

医療におけるデジタル画像と通信 (DICOM)

パート 19: アプリケーションホスティング

発行元

北米電子機器工業会
1300 N. 17th Street
Rosslyn, Virginia 22209 USA

© 著作権2011 北米電子機器工業会。他の言語への翻訳を含むすべての権利が国際著作権条約、リテラシーと芸術的作品の保護のためのベルン協定および国際・全米著作権協定の下で留保されている。

Disclaimer 免責事項

DICOM is the worldwide Standard for medical imaging and related information. It is published and copyright by the National Electrical Manufacturers Association (NEMA). The normative DICOM Standard is published in English, and is available free on the official website at <http://dicom.nema.org/standard.html>.

This document is a translation prepared by the Japan Medical Imaging and Radiological Systems Industries Association (JIRA) under agreement with NEMA, with the intention to help Japanese readers understand the DICOM Standard more readily.

This translation represents a “best effort”; however, differences in meaning may exist between this translation and the normative DICOM Standard. Further, the DICOM Standard is under continuous maintenance and extension, so readers should expect that there are changes that are not reflected in this translation.

In the event of any difference between this translation and the DICOM Standard published in English by NEMA, the English version is normative and takes precedence.

Implementations shall claim conformance to the normative DICOM Standard. Users are advised to obtain the most current documents of the DICOM Standard directly from the official website.

DICOM は医用画像と関連する情報に関する国際標準規格です。DICOM 規格は米国電機工業会 (NEMA) が発行し著作権を有します。DICOM 規格の規范文書は英語で出版され、公式サイト <http://dicom.nema.org/standard.html> から無償でダウンロードが可能です。

この文書は日本語を好む読者が DICOM 規格をより容易に理解するための手助けを意図して、NEMA の許可を得て一般社団法人日本画像医療システム工業会 (JIRA) が提供する翻訳です。

この翻訳は最善の努力を以て提供されていますが、この翻訳と規範 DICOM 規格の間に意味の違いが存在するかもしれません。更に、DICOM 規格は継続的な保守と拡張が施されているので、読者はこの翻訳に反映されていない変更が存在することに留意する必要があります。

この翻訳と NEMA が発行する英語版の DICOM 規格との間に差が生じた場合は、英語版が規範であり優先します。

実装は規範 DICOM 規格への適合性を宣言しなければなりません。使用者は DICOM 規格の最新の文書を公式サイトから直接入手することが要望されます。

通知および免責条項

この出版物での情報は、開発当時は、文書の開発および承認に従事していた人のコンセンサスによって技術的に正常であると考えられた。コンセンサスは、この文書の開発に参加するすべての人による満場一致を必ずしも意味しない。

NEMA規格および指針出版物は、自発的なコンセンサス規格開発プロセスを通じて開発されている。本書もその一つである。このプロセスではボランティアを集め、この出版物の対象となるトピックに関心をもつ人の見解を求める。**NEMA**はプロセスを処理し、コンセンサスの開発での公平を促進する規則を確立するが、文書の執筆はしない。また、**NEMA**は、規格と指針出版物に含まれる情報の正確さ若しくは完全性または判断の健全性を、独立して試験したり、評価したり、確認したりしない。

NEMAは、特別、間接、必然か補償かにかかわらず、直接的または間接的にこの出版物、この文書の使用、適用または依存に起因する身体傷害、財産または他の損害に対し免責とする。**NEMA**は、明示か黙示かを問わず、ここに出版された情報の正確さと完全性について免責とし保証はしない。またこの文書中の情報が読者の特定の目的またはニーズを満たすことを免責とし保証はしない。**NEMA**は、個々のメーカーまたは販売業者の製品または役務の性能を、この規格またはガイドにより保証したりしない。

この文書を出版し利用可能にする際に、**NEMA**は、個人または組織のために、またはそれらを代表して専門的その他の役務を与えることを試みていない。また**NEMA**は個人または組織が他の者に対し負う義務を行うものではない。この文書を使用する人は誰でも、自分自身の判断に頼るべきである。または、適切な場合、所定の状況での合理的な行為を決定する際に有能な専門家に対し助言を求めるべきである。この出版物の対象のトピックについての情報および他の規格は、他の情報源から入手できることがある。この出版物の対象でない追加の見解または情報を求めて、ユーザは他の情報源を調べる必要がある。

NEMAはこの文書の内容への適合を監視または強制する権限を持っていない。**NEMA**は安全または健康の目的のために、製品、設計または設置を認証したり、試験したり、検査したりしない。健康または安全関連の情報への適合の認証または他の言明は、いかなるものにも**NEMA**は免責とし、その言明を認証し実行した者が全責任を負う。

目次

目次.....	3
まえがき	6
1 適用範囲および応用分野	7
2 引用規格	9
3 定義	9
3.1 プレゼンテーションサービス定義	10
3.2 XML INFOSET定義	10
3.3 DICOM序論および概観定義	10
3.4 DICOM情報オブジェクト定義	10
3.5 DICOMデータ構造および符号化	10
3.6 コードおよび管理された用語定義	11
3.7 アプリケーションホスティング定義	11
4 記号および略語	11
5 規約	12
6 アプリケーションホスティング概観	12
7 ホスト型アプリケーションライフサイクル	14
7.1 初期化	14
7.2 状態	16
8 インタフェース	18
8.1 アプリケーションインタフェース	20
8.1.1 getState(): 状態	20
8.1.2 setState(newState : State) : ブール	20
8.1.3 bringToFront(requestedScreenArea : Rectangle) : ブール	20
8.2 ホストインタフェース	21
8.2.1 generateUID(): UID	21
8.2.2 getAvailableScreen(appPreferredScreen: 長方形): 長方形	21
8.2.3 getOutputLocation(preferredProtocols: ArrayOfString) : 文字列	21
8.2.4 notifyStateChanged(state : State) : void	22
8.2.5 notifyStatus(status : Status) : void	22
8.3 DataExchange Interface	22
8.3.1 notifyDataAvailable(data : AvailableData, lastData : boolean) : ブール	27
8.3.2 getData(objectUUIDs : ArrayOfUUID, acceptableTransferSyntaxUIDs : ArrayOfUUID, includeBulkData : boolean) : ArrayOfObjectLocator	27
8.3.3 getAsModels(objectUUIDs : ArrayOfUUID, classUID : UID, supportedInfosetTypes : ArrayOfMimeType) : ModelSetDescriptor	28
8.3.4 queryModel(models : ArrayOfUUID, xpaths : ArrayOfString) : ArrayOfQueryResult... ..	29
8.3.5 queryInfoset(models : ArrayOfUUID, xpaths : ArrayOfString) : ArrayOfQueryResultInfoset	29

8.3.6	releaseData(objectUUIDs : ArrayOfUUID): void.....	30
8.3.7	releaseModels(objectUUIDs : ArrayOfUUID): void	30
9	データ型および構造	30
9.1	ArrayOf[Type]	30
9.2	AvailableData	30
9.2.1	ObjectDescriptor	31
9.2.2	患者	31
9.2.3	研究	32
9.2.4	シリーズ	32
9.3	MimeType	32
9.4	ModelSetDescriptor	32
9.5	ObjectLocator	33
9.6	QueryResult	34
9.7	QueryResultInfoSet	34
9.8	長方形	34
9.9	状態	34
9.10	ステータス	35
9.10.1	StatusType	35
9.11	UID	35
9.12	UUID	35
9.13	XPathNode	35
9.14	XPathNodeInfoSet	36
9.15	XPathNodeType	36
10	データ交換モデル規約	36
10.1	符号化された用語	38
10.2	人名コンポーネント	39
附属書A	データ交換モデル	40
A.1	ネイティブのDICOMモデル	40
A.1.1	使用法	40
A.1.2	識別	40
A.1.3	サポート	40
A.1.4	情報モデル	40
A.1.5	説明	42
A.1.6	スキーマ	46
A.1.7	例	47
A.2	抽象の多次元画像モデル	48
A.2.1	使用法	48
A.2.2	識別	49
A.2.3	サポート	49
A.2.4	情報モデル	49
A.2.5	説明	50
A.2.6	スキーマ	57
A.2.7	例	60

