

**第 4 章**

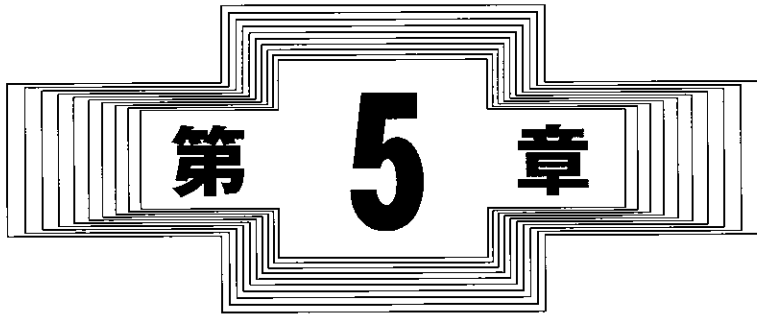
**ストーリーボード**



**Storyboards**

けられます。

UMLモデリングは、要求を表現する、より形式的な方法であるユースケースという概念を有しています。ユースケースは、ある特定のシステムの動きを完全に一方向の道筋として定義します。ユースケースは、事前条件と事後条件を伴ったイベントを定義することも含めて、特定された1組の参加者（システム利用者）による数々の動作によって引き起こされる帰結の集合を包含します。HL7のインタラクション、トリガイベント、およびアプリケーションロールから構成されるそれぞれの組は、ある特定のユースケースの対象領域を記述していることとなります。

A decorative frame consisting of multiple concentric, stepped rectangular lines that form a cross-like shape. The frame is centered on the page and contains the chapter title.

**第 5 章**

**アプリケーションロール**



**Application Roles**

なるでしょう。

HL7は、これらのアプリケーションロールに適合するシステムコンポーネントが、どのようにしてそれらを実行するかという具体的方法は定義しません。言い換えれば、HL7はシステムの内部動作を標準化するわけではありません。しかしながら、HL7方法論は、アプリケーションロールという観点から、HL7適合（HL7 Conformance）そのものを定義しようとしています。

ソフトウェアがアプリケーションロールのサポートを適合宣言しているということは、そのソフトウェアが以下の条件を満たしていることを意味します。

- 送信を要求されているすべてのメッセージを送信する。
- そのアプリケーションロールが受信することを要求されているすべてのメッセージを受信する。
- これらのすべてのインタラクションに関連付けられている受信者責務を遂行する。

同じメッセージ型が、異なるトリガイベント、異なるアプリケーションロールの集合、あるいはその両者に対応して使用されることがあります。しかし、その場合でも、インタラクションはすべて固有であり、固有の識別子を持つこととなります。

V3アプリケーションロールのこれらの適合に関する事項は、V3メッセージと方法論の第1段階の承認には盛り込まない予定です。

## 5.1 トリガイベント

トリガイベントは、システムコンポーネント間の情報転送のきっかけとなる条件の明示的な集合です。トリガイベントは、例えば、臨床医が検査オーダーを発行する、薬剤を調剤するなどの実世界のイベントです。トリガイベントは、自動化されたシステムによって認識可能でなければなりません。

例

■説明

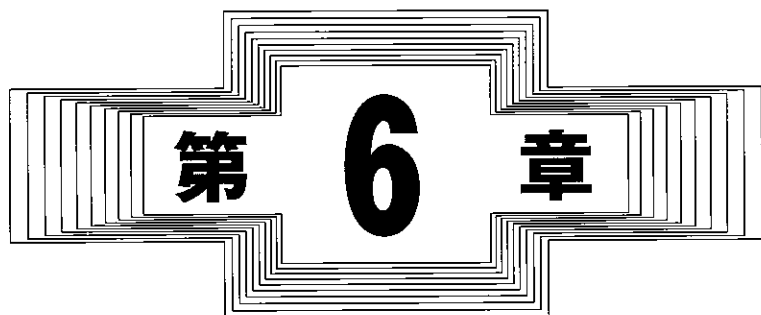
構造化名：複合薬剤部門イベント完了、通知 (Combined Pharmacy Event Complete, Notification)

