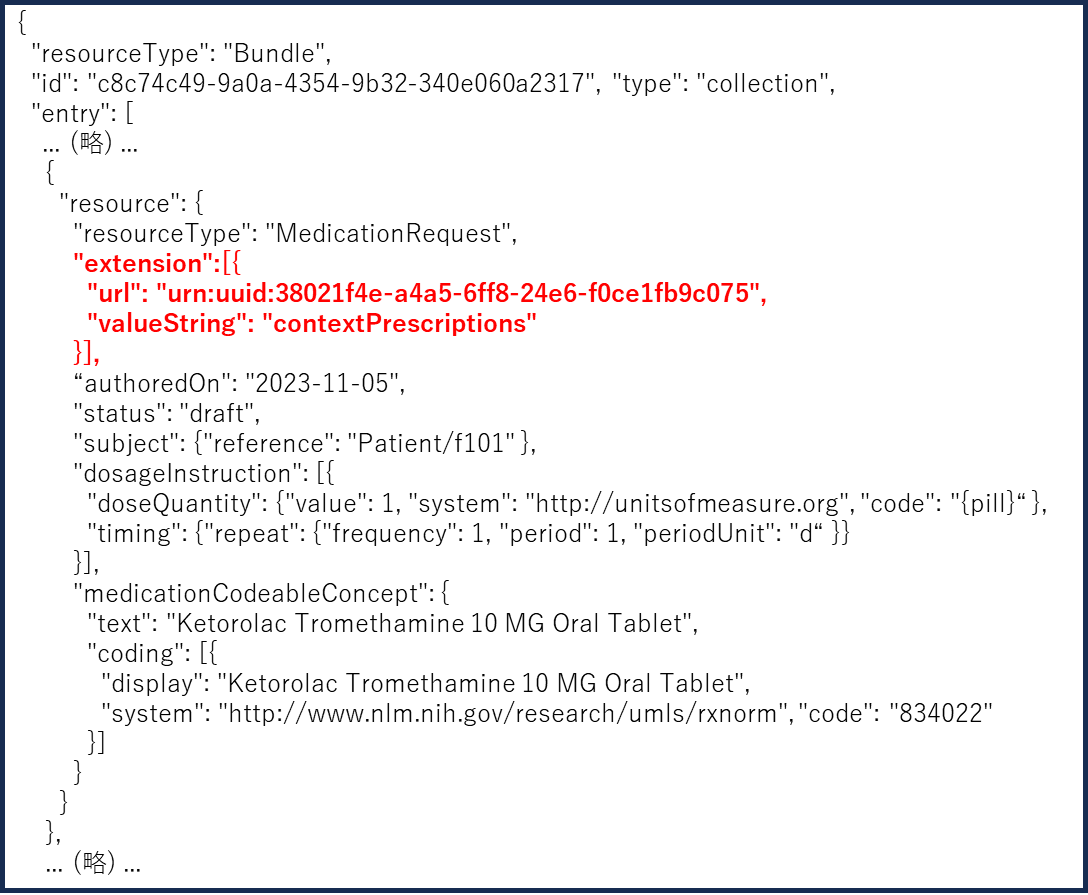


図 　フラグの付与

作成されたBundleリソースは、次節で説明するCQL解析（具体的には解析に使用するオープンソースcql-execution）の入力データとして使用する。

### CQL解析

CDSHooksリクエストに含まれるFHIRリソースを入力とし、これにCQLに基づく解析を実施する。CQL解析では以下に示すオープンソースを利用する。

* cql-execution [7]

CQLを実行するためのJavaScriptフレームワーク。このフレームワークは、JSON ELM として表現された CQL アーティファクトを実行するための TypeScript/JavaScript ライブラリを提供する。

* cql-exec-fhir [8]

cql-executionで使用する FHIR ベースのデータ ソース モジュール。

これらのオープンソースを利用して、本機能は以下の順番で実行する。

1. ELM及びFHIRHelpers.jsonの読み込み

指定のエンドポイントURLに記載の[id]に対応するJSON形式のELMをストレージから読み込む。加えてシステムが保持するFHIRHelpers.jsonを読み込む。ELMとFHIRHelpers.jsonについては3.2.2節参照のこと。

1. Valueset読み込み

ELMのlibrary.valueSets（CQLにおけるvalueset）に指定されているFHIR ValueSetリソースに含まれるコードデータを読み込む.。

1. 解析に使う入力データの読み込み

入力データは「CQL解析準備」（3.3.1節参照）で作成した、CDSHooksリクエストに含まれるFHIRリソースをまとめたFHIR Bundle リソースとなる。またこのFHIR Bundleリソースの読み込みにcql-exec-fhirを利用する。

1. cql-executionの実行
   * 1. ～③で読み込んだ各種データを利用してcql-executionを実行する。

上記④まで実行すると、cql-executionの実行結果（図 16に例を示す）が得られる。これをCQL解析結果として、次節の「CarePlan、RequestGroup作成」機能で使用される。

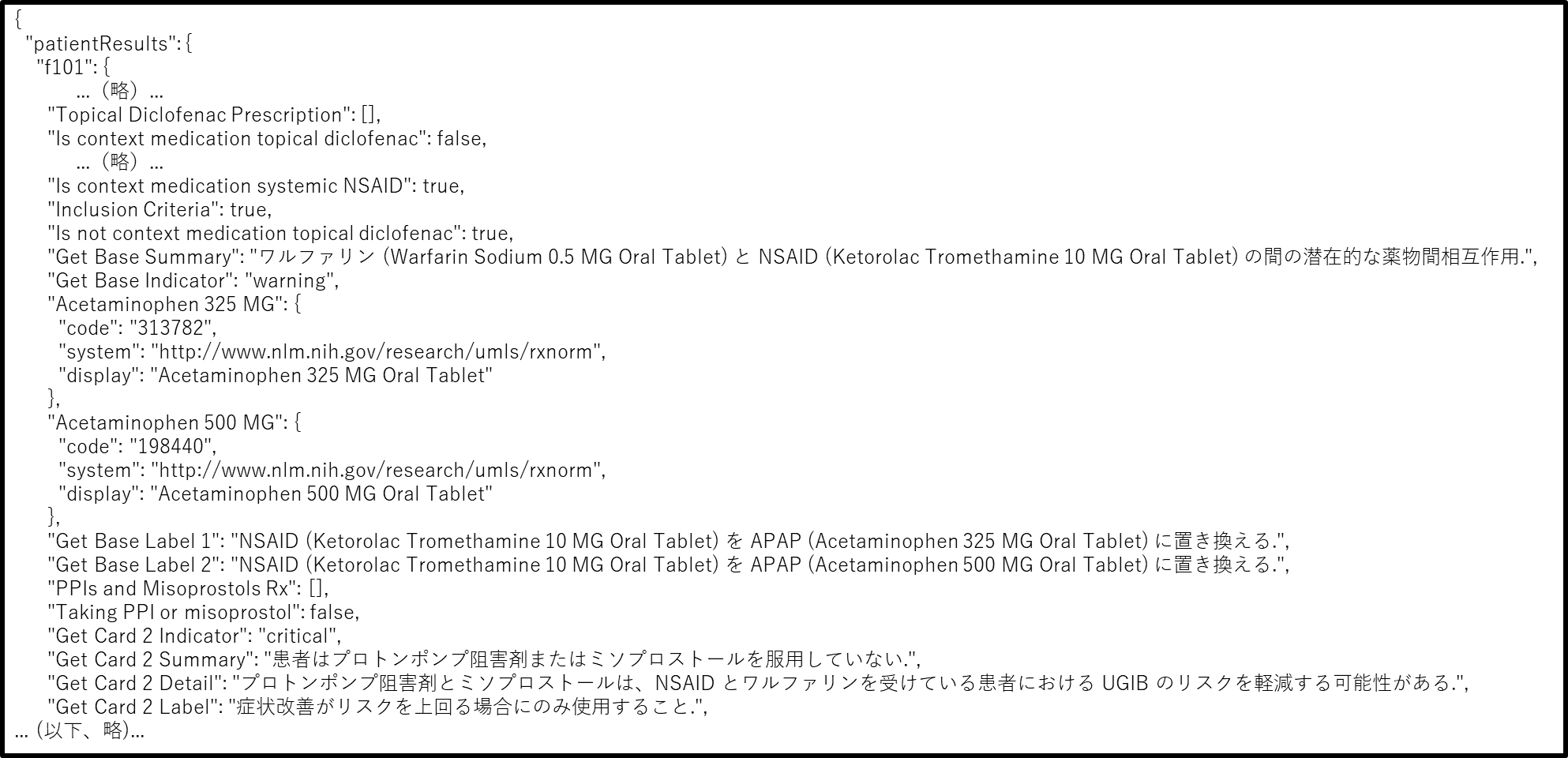


図 　CQL解析結果（Warfarin+NSAIDs、order-sign）の例

なお、図 16の例の全文は「PDDI-CDSサービス-機能仕様書\_附属書\_各種メッセージ.xlsx」のタブ「CQL解析結果」を参照のこと。

### CarePlan, RequestGroup作成

FHIR　PlanDefinitionリソースをベースにし、かつCQL解析結果を反映して、CardのもととなるFHIR CarePlanリソースとFHIR RequestGroupリソースを作成する。簡単な概要図を図 17に示す。