附属書B	インタフェース定義	64
B.1	アプリケーションインタフェースー VERSION 20100825.....	64
B.1.1	インタフェースのWSDL定義.....	64
B.1.2	使用されたデータフレームの定義.....	69
B.2	ホストインタフェースー VERSION 20100825.....	77
B.2.1	インタフェースのWSDL定義.....	77
B.2.2	使用されたデータ構造の定義.....	83

まえがき

このDICOM規格は、DICOM規格委員会の手続きに従って開発された。

DICOM規格は次の文書の中で確立された指針を使用して、複数のパートから成る文書として組立てられる：

ISO/IEC専門業務指針、1989年パート3：国際規格の起草および様式

PS3.1はこの規格の現在のパートの基本的な参照として使用されるのがよい。

1 適用範囲および応用分野

DICOM規格のこのパートは、2つのソフトウェアアプリケーション間のインタフェースを定義する。第1のアプリケーション(ホスティングシステム)は、1組の画像および関連するデータのようなデータに第2のアプリケーションを提供する。第2のアプリケーション(ホスト型アプリケーション)は、そのデータを分析し、その分析結果を潜在的に戻す。例えば、画像および/または構造化した報告書の別の組の形にして、第1のアプリケーションに戻す。そのようなアプリケーション業務プログラム・インタフェース(API)は、DICOM規格の他のパートとは適用範囲が異なる。同系統上のソフトウェアコンポーネント間のデータ交換を標準化し、異系統間のデータ交換をしない。その標準化されたインタフェースに書き出されたホスト型アプリケーション・プログラムは、ホスティング・システムに「プラグイン」できる(図1-1を参照)。ソフトウェアのアドオンまたは「プラグイン」についての概念は、コンピュータの世界においては全く一般的であり、ウェブブラウザ、メディアプレイヤー、グラフ式のエディタ、出版プログラムなどの能力を拡張するために成功裡に使用されてきた。ホスティングシステム作成者は、種々様々のアドオンホスト型アプリケーションをサポートするために、標準化されたAPIを一度だけ作成する必要がある。

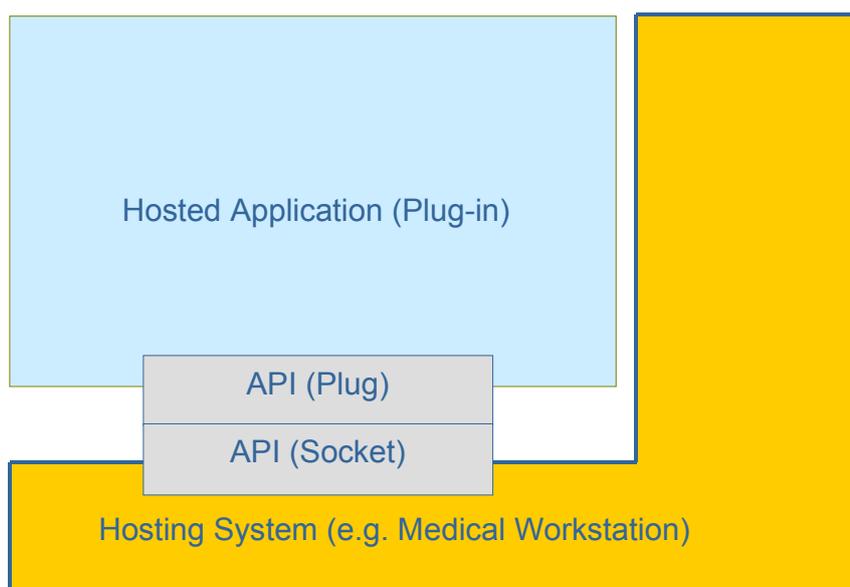


図1-1 ホスト型アプリケーションとホスティングシステムとの間のインタフェース

従来の「プラグインの」モデルでは、「プラグイン」は、特定のホスト・システム(例えばウェブ閲覧プログラム)に専用であり、他のホスト・システム(例えば他のウェブ閲覧プログラム)の下では実行しないかもしれない。PS3.19は、任意のホスティングシステムによって実装される標準化されたAPIを定義する。標準化されたAPIに「プラグインの」ホスト型アプリケーションに書き出されれば、その標準化されたAPIを実装するすべてのホスティングシステム上で実行することができる(図1-2を参照)。

The same Hosted Application can run on any platform (Hosting System) that supports the API.

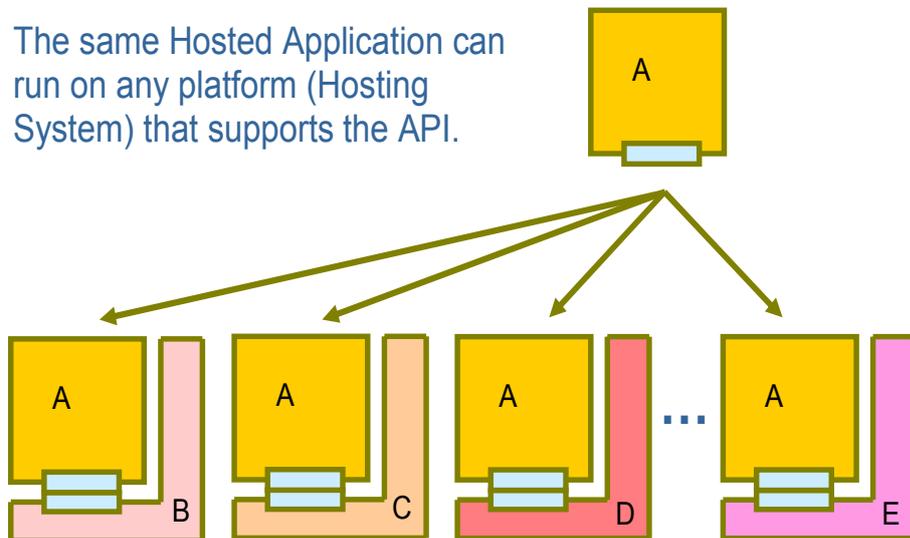


図1-2 ホスト型アプリケーションアーキテクチャによるプラットフォーム独立の概略図

APIのための設計目標および仮定は次のものを含んでいる：

- 言語独立—API は、任意の共通のプログラミング言語で書かれたプログラムがそれを利用できる方法で定義される。
- プラットフォーム独立— API は、それが、特定の計算プラットフォームまたはオペレーティングシステムに依存しない方法で定義される。
- 拡張可能である— API は後方への互換性をもつ方法で拡張できる。古いアプリケーションは、新しい拡張において働く。その一方で拡張に気づいている新しいアプリケーションは、機能性のより豊富なセットへアクセスできる。
- 保護された— API 設計は、知的所有権を保護するメカニズムの事後の追加分と一致している。このメカニズムは、適切な許可およびライセンスの保証を整備する。API は通常のライセンス制に干渉するべきでない。
- 安全である— ホスト型アプリケーションがホスティングシステム上のデータへアクセスすれば、ホスティングシステムによる API を介してコントロールされる。ホスティングシステムはホスト型アプリケーションにデータを供給するシステムであるから、アクセス管理および監査ロギングに責任を持つ。
- 既存の技術の活用—API 定義は通常使用される既存の技術を、できるだけ利用し、新しい方法論を定義しない。
- 同時開始— ホスティングシステムは、同じ、または異なるホスト型アプリケーションのいくつかのインスタンスを、同時に始めることができるであろう。