型：状態遷移型

薬剤などの投薬や調剤が完了（充足）したこと、そして、応答要求なしで受信アプリケーションにそれを通知していることを示します。

このトリガイメントには、これまでにHL7で定義されているすべての種類の薬剤部門イベントと調和するように構造化名が割り当てられています。

このトリガイメントは、薬剤の調剤の完了を知らせる合図となっていますので、状態遷移型と定義されています。



**第 6 章**

**HL7 V3メッセージの作成  
—V3モデリング手法—**



**Making HL7 V3 Messages  
—the V3 Modelling Approach—**

### 6.1.2 関連 (Associations)

UMLにおいては、関連はクラス間の論理的な関係を記述します。異なったクラスの間に関連があれば、同じクラスの2つのインスタンスの間にも関連があることもあります。

HL7 RIMでの関連は、線を結びつけることによってグラフィック表記されています。ある関連の各終端は、必ず唯一つのクラスに結合されています。しばしば、**末端側 (distal)** という用語が、ある関連の反対の終端に接続されているクラスについて言及するのに使用されます。

ある関連のそれぞれの終端は、その関連によって関係付けられ、そこに接続されているクラスのインスタンスの最小数と最大数を示す特性 (property) を有しています。この特性は、多重度 (multiplicity) あるいは **cardinality (基数)** と呼ばれて、2個の点 (..) によって区切られた下限と上限を表す数字によって表記されます。2番目の数字は、上限がないことを意味する "\*" によって置き換えられる場合もあります。

中核 RIM クラスとそれらの関連は、RIM バックボーンと呼ばれます。RIM バックボーンを表した UML 図 (図2) を以下に示します。図2には、RIM の6つの中核クラス間の8つの関連が示されています。

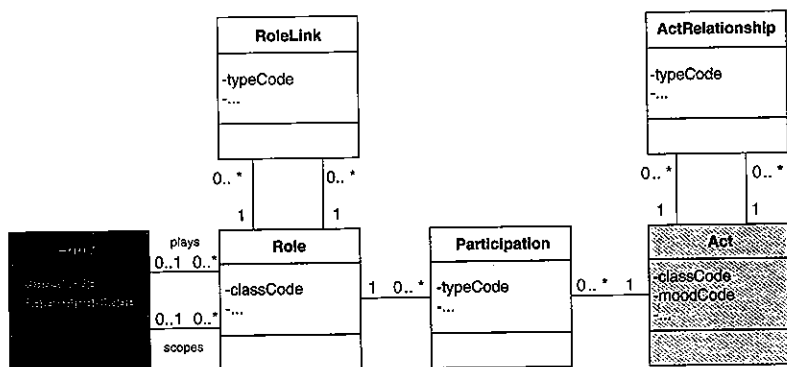


図2 RIMバックボーン

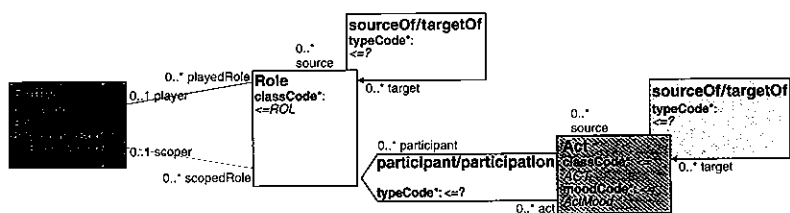


図3 R-MIM形式でのモデル表現例

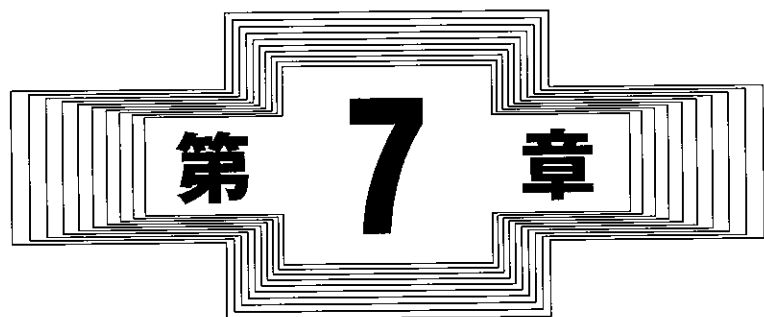
UML形式からR-MIM形式への変更をまとめると以下のとおりです。

- **Participation** は矢印形四角として示されます。
- **RoleLink** および **ActRelationship** は、それらがそれぞれ関係する **Role** と **Act** に重ね合わせて現れます。
- 関連に関しては、関連元 (source) と関連先 (target) を示すために、付加的なテキストが追加されます。  
すなわち、**ActRelationship** を例にとれば、srcAct および tgtAct がそれです。
- plays は playedRole に、scopes は scopedRole になります。詳細な解説は7.2.3節 (34ページ～) をご覧ください。