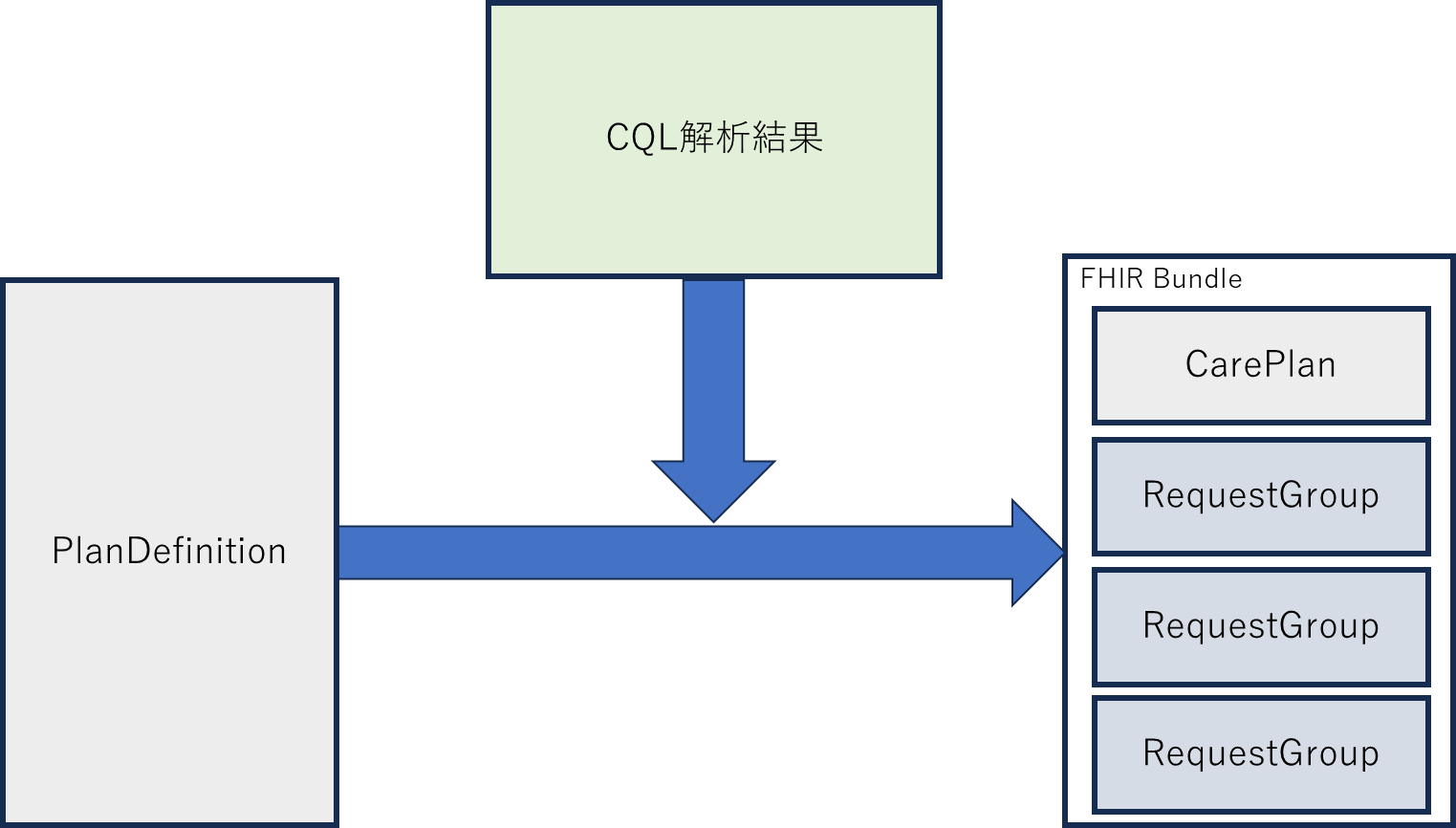


図 　CarePlan、RequestGroup作成の概略

CarePlan及びRequestGroupリソースを作成するにあたり、以下の2点を実行する。

1. PlanDefinitionリソースからCarePlan、RequestGroupリソースを作成
2. CQL解析結果をCarePlan、RequestGroupリソースへ反映

以下、Ⅰ、Ⅱについて、説明する。

1. PlanDefinitionリソースからCarePlan、RequestGroupリソースを作成

図 18にPlanDefinitionリソースとそれが参照するActivityDefinitionリソースから、CarePlan、RequestGroup及びRequestGroupが参照するFHIRリソースを作成するイメージ図を示す。

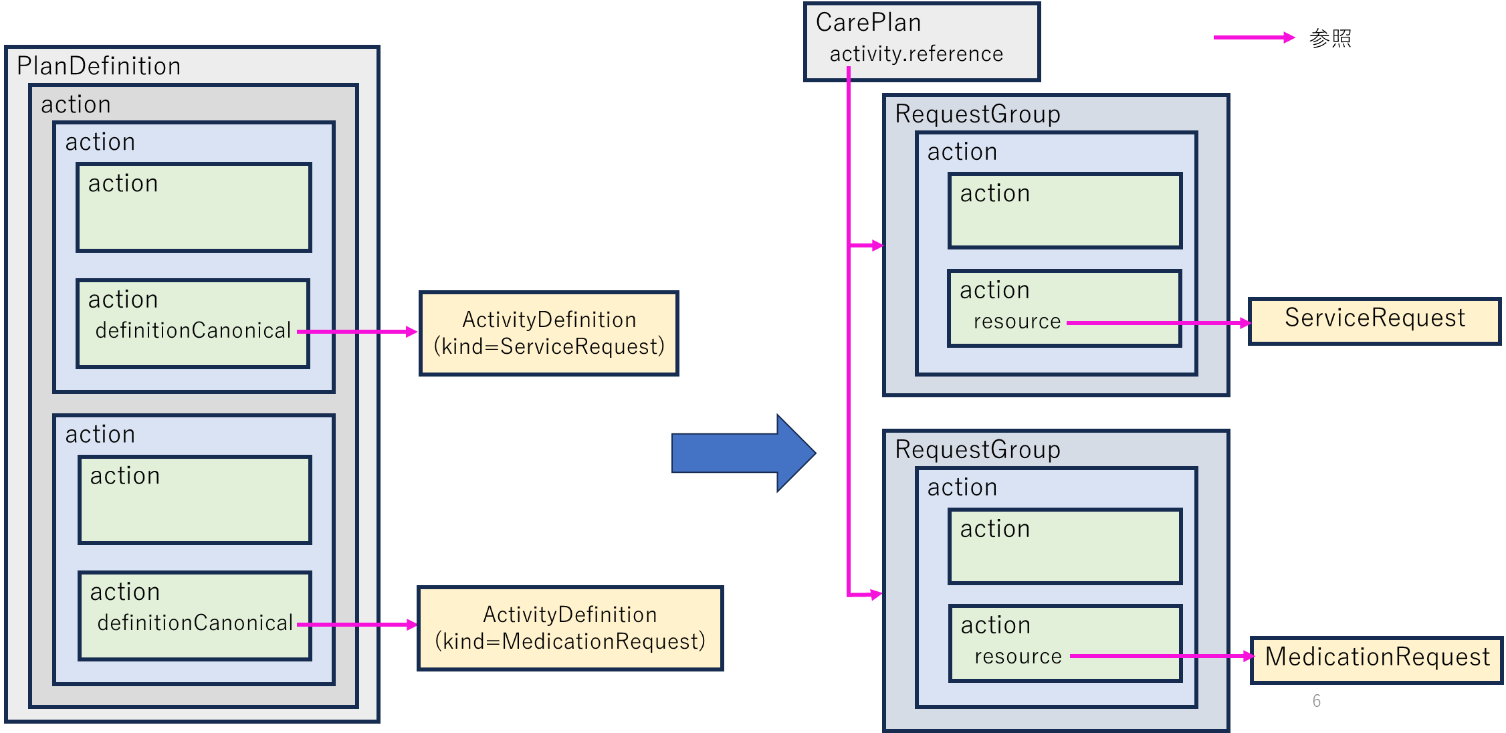


図 　PlanDefinitionからCarePlan, RequestGroupの作成

本作成機能ではPlanDefinition、ActivityDefinitionの各リソースからの変換を実施する。

変換に際しての対応関係を以下に示す。

* PlanDefinition.action.actionとRequestGroup.action。ただしRequestGroup作成時にid（RequestGroup.id）の新規設定を必ず行う。
* PlanDefinition.action.action.actionとRequestGroup.action.action
* ActivityDefinitionとServiceRequestまたはMedicationRequest。どちらに変換されるかはActivityDefinition.kindが”ServiceRequest”か”MedicationRequest”かのどちらに設定されているかによる。またServiceRequest、MeficationRequest作成時にidの新規設定を行う。
* PlanDefinition.actionとCarePlan。PlanDefinition.action.actionから作成したRequestGroupへの参照情報はCarePlan.activity.reference.referenceへ設定する。またCardに挿入するインジケーターの値はCarePlan.activity.extensionへそれぞれ設定する（後述。Ⅱを参照のこと）

対応関係のイメージ図を図 19に示す。

「PDDI-CDSサービス-機能仕様書\_附属書\_リソース変換ルール.xlsx」にはFHIRリソース間の変換を行うために属性間の対応が定義されている。PlanDefinitionからCarePlan及びRequestGroupへの属性間対応は同附属書のタブ「PlanDefinition->CP,RG」を参照のこと。また、ActivityDefinitionからActivityDefinition.kind属性で指定されたFHIRリソースへの属性間対応は同附属書のタブ「ActivityDefinition変換」を参照のこと。