- 分散実行 — API はローカルの実行のために設計されているが、それは遠隔の実行を妨げない。アプリケーションがホストと異なるシステム上で実行されている場合である。

PS3.19は、ホスティング・システムおよびホスト型アプリケーションの間で、相互作用とアプリケーションプログラムインタフェース(API)の両方を指定する。PS3.19は、このパートで定義されるAPIによって使用されるデータモデルを定義する規定附属書および参考附属書も含んでいる。

APIでは、ワークフロー管理を直接には扱わない。ワークフロー管理は他のDICOMサービスの対象である。

2 引用規格

下記の規格はこのテキスト中に引用されることによって、この規格の条文を構成する。出版の時に、下記の版は有効であった。規格はすべて改正されることがあるので、この規格に基づいて契約をする当事者は、下記の規格の最新版を適用できるかを調査することが望ましい。

IETF RFC 2045,2046,2048	MIME 多目的インターネットメール拡張
IETF RFC 2396	ユニフォームリソース識別子(URI): 総括的なシンタックス
IETF RFC 3240	アプリケーション/dicom MIME サブタイプ登録
ISO 8822:1988	情報処理システム--開放型システム間相互接続 - 接続志向プレゼンテーションサービス定義
ISO/IEC 19757	DSDL 文書スキーマ定義言語(DSDL)
ITU-T勧告 X.667	UUID (さらに IETF RFC 4122)
W3C勧告	ウェブサービス記述言語サービス(WSDL) 1.1

注: WSDL W3C勧告は次のサイトにある:
<http://www.w3.org/TR/wsd/>

W3C勧告	XMLパス言語 (XPath) 2.0
-------	---------------------

注: XPath W3C勧告は次のサイトにある:
<http://www.w3.org/TR/2007/REC-xpath20-20070123/>

W3C勧告	XML 情報集合
-------	----------

注: XML情報集合W3C勧告は次のサイトにある:
<http://www.w3.org/TR/xml-infoset/>

3 定義

この規格のために、次の定義を適用する。

3.1 プレゼンテーションサービス定義

規格のこのパートはISO 8822の中で定義された次の用語を使用する：

- a. 転送構文
- b. 転送構文名

3.2 XML INFOSET定義

規格のこのパートはW3C勧告XML情報集合に定義された次の用語を使用する：

- a. InfosetまたはXML Infoset
- b. 要素またはXML要素
- c. 属性またはXML属性

注： 1 XML属性の概念は、DICOM属性の概念とは全く異なる。
2 同様の名前を持つDICOM用語との混乱を回避するために、DICOM規格のテキストは、これらのXML Infoset概念を参照する時、XML要素およびXML属性を使用する。要素または属性の前に用語XMLをつけない場合は、一般にXML Infoset概念の代りにDICOM概念を参照する。

3.3 DICOM序論および概観定義

規格のこのパートはPS3.1に定義された次の用語を使用する：

- a. 属性

3.4 DICOM情報オブジェクト定義

規格のこのパートはPS3.3に定義された次の用語を使用する：

- a. 属性タグ

3.5 DICOMデータ構造および符号化

規格のこのパートは、PS3.5に定義された次の用語を使用する：

- a. データ要素
- b. データ要素タグ
- c. データ要素タイプ
- d. データセット
- e. 定義された用語
- f. 数値
- g. アイテムのシーケンス
- h. 一意的識別子(UID)
- i. 値多重性(VM)
- j. 値表現(VR)

3.6 コードおよび管理された用語定義

規格のこのパートはPS3.16に定義された次の用語を使用する：

- a. 基線コンテキストグループ識別子(BCID)
- b. 定義されたコンテキストグループ識別子(DCID)
- c. コンテキストグループ
- d. コンテキストグループバージョン
- e. コンテキストID(CID)
- f. マッピングリソース
- g. DICOMコンテンツマッピングリソース(DCMR)
- h. 値セット
- i. コード体系

3.7 アプリケーションホスティング定義

次の定義が、規格のこの一部で一般に使用される：

アプリケーションプログラミングインタフェース： アプリケーションおよびホスティング・システムが互いに通信するために使用する1セットのインタフェース方法。

ホスト型アプリケーション： ホスティングシステムによって始められるコントロールされたアプリケーション。ホスト型アプリケーションは、ホスティングシステムによって提示されたサービスを利用できる。

ホスティングシステム： ホスト型アプリケーションを開始しコントロールするアプリケーション。ホスティングシステムは、様々なサービス、例えばDICOMオブジェクト検索および記憶をホスト型アプリケーションのため提供する。ホスティングシステムは、ホスト型アプリケーションが実行し外的環境と対話するインフラストラクチャーを提供する。これはネットワークアクセス、データベースおよびセキュリティを含んでいる。

4 記号および略語

次の記号および略語が規格のこのパートで使用される。

ACR	米国放射線学会
ASCII	米国標準情報交換用符号(米国標準情報交換用の記号)
ANSI	米国規格協会
API	アプリケーションプログラミングインタフェース
BCID	基線コンテキストグループ識別子
CID	コンテキストID
DCID	定義されたコンテキストグループ識別子

DCMR	DICOMコンテンツマッピングリソース
DICOM	医用デジタル画像と通信に関する標準規格
DSDL	文書スキーマ定義言語
IEC	国際電気標準会議
IOD	情報オブジェクト定義
IANA	インターネット番号割当機関
ISO	国際標準化機構
LUT	ルックアップテーブル
MIME	多目的インターネットメール拡張
NEMA	米国電気機器製造工業会
OID	オブジェクト識別子(ISO 8824))
ROI	関心領域
SOP	サービスオブジェクトペア
SR	構造化報告書
UID	固有識別子
UUID	普遍的固有識別子 (ISO/IEC 11578)
URL/URI	ユニフォーム資源ロケータ/識別子
VM	値の多重度
VR	値表現
WSDL	ウェブサービス記述言語
XSD	XMLスキーマ定義
XML	拡張可能マークアップ言語
XPath	XMLパス言語

5 規約

セクション3の定義に列記された用語は、この文書の全体にわたって(英語の)大文字で表記する。

6 アプリケーションホスティング概観

このセクションは、APIの能力について記述し、オペレーションの順序の例を挙げて、このパートの以下の残りのセクションを要約する。

APIは、ホスティングシステムおよび1つ以上のホスト型アプリケーションによって共有される。

APIは、ハードウェアプラットフォーム、オペレーティングシステムおよびGUIにとらわれない。利用可能な場合、APIはGUIの中の要求するスペースをサポートする。APIはヘッドレスオペレーションをサポートする(つまり、GUIはない)。

APIは、プログラミング言語、プラットフォームおよび技術中立であるためにウェブサービス定義言語 (WSDL)を使用して定義される。APIは、言語独立を最大限にすると同時に、他方でウェブサービス技術を利用する効率への影響を最小限にするよう設計されている。インタフェースは、ネットワークファイルベースの相互作用モデルと、共有メモリ相互作用モデルの両方をサポートする。APIはマニュアル構成をサポートするが、ディレクトリをサポートしない。

APIは、DICOMデータセットおよび他のデータをホスト型アプリケーションに供給できる。またホスト型アプリケーションによって作成されたDICOMデータセットおよび他のデータを、漸増的にまたは完成時に受取できる。ホスト型アプリケーションは、ホスティングシステムによって提供されるデータ(例えば単一の属性、画素データの部分集合、など)への粒度の細かいアクセスをもち、またそのデータだけにアクセスする。APIはDICOM意味論を利用する。しかし、必ずしもDICOMネットワーク転送構文を利用しない。ホスティングシステムは、メカニズムをホスト型アプリケーションにUIDを生成するため供給する。