## 6.2 制約と詳細化

V3方法論の非常に重要な部分は、ある特定の要求事項を満たすために伝達される必要がある情報を正確に表すために、RIMクラスとそれらの関連に対して、制約を加えたり (constrain)、詳細化を行ったり (refine) するための手段と技法を提供していることです。本書の多くの部分が、この過程に焦点を合わせています。15章 (81ページ～) には、使用される正確な方法に関するより詳細な解説があります。





第 7 章

# R-MIMとD-MIM



R-MIMs and D-MIMs

## 7.1 詳細化メッセージ情報モデル (R-MIM) の例

図4は、単純なメッセージを定義するR-MIMです。この形式の図は、HL7 Visio ツールを使用することで作成されます (18章参照)。一度R-MIMを作り上げると、HL7ツールを用いて、メッセージ定義やさまざまな形式の他の補助資料を自動的に生成することができるようになります。それらについては7.2節 (32ページ~) で解説します。

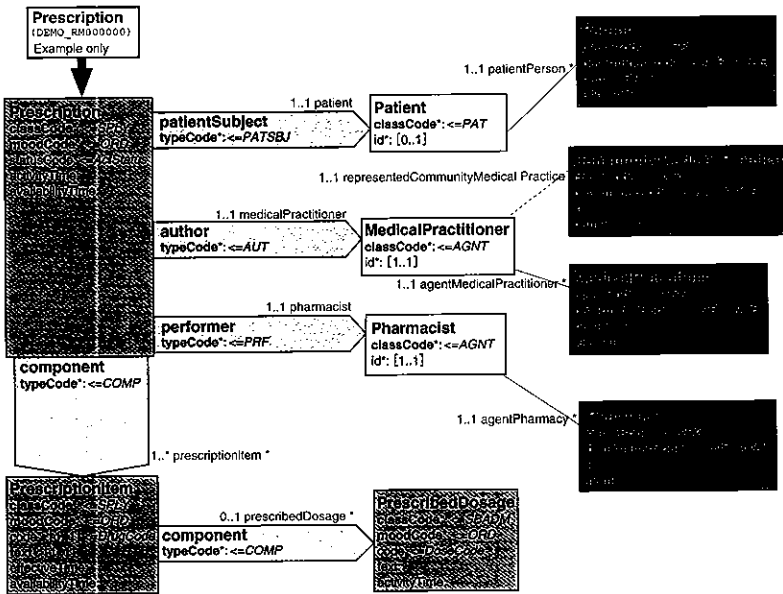


図4 処方に関するR-MIMの例

R-MIM図 (図4) では、すべての正方形、あるいは長方形はHL7クラスを表します。この例は、あるPatient (患者) のために、あるMedicalPractitioner (臨床医) が処方せんを発行するというAct (行為) をモデリングしています。例に示されるクラスは、以下の節でそれぞれ解説しています。

HL7はモデリングに際しては、UMLの手法と表記規則を採用してい

7.2節におけるR-MIMの例に戻って、PrescriptionItem（処方項目）クラスもまた、Actの1つです。Prescriptionはある処方全体を扱っていますが、PrescriptionItemはその処方に含まれる個々の項目を扱っています。

**定義：**Actは、関心をもって取り上げるあらゆる行為を表します。Actはすでに起こった出来事でもかまいませんし、計画や指示、あるいは何かを起こそうとする意図であることもあります。予約、診察、オーダー、検査、手技、紹介、そして、事務処理など、すべてActの実例です。

### 7.2.2 Entity (実体)

Entityクラスは、RIM, D-MIM, R-MIMでは緑色で示されます。

**例** (図6)



この例では、**Person**はclassコードとして“PSN”を持っています。すべてのEntityはclassCode属性を持っています。classCode属性は、HL7で言うところの構造的属性の1つであり、9.2節（48ページ〜）で解説します。このPersonはまた、**determiner**（限定詞）コードとして“INSTANCE”も持っています。1つのインスタンスか、あるいは定量化された値かを区別するために、現在のところ、3種類のdeterminerコードが使われています。