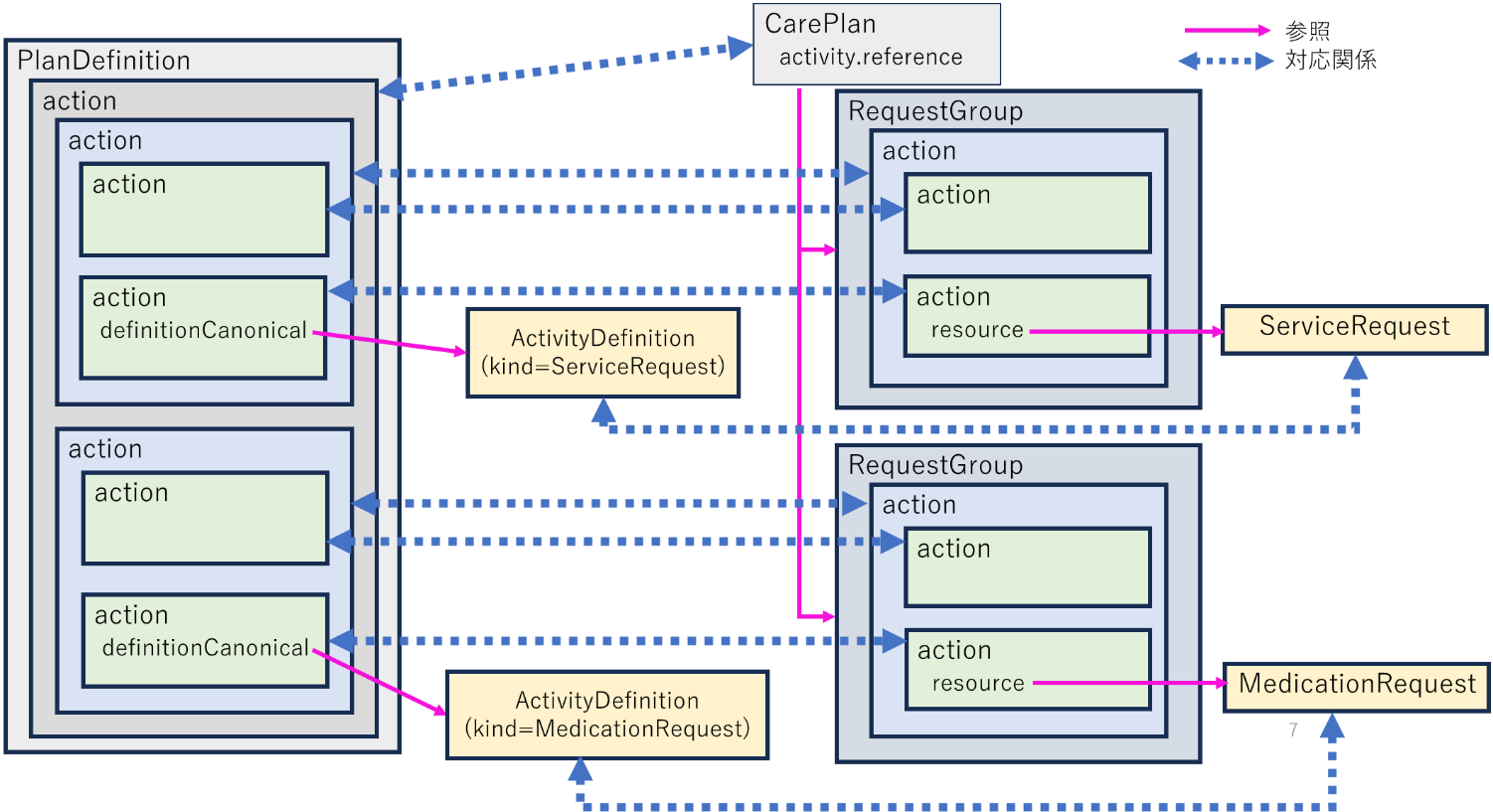


図 　リソース間の対応関係

1. CQL解析結果をCarePlan、RequestGroupリソースへ反映

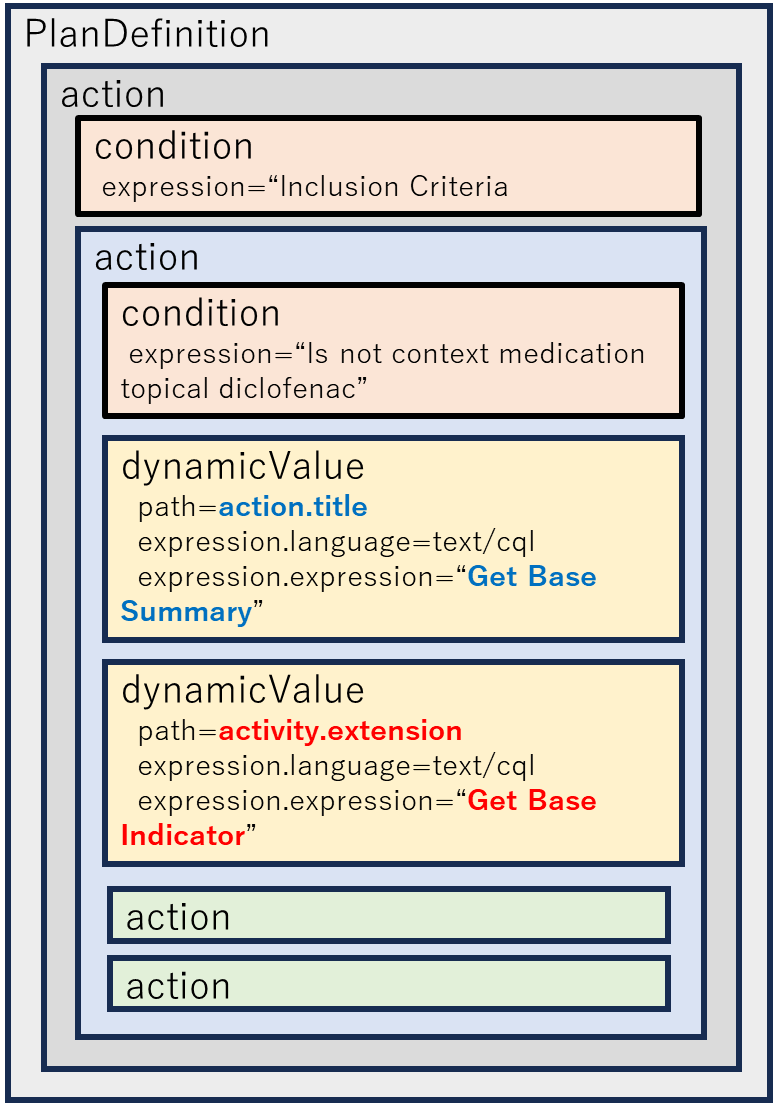


図 　condition、dynamicValueを含むPlanDefinitionリソースのイメージ図

CQL解析結果をCarePlan及びRequestGroupリソースに反映させるには、PlanDefinitionリソースのaction属性にcondition、dynamicValueといった属性を利用する（これらの属性を含んだPlanDefinitionリソースのイメージ図を図 20に示す）。

* condition

actionの子属性にconditionが設定されている場合、そのactionが適用されるかどうかを決定するために使用される。そのための条件はcondition.expressionにCQLで定義する条件式を記述する。CQL解析の結果、この条件がtrueとなる場合はconditionを含むactionは適用される。また一つのaction属性の子属性として複数のcondition属性が存在する場合は、これらのconditionがすべてtrueとなるときこのactionは適用となる。

図 21にconditionのイメージ図を示す。この図では、PlanDefinition.action (action0)の子要素に2つのaction（action01、action02）が存在する。また各actionの子要素にconditionがあり、CQL解析結果から、各conditionのtrue、falseが図 21に記入されているとおりとする。この時、action01のconditionがtrueとなっているので、action01に対応するRequestGroupは作成されるが、action02のconditionがfalseとなっているので、action02に対応するRequestGroupは作成されない。

