

# 資料2

---

## CRF Reporter と電子カルテの 連携システム仕様

リリース 0.41

大阪大学大学院医学系研究科  
情報統合医学講座 医療情報学

2015年04月08日

i

# 目次

第 1 章 概要	1
第 2 章 CRF Reporter システム構成	2
第 3 章 CRF Reporter Communication DLL インタフェース仕様	3
3.1 電子カルテと CrfReporterComm DLL の連携シーケンス	4
3.1.1 CRF Reporter アプリケーション起動シーケンス	4
3.1.2 CRF Reporter アプリケーション終了シーケンス	5
3.1.3 CRF Reporter イベントデータ入力シーケンス	6
3.1.4 CRF Reporter 新規イベントデータ入力シーケンス	7
3.1.5 CRF Reporter イベントデータ修正シーケンス	8
3.1.6 CRF Reporter イベントデータ削除シーケンス	9
3.2 ICrfReporterComm インタフェース	10
3.2.1 ICrfReporterComm メソッド	11
ExitApp メソッド	11
NotifyEventDataCanceled メソッド	11
NotifyEventDataCommitted メソッド	12
NotifyEventDataDeleted メソッド	12
ShowDialog メソッド	13
ShowModifyEventDataDialog メソッド	13
StartApp メソッド	14
StartApp メソッド (IntPtr)	14
StartApp メソッド (String, String)	14
3.2.2 ICrfReporterComm イベント	15
AppExiting イベント	15
EventEntryCompleted イベント	15
3.3 EventEntryCompletedEventArgs クラス	16
3.3.1 EventEntryCompletedEventArgs プロパティ	16
EventDataId プロパティ	16
Text プロパティ	17
3.4 CrfReporterException クラス	17
3.5 CrfReporterAuthFailedException クラス	17
3.6 CrfReporterRunningException クラス	18
第 4 章 EmrComm プラグイン DLL インタフェース仕様	19
4.1 IEmrComm インタフェース	20
4.1.1 Authenticate メソッド	21
4.1.2 PatientProfile メソッド	22
4.1.3 BodyMeasurement メソッド	23
4.1.4 Hospitalization メソッド	24
4.1.5 Operation メソッド	25
4.1.6 Laboratory メソッド	27
4.1.7 Prescription メソッド	29
4.2 データ取得メソッドが返す応答メッセージ	30
第 5 章 CRIT 応答メッセージ XML 仕様	31
5.1 概要	31
5.1.1 シンタックス記法	31
5.1.2 データフォーマット	31
5.2 PatientProfile 応答メッセージ	32
5.2.1 XML 要素	32
PatientProfile 要素	32
Name 要素	33
Full 要素	33
Family 要素	33
Given 要素	34

		Gender 要素	34
		BirthDate 要素	35
	5.2.2	PatientProfile 応答メッセージ例	35
	5.2.3	PatientProfile 応答メッセージスキーマ	35
5.3		BodyMeasurement 応答メッセージ	37
	5.3.1	XML 要素	37
		BodyMeasurement 要素	37
		Height 要素	38
		Height/Value 要素	38
		Height/DateTime 要素	38
		Weight 要素	39
		Weight/Value 要素	39
		Weight/DateTime 要素	40
	5.3.2	BodyMeasurement 応答メッセージ例	40
	5.3.3	BodyMeasurement 応答メッセージスキーマ	40
5.4		Hospitalization 応答メッセージ	42
	5.4.1	XML 要素	42
		Hospitalization 要素	43
		Admission 要素	43
		ChangeFrom 要素	43
		ChangeTo 要素	44
		Discharge 要素	44
		DateTime 要素	44
		Department 要素	45
	5.4.2	Hospitalization 応答メッセージ例	45
	5.4.3	Hospitalization 応答メッセージスキーマ	46
5.5		Operation 応答メッセージ	48
	5.5.1	XML 要素	48
		Procedures 要素	48
		Procedure 要素	49
		DateTime 要素	49
		Department 要素	49
		Method 要素	50
	5.5.2	Operation 応答メッセージ例	50
	5.5.3	Operation 応答メッセージスキーマ	51
5.6		Laboratory 応答メッセージ	52
	5.6.1	XML 要素	52
		Laboratory 要素	52
		Specimen 要素	53
		CollectionDateTime 要素	53
		Test 要素	53
		Observation 要素	54
		ValueQuantity 要素	54
		ReferenceRange 要素	55
		Text 要素	55
		Low 要素	55
		High 要素	56
	5.6.2	Laboratory 応答メッセージ例	56
	5.6.3	Laboratory 応答メッセージスキーマ	57
5.7		Prescription 応答メッセージ	60
	5.7.1	XML 要素	60
		Prescription 要素	60
		Ingredient 要素	61
		Medication 要素	61
		Form 要素	62
		DosageInstruction 要素	63

	AsNeeded 要素 .....	63
	FrequencyPerDay 要素 .....	64
	QuantityPerDay 要素 .....	64
	QuantityPerDose 要素 .....	64
	DosageInstruction/Period 要素 .....	65
	Site 要素 .....	66
	Route 要素 .....	66
	Method 要素 .....	66
	Dispense 要素 .....	67
	Dispense/Period 要素 .....	67
	Dispense/Quantity 要素 .....	68
5.7.2	Prescription 応答メッセージ例 .....	68
5.7.3	Prescription 応答メッセージスキーム .....	70
第 6 章 変更履歴		74



# 第 1 章 概要

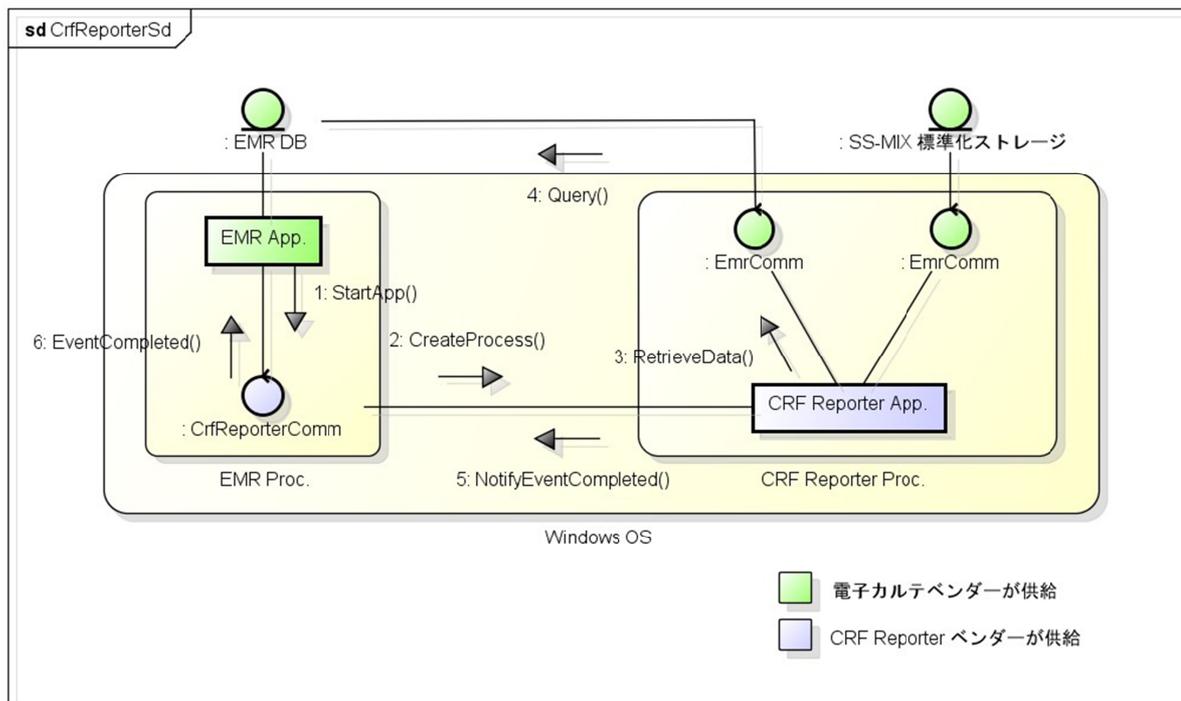
本文書は、厚生労働科学研究費補助金(医療技術実用化総合研究事業(臨床研究・治験推進研究事業))「臨床研究・治験の IT 化推進のための実施プラン策定に関する研究 (Clinical Research on IT: CRIT)」分担研究「病院情報システムと EDC の連動による症例報告書作成とデータ収集の支援に関する研究」において、CRF Reporter と電子カルテシステム間のデータ連携を実現するためのインタフェース仕様を検討した結果を記す。

CRF Reporter が電子カルテと連携するためのインタフェースとしては、以下の 3 種類を検討する。

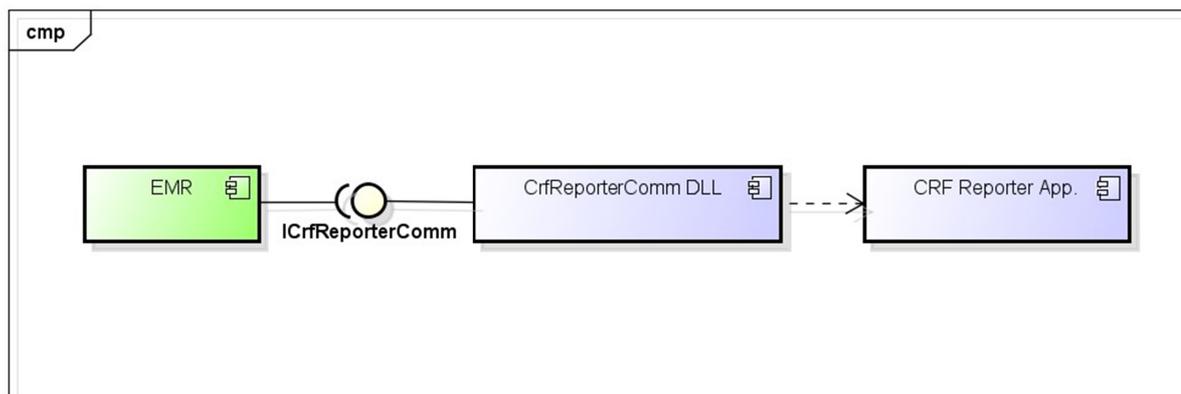
1. 電子カルテから CRF Reporter を起動するためのインタフェース
2. CRF Reporter が電子カルテ内のデータを取得するためのインタフェース
3. 電子カルテの記事と CRF Reporter を連携するためのインタフェース

本文書では、上記 2 番目の CRF Reporter が電子カルテ内のデータを取得するためのインタフェースに対する原案を提示する。

## 第2章 CRF Reporter システム構成



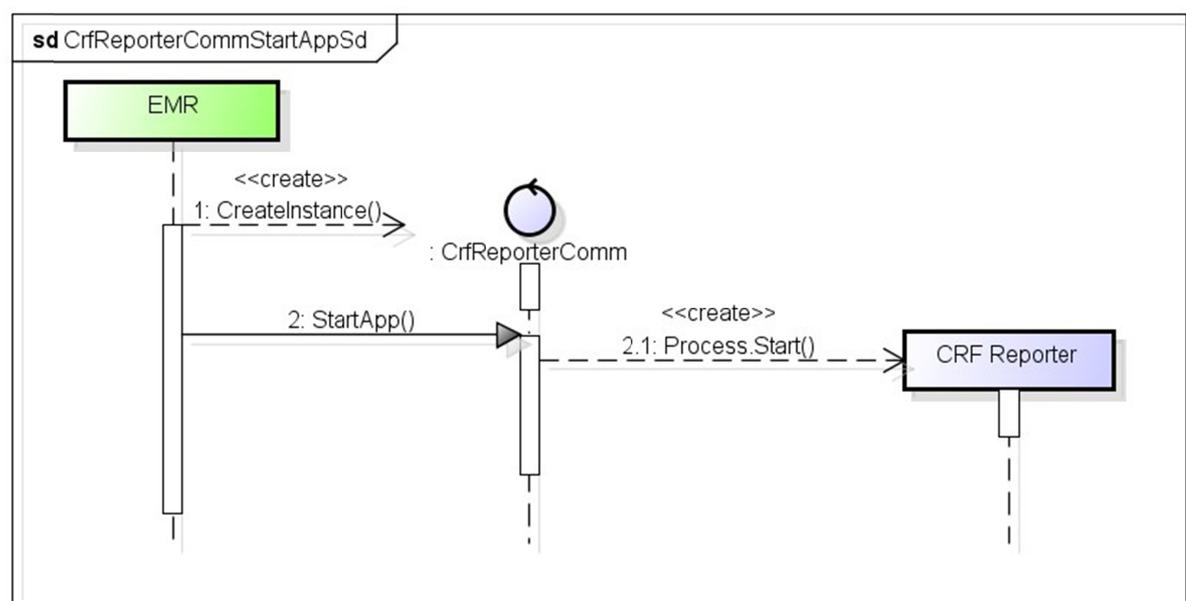
## 第 3 章 CRF Reporter Communication DLL インタフェース仕様



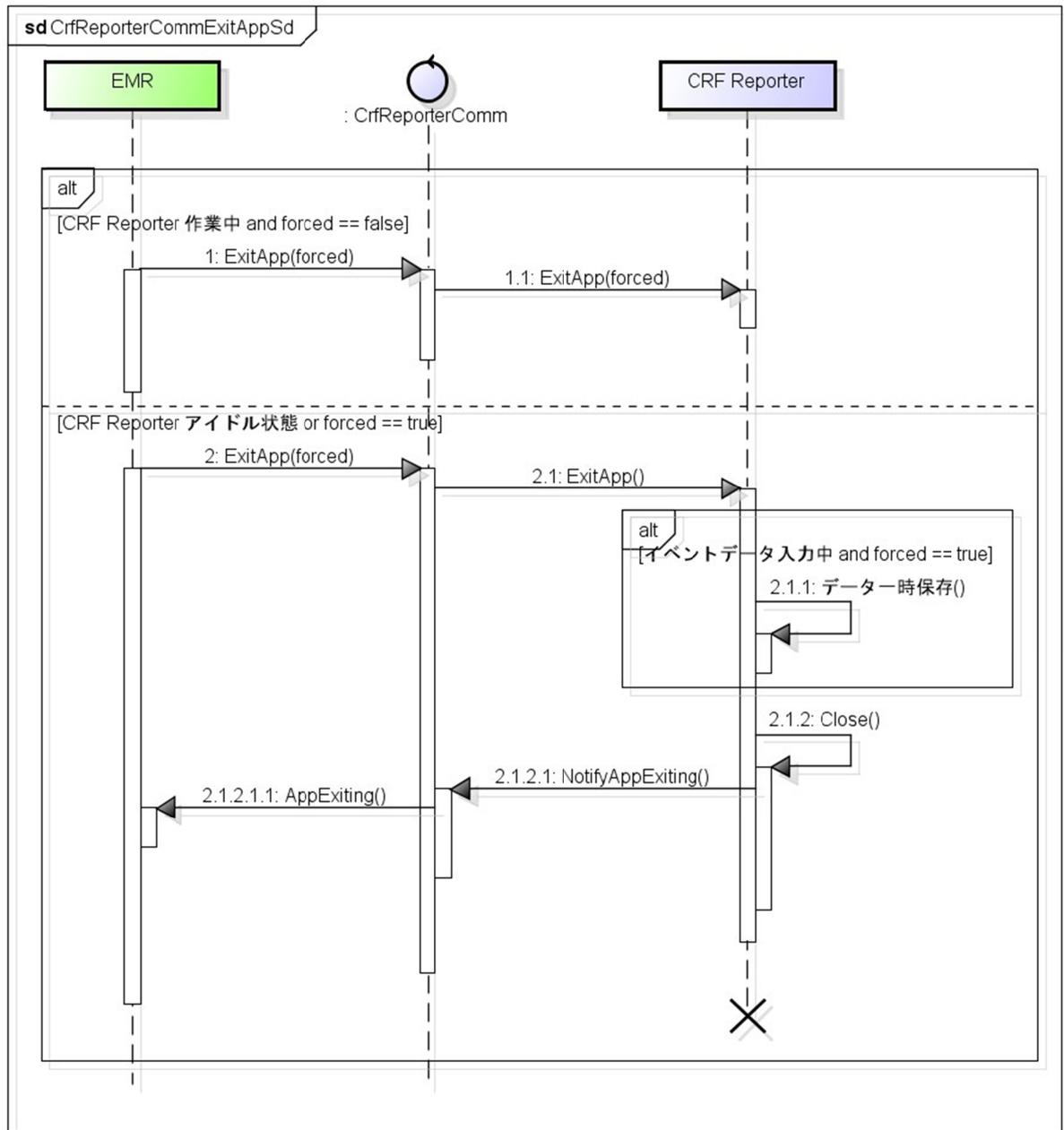
電子カルテアプリケーションが CRF Reporter と関係するための DLL (CRF Reporter Communication DLL) のインタフェース仕様を定義します。本 DLL は、CRF Reporter ベンダーが作成し、電子カルテベンダーに提供するものとします。