訳注：determiner（限定詞）とは、英文法で名詞の前に置かれる、冠詞、指示代名詞、不定代名詞などのことです。determinerコードを用いることにより、例えば、A型の血液（一般的な種類）、A型の血液3単位（具体的なもの）、ロット番号xxxのA型の血液（個別化された特定の実在）、などを区別して正確に記述することができるようになります。

例2はAgent（代表者）という役割をある MedicalPractitioner（臨床医）という人が果たしていることを示しています。人を表す2つのクラス（例1のPersonと例2のMedicalPractitioner）は、どちらも同一のRIMクラスに基づいています。しかしながら、これらは31ページの図4で別々に表記されています。これは、異なる役割の人を表現する際に、PersonクラスのRIM属性の異なる部分集合を利用できるようにするためです。

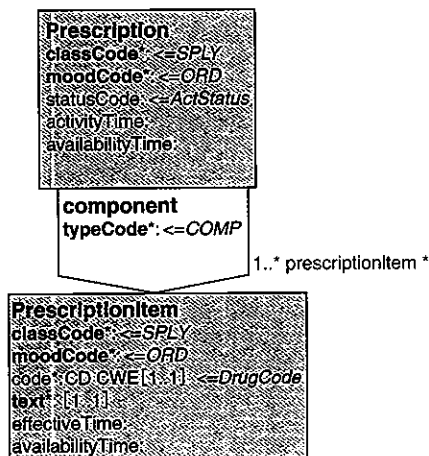
例2はまた、Agentの役割が代表する CommunityMedicalPractice（地域診療所）も示しています。代表された（represented）Entityというのは、それを代表するEntityがそこである特定のRoleを果たしているという状況を意味しています。この例の場合、代表された組織は、MedicalPractitionerが診療している医院、あるいは他の医療施設を示すことによって、Agentの役割をより明確に特定することを可能にしています。

**定義：Role（役割）**は、あるEntity（実体）がさまざまなAct（行為）（あるいは潜在的なAct）に参加（Participation）する際に、そのEntityが占める地位、果たす責任、演じる役割を表現します。

### 7.2.5 ActRelationship (行為間関係)

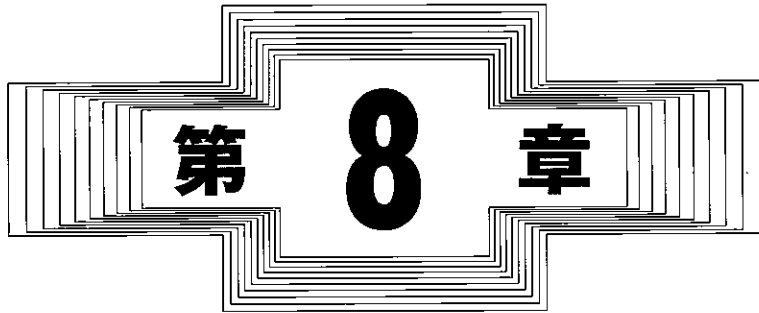
ActRelationship は、2つの Act の間の関係を表現します。ActRelationship クラスは、RIM, D-MIM, R-MIM ではピンクで示されます。

#### 例 (図10)



この例は、最もシンプルな ActRelationship が示されています。1つの Act, すなわち PrescriptionItem (処方項目) が、単に別の Act である Prescription (処方) の構成要素であることが、ActRelationship である component (コンポーネント) を用いて示されています。

訳注: ActRelationship は非常に強力なクラスです。typeCode の COMP (has component) を指定すれば、この例のような処方の組み合わせ、セット検査、処置の手順など、さまざまな複雑な構造を記述することができます。また、typeCode に RSON (has reason) を指定すれば診断や治療の根拠を表現でき、OUTC (has outcome) を指定すれば行為間の因果関係を表現することもできます。COST (has cost) を用いれば、診療報酬請求や原価計算が行えます。この typeCode としては、現在 76 のコード値が定義されています。

A decorative frame consisting of multiple concentric, stepped rectangular outlines. The frame is centered and contains the chapter title.