APIは、ホスティングシステムがホスト型アプリケーションのオペレーションを中止および/または取消をし、そしてユーザー・インターフェースコントロールを回復することを可能にする。APIは、ホスト型アプリケーションから状況情報をホスティングシステムへ返すことおよびホスト型アプリケーションの状態を追跡することをサポートする。

ホスティングシステムは、1つ以上のホスト型アプリケーションを開始するか接続する、ホスト型アプリケーションが成功裡に始まったことを確認する、次に、初期データオブジェクトをパスするメカニズムもっている。相互作用はすべてホスティングシステムで始まる。イベントの典型的なシーケンスは以下のとおりである：

1. ホスティングシステムは、ホスト固有の方法を使用するタスクとデータに適切なホスト型アプリケーションを識別し見つける。しばしば、希望のアプリケーションは、システムのユーザーによって選択されているか、またはワークリストエントリで識別される。
2. ホスティングシステムはアプリケーションを開始する。本質的「run」または「exec」コマンドを発行し、2つの間の相互通信を確立するためにホスト型アプリケーションが使用するパラメータをパスする。
3. ホスティングシステムは、ホスト型アプリケーションでの処理タスクを始めるためにその入力データをそれに通知するためにAPIを使用する。
4. ホスト型アプリケーションは、ホスティングシステムから入力データに関する情報(バルク画素データの位置を含む)を引き出すためにAPIを使用する。
5. ホスト型アプリケーションは、ファイルI/O、メモリマッピングまたはバルク画素データへのアクセスを獲得する他の適切な方法を使用できる。
6. ホスト型アプリケーションは、さらに、ホスティングシステムに処理のステータス、例えば進行、警告、遭遇したエラーを通知するためにAPIを使用できる。
7. ホスティングシステムは、ホスト型アプリケーションでの処理を中止するか取消するためにAPIを使用できる。
8. ホスティングシステムがホスト型アプリケーションでの処理を中止した場合、ホスティングシステムは、ホスト型アプリケーションに処理を再開するように命じるためにAPIを使用できる。

9. ホスト型アプリケーションは入力データを処理するとき、ホスティングシステムにそれらの存在を通知するために出力オブジェクトを作成し、APIを使用できる。
10. ホスティングシステムは、出力オブジェクトに関する情報(バルクデータの位置を含む)をホスト型アプリケーションから引き出すためにAPIを使用する。
11. ホスティングシステムは、ファイルI/O、メモリマッピングまたは出力バルクデータへのアクセスを獲得する他の適切な方法を、もし必要ならば、使用できる。
12. 一旦ホスティングシステムがホスト型アプリケーションから出力データを引き出したならば、ホスト型アプリケーションに次の処理タスクを待つように命じる（つまりホスト型アプリケーションにアイドルするように命じる）ために、ホスティングシステムはAPIを使用する。
13. ホスティングシステムが、ホスト型アプリケーションのために行うべき別のタスクをしている場合、セクション3で始めたイベントのシーケンスに続いて、そのタスクを始めるために、ホスティングシステムはAPIを使用できる。
14. ホスティングシステムがもはやホスト型アプリケーションを必要としない場合、ホスト型アプリケーションがエグジットすることを要求するためにAPIを使用できる。

セクション7は、ホスト型アプリケーションライフサイクルについて、更に詳しく記述する。

セクション8は、ホスティングシステムとホスト型アプリケーションの間の基礎インタフェースについて記述する。

セクション9は、インタフェースによって使用されるカスタムデータタイプおよびデータ構造について記述する。

セクション10は、モデルベースのインタフェースによって使用されるモデルの一般形態について、およびそれらのモデルを定義するのに使用される規約について記述する。この規格によって定義されたモデルは、附属書に記述されている。

7 ホスト型アプリケーションライフサイクル

7.1 初期化

ホスティングシステムは、インタフェースのために使用される終止点参照(URL)を指定するコマンド線パラメータを使用して実行命令またはその等価物(例えばC言語の中のexec機能)を出すことにより、ホスト型アプリケーションを初期化する。第1の終止点参照は、ホスティングシステムによって提供されるホストインタフェースにアクセスするために、ホスト型アプリケーションによって使用される。第2の終止点参照は、ホスト型アプリケーションによって提供されるアプリケーションインタフェースを、ホスティングシステムが探す場所である。ホストとアプリケーションのインタフェースは、セクション8に記述されている。もしコマンドプロンプトかシェルから出されれば、実行命令は、次のように現れるかもしれない:

```
app --hostURL url1 --applicationURL url2
```

- 注:
1. このスタートアップ方法論では、両方のURLを指定するのはホスティングシステムであり、ホスト型アプリケーションではない。ホスト型アプリケーションは、ホスティングシステムによってそれに割当てられたURLにおいて答えなければならない。
 2. ホスト型アプリケーション実装があり、そこではホスト型アプリケーションが遠隔にまたはアプリケーションサーバ上で実行する場合、その実装は、ホスティングシステムによって提供されるURLと、ホスト型アプリケーションが使用している実際のURLとの間において、適切に写像するためにスタートアップまたはプロキシのアプリケーションを利用するかもしれない。

図7.1-1は、ホスト型アプリケーション初期化のシーケンス線図を示す。一旦ホスト型アプリケーションが初期化しており、データを処理し始める準備ができれば、それはその状態をIDLEに変更し、ホスティングシステムに状態変更を通知する。それにはnotifyStateChanged()方法への呼出しを使用して、ホスト型アプリケーションの準備完了をホスティングシステムに通知する。

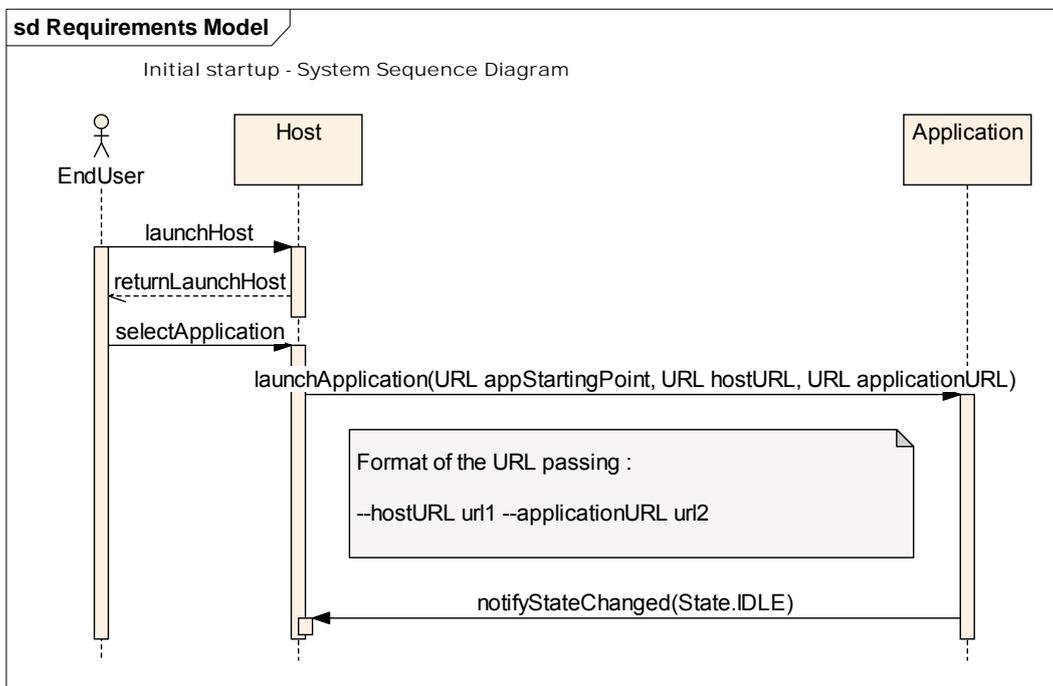


図7.1-1 ホスト型アプリケーション初期化シーケンス

7.2 状態

図7.2-1は、ホスト型アプリケーションのための状態線図を示す。次の状態が定義される:

状態	説明
IDLE	IDLE状態では、ホスト型アプリケーションは、ホスティングシステムからの新しいタスク割当を待っている。ホスト型アプリケーションが始まる場合、これは初期状態である。
INPROGRESS	ホスト型アプリケーションは、割当てられたタスクを実行している。
SUSPENDED	ホスト型アプリケーションは処理を停止しており、できるだけ多くの資源を解放している。他方で処理を再開できるようにまだ十分な状態を維持している。
COMPLETED	ホスト型アプリケーションは処理を終了しており、ホスティングシステムがアクセスし、かつホスト型アプリケーションからの出力データをすべて解放するのを待っている。
CANCELED	ホスト型アプリケーションは処理を停止しており、処理を再開する機会のない資源をすべて解放している。
EXIT	ホスト型アプリケーションの終了状態。

状態間の推移は次のとおりである:

状態	トリガー	新しい状態
開始しなかった。	ホスティングシステムはホスト型アプリケーションを始める (例えば, <code>run</code> , <code>exec</code>)。)	IDLE
IDLE	ホスティングシステムはアプリケーション <code>setState (EXIT)</code> を呼び出す。	EXIT
IDLE	ホスティングシステムはアプリケーション <code>setState (INPROGRESS)</code> を呼び出す。	INPROGRESS
INPROGRESS	ホスティングシステムはアプリケーション <code>setState (SUSPENDED)</code> を呼び出す。	SUSPENDED
INPROGRESS	ホスティングシステムはアプリケーション <code>setState (CANCELED)</code> を呼び出す。	CANCELED
INPROGRESS	ホスト型アプリケーションは、エラーが生じてこれ以上の処理はできないが、まだ、十分に稼働し恐らく別のタスクを始めることはできる。ホスト型アプリケーションは、このエラーを呼出しを通じて <code>notifyStatus()</code> に報告しなければならない。 <code>statusType</code> は <code>FATALERROR</code> であり、その後 <code>CANCELED</code> 状態に移る。	CANCELED
INPROGRESS	ホスト型アプリケーションはその処理を終了する。	COMPLETED
SUSPENDED	ホスティングシステムはアプリケーション <code>setState (INPROGRESS)</code> を呼び出す。	INPROGRESS
SUSPENDED	ホスト型アプリケーションは、エラーが生じて(例えばサスペンション中に)これ以上の処理はできないが、まだ、十分に稼働し恐らく別のタスクを始めることはできる。ホスト型アプリケーション	CANCELED

	は、このエラーを呼出しを通じてnotifyStatus()に報告しなければならない。statusTypeはFATALERRORであり、その後CANCELED状態に移る。	
SUSPENDED	ホスティングシステムはアプリケーションsetState (CANCELED)を呼び出す。	CANCELED
COMPLETED	ホスティングシステムはアプリケーションsetState (IDLE)を呼び出す。その前に、すべての適切な出力データをホスト型アプリケーションから捕獲する。	IDLE
CANCELED	ホスト型アプリケーションは資源をすべて解放し、次のタスクの準備ができる。	IDLE

ホスト型アプリケーションは、notifyStateChanged()方法と呼ぶことにより、ホスティングシステムにすべての状態推移について通知する。

注: ホスト型アプリケーションが、ホスティングシステムの出した状態変更要求に答えない場合、ホスティングシステムは、実装固有のやり方、例えばホスト型アプリケーションが実行しているプロセスを中断することで、ホスト型アプリケーションを「強制中断」するかもしれない。

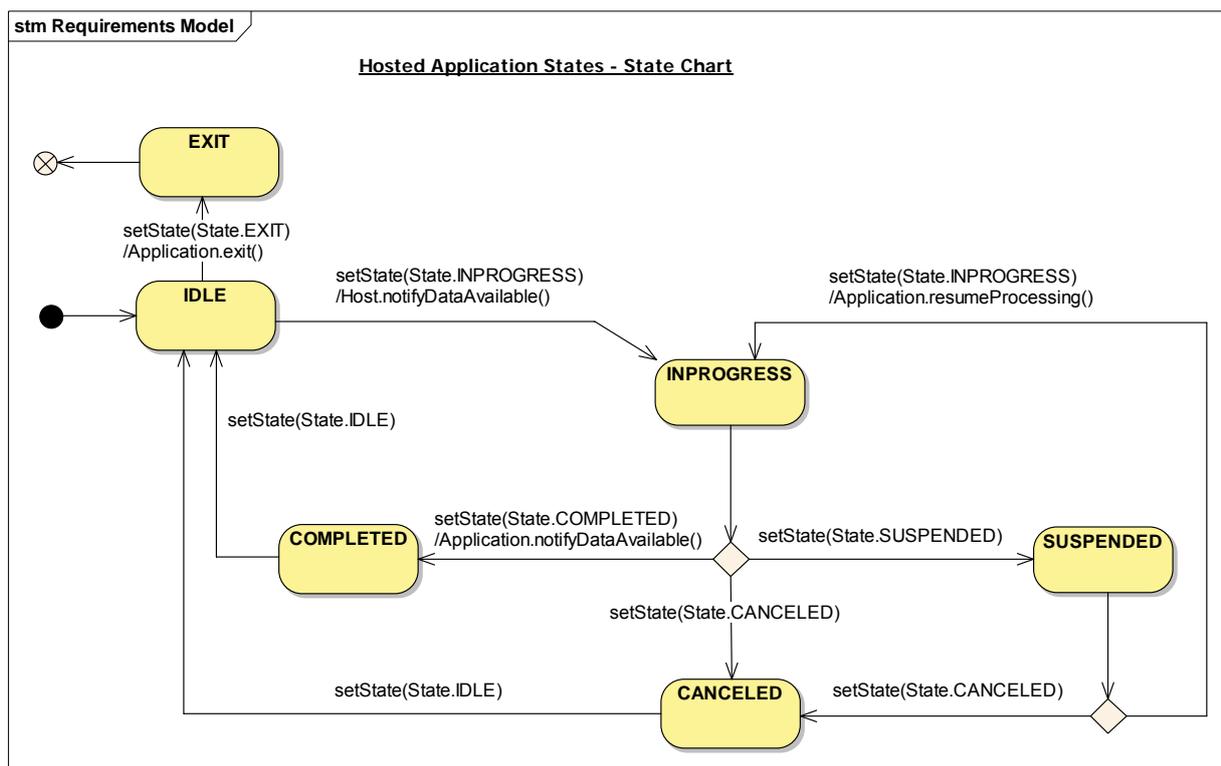


図7.2-1 ホスト型アプリケーションの状態図。

8 インタフェース

3つの基礎インタフェースがこのパートで定義されている。それを図8-1に示す。第1は、「アプリケーション」と称し、ホスト型アプリケーションを表し、ホスト型アプリケーションをコントロールするためにホスティングシステムによって利用される。第2は、「ホスト」と称し、ホスティングシステムを表し、サービスを要求し、ホスティングシステムに対して、ホスト型アプリケーションの実行中のイベントを通知するために、ホスト型アプリケーションによって利用される。第3は、「DataExchange」と称し、交換されるデータに関する情報を通信するために、ホスティングシステムおよびホスト型アプリケーションの両方によって使用されるインタフェースである。したがって、全体のホスト型アプリケーション(「ApplicationService」)実装は、「アプリケーション」および「DataExchange」基礎インタフェースの組み合わせから成り、その一方で全体のホスティングシステム(「HostService」)実装は、「ホスト」および「DataExchange」基礎インタフェースの組み合わせから成る。