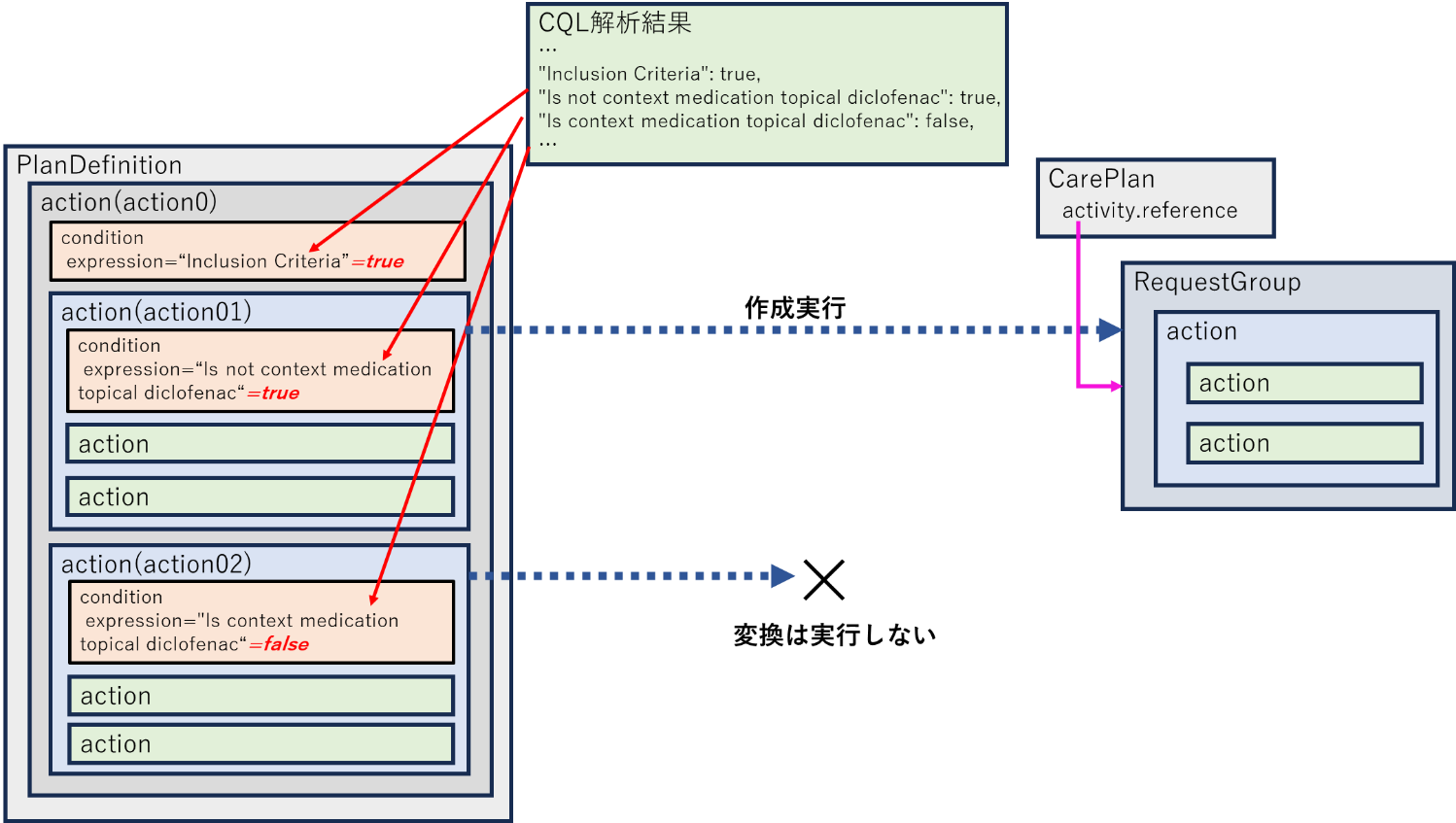


図 conditionの動作イメージ　その1

図 22にはconditionにおける特殊な例として、PlanDefinition.actionがaction0のみ存在し、このaction0のconditionがfalseとなるイメージ図を示す。この場合、action01やaction02のconditionの結果に関係なく、action0全体が適用されない。したがって本機能で作成されるCarePlan及びRequestGroupリソースはひとつも作成されない。この例のように、本機能実行の結果RequestGroupが一つも作成できなかった場合は、次に実行されるCard作成機能においてCardが作成できないため、PDDI-CDSサービスのレスポンスとして空のCard配列を返す。

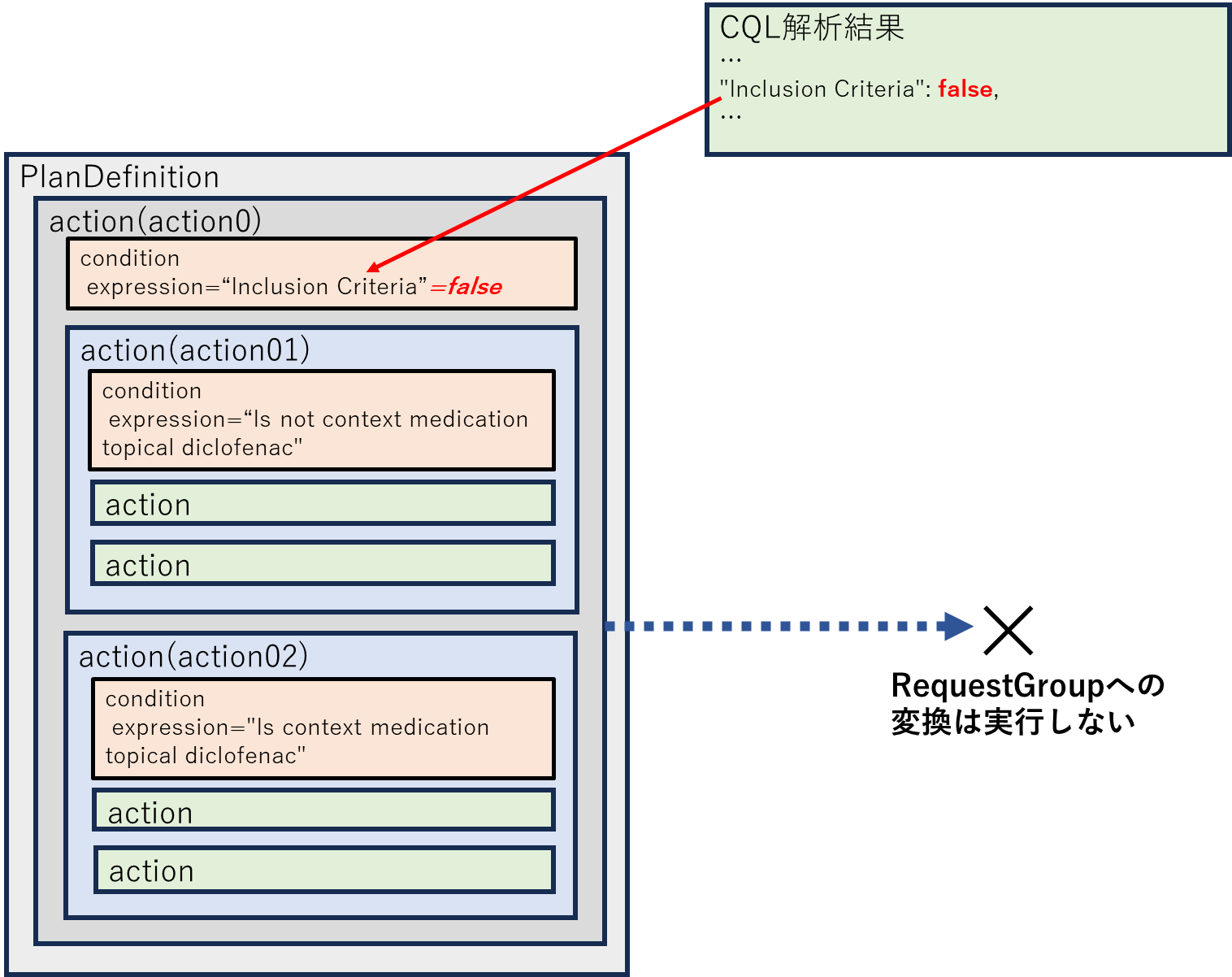


図 conditionの動作イメージ　その2

* dynamicValue

PlanDefinitionのaction属性の子属性であるdynamicValueを使うことで、静的に定義されたリソースのカスタマイズができるようになる。PDDI-CDSの各決定ブロックには1つ上の個別の情報コンポーネントがあるため、患者固有及び製品固有のデータをRequestGroupに反映できるようになる。図 23にdynamicValueを用いてCQL解析結果をRequestGroupへ反映させるイメージ図を示す。

dynamicValueはexpressionで指定した、CQLでの定義項目（図中の”Get Base Summary”や“Get Base Indicator”がこれにあたる）に対する解析結果値を、CarePlanやRequestGroupのどこに設定するかをpathで指定する。主なpath設定値に対応するCarePlan及びRequestGroupリソースの属性名を表 4に示す