## 3.1 電子カルテと CrfReporterComm DLL の連携シーケンス

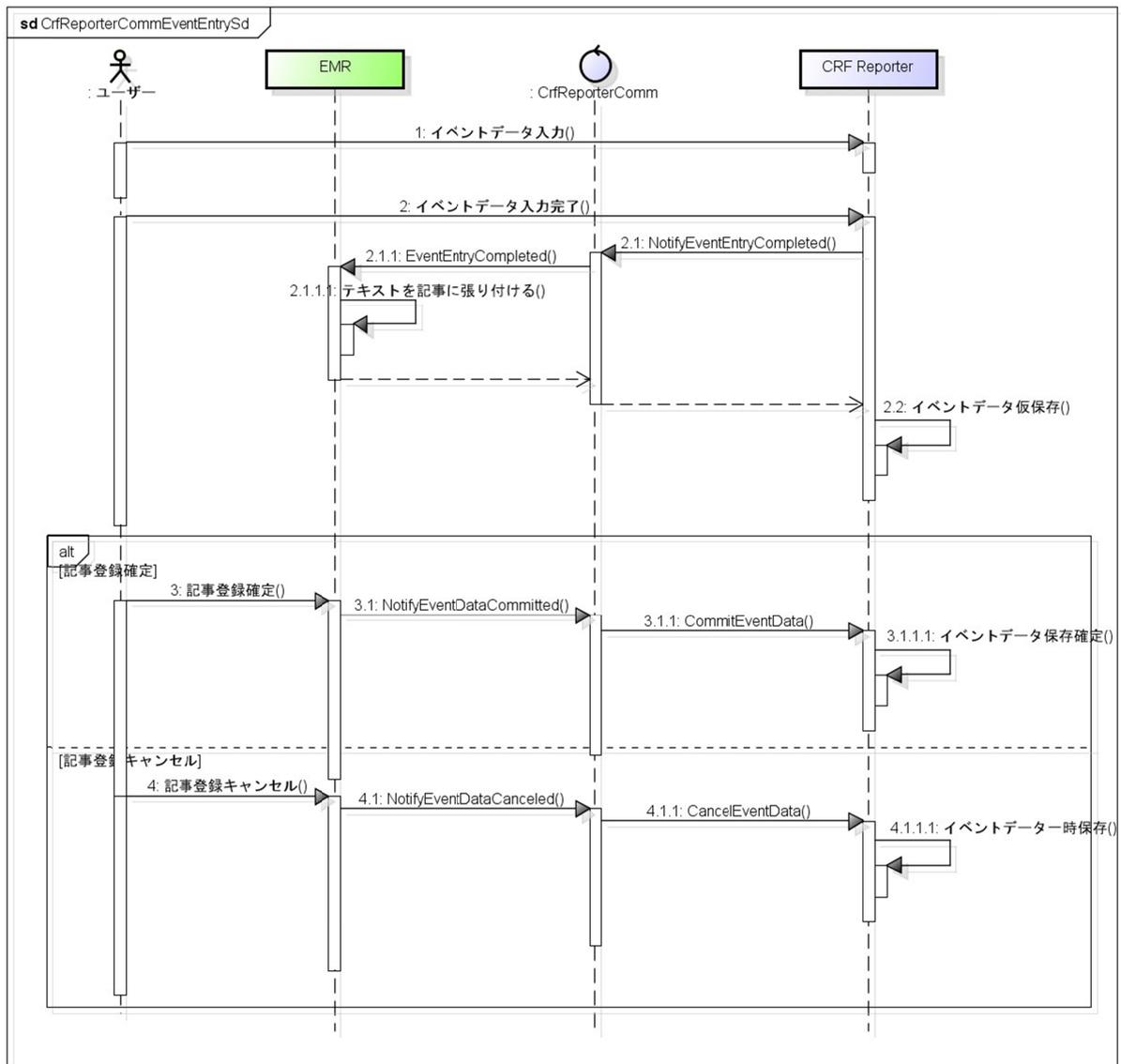
### 3.1.1 CRF Reporter アプリケーション起動シーケンス



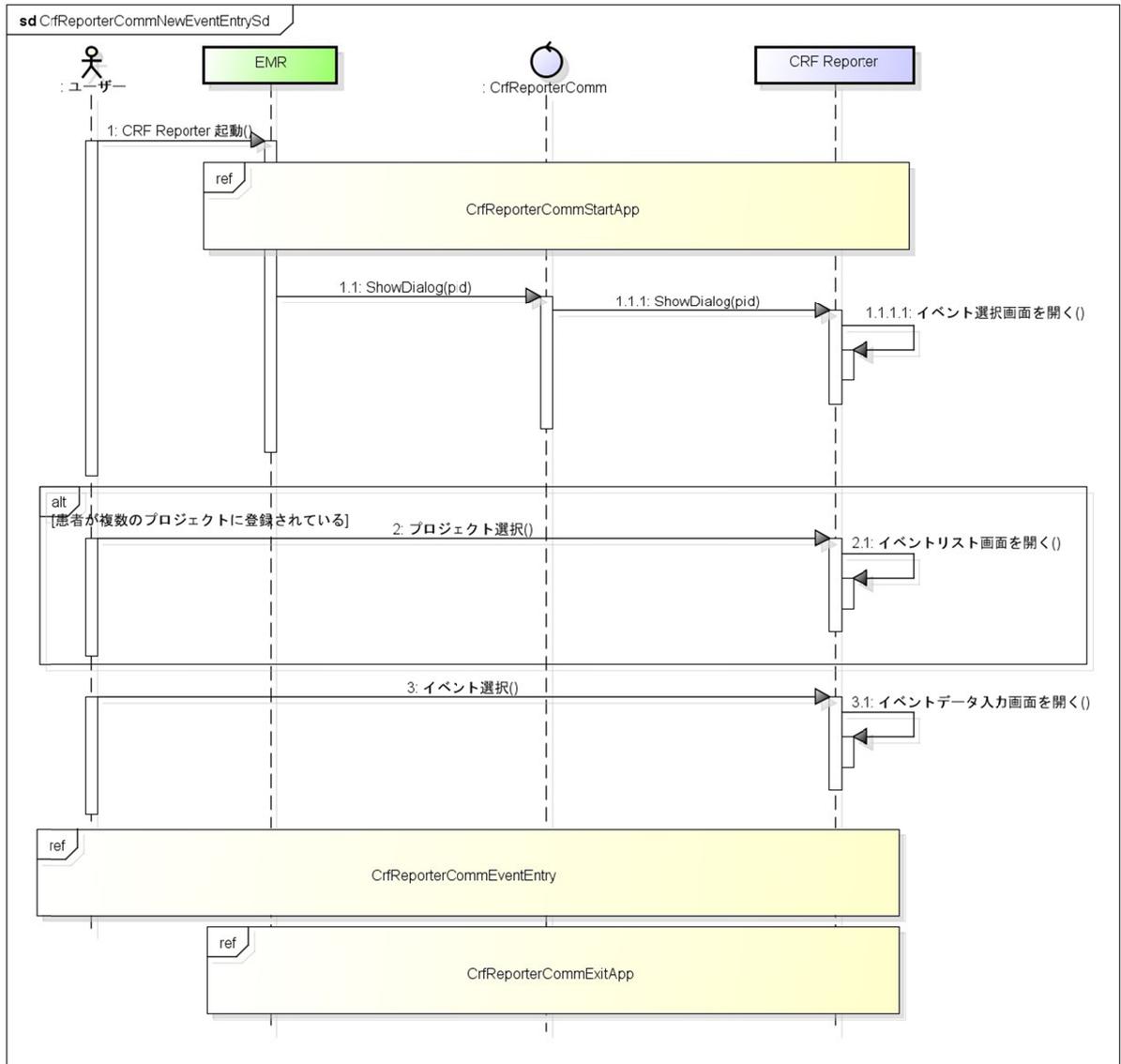
### 3.1.2 アプリケーション終了シーケンス



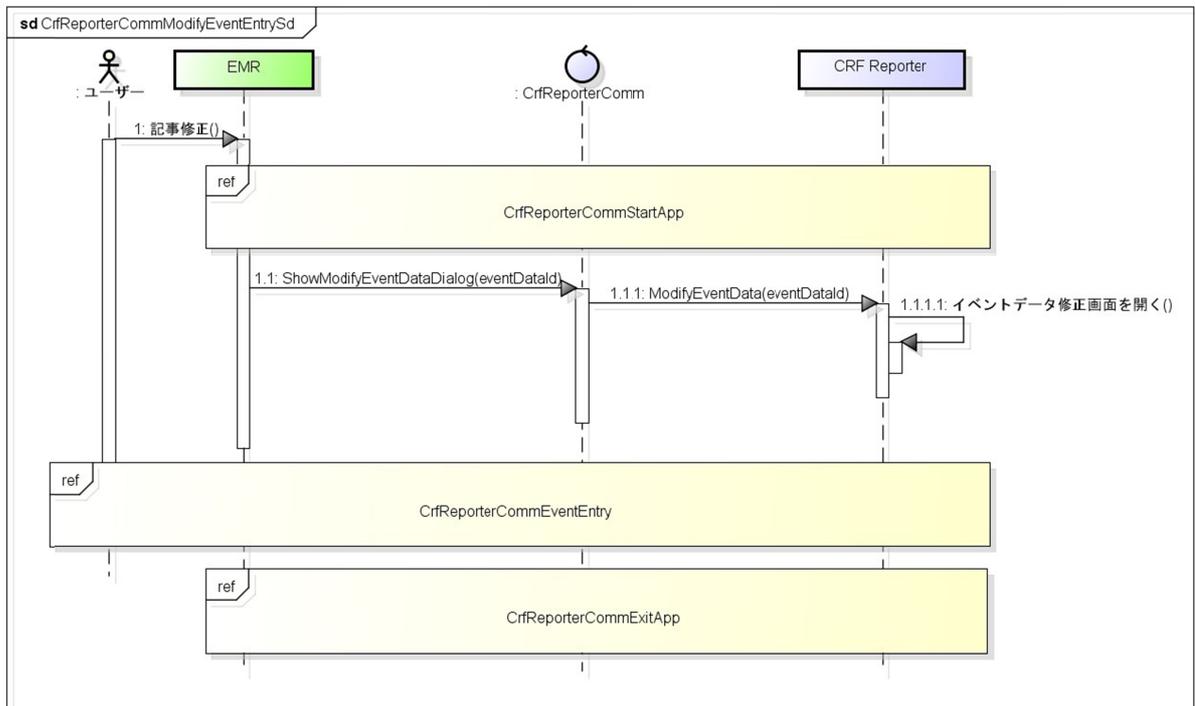
### 3.1.3 イベントデータ入力シーケンス



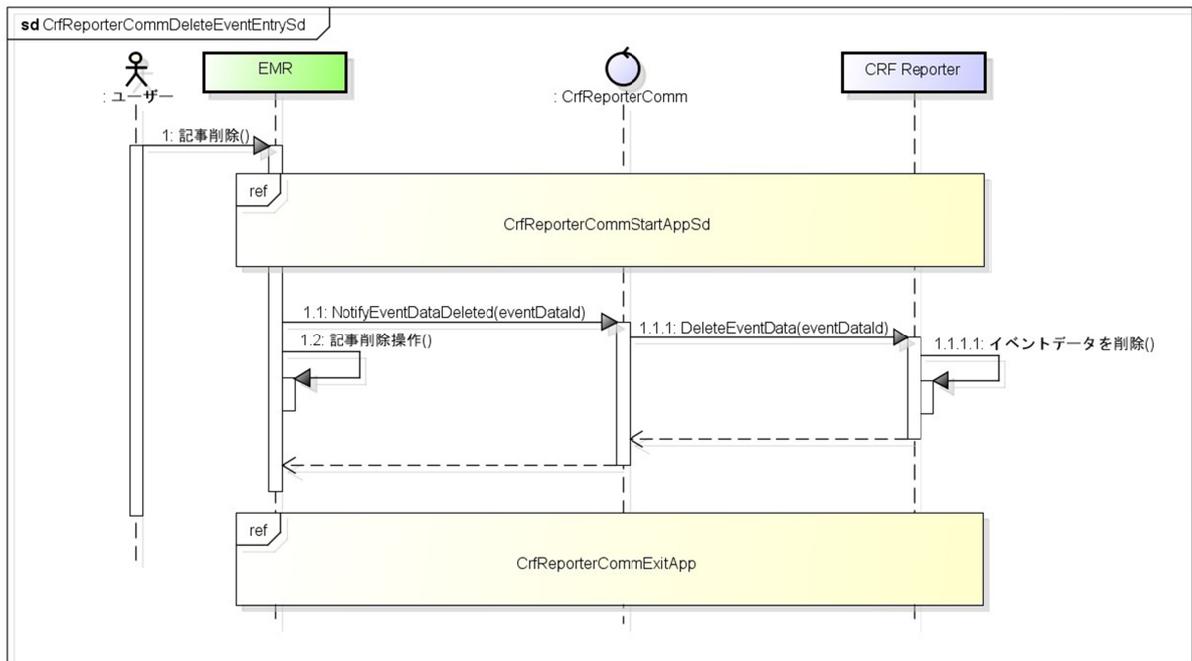
### 3.1.4 新規イベントデータ入力シーケンス



### 3.1.5 イベントデータ修正シーケンス



### 3.1.6 イベントデータ削除シーケンス



## 3.2 ICrfReporterComm インタフェース

CRF Reporter Communication DLL が実装するメソッド、イベントを定義します。

名前空間: OUH.Crit

構文

```
public interface ICrfReporterComm
```

ICrfReporterComm 型で公開されるメンバーは以下のとおりです。

メソッド

名前	説明
<i>NotifyEventDataCanceled</i>	<i>EventEntryCompleted</i> イベントで渡された CRF Reporter イベントデータの電子カルテ登録がキャンセルされたことを CRF Reporter に通知します。
<i>NotifyEventDataCommitted</i>	<i>EventEntryCompleted</i> イベントで渡された CRF Reporter イベントデータが電子カルテに登録されたことを CRF Reporter に通知します。
<i>NotifyEventDataDeleted</i>	電子カルテに登録された CRF Reporter イベントデータが削除されたことを CRF Reporter に通知します。
<i>ExitApp</i>	CRF Reporter を終了させます。
<i>ShowDialog</i>	ケース登録、イベントデータ入力のためのダイアログボックスを表示します。
<i>ShowModifyEventDataDialog</i>	指定されたイベントデータ識別子に紐づけられたイベント入力を修正するための CRF Reporter アプリケーション画面を表示します。
<i>StartApp()</i>	CRF Reporter アプリケーションを起動し、現在 Windows オペレーティングシステムにログインしているユーザー名で CRF Reporter アプリケーションにログインします。
<i>StartApp(IntPtr)</i>	CRF Reporter アプリケーションを起動し、トークンハンドルに紐づけられたユーザー名で CRF Reporter にログインします。
<i>StartApp(String, String)</i>	CRF Reporter アプリケーションを起動し、指定されたユーザー名で CRF Reporter アプリケーションにログインします。