**第 8 章**

**R-MIMおよびD-MIMの  
さらなる詳細**



**More R-MIM and D-MIM Details**

の種類については、15章(81ページ〜)に記載されています。

R-MIM中に現れるすべてのクラスは、それが共通のRIMクラスから派生しており、そのRIMクラスに準拠していることを示すため、クローンクラス(**cloned class**)と呼ばれます。1つのR-MIMが、同一のRIMクラスからクローン化(複製)されたクローンクラスをいくつか含んでいることもあります。これらのクローンクラスは、それぞれ別個の一連の制約を有しているでしょう。

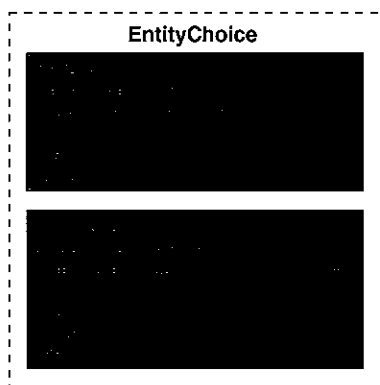
例えば、“Person” Entity(実体)は、Patient(患者)やPractitioner(臨床医)のRole(役割)を演じることができます。Personの属性の中には、“Patient” Roleを演じているEntityの観点からすれば伝送する必要があるが、“Practitioner” Roleを演じているEntityの観点からは重要でない属性もいくつかあるかもしれません。そこで、メッセージとして伝送されるデータは、“Person”の属性および関連の中で、妥当であり、かつ適切であるものだけに制約することができます。

クローン化という手法によって、RIMとして定義する必要のあるクラスの数を抑えることができ、その結果、小規模で安定したRIMの仕様となります。

ある1つのR-MIMにおいては、クローンクラスはすべて、それぞれ固有の名前を持たなければなりません。名前は、その特定の領域におけるクローンクラスの使われ方を反映しなければなりません。同じクローン名は、異なる領域、あるいは異なるD-MIM、その構成要素であるR-MIMを有する下位領域であれば、たとえ制約が同じでない場合でも再利用することができます。

これらの技法はすべて、V3方法論における中核定義が、すべての領域を通して首尾一貫して使用されることを確実にするための技術です。新しく開発されるメッセージはすべて、これまでにすでに定義されているものに基づくことになるので、新しいメッセージを設計し、実装することがより簡単になるでしょう。HL7 V3のメッセージ開発の全過程を通じて、こうした利点は明らかです。名前によってさらなる柔軟性を提供するために、Act, Role, およびEntityクラスにはクローン化に際し

## 例 (図 12)



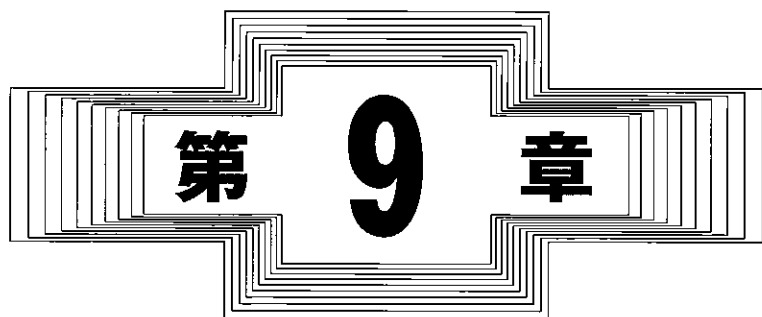
これは Patient の定義の一部を示した例ですが、1 個体の人 (person) と 1 個体の動物 (animal) を選択肢とする choice を提供することによって、この定義の範囲が獣医科へ適用できるように拡張されています。

### 8.4 成果物の識別 (Identifying Artifacts)

HL7 V3 方法論においては、識別する必要がある構造物はすべて、成果物 (artifact) として知られています。成果物にはストーリーボード、ストーリーボードの中の叙述的記述、アプリケーションロール、トリガイベント、インタラクション、D-MIM、R-MIM、HMD、およびメッセージ型が含まれます。HL7 はすべての成果物に対して、16桁のコードを使用して固有な識別子を割り当てます。以下に示す表 1 は、これらの識別子がどのように構成されているかを示しています。

すべての識別子は SSDD\_AAnnnnnRRVV という形式で表現されており、詳細は以下の表 1 に示すとおりです。



A decorative frame consisting of multiple concentric, stepped rectangular outlines that form a cross-like shape. The frame is centered on the page and contains the chapter title.