インタフェースは、ウェブサービス記述言語(WSDL)を使用して、1セットの方法として定義される。実装者は、WSDL仕様中で終了点参照(つまり「サービス」XML要素の「ポート」XML要素内の「アドレス」XML要素内の「位置」XML属性)を、これらのインタフェースを利用するホスト型アプリケーションおよびホスティング・システムを展開させるために必要とされるように変更しなければならない。

注: インタフェースのメジャー(上位互換なし)バージョンは、インタフェースのWSDL仕様中の「定義」XML要素の、「名前」および「targetNamespace」XML属性の値に反映される。後方互換性のないインタフェースの変更は、「定義」XML要素の「名前」および「targetNamespace」XML属性に対して新しい値を利用する。

インタフェースのマイナー(上位互換あり)バージョンは、新しい「スキーマ」XML要素の「targetNamespace」XML属性の値に反映される。この場合、新しい入力または出力データタイプが、WSDL仕様書に定義される。かつ/または「portType」および「サービス」XML要素の「名前」XML属性の値に反映される。この場合、新しいメッセージおよびオペレーションがインタフェースのWSDL仕様書中で新しいサービスとして関連させられる。上位互換を維持するために、インタフェースのWSDL仕様中の既存の要素、メッセージおよびオペレーションの名前は、同じままである。

これらの方法は、パラメータ(XMLスキーマを使用して定義される)を通信するために、1セットの基本的なデータ型および更に複雑なデータ構造を利用する。この文書の後のセクションでは、インタフェースとデータ構造の更に詳細な記述、およびインタフェースがどのように使用されるかを例証するシーケンス線図が提供される。

このインタフェースを指定する実際のWSDLコードおよびXMLスキーマは、附属書Bに定義される。

- 注:
1. WSDLは、2つの協力するアプリケーション間のインタフェースを指定するプラットフォームおよびプログラミング言語独立手段である。アプリケーションは同じプログラミング言語で書かれる必要はない。
 2. インタフェースは、SOAP通信問題の報告を直接扱わない。問題が、ホスティングシステムとホスティングアプリケーションの間の通信において、WSDLインタフェース呼出しの実行の間に生じる場合、これは実装によって利用されたSOAPライブラリによって、例えば、例外として報告されるのがよい。

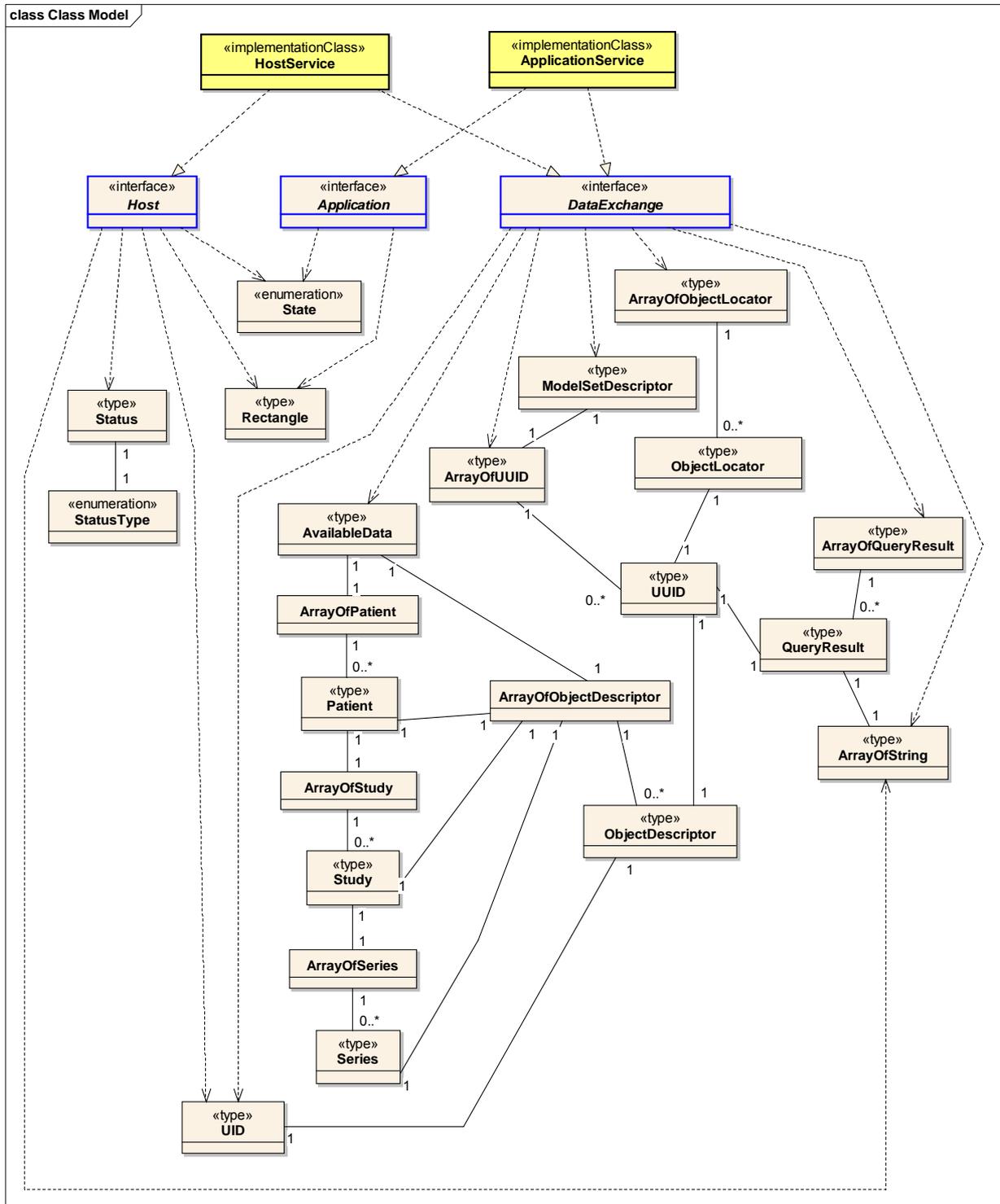


図8-1 ホスティングシステムとホスト型アプリケーションの間のインターフェースの図解

8.1 アプリケーションインタフェース

次のセクションは、アプリケーションインタフェースの方法について記述する。

8.1.1 getState(): 状態

ホスト型アプリケーションは、その現状を呼び出し元に返す。

この方法はいつ呼ばれるかもしれない。

- 注:
- 1 ホスティングシステムは、`notifyStateChanged()`方法呼出しによって報告されたホスト型アプリケーション状態変更をトラッキングする代わりとして、この方法を使用するかもしれない。
 2. ホスティングシステムは、ホスト型アプリケーションがまだ運転中かどうか決めるためにこの方法を使用するかもしれない (つまり、`notifyStateChanged()`方法をEXIT状態を用いて呼び出すことなく停止しなかった)。

8.1.2 setState(newState : State) : プール

ホスティングシステムは、ホスト型アプリケーションがnewStateに切り替わることを要求する。

ホスト型アプリケーションが要求を受取った場合、ホスト型アプリケーションは、方法からTRUEを返し、かつ要求された状態変更は、状態図の中で許可される。そうでなければ、方法はFALSEを返す。TRUEの返り値は、ホスト型アプリケーションの状態がnewStateに変わったことを示さない;それは、要求された状態変更が有効であり、すぐに変更されることを単に示す。一旦ホスト型アプリケーションが要求された状態に切り替われば、それは、ホストインタフェースのnotifyStateChanged()方法によってホスティングシステムに通知しなければならない。