	す。
--	----

## イベント

名前	説明
AppExiting	CRF Reporter が終了する直前に発生します。
EventEntryCompleted	CRF Reporter でのイベントデータ入力完了したときに発生します。

## 3.2.1 ICrfReporterComm メソッド

### ExitApp メソッド

CRF Reporter アプリケーションを終了させます。

#### 構文

```
public Boolean ExitApp(  
    Boolean forced  
)
```

パラメーター

*forced*

型: System.Boolean

強制的に CRF Reporter アプリケーションを停止させるかどうかを示す Boolean。

#### 戻り値

型: System.Boolean

CRF Reporter アプリケーションが正常に終了された場合には true 。 CRF Reporter アプリケーションが終了されなかった場合には false 。

#### 解説

ExitApp メソッドを呼び出すと、CRF Reporter アプリケーションに停止要求が送信されます。停止要求を受け、CRF Reporter アプリケーションはイベントデータ入力中などデータ喪失の危険性がない場合には停止し、戻り値として true が返されます。

CRF Reporter アプリケーションがイベントデータ入力中などデータ喪失の危険性があるときに ExitApp メソッドが呼び出されると、*forced* パラメータが false の場合には CRF Reporter アプリケーションは停止せず、戻り値として false が返されますが、*forced* パラメータが true の場合には、CRF Reporter アプリケーションで入力中のデータは一時保存され、CRF Reporter アプリケーションは強制終了し、戻り値として true が返されます。

## NotifyEventDataCanceled メソッド

*EventEntryCompleted* イベントで渡された CRF Reporter イベントデータの電子カルテ登録がキャンセルされたことを CRF Reporter に通知します。

### 構文

```
public void NotifyEventDataCanceled(  
  
    String eventDataId  
  
)
```

### パラメーター

*eventDataId*  
型:  
System.String

*EventEntryCompleted* イベントで渡された CRF Reporter イベントデータ識別子。

### 解説

*EventEntryCompleted* イベントで渡された CRF Reporter イベントデータを電子カルテに登録する手続きがキャンセルされた場合に、電子カルテアプリケーションはこのメソッドを呼び出します。

## NotifyEventDataCommitted メソッド

*EventEntryCompleted* イベントで渡された CRF Reporter イベントデータが電子カルテに登録されたことを CRF Reporter に通知します。

### 構文

```
public void
    NotifyEventDataCommitted( String
        eventDataId
    )
```

パラメーター

*eventId*  
型:  
System.St  
ring

*EventEntryCompleted* イベントで渡された CRF Reporter イベントデータ識別子。

解説

*EventEntryCompleted* イベントで渡された CRF Reporter イベントデータを電子カルテに登録する手続きが完了した時に、電子カルテアプリケーションはこのメソッドを呼び出します。

### NotifyEventDataDeleted メソッド

電子カルテに登録された CRF Reporter イベントデータが削除されたことを CRF Reporter に通知します。

構文

```
public void NotifyEventDataDeleted( String
    eventDataId
)
```

パラメーター

*eventId*  
型: System.String

*EventEntryCompleted* イベントで渡された CRF Reporter イベントデータ識別子。

解説

一旦登録された CRF Reporter のイベントデータが電子カルテから削除されたときに、電子カルテアプリケーションはこのメソッドを呼び出します。

## ShowDialog メソッド

ケース登録、イベントデータ入力などのための CRF Reporter アプリケーション画面を表示します。

### 構文

```
public void ShowDialog(  
    String pid  
)
```

パラメーター

*pid*

型: System.String

CRF Reporter アプリケーションでイベントデータ入力を実施する対象患者 ID。

### 解説

電子カルテの記事入力画面上に配置された CRF Reporter アプリケーション起動ボタンが押されたときに、電子カルテアプリケーションはこの ShowDialog メソッドを呼び出します。*pid* パラメータには、電子カルテの記事入力対象となっている患者 ID (電子カルテ内で設定されている患者識別子) を渡してください。

## ShowModifyEventDataDialog メソッド

指定されたイベントデータ識別子に紐づけられたイベント入力を修正するための CRF Reporter アプリケーション画面を表示します。

### 構文

```
public void ShowModifyEventDataDialog(  
    String eventDataId  
)
```

パラメーター

*eventDataId*

型: System.String

*EventEntryCompleted* イベントで渡された CRF Reporter イベントデータ識別子。

## StartApp メソッド

CRF Reporter アプリケーションを起動し、現在 Windows オペレーティングシステムにログインしているユーザー名で CRF Reporter アプリケーションにログインします。

構文

```
public void StartApp()
```

例外

例外の種類	条件
CrfReporterAuthFailedException	現在 Windows オペレーティングシステムにログインしているユーザー名が CRF Reporter アプリケーションに登録されていません。
CrfReporterRunningException	CRF Reporter が既に起動されています。

## StartApp メソッド (IntPtr)

CRF Reporter アプリケーションを起動し、トークンハンドルに紐づけられたユーザー名で CRF Reporter アプリケーションにログインします。

構文

```
public void StartApp(  
  
    IntPtr hToken  
  
)
```

パラメーター

*hToken*

型: System.IntPtr

ユーザーを表すトークンのハンドル。

例外

例外の種類	条件
-------	----

CrfReporterAuthFailedException	<i>hToken</i> パラメータで指定されたトークンハンドルに紐づけられたユーザー名が CRF Reporter アプリケーションに登録されていません。
CrfReporterRunningException	CRF Reporter が既に起動されています。

### StartApp メソッド (String, String)

CRF Reporter アプリケーションを起動し、指定されたユーザー名で CRF Reporter アプリケーションにログオンします。

#### 構文

```
public void StartApp(
    String userName,
    String password
)
```

#### パラメーター

*userName* 型:

System.String

ユーザー名を表す文字列。

*password* 型:

System.String

パスワードを表す文字列。

#### 例外

例外の種類	条件
CrfReporterAuthFailedException	<i>userName</i> パラメータで指定されたユーザー名が CRF Reporter アプリケーションに登録されていません。
CrfReporterRunningException	CRF Reporter が既に起動されています。

## 3.2.2 ICrfReporterComm イベント

### AppExiting イベント

CRF Reporter アプリケーションが終了する直前に発生します。

構文

```
public event EventHandler AppExiting
```

解説イベントは、CRF Reporter アプリケーションが終了する直前に発生します。

### EventEntryCompleted イベント

CRF Reporter でのイベントデータ入力完了したとき (イベント入力画面で [完了] ボタンが押されたとき) に発生します。イベントデータ入力が保留 (一時保存) されたとき (イベント入力画面で [保留] ボタンが押されたとき) には発生しません。

構文

```
public event EventHandler<EventEntryCompletedEventArgs> EventEntryCompleted
```

解説

CRF Reporter が電子カルテから起動された場合には、イベント入力画面で [完了] ボタンが押されても、イベントデータは入力確定状態にはなりません。CRF Reporter から出力されたイベントデータのテキストが張り付けられた電子カルテ記事が確定した際に CommitEventEntry メソッドが呼び出されてようやく、CRF Reporter のイベントデータの入力が確定します。イベントデータのテキストが張り付けられた電子カルテ記事がキャンセルされた際に CancelEventEntry メソッドが呼び出されると、CRF Reporter のイベントデータは一時保留状態に移行します。

## 3.3 EventEntryCompletedEventArgs クラス

*EventEntryCompleted* イベントにデータを供給します。

継承階層

System.Object

System.EventArgs

OUH.Crit.Comm.EventEntryCompletedEventArgs

名前空間: OUH.Crit.Comm

構文

```
public class EventEntryCompletedEventArgs : EventArgs
```

プロパティ

名前	説明
EventDataId	入力されたイベントデータの識別子を取得します。
Text	入力されたイベントデータを表す文字列を取得します。

### 3.3.1 EventEntryCompletedEventArgs プロパティ

#### EventDataId プロパティ

CRF Reporter に入力されたイベントデータの識別子を取得します。

構文

```
public String EventDataId { get; }
```

プロパティ値

型: System.String イベントデータ識別子。

#### Text プロパティ

CRF Reporter に入力されたイベントデータのを表す文字列を取得します。

構文

```
public String Text { get; }
```

プロパティ値

型: System.String イベントデータを表す文字列。

## 3.4 CrfReporterException クラス

ICrfReporterComm インタフェースのメソッドを実行したときに発生するエラーを表します。

継承階層

System.Object

System.Exception

OUH.Crit.Comm.CrfReporterException

名前空間: OUH.Crit.Comm

構文

```
public class CrfReporterException : Exception
```

## 3.5 CrfReporterAuthFailedException クラス

CRF Reporter アプリケーションのユーザー認証に失敗した場合にスローされる例外。

継承階層

System.Object

System.Exception

Ouh.Crit.Comm.CrfReporterException

Ouh.Crit.Comm.CrfReporterAuthFailedException

名前空間: Ouh.Crit.Comm

構文

```
[SerializableAttribute] public class CrfReporterAuthFailedException :  
CrfReporterException
```

## 3.6 CrfReporterRunningException クラス

CRF Reporter アプリケーションが起動中に、CRF Reporter アプリケーション起動メソッドが呼び出された場合にスローされる例外。

継承階層

System.Object

System.Exception

OUH.Crit.Comm.CrfReporterException

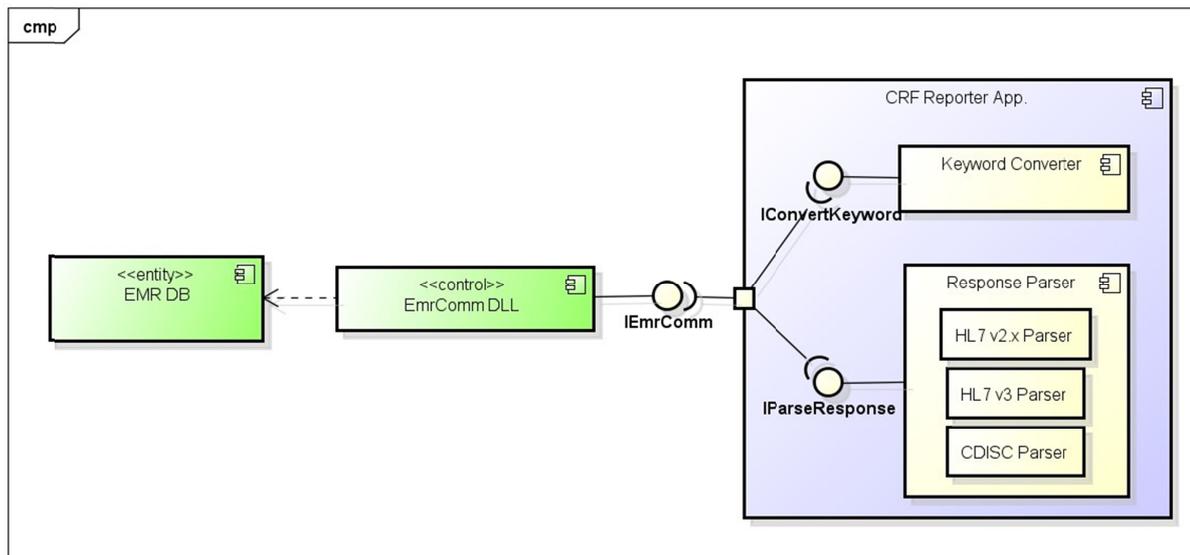
OUH.Crit.Comm.CrfReporterRunningException

名前空間: OUH.Crit.Comm

構文

```
[SerializableAttribute] public class CrfReporterRunningException :  
CrfReporterException
```

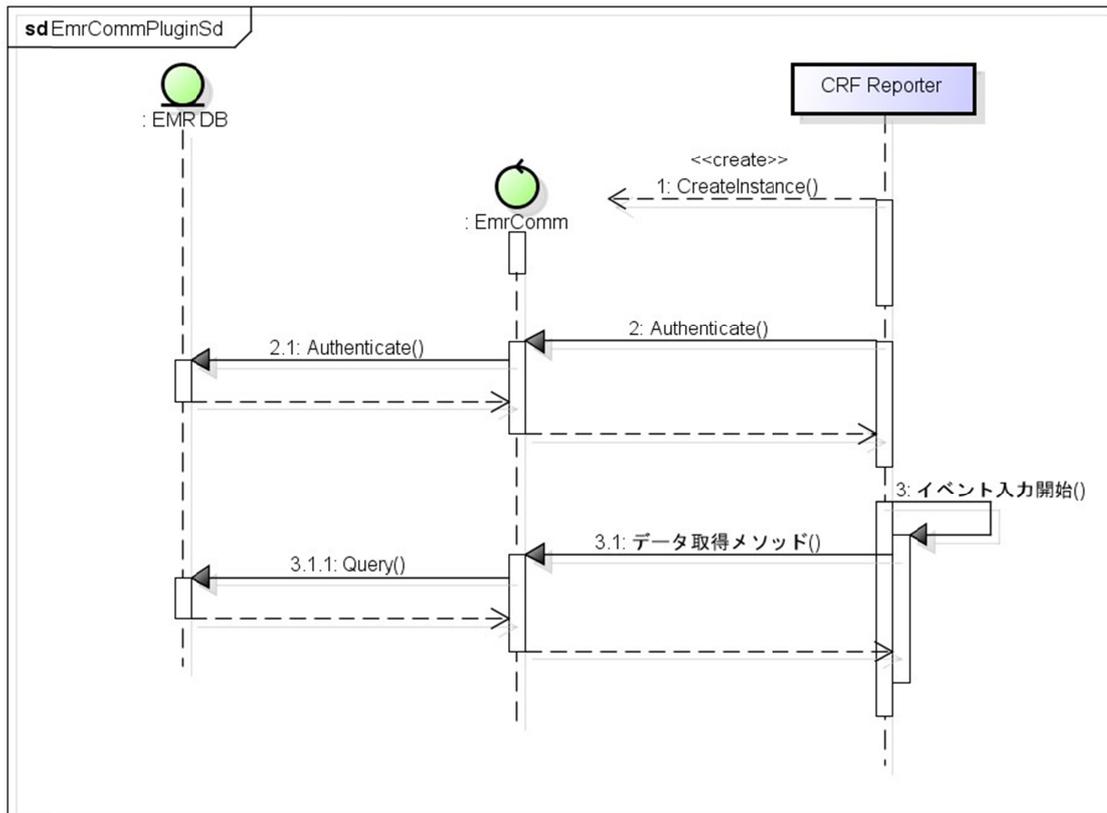
# 第4章 EmrComm プラグイン DLL インタフェース仕様



電子カルテと連携するためのプラグインモジュールを DLL の形で電子カルテベンダーから提供してもらうことにより、CRF Reporter が電子カルテと連携することを実現します。