# 第 9 章

## 属 性



## Attributes

属性の主な特性は以下のとおりです。

- **必須 (Mandatory)** : 必須属性は、属性名を太字で表記します。必須属性は、そのクラスが使用されるすべての場面において、必ず存在しなければなりません。
- **要求適合 (Conformance)** : 属性名の後のアスタリスク (\*) は、すべてのメッセージのインスタンスにおいて、この属性に値が存在することが要求されている (required) ことを示しています (必須属性においてはこの指定は不要ですが、完全性のために含まれています)。要求適合以外の属性は選択可 (optional) です。RIMクラスで宣言されていて、R-MIMでは必要とされない属性は、詳細化の通常の過程においては削除されます。もし、何らかの理由でこれらの不要な属性を含めておく必要がある場合は、# 指定して、使用不許可 (not permitted) であることを示します。
- **ドメイン (Domain)** : ある属性インスタンスに関連付けられたボキャブラリドメインは、記号 "<=" によって指定されます。これは、コード自体が明示されているときはそのコードが、ボキャブラリドメインが明示されている場合は、その明示されたボキャブラリドメインに含まれるすべてのコードが、使用可能であることを意味します。
- **データ型 (Data Type)** : 属性データ型は、定義された属性すべてに対して明示する必要があります (例えば、"CE" は Coded with Equivalentents (同義語を伴うコード化) データ型をその属性に割り当てます)。クローンクラスで指定するデータ型は、その属性のRIMデータ型と同じか、あるいはそれに妥当な制約を課したものでなければなりません。HL7 V3で使用されている主要なデータ型については、12章 (61ページ~) で概説しています。本書のVisioツールの図には、データ型はRIMからの変更がない場合は表示していません。そのような変更は、詳細化の過程で起こることがあり、15.4節 (83ページ~) で議論しています。
- **コード化強度 (Coding Strength)** : コード化強度は、CWE-拡張可能コード化 (coded with extension : 適当なコードが存在しない

**typeCode** : Participation, ActRelationship, あるいは RoleLink の一般的な特徴を示します。

**moodCode** : ある Act がすでに発生した出来事を表しているのか、あるいはオーダされたものを表しているのか、あるいは将来行うことの計画を表しているのか、を示します。

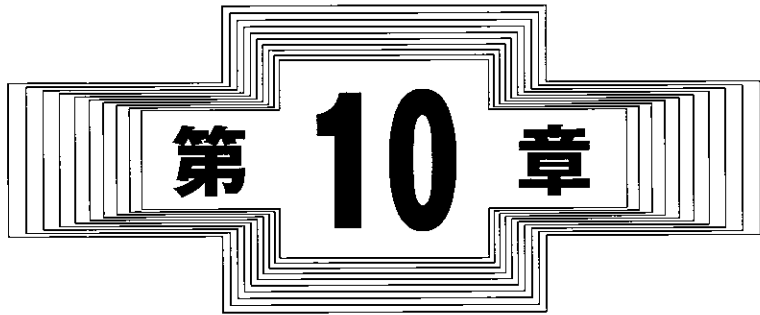
**determinerCode** : ある Entity が実在の識別可能なインスタンスを表しているのか、その実在の一般的な種類を表しているのか、を示します。例えば、“Kari Kidd 医師”は Person の実在するインスタンスですが、“ネコ”は、LivingSubject の一般的な種類です。

### 9.3 よく用いられる属性

特に注意を払っておくべき、よく用いられる一連の属性を表2に挙げておきます。

表2 代表的な属性

属性	説明
code	現実世界の Entity, Act, Role, Participation あるいは Act Relationship あるいは RoleLink の詳細な特性を表現するための特定のコード。この属性にはしばしば外部で定義されたコード体系が適用されます。
statusCode	表現されている Act の状態、すなわち、有効 (active)、完了 (completed)、保留 (suspended)、中止 (aborted) を特定します。statusCode は、構造的属性の moodCode とははっきりと異なることに注意してください。
id	固有の識別子
text	コンピュータが処理するのではなく、人間が読むことを意図した自由テキスト表現
name	構造化された形式、あるいは構造化されていない形式で表現される、人、組織あるいはその他の Entity の名前
address	構造化された形式、あるいは構造化されていない形式で表現される住所
telecom	電話番号、ファックス番号、ポケットベル番号、あるいは電子メールアドレス
time	時刻と日付、あるいは日時の間言及する、さまざまな time 属性があります。これらには、effectiveTime, activityTime, validTime, existenceTime, availabilityTime が含まれます。

A decorative frame consisting of multiple concentric, stepped rectangular lines that form a cross-like shape, enclosing the chapter title.

第 10 章

ボキャブラリ



Vocabularies