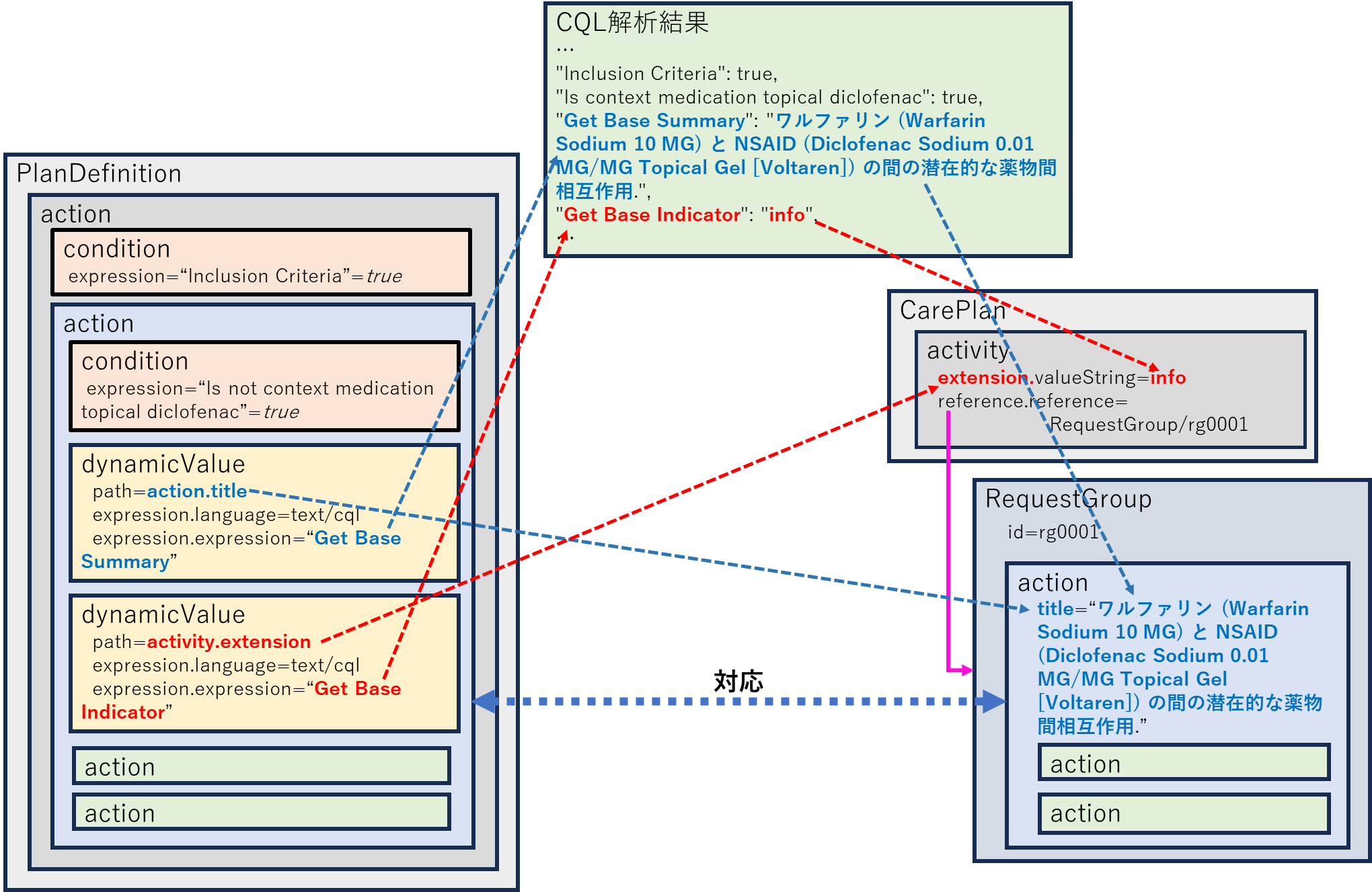


図 dynamicValueの動作イメージ

表 path設定値に対応するCarePlan及びRequestGroupリソースの属性名

|  |  |  |
| --- | --- | --- |
| dynamicValue.path | 設定先FHIRリソース | 設定先属性名 |
| action.title | RequestGroup | 対応するactionのtitle属性 |
| action.label | RequestGroup | 対応するactionのprefix属性 |
| action.prefix | RequestGroup | 対応するactionのprefix属性 |
| action.description | RequestGroup | 対応するactionのdescription属性 |
| activity.extension | CarePlan | activity.extension.valueString |

本機能で作成されたCarePlan、RequestGroupなどのFHIRリソースはFHIR　Bundleリソースにまとめられて（）、次に実行される「Card作成」機能へ渡される。このBundleリソースの例を図 25に示す。この例の全文は「PDDI-CDSサービス-機能仕様書\_附属書\_各種メッセージ.xlsx」のタブ「CarePlan＋RequestGroup」を参照のこと。