電子カルテ連携プラグインが実装すべきインタフェースを、以下に定義します。

## 4.1 IEmrComm インタフェース



電子カルテ内のデータを取得するためのメソッドを定義します。

名前空間: OUH.Crit.Comm アセンブリ: IEmrComm.dll

構文

```
public interface IEmrComm
```

IEmrComm 型で公開されるメンバーは以下のとおりです。

メソッド

名前	説明
<i>Authenticate</i>	指定されたユーザー名とパスワードを認証します。
<i>BodyMeasurement</i>	指定された患者の身長および体重の測定結果を返します。
<i>Hospitalization</i>	指定された患者の入退院歴(転科を含む)を返します。
<i>Laboratory</i>	指定された患者の検体検査結果を返します。

<i>Operation</i>	指定された患者の手術歴を返します。
<i>PatientProfile</i>	指定された患者の患者情報を返します。
<i>Prescription</i>	指定された患者の処方データを返します。

## 4.1.1 Authenticate メソッド

指定されたユーザー名とパスワードを認証します。

名前空間: OUH.Crit.Comm アセンブリ: IEmrComm.dll

構文

```
public Object Authenticate(
    String userName,
    String password
)
```

パラメーター

*userName* 型:  
System.String

電子カルテシステムに登録されているユーザー名。

*password* 型:  
System.String

電子カルテシステムに登録されているユーザー名に対応するパスワード。

戻り値

型: System.Object

ユーザー名とパスワードが有効な場合は認証情報 (アクセストークン) を表すオブジェクト。それ以外の場合は null。

## 4.1.2 PatientProfile メソッド

指定された患者の患者情報を返します。

## 構文

```
public Byte[] PatientProfile(
```

```
    Object token,
```

```
    String pid
```

```
)
```

パラメーター

*token*

型: System.Object

[Authenticate](#) メソッドで返された認証情報 (アクセストークン) を表すオブジェクト。

*pid*

型: System.String

患者 ID。

## 戻り値

型: System.Byte[]

患者情報を表すデータを格納しているバイト配列。解説

返される患者情報には、以下の情報が含まれます。

- ・ 氏名
- ・ 性別
- ・ 生年月日

戻り値のデータは、以下の書式のいずれかに準拠するものとします。

- ・ CRIT 標準書式
- ・ HL7 v2.x
- ・ HL7 v3
- ・ HL7 FHIR

CRIT 標準書式に関する詳細は、[PatientProfile 応答メッセージ](#)を参照してください。

## 4.1.3 BodyMeasurement メソッド

指定された患者の身長および体重の測定結果を返します。

構文

```
public Byte[] BodyMeasurement(  
    Object token,  
    String pid  
    DateTime fromDate,  
    DateTime toDate,  
    Nullable<Int32> maxCount  
)
```

パラメーター

*token* 型:

System.Object

*Authenticate* メソッドで返された認証情報 (アクセストークン) を表すオブジェクト。

*pid* 型:

System.String

患者 ID。

*fromDate*

型:

DateTime

取得する測定結果の期間の開始日。

*toDate* 型:

DateTime

取得する測定結果の期間の終了日。

*maxCount* 型:

Nullable<Int  
32>

戻り値データに含まれる測定結果の最大データ個数。

戻り値

型: System.Byte[]

身長、体重の測定結果を格納しているバイト配列。

解説戻り値のデータには、以下の情報が含まれます。

- ・ 身長
- ・ 身長測定日
- ・ 体重
- ・ 体重測定日

*fromDate* パラメータおよび *toDate* パラメータに設定された日に測定された測定結果は戻り値のデータに含まれます。

戻り値のデータは、以下の書式のいずれかに準拠するものとします。

- ・ CRIT 標準書式
- ・ HL7 v2.x
- ・ HL7 v3
- ・ HL7 FHIR

CRIT 標準書式に関する詳細は、[BodyMeasurement 応答メッセージ](#)を参照してください。

## 4.1.4 Hospitalization メソッド

指定された患者の入退院歴(転科を含む)を返します。

構文

```
public Byte[] Hospitalization(  
    Object token,  
    String pid,  
    Nullable<DateTime> fromDate,
```

Nullable<DateTime> toDate

)

パラメーター

*token* 型:

System.Object

*Authenticate* メソッドで返された認証情報 (アクセストークン) を表すオブジェクト。

*pid* 型:

System.String

患者 ID。

*fromDate* 型:

Nullable<DateTime>

指定された日付以降 (指定日付を含む) の入院/退院/転科イベントを返してください。null の場合には、開始日制限はかかりません。

*toDate*

型: Nullable<DateTime>

指定された日付以前 (指定日付を含む) の入院/退院/転科イベントを返してください。null の場合には、終了日制限はかかりません。

戻り値

型: System.Byte[]

入退院歴を表すデータを格納しているバイト配列。解説

返される入退院歴には、以下の情報が含まれます。

- ・ 入院、退院、転科区分
- ・ 診療科コード
- ・ 診療科名
- ・ 入退院、転科日

*fromDate* パラメータおよび *toDate* パラメータに *DateTime* 値が設定された場合、その日に発生した入退院および転科イベントは戻り値のデータに含まれます。また、*fromDate* パラメータに *null* が指定された場合、開始日付制限はかかりません。同様に、*toDate* パラメータに *null* が指定された場合、終了日付制限はかかりません。ですので、例えば *fromDate* パラメータと *toDate* パラメータの両者に *null* が指定された場合、電子カルテに記録されているすべての入退院歴が返されます。

返される入院、退院、転科歴は実施済みのイベントのみであり、将来の予定情報は含まれません。

戻り値のデータは、以下の書式のいずれかに準拠するものとします。

- ・ CRIT 標準書式
- ・ HL7 v2.x
- ・ HL7 v3
- ・ HL7 FHIR

CRIT 標準書式に関する詳細は、[Hospitalization 応答メッセージ](#)を参照してください。

## 4.1.5 Operation メソッド

指定された患者の手術歴を返します。

構文

```
public Byte[] Operation(  
    Object token,  
    String pid,  
    Nullable<DateTime> fromDate,  
    Nullable<DateTime> toDate  
)
```

パラメーター

*token* 型:  
System.Object

[Authenticate](#) メソッドで返された認証情報 (アクセストークン) を表すオブジェクト。

*pid* 型:

System.String

患者 ID。

*fromDate* 型:

Nullable<DateTime>

取得する手術歴の期間の開始日。

*toDate* 型:

Nullable<DateTime>

取得する手術歴の期間の終了日。

戻り値

型: System.Byte[]

手術歴を表すデータを格納しているバイト配列。

解説返される手術歴には、以下の情報が含まれます。

- ・ 手術日 (入室日時)
- ・ 診療科コード
- ・ 診療科名
- ・ 術式

*fromDate* パラメータおよび *toDate* パラメータに DateTime 値が設定された場合、その日に発生した手術イベントは戻り値のデータに含まれます。また、*fromDate* パラメータに null が指定された場合、開始日付制限はかかりません。 *toDate* パラメータに null が指定された場合、終了日付制限はかかりません。 ですので、例えば *fromDate* パラメータと *toDate* パラメータの両者に null が指定された場合、電子カルテに記録されているすべての手術歴が返されます。

返される手術歴は実施済みのもののみであり、未実施の予定情報は返されません。

術式として、電子カルテ内に実施術式と請求用術式の 2 種類が記録されている場合、実施術式が返されます。

戻り値のデータは、以下の書式のいずれかに準拠するものとします。

- ・ CRIT 標準書式

- ・ HL7 v2.x
- ・ HL7 v3
- ・ HL7 FHIR

CRIT 標準書式に関する詳細は、[Operation 応答メッセージ](#)を参照してください。

## 4.1.6 Laboratory メソッド

指定された患者の検体検査結果を返します。

構文

```
public Byte[] Laboratory(  
    Object token,  
    String pid,  
    String[] keywords,  
    DateTime fromDate,  
    DateTime toDate,  
    Nullable<Int32> maxCount  
)
```

パラメーター

*token*

型: System.Object

[Authenticate](#) メソッドで返された認証情報 (アクセストークン) を表すオブジェクト。

*pid*

型: System.String

患者 ID。

*keywords*

型: System.String[]

取得する検査項目に対応するキーワードのリスト。

*fromDate*

型: DateTime

取得する検査結果の期間の開始日。

*toDate*

型: DateTime

取得する検査結果の期間の終了日。

*maxCount*

型: Nullable<Int32>

戻り値データに含まれる検査値の最大データ個数。

戻り値

型: System.Byte[]

検体検査結果を表すデータを格納しているバイト配列。解説

返される検体検査結果には、以下の情報が含まれます。

- ・(各施設で定義された)検査項目コード
- ・検査日時(検体採取日時)
- ・検査値
- ・検査値単位
- ・正常値上限
- ・正常値下限

*fromDate* パラメータおよび *toDate* パラメータに設定された日に採取された検体の検査結果は戻り値のデータに含まれます。

*maxCount* パラメータが指定された場合、新しい検査日時順に指定個数以内の検査結果が返されます。 *maxCount* パラメータに null が指定された場合、戻り値データの個数制限はかかりません。

検査結果が未登録の検査情報(検査中の検査情報)は返されません。

戻り値のデータは、以下の書式のいずれかに準拠するものとします。

- ・ CRIT 標準書式

- ・ HL7 v2.x
- ・ HL7 v3
- ・ HL7 FHIR
- ・ CDISC Laboratory Data Model (LAB)

CRIT 標準書式に関する詳細は、[Laboratory 応答メッセージ](#)を参照してください。

## 4.1.7 Prescription メソッド

指定された患者の処方歴を返します。

構文

```
public Byte[] Prescription(  
    Object token,  
    String pid,  
    DateTime fromDate,  
    DateTime toDate  
)  
パラメーター
```

*token* 型:  
System.Object

*Authenticate* メソッドで返された認証情報 (アクセストークン) を表すオブジェクト。

*pid* 型:  
System.String

患者 ID。

*fromDate*  
型:  
DateTime

取得する処方歴の期間の開始日。

*toDate* 型:  
DateTime

取得する処方歴の期間の終了日。

## 戻り値

型: System.Byte[]

処方歴を表すデータを格納しているバイト配列。

## 解説

*fromData* パラメータに指定された日付 (指定日を含む) から *toDate* パラメータに指定された日付 (指定日を含む) の期間内に処方が開始され、かつ処方が終了する処方データが返されます。

返される処方データには、以下の情報が含まれます。

・(各施設で定義された)薬剤コード

- ・ 薬剤名
- ・ 処方開始日
- ・ 処方終了日
- ・ 分量
- ・ 用法
- ・ 用法種
- ・ 用量
- ・ 経路戻り値のデータは、以下の書式のいずれかに準拠するものとします。
- ・ CRIT 標準書式
- ・ HL7 v2.x
- ・ HL7 v3
- ・ HL7 FHIR

CRIT 標準書式に関する詳細は、[Prescription 応答メッセージ](#)を参照してください。

## 4.2 データ取得メソッドが返す応答メッセージ

データ取得メソッド (PatientProfile, BodyMeasurement, Hospitalization, Operation, Laboratory, Prescription) が返す応答メッセージは、本プロジェクトで定める CRIT 標準書式、もしくは以下の国際標準に従うものとします。

- ・ HL7 Version 2.x Messaging Standard
- ・ HL7 Version 3 Clinical Document Architecture (CDA)
- ・ HL7 FHIR (Fast Healthcare Interoperability Resources) Specification
- ・ CDISC Laboratory Data Model (LAB)

本プロジェクトで定める CRIT 標準書式に関しては、次章の [CRIT 応答メッセージ XML 仕様](#) を参照してください。

# 第5章 CRIT 応答メッセージ XML 仕様

## 5.1 概要

### 5.1.1 シンタックス記法

XML 要素のシンタックスは、以下の形式で記述します。

要素名

子要素:

子要素のグループ

属性:

名称	型	詳細
...		

子要素のグループは、括弧で囲まれたひとつ以上の要素名から成り、コンマ (,) もしくはパーティカルバー (|) で区切られます。コンマは要素が順番に現れることを表します。パーティカルバーはその中のひとつが現れることを表します。要素名の後ろには ? (ゼロ回もしくは 1 回出現する)、\* (ゼロ回以上出現する)、+ (1 回以上出現する) の記号が付く場合があります。

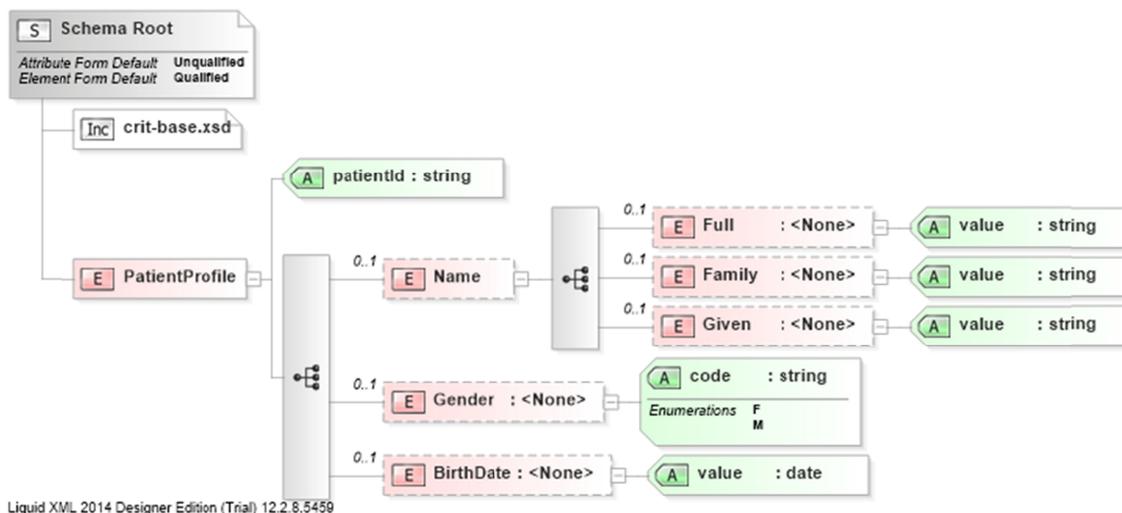
### 5.1.2 データフォーマット

XML 属性値は文字列で表されます。この文字列の書式はデータフォーマットで規定されま  
す。

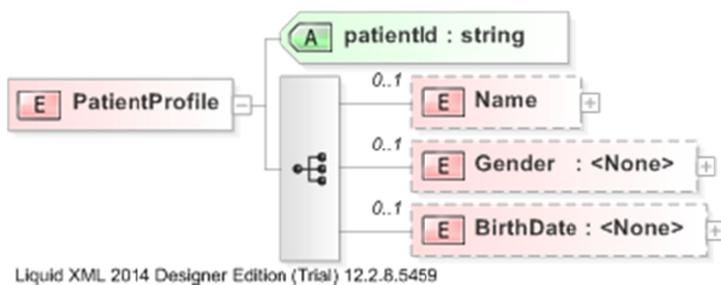
フォーマット名	スキーマ型	説明
string	xs:string	ユニコード文字列。
date	xs:date	日付。
dateTime	xs:date, xs:dateTime	日付もしくは日時。
	のユニオン	
decimal	xs:decimal	有理数。

## 5.2 PatientProfile 応答メッセージ

### 5.2.1 XML 要素



#### PatientProfile 要素



子要素:

(Name?, Gender?, BirthDate?)