- 注: 状態変化に対する非同期レスポンスは、アプリケーションにおいて潜在的に時間のかかる状態変化を待つ間に、ホスティングシステムをブロックするのを最小限にするように意図される。

ホスト型アプリケーションはどんなsetState()も無視して、ホスト型アプリケーションが既に要求された状態である場合、TRUEを返さなければならない(つまり、これは同じ状態を用いての再呼び出しである)。

ホスト型アプリケーションが前の要求を終えるに先立ち、異なる状態の第2のsetState()要求を受取る場合、ホスト型アプリケーションは異常終了するか、または前の要求を無視して最新の要求を処理し始めなければならない。

この方法はいつ呼ばれるかもしれない。しかしながら、要求された新しい状態が現状からの許容転移でない場合、どんな効果(FALSEのリターン以外の)も生じないかもしれない。

8.1.3 bringToFront(requestedScreenArea : Rectangle) : プール

この方法呼び出すことによって、ホスティングシステムは、ホスト型アプリケーションに、必要なステップを取り、そのGUIを最上部ウィンドウとして見えるようにし、かつ焦点を合わせることを依頼する。

できれば、ホスト型アプリケーションはrequestedScreenAreaにそれ自体をサイズ変更し位置変更しなければならない。requestedScreenAreaが見当たらないか無効の場合、ホスト型アプリケーションは、その現在のサイズおよび位置をスクリーン上に保持するかもしれない。

ホスト型アプリケーションが要求を受け取った場合、方法はTRUEを返し、それに作用する。そうでなければ、それはFALSEを返す。

ホスト型アプリケーションがIDLEかINPROGRESS状態である場合、ホスト型アプリケーションはこの方法に作用しなければならない。ホスト型アプリケーションは、ホスト型アプリケーションがIDLEかINPROGRESS状態でない場合に、この方法に作用することを要求されない。

GUIを持たないホスト型アプリケーション(例えばヘッドレス分析アプリケーション)は、可視および焦点合わせが意味を持たない場合、常にTRUEをこの方法から返さなければならない。

8.2 ホストインタフェース

次のセクションは、ホストインタフェースの方法について記述する。

8.2.1 generateUID(): UID

例えば、新しいデータオブジェクトおよび構造を作成するために、ホスト型アプリケーションが使用するかもしれない新しく作成されたDICOM UIDを返す。

この方法はいつ呼ばれるかもしれない。

8.2.2 getAvailableScreen(appPreferredScreen: 長方形): 長方形

ホスト型アプリケーションは、その好ましい画面サイズをappPreferredScreenパラメータ中で供給する。ホスティングシステムはヒントとしてこの情報を利用するかもしれないが、ホスティングシステムのGUIに最適のウィンドウ位置およびサイズを返すかもしれない。

この方法は、ホスティングシステムがホスト型アプリケーションに使ってほしいと好むウィンドウ位置およびサイズを返す。しかし、ホスト型アプリケーションがその情報に基づいて行動するようにという要求事項はない。

この方法はいつ呼ばれるかもしれない。

8.2.3 getOutputLocation(preferredProtocols: ArrayOfString) : 文字列

この方法は、ホスト型アプリケーションがホスティングシステムに返却するかもしれない出力を格納するために、ホスト型アプリケーションが使用するかもしれないURIを返す(例えば、getData()呼出しに応じて)。

ホスト型アプリケーションは、優先順に、データを格納するため使用できるプロトコルを示す。ホスト型アプリケーションは少なくともhttp: および the file: プロトコルの両方をサポートしなければならない。ホスティングシステムは、最も適切なプロトコルを選択する。その際に考慮するのはシステムまたはセキュリティおよび優先順である。ホスティングシステムは、選択されたプロトコルを使用して、リソースを設定し、ホスト型アプリケーションに返されるURIを生成する。

- 注:
1. the http: protocolを使用する場合、the file: protocolと比較して、時制限があるかもしれない。例えばseek, rewrite, deleteなどのa file: protocolで作用する機能には、the http: protocolで働かないものもある。ホスト型アプリケーションは、返された出力位置がthe http: protocolを使用する場合、それが連続する順で書き出せるのは一度だけということを仮定するのがよい。
 2. いずれかの認証情報がデータにアクセスするために必要な場合、この認証情報はURIに含まれているかもしれない。

ホスティングシステムは、ホスト型アプリケーションがIDLEまたはEXIT以外の任意の状態にある間、またはホスト型アプリケーションがホスティングシステムにURIを返すまでの時間内は、URIをアクティブにしておかなければならない(例えば、getData()呼出しに応じてホスティングシステムに返されたObjectLocatorの中で)。ホスト型アプリケーションがこのURIに送信するデータの配置は、ホスト型アプ

リケーションがIDLE状態へ移った後、またはホスト型アプリケーションがURIをホスティングシステムに返した後（例えば、`getData()`呼出しに応じてホスティングシステムに返されたObjectLocatorの中で）は、ホスティングシステムの責任である。ホスト型アプリケーションがIDLE状態へ移った後は、ホスティングシステムはURIをアクティブにしておく必要はない。

ホスト型アプリケーションがINPROGRESSまたはCOMPLETEDの状態にある場合、この方法呼び出すだけにしなければならない。

8.2.4 notifyStateChanged(state : State) : void

ホスト型アプリケーションが新しい状態に成功裡に移る度ごとに、この方法を起動しなければならない。新しい状態は状態パラメーター中で渡される。

注: ホスティング・システムはみなこのインタフェース呼出し方法を受理している必要があるが、それらは、例えばアプリケーション`getState()`方法を使用して状態をポーリングする方法などの他の方法で現在のアプリケーション状態を追跡するかもしれない。

8.2.5 notifyStatus(status : Status) : void

ホスト型アプリケーションは、この方法を起動し、ステータスパラメータ中の情報をパスすることによって、実行中に生じる顕著なイベントをホスティングシステムに通知するかもしれない。

注: ホスティングシステムは、デバッグを促進するために典型的にはこれらのイベントを記録するのである。それは、その分別で、情報をユーザーへ表示するかもしれない。

この方法はいつ呼ばれるかもしれない。

8.3 DataExchange Interface

インタフェースは、情報源と受信者との間で転送されているデータに関する情報を交換するため使用されるが、そのインタフェースはホスティングシステムおよびホスト型アプリケーションの両方に対して同様である。アプリケーションインタフェースの実装はさらにDataExchangeインタフェースを含まなければならない。ホストインタフェースの実装はさらにDataExchangeインタフェースも含まなければならない。言い換えれば、DataExchangeインタフェースは、ホスティングシステムおよびホスティングアプリケーションに関して対称である。

ホスティングシステムとホスト型アプリケーションの間で交換されているデータは、ファイルとして渡され得るか、または受信者によって照会されるモデルに記述されるかもしれない。

DICOMオブジェクトを解析できる受信者は、ファイルベースの方法を要求できる。図8.3-1の中のシーケンス線図は、ファイルベースの方法を使用する潜在的な交換の1つを図解する。