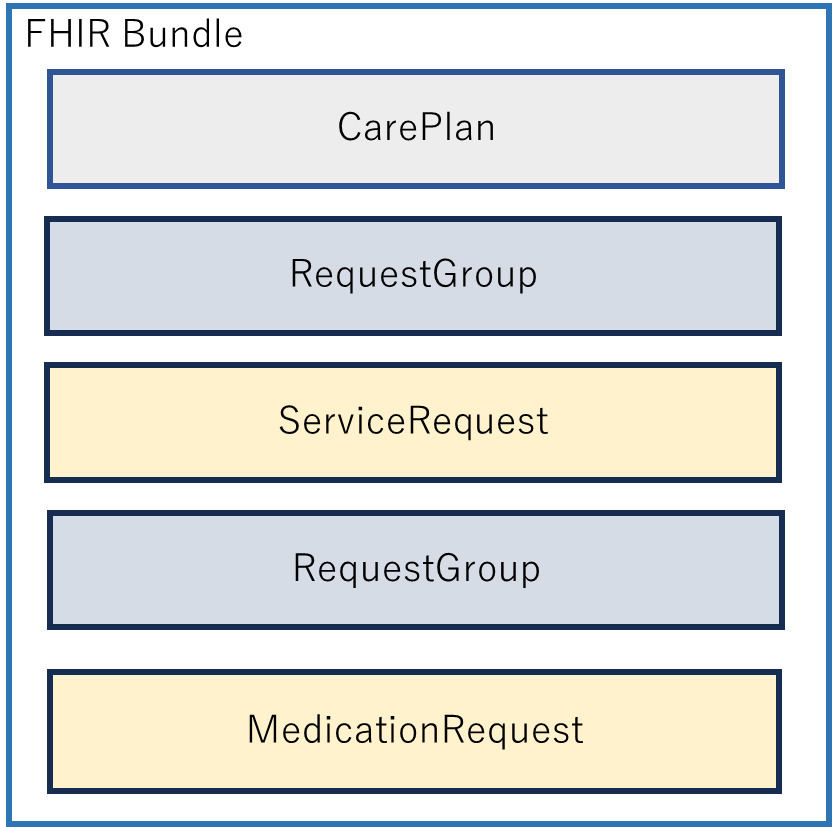


図 　CarePlan、RequestGroup作成機能での出力となるFHIR Bundleリソース

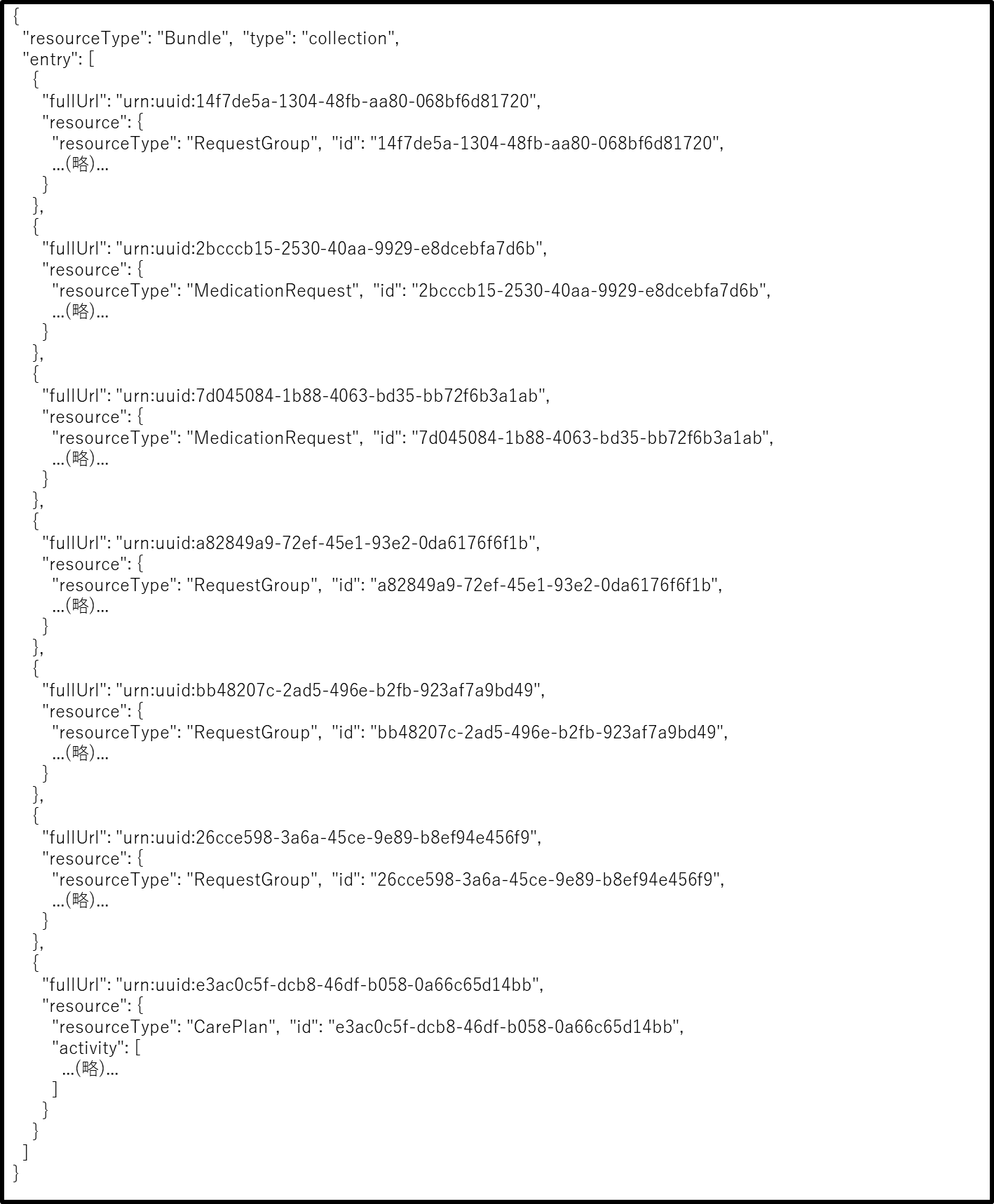


図 CarePlan、RequestGroup作成機能での出力となるFHIR Bundleリソースの例

### Card作成

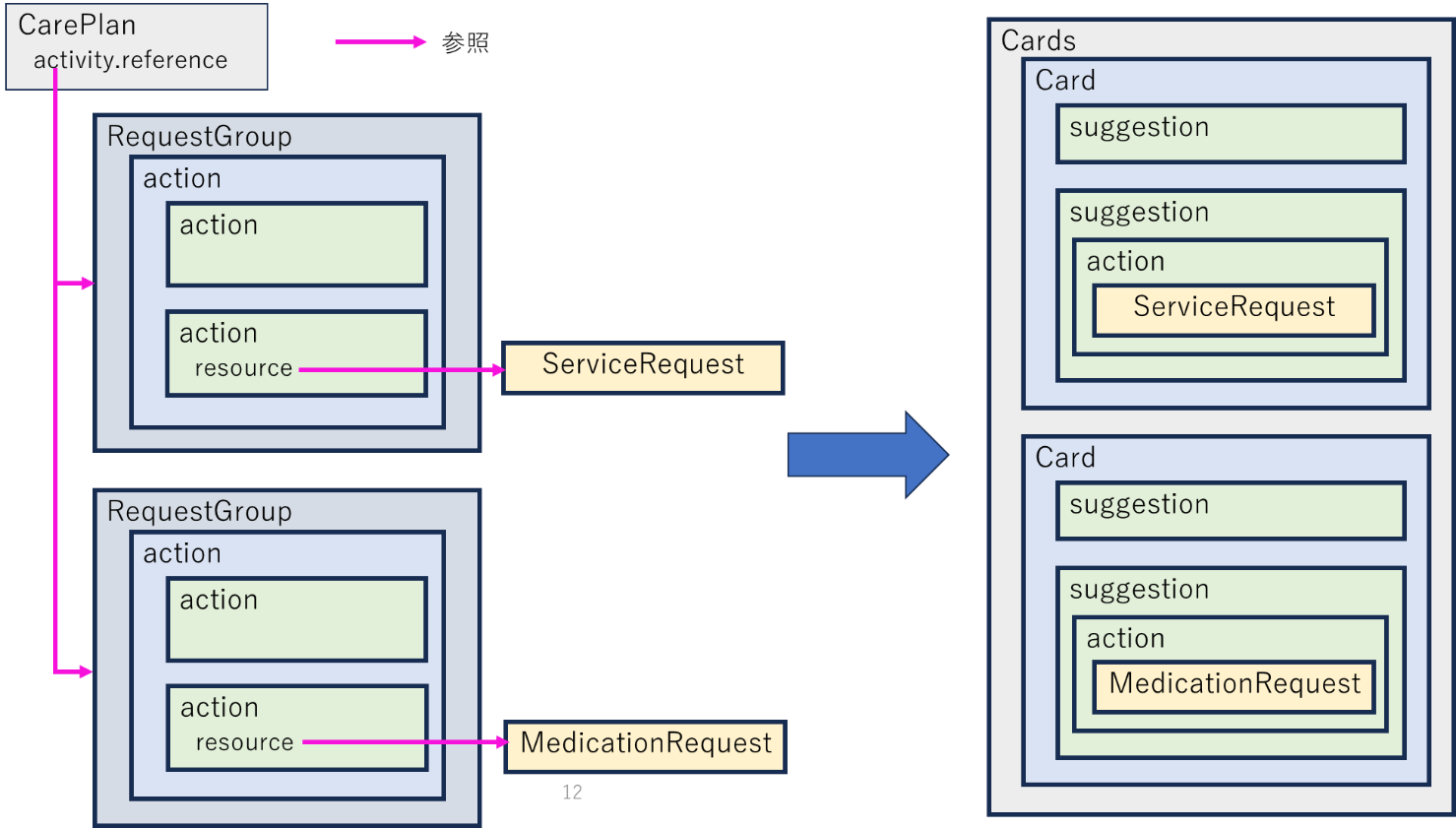


図 　CarePlan, RequestGroupリソースからCard作成のイメージ

「CarePlan、RequestGroup作成」機能で作成したFHIRリソースを用いて、PDDI-CDSサービスのレスポンスとなるCard配列（Cards）を作成する。図 26にそのイメージ図を示す。

本作成機能ではCarePlan、RequestGroupなどのFHIRリソースからCardへの変換を行うことで、Card作成を実施する。変換に際しての対応関係を以下に示す。

* RequestGroup.action と　Card
* RequestGroup.action.actionとCard.suggestion
* CarePlan.activity
* RequestGroup.action.action.resourceで参照されるFHIRリソース（ServiceRequest、MedicationRequest）はCard.suggestion.actions.resourceへ埋め込む。

CarePlan、RequestGroupなどのFHIRリソースからCardを作るための変換ルールは「PDDI-CDSサービス-機能仕様書\_附属書\_リソース変換ルール.xlsx」のタブ「CP,RG->Card」を参照のこと。作成されたCard配列の例を図 27に示す。この例の全文は「PDDI-CDSサービス-機能仕様書\_附属書\_各種メッセージ.xlsx」のタブ「Cards」を参照のこと。

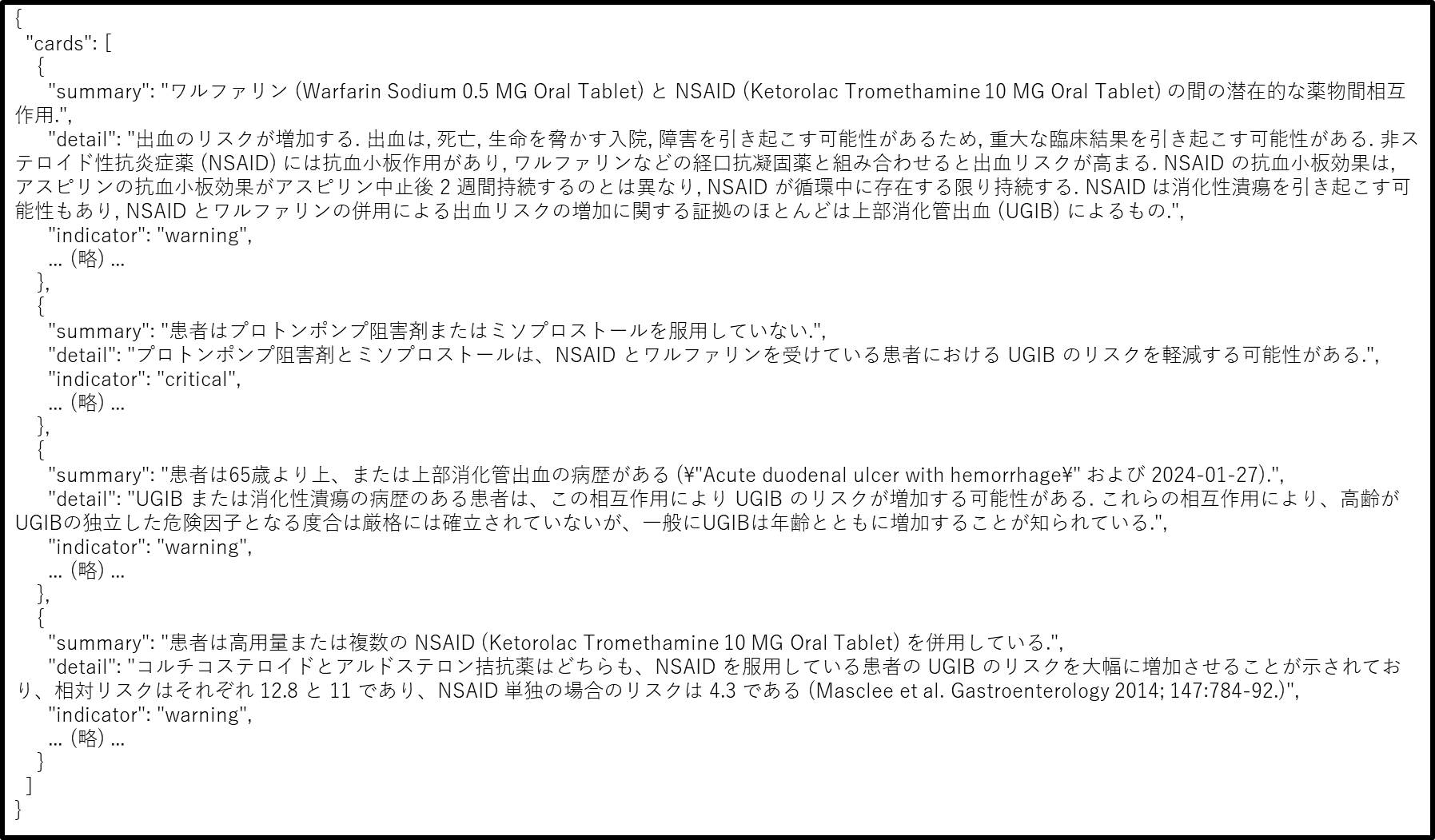


図 Card配列の例

また前節（3.3.3節）でも触れたように、CQL解析結果反映の結果、CarePlan及びRequestGroupが一つも作成されなかった場合は、PDDI-CDSシステムのレスポンスとして、ステータスコード200で空のCard配列（図 28）を返却する。

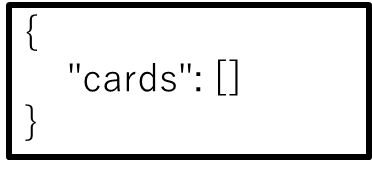


図 　空のCard配列

# 未実装及び未対応事項

## 未実装・未対応事項

本PDDI-CDSサービスにおいて、未実装事項及び未対応事項は以下の通り

* 本サービスはWebサービスとして機能するが、SMARTonFHIR及び認証機能はサポートしていない。
* CSRF対策、XSS対策などのセキュリティ対策は講じていない。
* SSL、TLSのような通信路暗号化には対応していない。
* 監査ログの出力は行っていない。
* PDDI-CDSサービスとして、FHIRサーバへのアクセスは考慮していない。CQL解析に使用するコードはストレージに入っているものを使う。
* PDDI-CDS実装ガイド、3.CDS Service Specificationの3.4.3 CDS Hooks Requestに記されているExtensionについて、これに関する実装は行っていない。