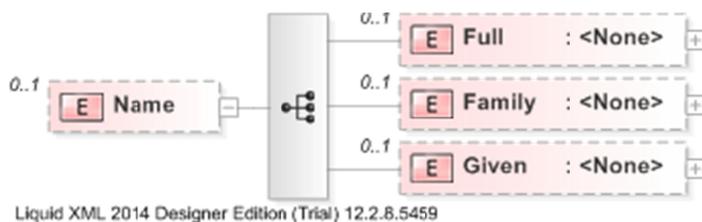
属性:

名称	型	詳細
patientId	string	患者 ID

PatientProfile 要素が PatientProfile 応答メッセージのルート要素です。

patientId 属性値には、PatientProfile メソッドの pid 引数として渡された値が設定されます。

Name 要素



子要素:

(Full?, Family?, Given?)

属性:

なし

Name 要素には、患者の名前情報が入ります。

### Full 要素



子要素:

なし

属性:

名称	型	詳細
value	string	姓名

value 属性値には、姓名のフルネーム値が入ります。患者基本データベースに姓と名が独立して管理されている場合には、姓と名を連結して設定してください。その連結時に空白文字を区切り記号として挿入しても構いませんし、また区切り記号なしに連結しても構いません。

### Family 要素



子要素:

なし

属性:

名称	型	詳細
value	string	姓

value 属性値には、姓が入ります。患者基本データベースに姓と名が独立して管理されていない場合には、この要素を記述しないでください。

### Given 要素



子要素:

なし

属性:

名称	型	詳細
value	string	名

value 属性値には、名が入ります。患者基本データベースに姓と名が独立して管理されていない場合には、この要素を記述しないでください。

### Gender 要素



子要素:

なし

属性:

名称	型	詳細
code	string	性別 (男性: "M"、女性: "F")

code 属性値には、性別を表す文字列が設定されます。男性の場合には "M" が、女性の場合には "F" が設定されます。

## BirthDate 要素



子要素:

なし

属性:

名称	型	詳細
value	date	生年月日

value 属性値には、ISO 8601 date format に準拠した生年月日を表す文字列が設定されます。

## 5.2.2 PatientProfile 応答メッセージ例

```
<?xml version="1.0" encoding="UTF-8"?>
<PatientProfile patientId="12345" xmlns="http://med.osaka-u.ac.jp/crit">
  <Name>
    <Full value="大阪太郎"/>
    <Family value="大阪"/>
    <Given value="太郎"/>
  </Name>
  <Gender code="M"/>
  <BirthDate value="1974-12-25"/>
</PatientProfile>
```

## 5.2.3 PatientProfile 応答メッセージスキーマ

```
<?xml version="1.0" encoding="utf-8" ?>
<xs:schema xmlns="http://med.osaka-u.ac.jp/crit"
  attributeFormDefault="unqualified"
  elementFormDefault="qualified"
  targetNamespace="http://med.osaka-u.ac.jp/crit"
  xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:include schemaLocation="crit-base.xsd" />
  <xs:element name="PatientProfile">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="Name" minOccurs="0">
```

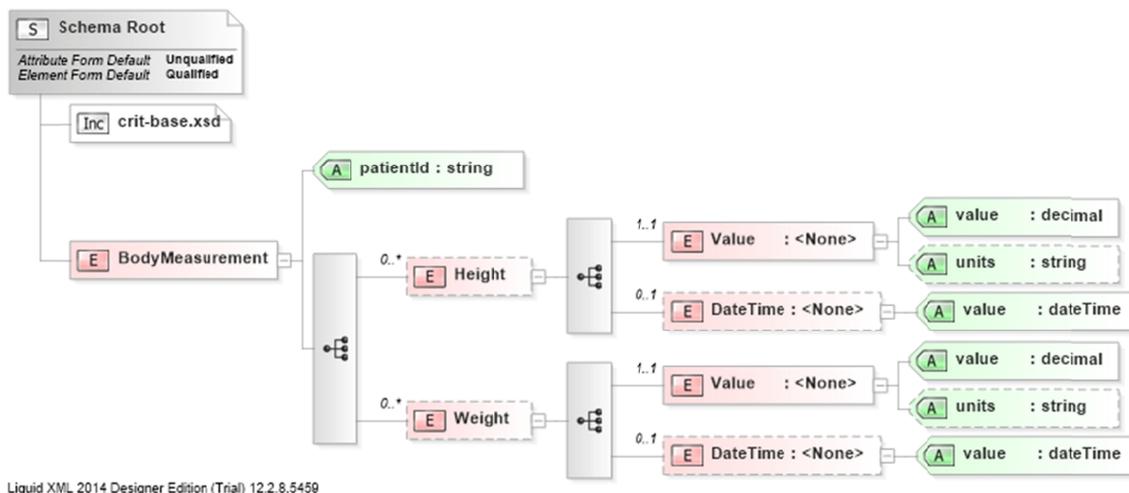
```

<xs:complexType>
  <xs:sequence>
    <xs:element name="Full" minOccurs="0">
      <xs:complexType>
        <xs:attribute name="value" type="xs:string"
          use="required" />
      </xs:complexType>
    </xs:element>
    <xs:element name="Family"
      minOccurs="0">
      <xs:complexType>
        <xs:attribute name="value" type="xs:string" use="required" />
      </xs:complexType>
    </xs:element>
    <xs:element name="Given"
      minOccurs="0">
      <xs:complexType>
        <xs:attribute name="value" type="xs:string" use="required" />
      </xs:complexType>
    </xs:element>
  </xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="Gender"
  minOccurs="0">
  <xs:complexType>
    <xs:attribute name="code"
      use="required">
      <xs:simpleType>
        <xs:restriction base="xs:string">
          <xs:enumeration value="F" />
          <xs:enumeration value="M" />
        </xs:restriction>
      </xs:simpleType>
    </xs:attribute>
  </xs:complexType>
</xs:element>
<xs:element name="BirthDate"
  minOccurs="0">
  <xs:complexType>
    <xs:attribute name="value" type="xs:date" use="required" />
  </xs:complexType>
</xs:element>
</xs:sequence>
<xs:attribute name="patientId" type="xs:string" use="required" />
</xs:complexType>
</xs:element>
</xs:schema>

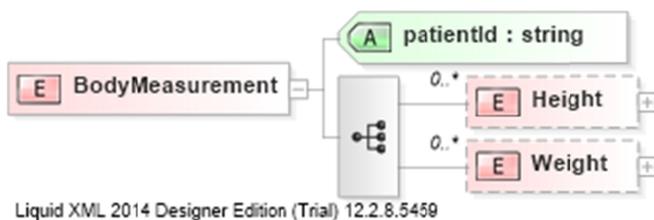
```

## 5.3 BodyMeasurement 応答メッセージ

### 5.3.1 XML 要素



#### BodyMeasurement 要素



子要素:

*(Height\*, Weight\*)*

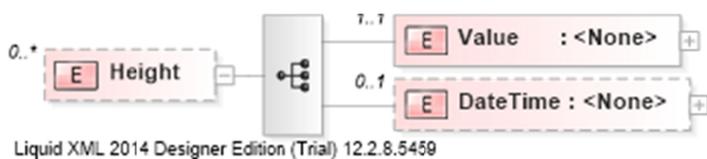
属性:

名称	型	詳細
patientId	string	患者 ID

BodyMeasurement 要素が BodyMeasurement 応答メッセージのルート要素です。

patientId 属性値には、BodyMeasurement メソッドの pid 引数として渡された値が設定されます。

## Height 要素



子要素:

(*Value, DateTime?*)

属性:

なし

Height 要素には、患者の身長情報が入ります。

## Height/Value 要素



子要素:

なし

属性:

名称	型	詳細
value	decimal	身長測定値
units	string	<i>(optional)</i> 身長測定値単位

Value 要素には、患者の身長測定値が記述されます。

value 属性値 (身長測定値) は “cm” 単位の数値を設定することを基本とします。

units 属性値は、国際単位系 (SI) の表記法に従うものとします。

## Height/DateTime 要素



子要素:

なし

属性:

名称	型	詳細
value	dateTime	身長測定日

DateTime 要素には、患者の身長測定日時が記述されます。value 属性値には、日時を表す ISO 8601 date format に準拠した文字列が設定されます。

### Weight 要素



子要素:

(*Value, DateTime?*)

属性:

なし

Weight 要素には、患者の体重情報が記述されます。

### Weight/Value 要素



子要素:

なし

属性:

名称	型	詳細
value	decimal	体重測定値
units	string	<i>(optional)</i> 体重測定値単位

Value 要素には、患者の体重測定値が記述されます。

value 属性値 (身長測定値) は “kg” 単位の数値を設定することを基本としますが、”g” 単位の数値も可とします。

units 属性値は、国際単位系 (SI) の表記法に従うものとします。

### Weight/DateTime 要素



子要素:

なし

属性:

名称	型	詳細
value	dateTime	体重測定日

DateTime 要素には、患者の体重測定日時が記述されます。value 属性値には、日を表す ISO 8601 date format に準拠した文字列が設定されます。

## 5.3.2 BodyMeasurement 応答メッセージ例

```
<BodyMeasurement patientId="12345" xmlns="http://med.osaka-u.ac.jp/crit">  
  <Height>  
    <Value value="172" units="cm" />  
    <DateTime value="2014-08-16" />  
  </Height>  
  <Weight>  
    <Value value="65.3" units="kg" />  
    <DateTime value="2014-08-16" />  
  </Weight>  
  <Weight>  
    <Value value="65.8" units="kg" />  
    <DateTime value="2014-08-30T11:57:00" />  
  </Weight>  
</BodyMeasurement>
```

## 5.3.3 BodyMeasurement 応答メッセージスキーマ

```
<?xml version="1.0" encoding="utf-8" ?>  
<xs:schema xmlns="http://med.osaka-u.ac.jp/crit"
```

```

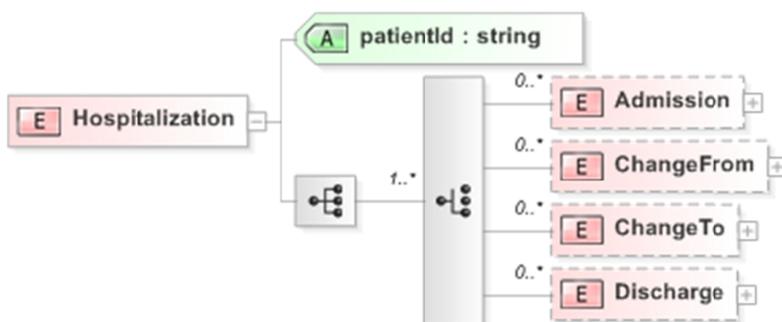
        attributeFormDefault="unqualified"
        elementFormDefault="qualified"
        targetNamespace="http://med.osaka-u.ac.jp/crit"
        xmlns:xs="http://www.w3.org/2001/XMLSchema">
<xs:include schemaLocation="crit-base.xsd" />
<xs:element name="BodyMeasurement">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="Height" minOccurs="0" maxOccurs="unbounded">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="Value" minOccurs="1">
              <xs:complexType>
                <xs:attribute name="value" type="xs:decimal" use="required" />
                <xs:attribute name="units" type="xs:string" use="optional" />
              </xs:complexType>
            </xs:element>
            <xs:element name="DateTime" minOccurs="0">
              <xs:complexType>
                <xs:attribute name="value" type="dateTime" use="required" />
              </xs:complexType>
            </xs:element>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:element name="Weight" minOccurs="0" maxOccurs="unbounded">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="Value" minOccurs="1">
              <xs:complexType>
                <xs:attribute name="value" type="xs:decimal" use="required" />
                <xs:attribute name="units" type="xs:string" use="optional" />
              </xs:complexType>
            </xs:element>
            <xs:element name="DateTime" minOccurs="0">
              <xs:complexType>
                <xs:attribute name="value" type="dateTime" use="required" />
              </xs:complexType>
            </xs:element>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
</xs:element>

```

```
    </xs:element>
  </xs:sequence>
  <xs:attribute name="patientId" type="xs:string" use="required" />
</xs:complexType>
</xs:element>
</xs:schema>
```



## Hospitalization 要素



Liquid XML 2014 Designer Edition (Trial) 12.2.8.5459

子要素:

*(Admission?, ChangeFrom?, ChangeTo?, Discharge?)*

属性:

名称	型	詳細
patientId	string	患者 ID

Hospitalization 要素が Hospitalization 応答メッセージのルート要素です。

patientId 属性値には、Hospitalization メソッドの pid 引数として渡された値が設定されます。

## Admission 要素



Liquid XML 2014 Designer Edition (Trial) 12.2.8.5459

子要素:

*(DateTime, Department)*

属性:

なし

Admission 要素には、患者の入院情報が記述されます。

## ChangeFrom 要素



子要素:

*(DateTime, Department)*

属性:

なし

ChangeFrom 要素には、患者の転出情報が記述されます。

## ChangeTo 要素



子要素:

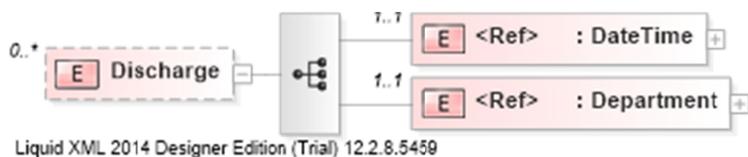
*(DateTime, Department)*

属性:

なし

ChangeTo 要素には、患者の転入情報が記述されます。

## Discharge 要素



子要素:

*(DateTime, Department)*

属性:

なし

Discharge 要素には、患者の退院情報が記述されます。

### DateTime 要素



子要素:

なし

属性:

名称	型	詳細
value	dateTime	入院日時/転出日時/転入日時/退院日時

DateTime 要素には、入院日時/転出日時/転入日時/退院日が時記述されます。value 属性値には、ISO 8601 date format に準拠した入院日時を表す文字列が設定されます。

### Department 要素



子要素: なし

属性:

名称	型	詳細
code	string	(optional) 診療科コード
text	string	診療科名

Department 要素には、入院/転出/転入/退院時の診療科情報が記述されます。code 属性値には、各診療施設で定義された診療科コードを設定してください。

## 5.4.2 Hospitalization 応答メッセージ例

```
<?xml version="1.0" encoding="utf-8" ?>
<Hospitalization patientId="12345" xmlns="http://med.osaka-u.ac.jp/crit">
  <Admission>
    <DateTime value="2011-06-26"/>
    <Department code="10" text="外科"/>
  </Admission>
  <ChangeFrom>
    <DateTime value="2011-07-11"/>
    <Department code="10" text="外科"/>
  </ChangeFrom>
  <ChangeTo>
    <DateTime value="2011-07-11"/>
    <Department code="11" text="整形外科"/>
  </ChangeTo>
  <Discharge>
    <DateTime value="2011-07-22"/>
    <Department code="11" text="整形外科"/>
  </Discharge>
  <Admission>
    <DateTime value="2012-03-13"/>
    <Department code="10" text="外科"/>
  </Admission>
  <Discharge>
    <DateTime value="2011-03-24"/>
    <Department code="10" text="外科"/>
  </Discharge>
</Hospitalization>
```

## 5.4.3 Hospitalization 応答メッセージスキーマ

```
<?xml version="1.0" encoding="utf-8" ?>
<xs:schema xmlns="http://med.osaka-u.ac.jp/crit"
  attributeFormDefault="unqualified" elementFormDefault="qualified"
  targetNamespace="http://med.osaka-u.ac.jp/crit"
  xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:include schemaLocation="crit-base.xsd" />
  <xs:element name="Hospitalization">
    <xs:complexType>
      <xs:sequence>
        <xs:choice maxOccurs="unbounded">
          <xs:element name="Admission" minOccurs="0" maxOccurs="unbounded">
            <xs:complexType>
              <xs:sequence>
```

```

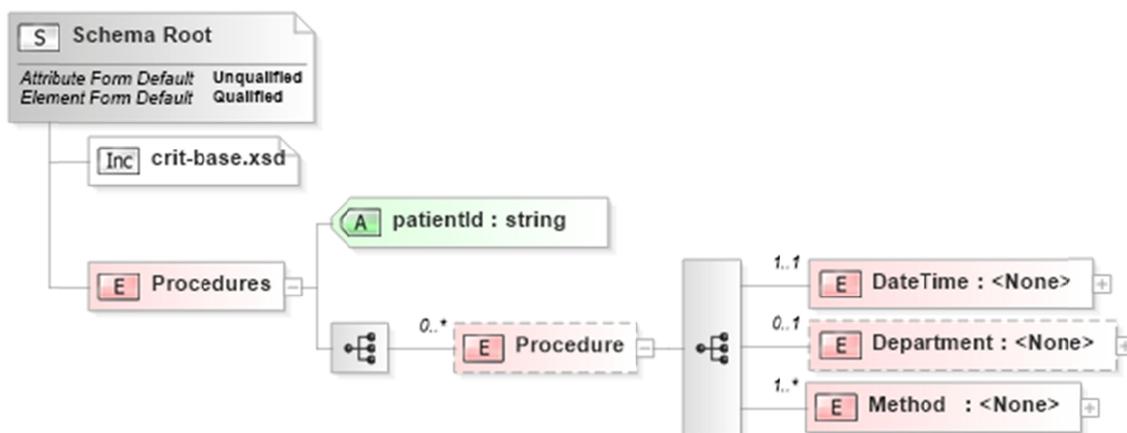
        <xs:element xmlns:q1="http://med.osaka-u.ac.jp/crit"
            ref="DateTime" minOccurs="1" />
        <xs:element xmlns:q2="http://med.osaka-u.ac.jp/crit"
            ref="Department" minOccurs="1" />
    </xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="ChangeFrom" minOccurs="0" maxOccurs="unbounded">
    <xs:complexType>
        <xs:sequence>
            <xs:element xmlns:q1="http://med.osaka-u.ac.jp/crit"
                ref="DateTime" minOccurs="1" />
            <xs:element xmlns:q2="http://med.osaka-u.ac.jp/crit"
                ref="Department" minOccurs="1" />
        </xs:sequence>
    </xs:complexType>
</xs:element>
<xs:element name="ChangeTo"
    minOccurs="0" maxOccurs="unbounded">
    <xs:complexType>
        <xs:sequence>
            <xs:element xmlns:q1="http://med.osaka-u.ac.jp/crit"
                ref="DateTime" minOccurs="1" />
            <xs:element xmlns:q2="http://med.osaka-u.ac.jp/crit"
                ref="Department" minOccurs="1" />
        </xs:sequence>
    </xs:complexType>
</xs:element>
<xs:element name="Discharge" minOccurs="0" maxOccurs="unbounded">
    <xs:complexType>
        <xs:sequence>
            <xs:element xmlns:q1="http://med.osaka-u.ac.jp/crit"
                ref="DateTime" minOccurs="1" />
            <xs:element xmlns:q2="http://med.osaka-u.ac.jp/crit"
                ref="Department" minOccurs="1" />
        </xs:sequence>
    </xs:complexType>
</xs:element>
</xs:choice>
</xs:sequence>
<xs:attribute name="patientId" type="xs:string" use="required" />
</xs:complexType>
</xs:element>
<xs:element name="DateTime">
    <xs:complexType>
        <xs:attribute name="value" type="dateTime" use="required" />
    </xs:complexType>
</xs:element>

```

```
<xs:element name="Department">
  <xs:complexType>
    <xs:attribute name="code" type="xs:string" use="optional" />
    <xs:attribute name="text" type="xs:string" use="required" />
  </xs:complexType>
</xs:element>
</xs:schema>
```

## 5.5 Operation 応答メッセージ

### 5.5.1 XML 要素



#### Procedures 要素



子要素:

(*Procedure*\*)

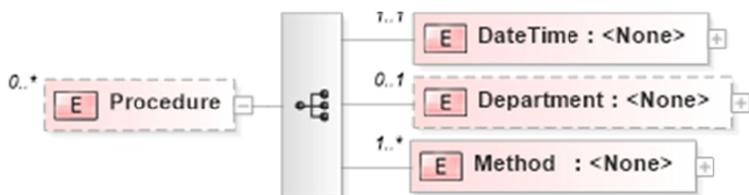
属性:

名称	型	詳細
patientId	string	患者 ID

Procedures 要素が Operation 応答メッセージのルート要素です。

patientId 属性値には、Operation メソッドの pid 引数として渡された値が設定されます。

## Procedure 要素



子要素:

(*DateTime*, *Department?*, *Method*)

属性:

なし

Procedure 要素には、患者の手術情報が記述されます。

## DateTime 要素



Liquid XML 2014 Designer Edition (Trial) 12.2.8.5459

子要素:

なし

属性:

名称	型	詳細
value	dateTime	手術実施日

DateTime 要素には、手術実施日時が記述されます。value 属性値には、ISO 8601 date format に準拠した手術実施日時を表す文字列が設定されます。

## Department 要素



Liquid XML 2014 Designer Edition (Trial) 12.2.8.5459

子要素:

なし

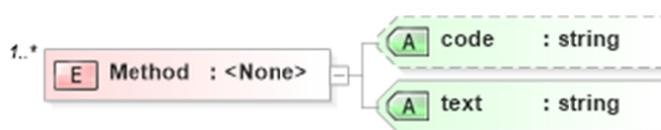
属性:

名称	型	詳細
code	string (Optional)	診療科コード
text	string	診療科名

Department 要素には、手術を実施した診療科情報が記述されます。

code 属性値には、各診療施設で定義された診療科コードを設定してください。

### Method 要素



子要素:

なし

属性:

名称	型	詳細
code	string (Optional)	手術コード
text	string	術式名称

Method 要素には、術式情報が記述されます。

code 属性値には、各診療施設で定義された術式コードを設定してください。

## 5.5.2 Operation 応答メッセージ例

```
<?xml version="1.0" encoding="utf-8" ?>
<Procedures patientId="12345" xmlns="http://med.osaka-u.ac.jp/crit">
  <Procedure>
    <DateTime value="2011-06-26"/>
    <Department code="10" text="外科"/>
    <Method code="34068001" text="心臓弁置換"/>
  </Procedure>
  <Procedure>
    <DateTime value="2012-08-03"/>
    <Department code="10" text="外科"/>
    <Method code="48387007" text="気管切開"/>
  </Procedure>
</Procedures>
```

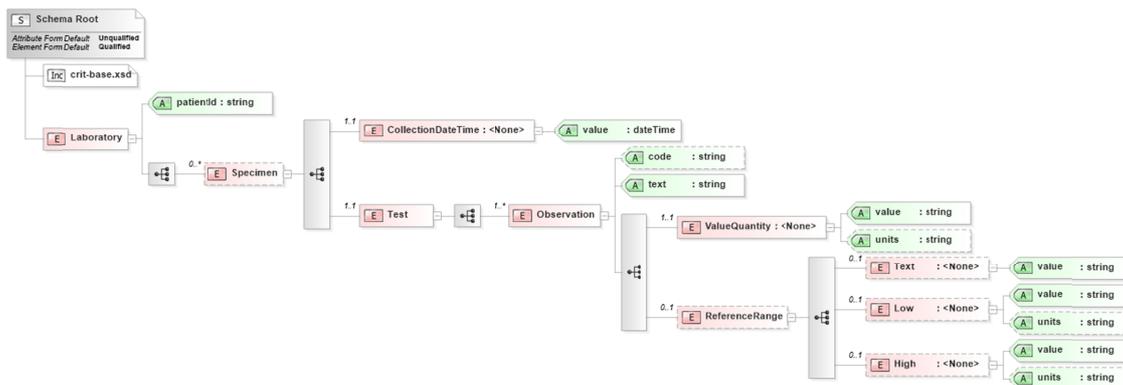
```
</Procedure>
</Procedures>
```

### 5.5.3 Operation 応答メッセージスキーマ

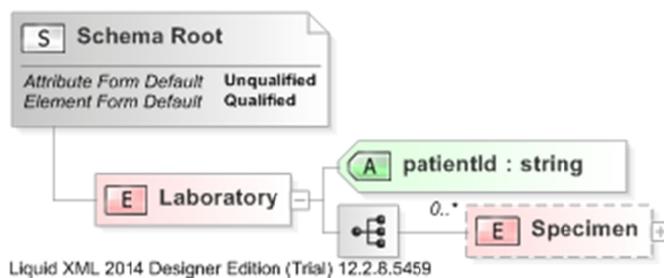
```
<?xml version="1.0" encoding="utf-8" ?>
<xs:schema xmlns="http://med.osaka-u.ac.jp/crit"
  attributeFormDefault="unqualified"
  elementFormDefault="qualified"
  targetNamespace="http://med.osaka-u.ac.jp/crit"
  xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:include schemaLocation="crit-base.xsd" />
  <xs:element name="Procedures">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="Procedure"
          minOccurs="0" maxOccurs="unbounded">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="DateTime" minOccurs="1">
                <xs:complexType>
                  <xs:attribute name="value" type="dateTime" use="required" />
                </xs:complexType>
              </xs:element>
              <xs:element name="Department" minOccurs="0">
                <xs:complexType>
                  <xs:attribute name="code" type="xs:string" use="optional" />
                  <xs:attribute name="text" type="xs:string" use="required" />
                </xs:complexType>
              </xs:element>
              <xs:element name="Method" minOccurs="1" maxOccurs="unbounded">
                <xs:complexType>
                  <xs:attribute name="code" type="xs:string" use="optional" />
                  <xs:attribute name="text" type="xs:string" use="required" />
                </xs:complexType>
              </xs:element>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
      </xs:sequence>
      <xs:attribute name="patientId" type="xs:string" use="required" />
    </xs:complexType>
  </xs:element>
</xs:schema>
```

## 5.6 Laboratory 応答メッセージ

### 5.6.1 XML 要素



#### Laboratory 要素



子要素:

(*Specimen\**)

属性:

名称	型	詳細
patientId	string	患者 ID

Laboratory 要素が Laboratory 応答メッセージのルート要素です。patientId 属性値には、Laboratory メソッドの pid 引数として渡された値が設定されます。

## Specimen 要素



子要素:

(*CollectionDateTime*, *Test*)

属性:

なし

Specimen 要素は検体を表し、検体採取日時情報とその検体から測定された検査値を子要素として持ちます。

## CollectionDateTime 要素



子要素:

なし

属性:

名称	型	詳細
value	dateTime	検体採取日時

CollectionDateTime 要素には、検体の採取日時が記述されます。value 属性値には、ISO 8601 date format に準拠した日時を表す文字列が設定されます。

## Test 要素



子要素:

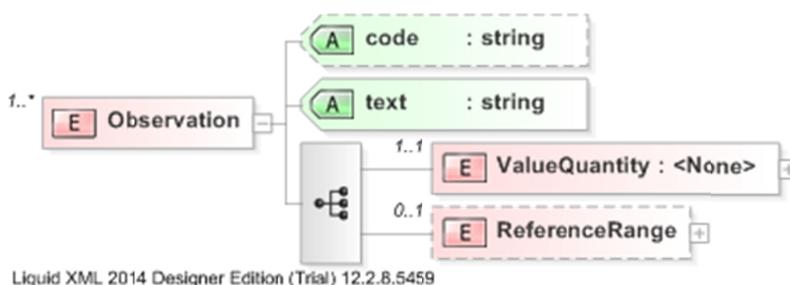
(*Observation*\*)

属性:

なし

Test 要素は検体検査を表し、子要素として検査項目を持ちます。

### Observation 要素



子要素:

(*ValueQuantity, ReferenceRange?*)

属性:

名称	型	詳細
code	string	(Optional) 検査項目コード
text	string	検査項目名称

Observation 要素には、検査項目情報が記述されます。

code 属性値には、各診療施設で定義された検査項目コードを設定してください。

### ValueQuantity 要素



子要素:

なし

属性:

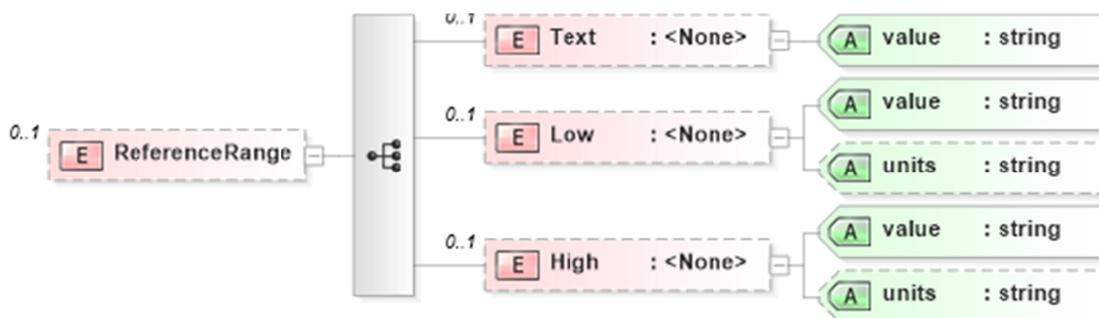
名称	型	詳細
value	string	検査値
units	string	(optional) 検査値単位

ValueQuantity 要素には、検査値が記述されます。

value 属性値には、検査値が文字列で記述されます。測定値が“感度以下”などの数値ではない場合にも、その文字列がそのまま value 属性値としてセットされます。

units 属性値には、施設で定義されている単位名称が記述されます。

### ReferenceRange 要素



子要素:

(Text?, Low?, High?)

属性:

なし

ReferenceRange 要素には、検査値の正常値情報が記述されます。

### Text 要素



子要素:

なし

属性:

名称	型	詳細
value	string	正常域説明文

Text 要素の value 属性値には、検査値の正常域を表す文字列が記述されます。

正常域の上限値および下限値が定義されている場合には、後述の Low 要素および High 要素の設定で十分ですが、数値で正常域が定義されない場合には、この Text 要素に正常域情報を設定してください。

また、正常域の上限値および下限値が定義されている場合でも、正常域を文字で表現可能な場合には、この Text 要素を設定しても構いません。

## Low 要素



子要素:

なし

属性:

名称	型	詳細
value	String	正常値下限値
units	String	<i>(optional)</i> 正常値下限値単位

Low 要素には、検査値の正常値下限値が記述されます。  
units 属性値には、施設で定義されている単位名称が記述されます。

## High 要素



子要素:

なし

属性:

名称	型	詳細
value	String	正常値上限値
units	String	<i>(optional)</i> 正常値上限値単位

High 要素には、検査値の正常値上限値が記述されます。

units 属性値には、施設で定義されている単位名称が記述されます。

## 5.6.2 Laboratory 応答メッセージ例

```
<?xml version="1.0" encoding="utf-8" ?>
<Laboratory patientId="12345" xmlns="http://med.osaka-u.ac.jp/crit">
  <Specimen>
    <CollectionDateTime value="2014-03-04T08:30:00+09:00" />
    <Test>
      <Observation code="718-7" text="Haemoglobin">
        <ValueQuantity value="176" units="g/L" />
        <ReferenceRange>
          <Low value="135" units="g/L" />
          <High value="180" units="g/L" />
        </ReferenceRange>
      </Observation>
      <Observation code="718-7" text="Haemoglobin">
        <ValueQuantity value="176" units="g/L" />
        <ReferenceRange>
          <Low value="135" units="g/L" />
          <High value="180" units="g/L" />
        </ReferenceRange>
      </Observation>
    </Test>
  </Specimen>
  <Specimen>
    <CollectionDateTime value="2014-03-04T08:30:00+09:00" />
    <Test>
      <Observation code="718-7" text="Haemoglobin">
        <ValueQuantity value="176" units="g/L" />
        <ReferenceRange>
          <Low value="135" units="g/L" />
          <High value="180" units="g/L" />
        </ReferenceRange>
      </Observation>
      <Observation code="718-7" text="Haemoglobin">
        <ValueQuantity value="176" units="g/L" />
        <ReferenceRange>
          <Low value="135" units="g/L" />
          <High value="180" units="g/L" />
        </ReferenceRange>
      </Observation>
    </Test>
  </Specimen>
</Laboratory>
```

## 5.6.3 Laboratory 応答メッセージスキーマ

```
<?xml version="1.0" encoding="utf-8" ?>
<xs:schema xmlns="http://med.osaka-u.ac.jp/crit"
  attributeFormDefault="unqualified"
  elementFormDefault="qualified"
  targetNamespace="http://med.osaka-u.ac.jp/crit"
  xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:include schemaLocation="crit-base.xsd" />
  <xs:element name="Laboratory">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="Specimen"
          minOccurs="0"
          maxOccurs="unbounded">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="CollectionDateTime"
                minOccurs="1">
              <xs:complexType>
                <xs:attribute name="value"
                  type="dateTime"
                  use="required" />
              </xs:complexType>
            </xs:sequence>
          </xs:element>
          <xs:element name="Test"
            minOccurs="1">
            <xs:complexType>
              <xs:sequence>
                <xs:element name="Observation" minOccurs="1"
                  maxOccurs="unbounded">
                <xs:complexType>
                  <xs:sequence>
                    <xs:element name="ValueQuantity" minOccurs="1">
                    <xs:complexType>
                      <xs:attribute name="value"
                        type="xs:string"
                        use="required" />
                      <xs:attribute name="units"
                        type="xs:string" use="optional" />
                    </xs:complexType>
                  </xs:sequence>
                </xs:element>
                <xs:element name="ReferenceRange"
                  minOccurs="0">
                <xs:complexType>
                  <xs:sequence>
```

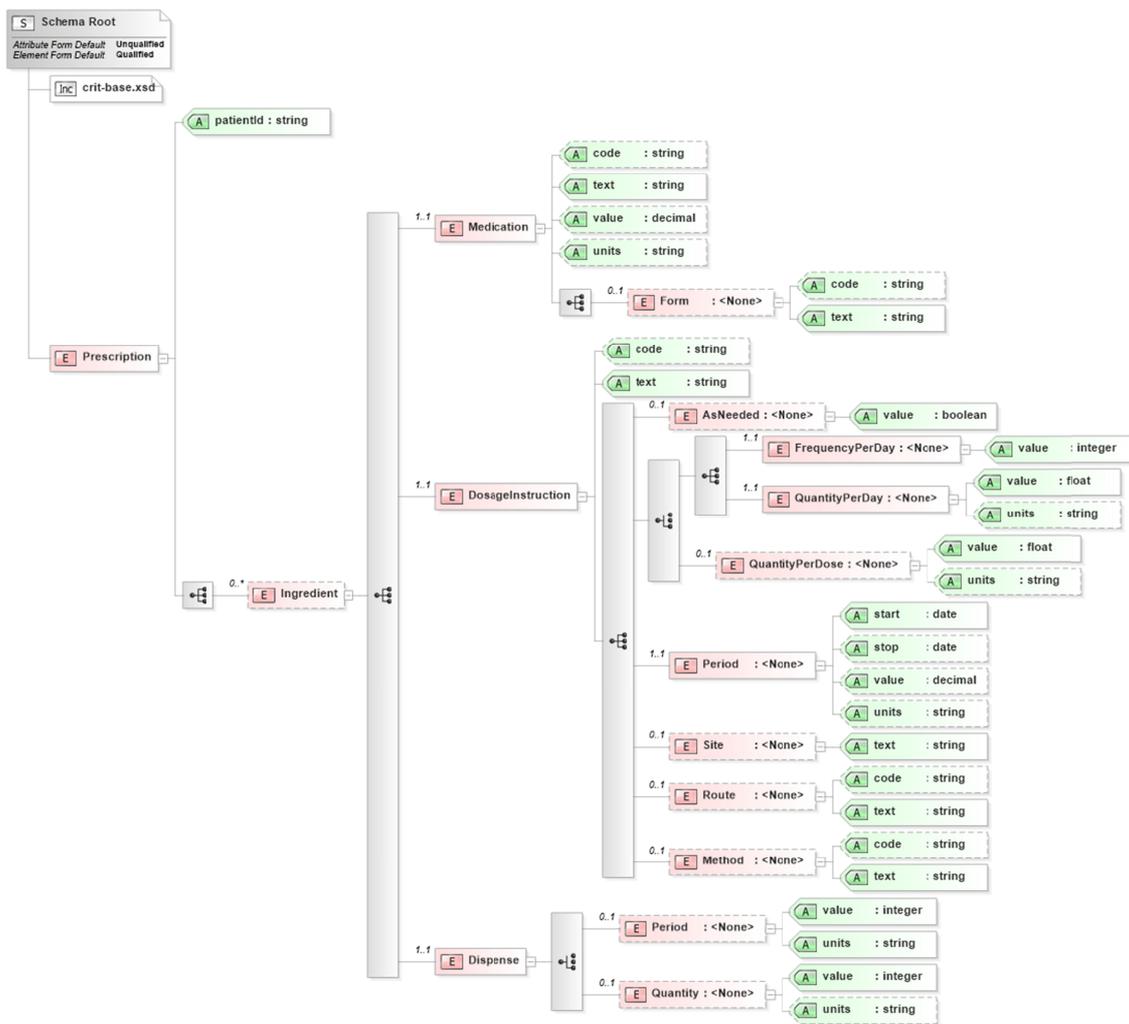
```

        <xs:element name="Text"
            minOccurs="0">
        <xs:complexType>
            <xs:attribute name="value"
                type="xs:string"
                use="required" />
        </xs:complexType>
    </xs:element>
    <xs:element name="Low" minOccurs="0">
        <xs:complexType>
            <xs:attribute name="value"
                type="xs:string"
                use="required" />
            <xs:attribute name="units"
                type="xs:string"
                use="optional" />
        </xs:complexType>
    </xs:element>
    <xs:element name="High" minOccurs="0">
        <xs:complexType>
            <xs:attribute name="value"
                type="xs:string"
                use="required" />
            <xs:attribute name="units"
                type="xs:string"
                use="optional" />
        </xs:complexType>
    </xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:sequence>
<xs:attribute name="patientId" type="xs:string" use="required" />
</xs:complexType>
</xs:element>
</xs:schema>

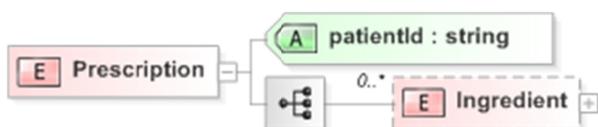
```

# 5.7 Prescription 応答メッセージ

## 5.7.1 XML 要素



### Prescription 要素



Liquid XML 2014 Designer Edition (Trial) 12.2.8.5459

子要素:

*(Ingredient\*)*

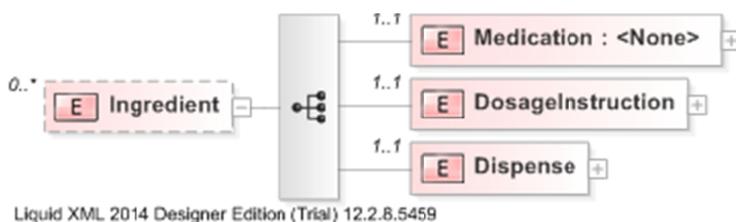
属性:

名称	型	詳細
patientId	string	患者 ID

Prescription 要素が Prescription 応答メッセージのルート要素です。

patientId 属性値には、Prescription メソッドの pid 引数として渡された値が設定されます。

### Ingredient 要素



子要素:

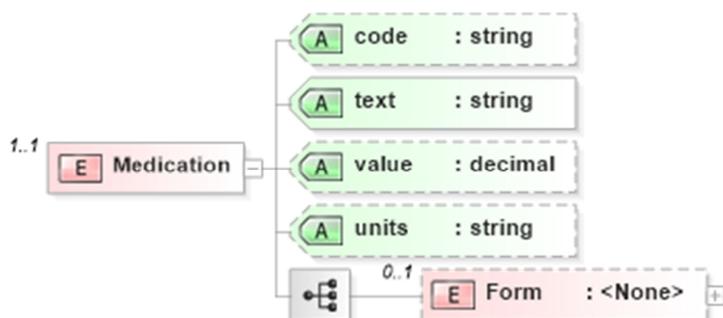
*(Medication, DosageInstruction, Dispense)*

属性:

なし

Ingredient 要素は処方項目を表し、薬剤情報、用法、用量を子要素として持ちます。

### Medication 要素



子要素:

*(Form <PrescriptionMedicationFormElement>?)*

属性:

名称	型	詳細
code	string	<i>(Optional)</i> 薬剤コード

text	string		薬剤名称
value	decimal	(optional)	有効成分量
units	string	(optional)	有効成分量単位

Medication 要素には薬剤情報が記述されます。

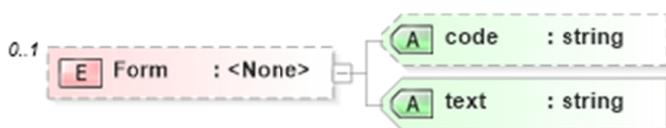
code 属性値には、各診療施設で定義された薬剤コードを設定してください。

text 属性値には、薬剤名称を設定してください。

value 属性値には、薬剤の有効成分量を設定してください。

units 属性値には、施設で定義されている単位名称が記述されます。

## Form 要素



子要素:

なし

属性:

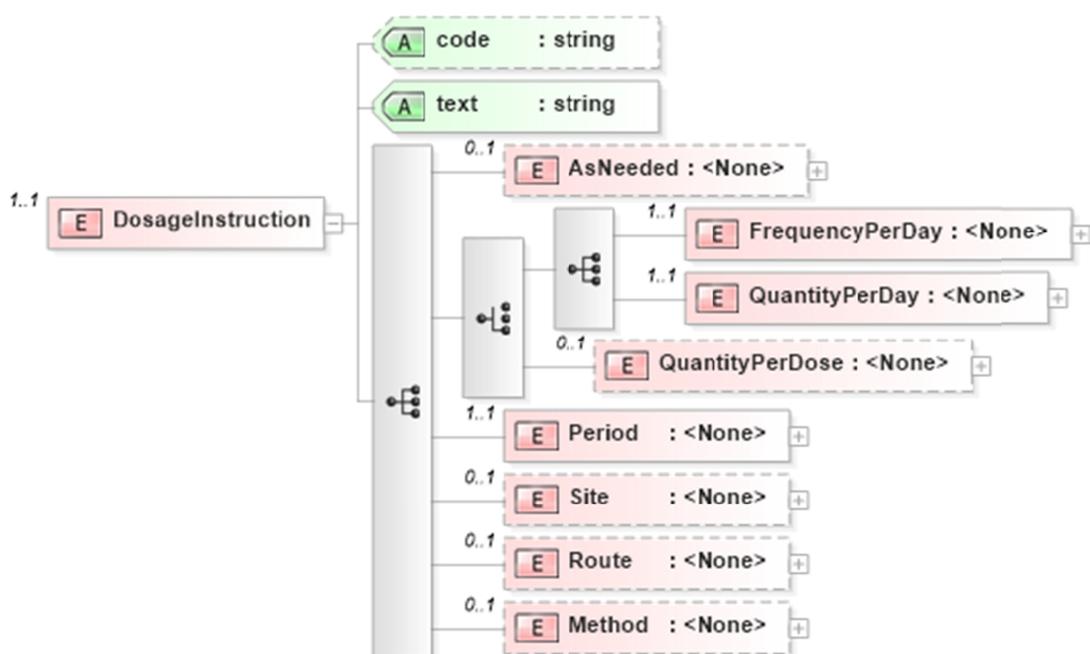
名称	型	詳細
code	string	(Optional) 剤形コード
text	string	剤形名称

Form 要素には薬剤の剤形情報が記述されます。

code 属性値には、各診療施設で定義された剤形コードを設定してください。

text 属性値には、各診療施設で定義された剤形名称を設定してください。

## DosageInstruction 要素



子要素:

(*AsNeeded?*, *FrequencyPerDay?*, *QuantityPerDay?*, *QuantityPerDose?*, *Period*, *Site?*, *Route?*, *Method?*)

属性:

名称	型	詳細
code	string	(Optional) 用法コード
text	string	用法名称

DosageInstruction 要素には、用法が記述されます。

code 属性値には、各診療施設で定義された用法コードを設定してください。text 属性値には、用法を表す文字列を設定してください。

## AsNeeded 要素



子要素:

なし

属性:

名称	型	詳細
value	boolean	頓用の場合 true、そうでない場合 false。

頓用か頓用でないかを記述します。処方が頓用か頓用でないかを正確に確定できない場合には、

AsNeeded 要素は省略してください。

### FrequencyPerDay 要素



子要素:

なし

属性:

名称	型	詳細
value	integer	1 日回数

頓用でない場合の、1 日当たりの投与回数を記述します。

### QuantityPerDay 要素



子要素:

なし

属性:

名称	型	詳細
value	integer	1 日量
units	string	<i>(optional)</i> 1 日量単位

頓用でない場合の、1 日当たりの投与量を記述します。隔日投与や曜日指定投与の場合にも、平均値ではなく、投与日の 1 日当たりの投与量が記述されます。

### QuantityPerDose 要素



子要素:

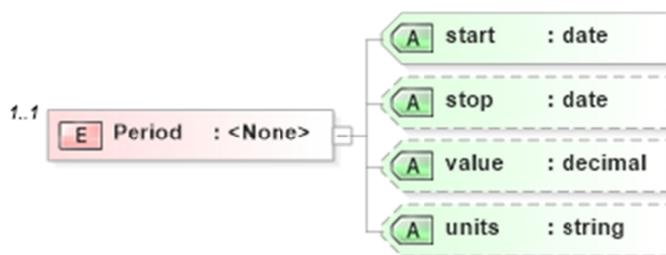
なし

属性:

名称	型	詳細
value	integer	1 回量
units	string	<i>(optional)</i> 1 回量単位

頓用の場合の、1 回当たりの投与量を記述します。

### DosageInstruction/Period 要素



子要素:

なし

属性:

名称	型	詳細
start	date	処方開始日
stop	date	<i>(optional)</i> 処方終了日
value	decimal	<i>(optional)</i> 投与日数
units	string	<i>(optional)</i> 投与日数単位

処方開始日と終了日を記述します。

start 属性値には、処方開始日 (投与開始日) を設定してください。

stop 属性値には、処方終了日 (投与終了日) を設定してください。頓服薬など処方終了日が設定できない場合には、stop 属性値を記述しないでください。

value 属性値には、投与日数を設定してください。頓服薬など投与日数が設定できない場合には、value 属性値を記述しないでください。

units 属性値には、投与日数の単位を設定してください。

## Site 要素



子要素:

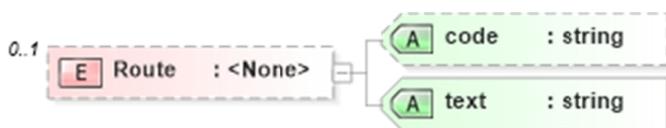
なし

属性:

名称	型	詳細
text	string	(optional) 部位

部位を記述します。

## Route 要素



子要素:

なし

属性:

名称	型	詳細
code	string	(Optional) 経路コード
text	string	経路名称

Route 要素には、摂取経路情報が記述されます。

code 属性値には、各診療施設で定義された経路コードを設定してください。

text 属性値には、経路を表す文字列を設定してください。

## Method 要素



子要素:

なし

属性:

名称	型	詳細
code	string	(Optional) 用法種コード
text	string	用法種名称

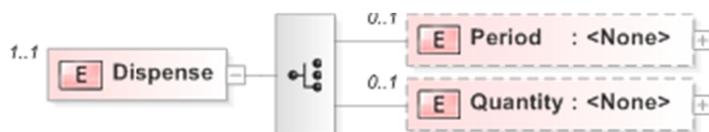
Method 要素には、用法の種類が記述されます。用法種は以下のどれかに分類されるものとします。

- ・ 内服
- ・ 外用
- ・ 注射
- ・ その他

code 属性値には、各診療施設で定義された用法種コードを設定してください。

text 属性値には、用法種を表す文字列 (内服|外用|注射|その他) を設定してください。

## Dispense 要素



Liquid XML 2014 Designer Edition (Trial) 12.2.8.5459

子要素:

(*Period?*, *Quantity?*)

属性:

なし

Dispense 要素には、投与日数もしくは投与回数を記述します。

### Dispense/Period 要素



子要素:

なし

属性:

名称	型	詳細
value	integer	投与実日数
units	string	<i>(optional)</i> 投与実日数単位

投与実日数 (処方総量の日数換算値) を記述します。

毎日定期的に投与される薬剤の場合、この数値は *DosageInstruction/Period* 要素の value 属性値と等しくなります。隔日投与される薬剤の場合、この数値は *DosageInstruction/Period* 要素の value 属性値の 1/2 の値が設定されます。

### Dispense/Quantity 要素



子要素:

なし

属性:

名称	型	詳細
value	integer	投与回数
units	string	<i>(optional)</i> 投与回数単位

頓用の場合の投与回数を記述します。

## 5.7.2 Prescription 応答メッセージ例

```
<?xml version="1.0" encoding="utf-8" ?>
<Prescription patientId="12345" xmlns="http://med.osaka-u.ac.jp/crit">
  <Ingredient>
    <!-- 内服薬例 -->
    <Medication code="DB00501" text="タガメット錠 200mg"
      value="200" units="mg">
      <Form code="1" text="錠剤" />
    </Medication>
    <DosageInstruction code="12A14" text="[分 2] 1 日 2 回:朝食後と寝る前">
      <FrequencyPerDay value="2" /> <!-- 分 2 -->
      <QuantityPerDay value="2" units="T" /> <!-- 1 日 2 錠 -->
      <Period start="2012-05-23" stop="2013-06-06" />
      <Route text="経口" />
      <Method text="内服" />
    </DosageInstruction>
    <Dispense>
      <Period value="14" units="days" /> <!-- 14 日分 -->
    </Dispense>
  </Ingredient>
  <Ingredient>
    <!-- 頓服薬 -->
    <Medication code="00028-0264" text="ボルタレン錠 25mg"
      value="25" units="mg">
      <Form code="1" text="錠剤" />
    </Medication>
    <DosageInstruction code="470113" text="発熱時">
      <AsNeeded value="true" /> <!-- 頓服 -->
      <QuantityPerDose value="2" units="T" /> <!-- 1 回 2 錠 -->
      <Period start="2012-05-23" /> <!-- 頓用なので処方開始日のみ指定 -->
      <Route text="経口" />
      <Method text="内服" />
    </DosageInstruction>
    <Dispense>
```

```

        <Quantity value="5" units="times" />                                <!-- 5回分 -->
    </Dispense>
</Ingredient>
<Ingredient>
    <!-- 外用薬例 -->
    <Medication code="25140" text="モーラステープ 20mg"
        value="20" units="mg">
        <Form code="29" text="貼付剤(湿布薬)" />
    </Medication>
    <DosageInstruction code="81E000" text="1日1回">
        <FrequencyPerDay value="1" />                                    <!-- 1日1回 -->
        <QuantityPerDay value="1" units="sheet" />                       <!-- 1日1枚 -->
        <Period start="2014-11-25"
            stop="2014-12-29" />
        <Site text="肩、腰" />
        <Route text="皮膚" />
        <Method text="外用" />
    </DosageInstruction>
    <Dispense>
        <Period value="35" units="days" />                               <!-- 35日分 -->
    </Dispense>
</Ingredient>
<Ingredient>
    <!-- 外用薬(頓用)例 -->
    <Medication code="DB00502" text="ボルタレンサポ 12.5mg"
        value="12.5" units="mg">
        <Form code="17" text="坐剤" />
    </Medication>
    <DosageInstruction code="9 J053" text="いたむ時">
        <AsNeeded value="true" />
        <QuantityPerDose value="1" units="T" />
        <Period start="2012-05-23" />                                     <!-- 非定期(頓用) -->
        <Route text="直腸" />                                             <!-- 1回1個 -->
        <Method text="外用" />                                           <!-- 頓用なので処方開始日のみ指定 -->
    </DosageInstruction>
    <Dispense>
        <Quantity value="3" units="times" />                               <!-- 3回分 -->
    </Dispense>
</Ingredient>
</Prescription>

```

### 5.7.3 Prescription 応答メッセージスキーマ

```

<?xml version="1.0" encoding="utf-8" ?>
<xs:schema xmlns="http://med.osaka-u.ac.jp/crit"
    attributeFormDefault="unqualified"
    elementFormDefault="qualified"

```

```

        targetNamespace="http://med.osaka-u.ac.jp/crit"
        xmlns:xs="http://www.w3.org/2001/XMLSchema">
<xs:include schemaLocation="crit-base.xsd" />
<xs:element name="Prescription">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="Ingredient"
        minOccurs="0" maxOccurs="unbounded">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="Medication" minOccurs="1">
              <xs:complexType>
                <xs:sequence>
                  <xs:element name="Form" minOccurs="0" maxOccurs="1">
                    <xs:complexType>
                      <xs:attribute name="code" type="xs:string" />
                      <xs:attribute name="text" type="xs:string" use="required" />
                    </xs:complexType>
                  </xs:element>
                </xs:sequence>
                <xs:attribute name="code" type="xs:string" use="optional" />
                <xs:attribute name="text" type="xs:string" use="required" />
                <xs:attribute name="value" type="xs:decimal" use="optional" />
                <xs:attribute name="units" type="xs:string" use="optional" />
              </xs:complexType>
            </xs:element>
          <xs:element name="DosageInstruction" minOccurs="1">
            <xs:complexType>
              <xs:sequence>
                <xs:element name="AsNeeded" minOccurs="0">
                  <xs:complexType>
                    <xs:attribute name="value" type="xs:boolean"
                      use="required" />
                  </xs:complexType>
                </xs:element>
                <xs:choice>
                  <xs:sequence>
                    <xs:element name="FrequencyPerDay" minOccurs="1">
                      <xs:complexType>
                        <xs:attribute name="value" type="xs:integer"
                          use="required" />
                      </xs:complexType>
                    </xs:element>
                    <xs:element name="QuantityPerDay" minOccurs="1">
                      <xs:complexType>
                        <xs:attribute name="value" type="xs:float"
                          use="required" />
                      </xs:complexType>
                    </xs:element>
                  </xs:sequence>
                </xs:choice>
              </xs:sequence>
            </xs:complexType>
          </xs:element>
        </xs:sequence>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
</xs:element>

```

```

                <xs:attribute name="units" type="xs:string"
                    use="optional" />
            </xs:complexType>
        </xs:element>
    </xs:sequence>
    <xs:element name="QuantityPerDose" minOccurs="0">
        <xs:complexType>
            <xs:attribute name="value" type="xs:float" use="required"
                />
            <xs:attribute name="units" type="xs:string"
                use="optional" />
        </xs:complexType>
    </xs:element>
</xs:choice>
<xs:element name="Period" minOccurs="1">
    <xs:complexType>
        <xs:attribute name="start" type="xs:date" use="required" />
        <xs:attribute name="stop" type="xs:date" use="optional" />
        <xs:attribute name="value" type="xs:decimal" use="optional"
            />
        <xs:attribute name="units" type="xs:string" use="optional"
            />
    </xs:complexType>
</xs:element>
<xs:element name="Site" minOccurs="0">
    <xs:complexType>
        <xs:attribute name="text" type="xs:string"
            use="required" />
    </xs:complexType>
</xs:element>
<xs:element name="Route" minOccurs="0">
    <xs:complexType>
        <xs:attribute name="code" type="xs:string"
            use="optional" />
        <xs:attribute name="text" type="xs:string"
            use="required" />
    </xs:complexType>
</xs:element>
<xs:element name="Method" minOccurs="0">
    <xs:complexType>
        <xs:attribute name="code" type="xs:string"
            use="optional" />
        <xs:attribute name="text" type="xs:string"
            use="required" />
    </xs:complexType>
</xs:element>
</xs:sequence>

```

```

        <xs:attribute name="code" type="xs:string"
                    use="optional" />
        <xs:attribute name="text" type="xs:string" use="required" />
    </xs:complexType>
</xs:element>
<xs:element name="Dispense" minOccurs="1">
    <xs:complexType>
        <xs:choice>
            <xs:element name="Period" minOccurs="0">
                <xs:complexType>
                    <xs:attribute name="value"
                                type="xs:integer" use="required" />
                    <xs:attribute name="units"
                                type="xs:string" use="required" />
                </xs:complexType>
            </xs:element>
            <xs:element name="Quantity" minOccurs="0">
                <xs:complexType>
                    <xs:attribute name="value"
                                type="xs:integer" use="required" />
                    <xs:attribute name="units"
                                type="xs:string" use="optional" />
                </xs:complexType>
            </xs:element>
        </xs:choice>
    </xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:sequence>
<xs:attribute name="patientId"
              type="xs:string" use="required" />
</xs:complexType>
</xs:element>
</xs:schema>

```

## 第6章 変更履歴

- ・ v0.41 <2015-4-8>
  - 以下の要素の説明文中にある“unit 属性値”の記述を“units 属性値”に訂正。
    - \* *Height/Value* 要素
    - \* *Weight/Value* 要素
    - \* *DosageInstruction/Period* 要素
  - 以下の XML スキーマダイアグラム中の  
/Prescription/Ingredient/DosageInstruction/Period の unit 属性名を units に訂正。
    - \* *Prescription* 応答メッセージ
    - \* *DosageInstruction/Period* 要素
  - *Prescription* 応答メッセージスキーマ
    - \* /Prescription/Ingredient/DosageInstruction/Period の unit 属性名を units に訂正。
  - *Given* 要素
    - \* 属性表中の value 属性の詳細を“姓”から“名”に訂正。
- ・ v0.4 <2015-02-16>
  - *ICrfReporterComm* インタフェース
    - \* *StartApp* メソッドの説明文から CRF Reporter 起動時のトークンに関する記述を削除。
    - \* *StartApp* メソッド (*IntPtr*)
      - ・ パラメータ *hToken* のデータ型を *Int32* から *IntPtr* に変更。
    - \* *StartApp* メソッド (*String, String*)
      - ・ *password* パラメータを追加。
  - *IEmrComm* インタフェース
    - \* *BodyMeasurement* メソッド
      - ・ *fromDate*, *toDate*, *maxCount* パラメータを追加。
    - \* *Laboratory* メソッド
      - ・ *fromDate* と *toDate* パラメータのデータ型を *Nullable<DateTime>* から *DateTime* に変更
    - \* *Prescription* メソッド
      - ・ 返されるデータに経路と用法種を追加。
  - *CRIT* 応答メッセージ XML 仕様
    - \* *Operation* 応答メッセージ
      - ・ *Method* 要素を複数設定可能 (*maxOccurs*="unbounded") に変更。
    - \* *Laboratory* 応答メッセージ・*Text* 要素を追加。
    - \* *Prescription* 応答メッセージ
      - ・ *Medication* 要素の *form* 属性を削除し、*Medication* 要素の子要素として *Form* 要素を追加。
      - ・ *DosageInstruction* 要素の子要素の *Text* 要素を削除し、*DosageInstruction* 要素の *code* 属性と *text* 属性を追加。
      - ・ *AsNeeded* 要素の説明文変更 (頓用 / 非頓用を判断できない場合には *AsNeeded* 要素を省略可能)。
      - ・ *AsNeeded* 要素の *value* 属性を必須属性に変更 (*use*="required")。
      - ・ *DosageInstruction/Period* 要素に *value* 属性と *unit* 属性を追加。

- *Route* 要素を追加。
  - *Method* 要素を追加。
  - *Dispense/Period* 要素が投与実日数 (処方総量の日数換算値) であることを明記。
  - *Prescription* 応答メッセージ例変更 (頓用薬の /Prescription/Ingredient/Dispense/Quantity の単位を回数に統一)。
- v0.3 <2014-12-18>
  - v0.2 <2014-10-30>