以下については現時点で対応していないが、システム改変を行わずに必要なデータの追加を行うことで利用可能となる。なおこれらの事項に対して、確認、テストは行っていない

* order-selectには対応していない
* WarfarinとNSAID、DigoxinとCyclosporineの2つの対象薬品のペア以外では対応していない。

## 必要な対策を講じた事項

* **ContextPrescriotionsについて**

PDDI-CDS実装ガイドの「5. Knowledge Representation for PDDI CDS」 [3]において、CQLルールの中で

parameter ContextPrescriptions List<MedicationRequest>

と定義することで、CDSHooksリクエストのcontext.draftOrders要素に含まれるMedicationRequestリソースをContextPrescriptionsパラメータに割り当てることができる。

しかし、cql-executionによるCQL解析の実行において、CDSHooksリクエストのcontext.要素にMedicationRequestリソースを含んでいる場合でも、ContextPrescriptionsパラメータにMedicationRequestリソースが割り当てられていないことが判明した。

　そこでcql-executionによるCQL解析を行う上での回避策として、以下を行う。

1. CQL解析準備機能において、CDSHooksリクエストに含まれるFHIRリソースをFHIR Bundleリソースにまとめるが、このFHIR Bundleリソースに含まれる各FHIRリソースはもともとCDSHooksリクエストのcontext.draftOrders以下にあったのか、prefetch以下にあったのか区別がつかなくなるため、context.draftOrders以下にあったFHIRリソースにはextensionを利用して以下に示すフラグ（図 29赤字の部分）を付与する。なお図 17赤字部分のextension.urlとしてUUIDが設定されているが、このUUID値はPDDI-CDSサービスを表す固有値として設定される。

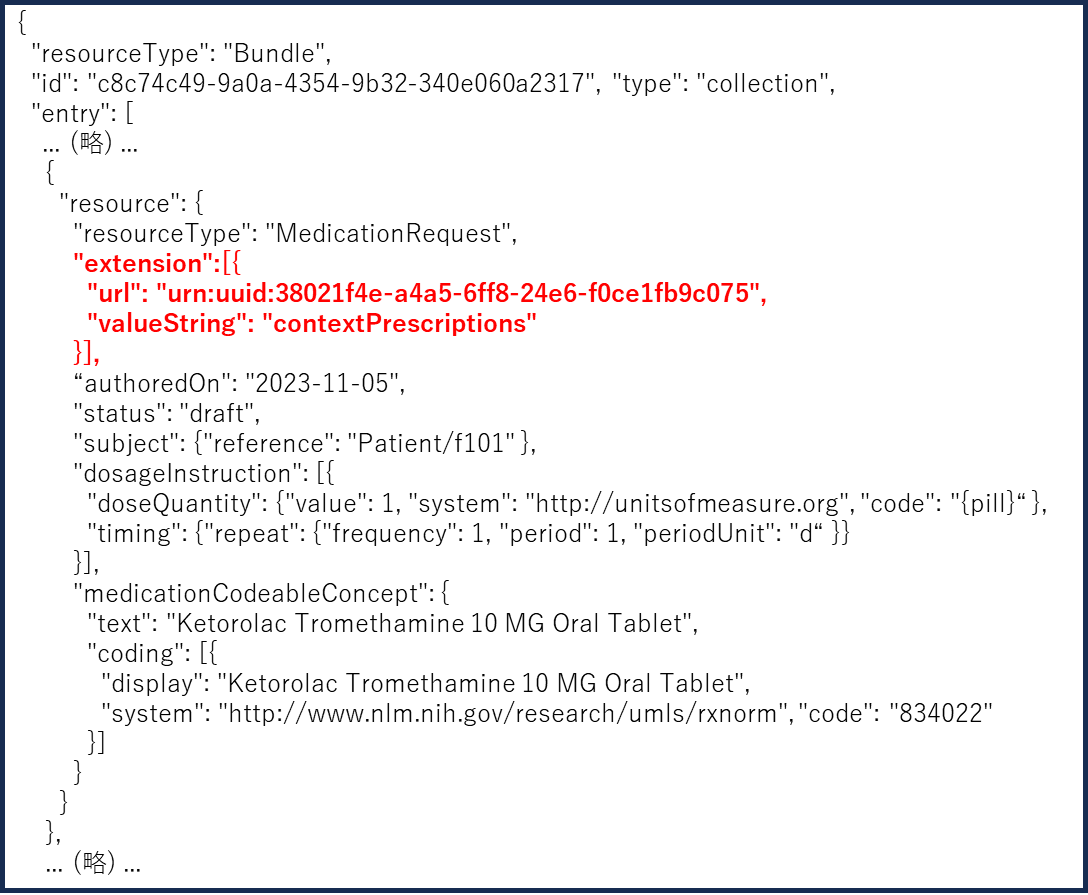


図 contextPrescriotionsフラグの付与

1. CQLルールにおいてContextPrescriptionsによる記述をやめ、図 30のように、リソースの集合（図 30での[MedicationRequest]および[MedicationRequest： “Topical Diclofenac”]）を用いた記述に書き換える。書き換えは「代替案1」と「代替案2」のどちらかに基づいて行う。

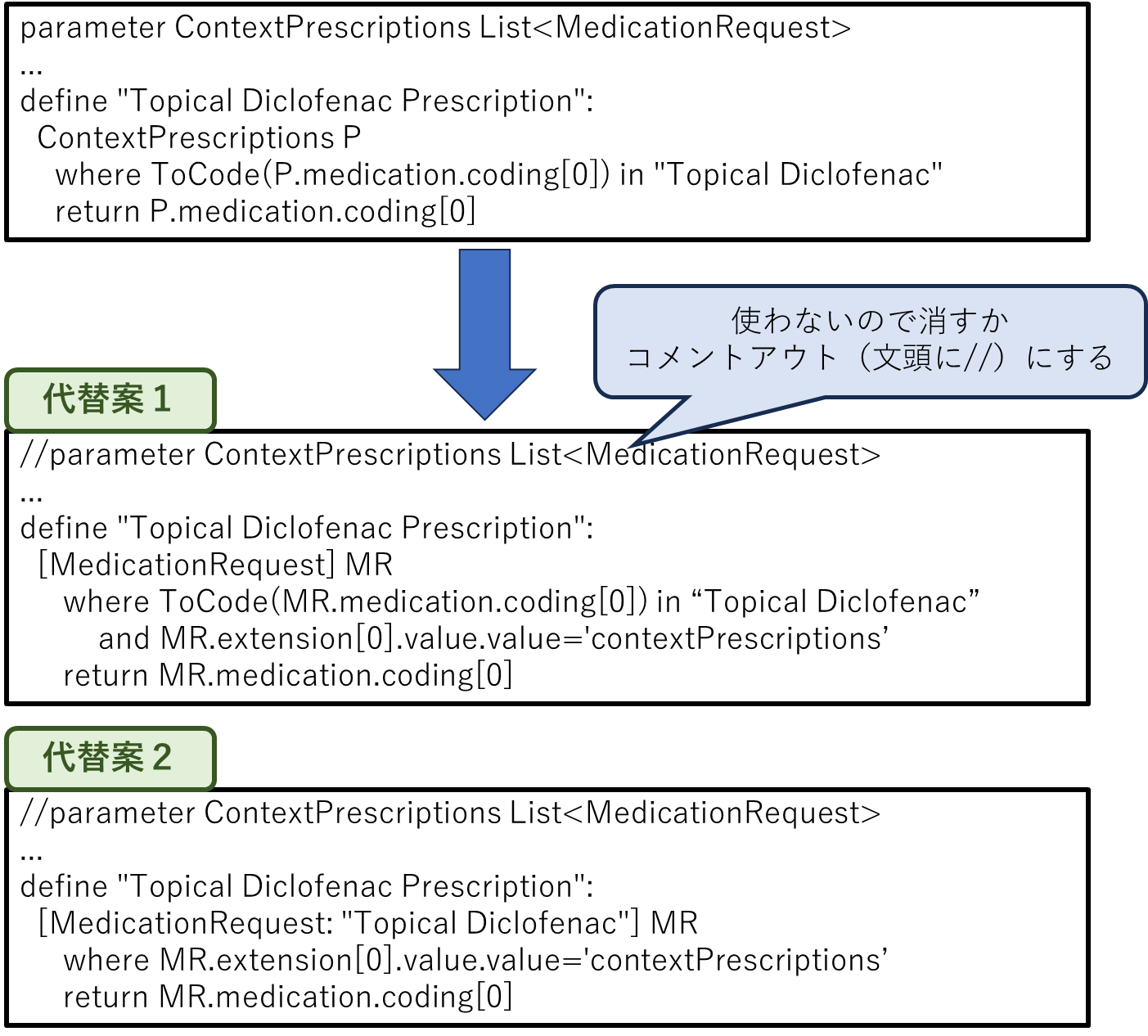


図 CQLにおけるContextPrescriptionsの代替

* **CQLで使用される単位系**

図 31にあるコードを含むCQLルールをCQL to ELM Translator [4]によりELMに変換しようとしたところ、以下のエラーが出力された。

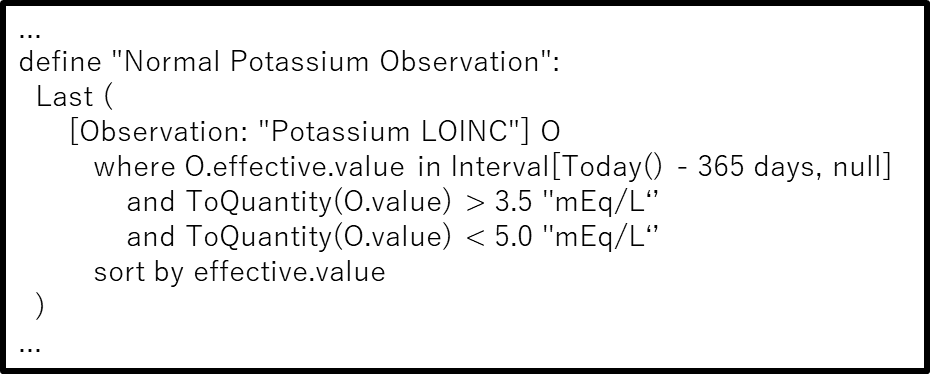


図 単位「mEq」を含むCQL

**Error Processing unit ’ｍEq/L’:**

**The unit ’mEq’ is unknown at position O**

CQL to ELM Translatorによる変換では、単位系のチェックにCQL to ELM Translatorで利用しているライブラリ「ucum-1.0.3.jar」に含まれるucum-essential.xmlを参照する。では「Eq」ではなく「eq」が用いられている(図 32)ことから、図 31において

　　 mEq/L →　meq/L

と修正することでエラーは出なくなった。

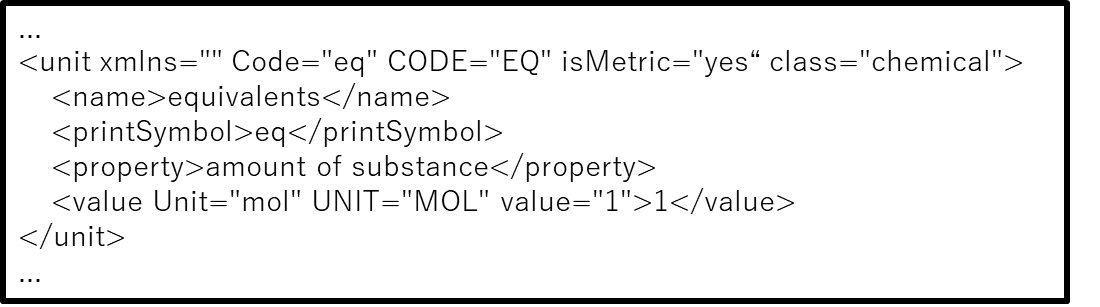


図 ucum-essential.xml

そこで、本件での対処法として、以下の２つのいずれかを適用する。

1. CQLルール、CDSHooksリクエストの両方で、単位系をCQL to ELM Translatorで利用しているライブラリに合わせる。

「mEq/L」を、ucum-1.0.3.jarのucum-essential.xmlでの記載に合わせて、「meq/L」に修正したように、使用ライブラリで定義されている書式に合わせる。

（今回はこちらの対処法を適用する）

1. CQL to ELM Translatorで利用しているライブラリに、CQLルールで使用する単位や記法を追加する。

例に即していえば、「mEq/L」でもエラーを出さないようにするため、ucum-1.0.3.jarのucum-essential.xmlに記号「Eq」の定義を追加する。

## 対策が必要な事項

* **PDDI-CDSサービスにおいて、エラー発生時のレスポンスについて**

PDDI-CDSサービスからのレスポンスは、成功時はHTTPステータスコード200でCard配列を返却する。しかし、PDDI-CDSサービスでエラー発生により処理続行が困難となった場合に、CDSHooksリクエスト送信元にエラーメッセージを通知する方法がPDDI-CDS実装ガイドには定義がなく、CDS-HooksにはHTTPステータスコードとして400番台ないし500番台を返す場合があるとするだけで、そのほかにエラーメッセージの通知に関する記述がない。

このため、PDDI-CDSサービスにおいてエラーが発生した場合にCDSHooksリクエスト送信元に対してCard配列ではなくエラーメッセージを送るためのメッセージ形式を定義する必要がある。

一案として、FHIRのOperationOutcomeリソース [9]の利用を考える。エラーメッセージを設定したOperationOutcomeリソースの例を図 33に示す。

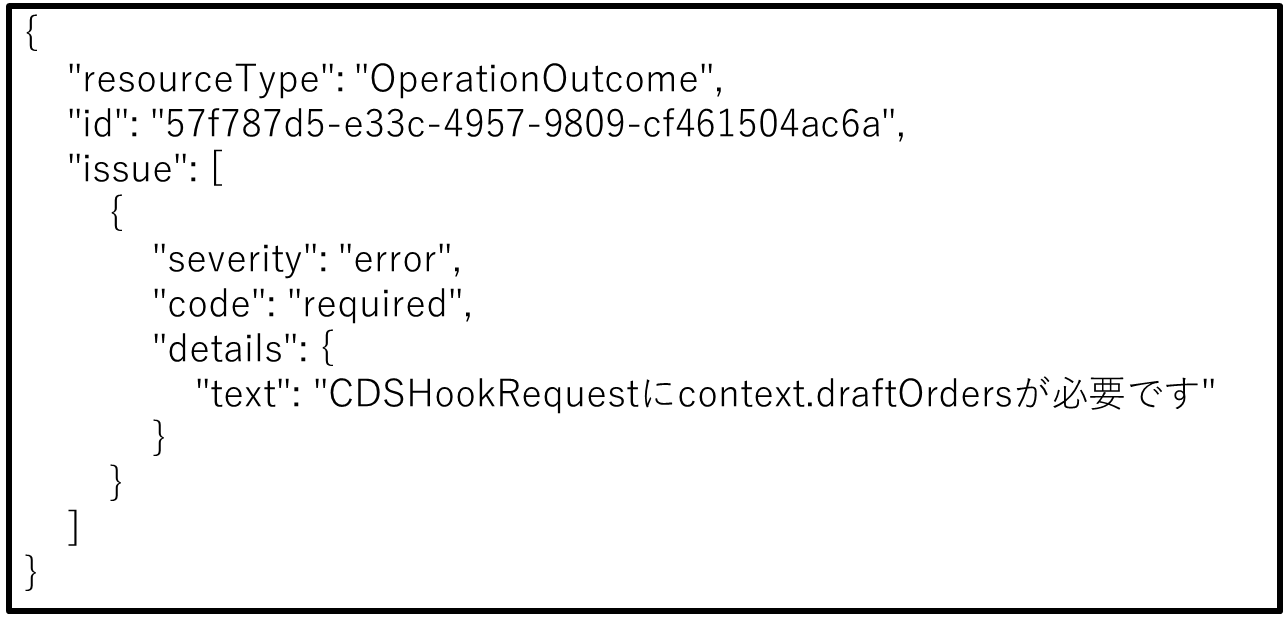


図 OperationOutcomeリソースによるPDDI-CDSサービスのエラーレスポンス例

OperationOutcomeリソースはクライアントからのリクエストに対して、FHIRサーバでの処理中にエラーが発生した場合に、エラーの内容をクライアントへのレスポンスとして含めるときに利用することから、PDDI-CDSサービスにおいても同様に利用できると考えられる。

# 参考文献

|  |  |
| --- | --- |
| [1] | “HL7 CDS Hooks,” https://cds-hooks.org/. |
| [2] | HL7, “Clinical Quality Language Release 1,” https://cql.hl7.org/. |
| [3] | HL7, “Potential Drug-Drug Interaction (PDDI) CDS IG : STU1 Ballot 2, 5. Knowledge-artifacts,” https://hl7.org/fhir/uv/pddi/2023Jan/knowledge-artifacts.html. |
| [4] | “cqframework/clinical\_quality\_language,” https://github.com/cqframework/clinical\_quality\_language. |
| [5] | “HL7 FHIR Release 4, 12.18 Resource PlanDefinition - Content,” https://hl7.org/fhir/R4/plandefinition.html. |
| [6] | “HL7 FHIR Release 4, 14.11 Resource Library - Content,” https://hl7.org/fhir/R4/library.html. |
| [7] | “cqframework / cql-execution,” https://github.com/cqframework/cql-execution. |
| [8] | “cqframework / cql-exec-fhir,” https://github.com/cqframework/cql-exec-fhir. |
| [9] | “HL7 FHIR Release 4, 2.44 Resource OperationOutcome - Content,” https://hl7.org/fhir/R4/operationoutcome.html. |

1. PDDI-CDSサービスで発生したエラーが発生しても、その内容の伝達についてPDDI-CDS実装ガイドでは定義がなく、CDS-HooksにはHTTPステータスコードとして400番台ないし500番台を返す場合があるとの記載されているだけで、そのためエラー発生時のCDSルール実行レスポンスにはHTTPステータスコードとHTTPヘッダのみ値を設定し、レスポンスボディは何も設定しないとした。しかし、実際HTTPステータスコードだけでエラーの内容までは判定できないことから、前述のレスポンスボディにはエラーの内容を記述したメッセージが必要と考えられる。後述の4.3節を参照のこと。 [↑](#footnote-ref-1)