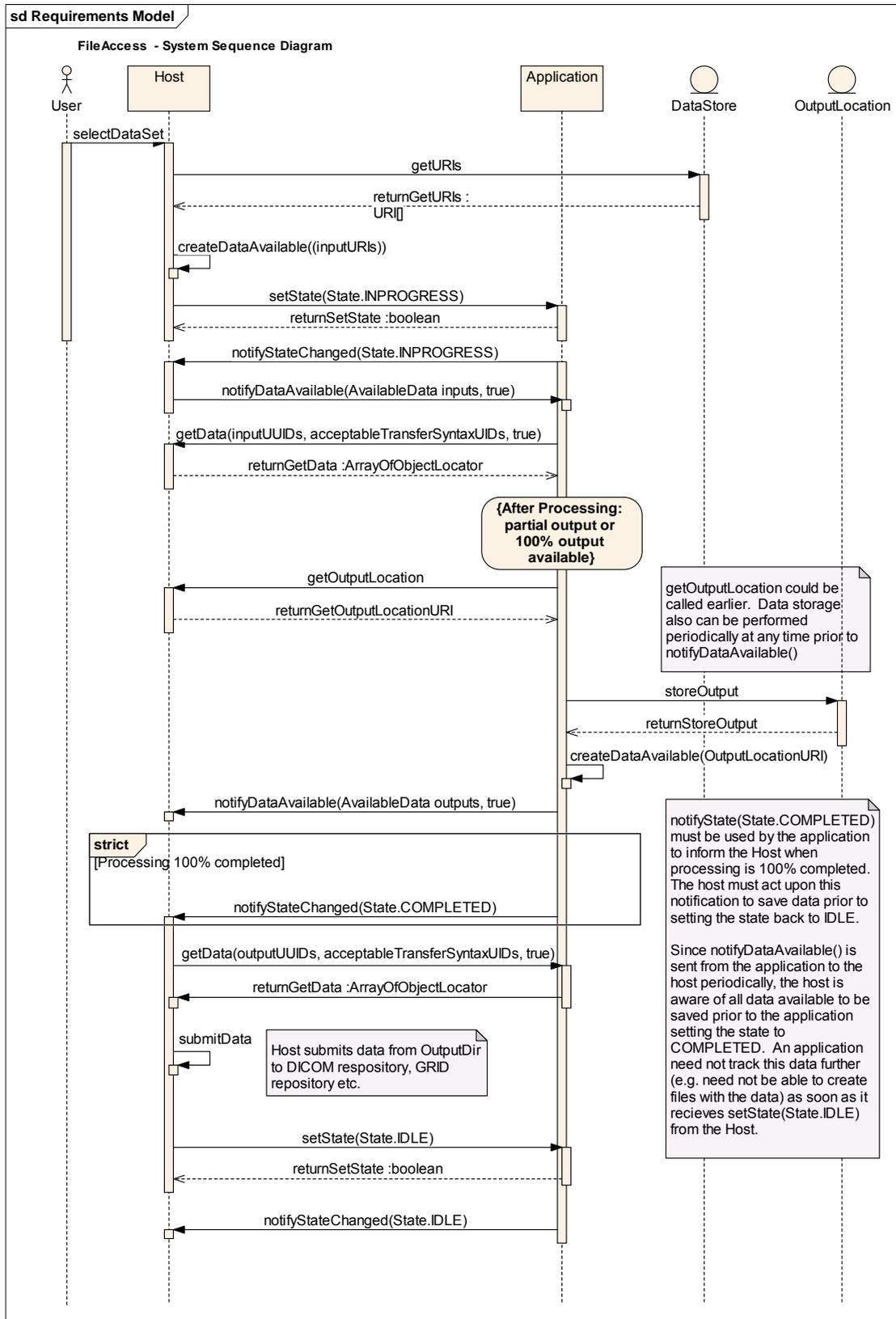


図8.3-1 ファイルベースのデータ交換シーケンスの例

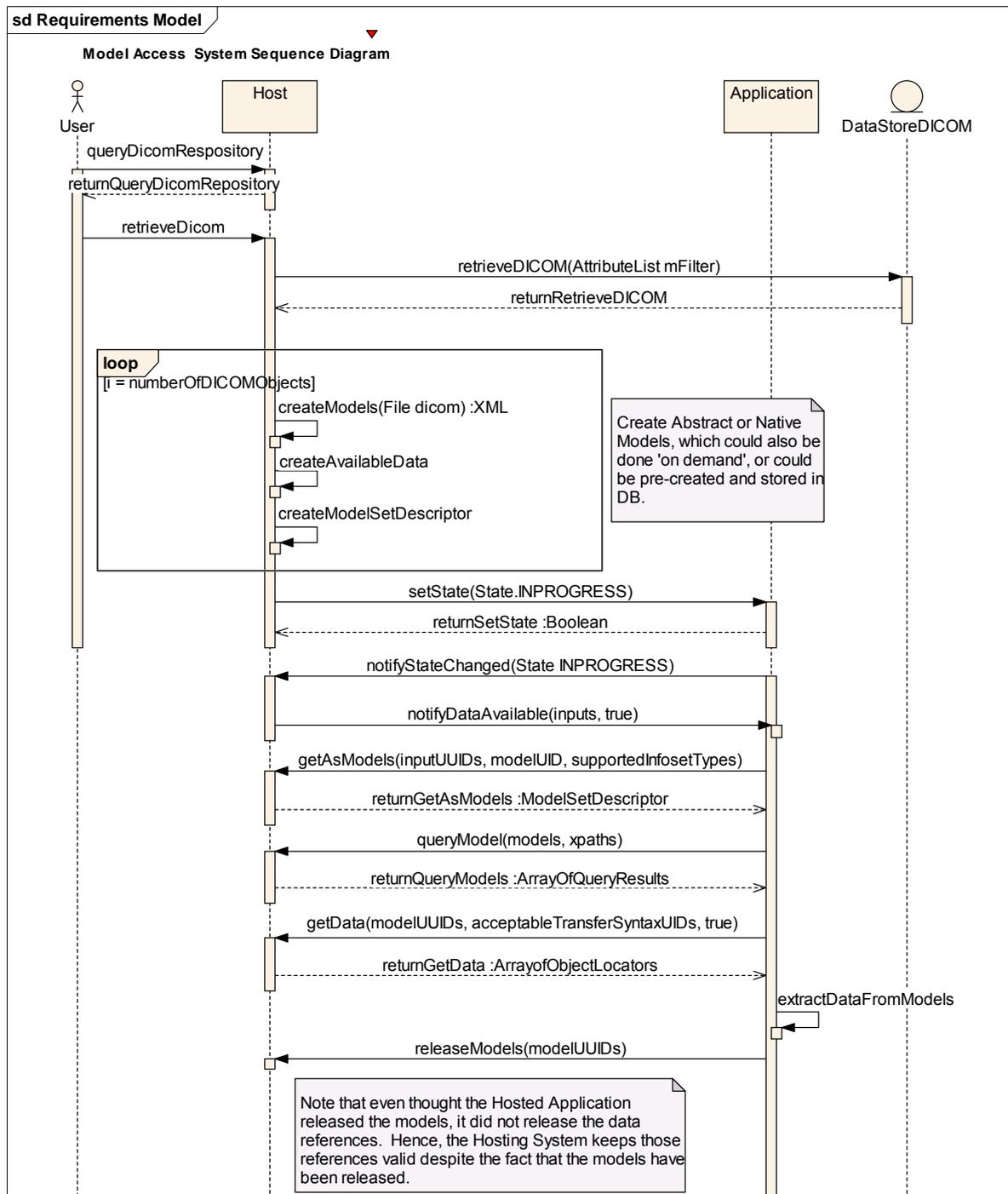
モデルベースの方法を使用する利点は、受信者がデータフォーマットを解析する方法を知る必要がないということである。その代り、受信者は、データをモデルから抽出するためにXML Infosetsを操作する一般に利用可能なツールを使用できる。

モデルベースのインタフェースは、様々なモデルで働くことができる。特定のモデルがUIDによって識別される。モデルはデータの抽象的概念か、またはネイティブフォーマットのモデルである。DICOM規格によって定義されたモデルは、附属書Aに記述されている。モデルは、たとえオリジナルデータがXMLの形で実際には表されない場合でも、XML Infosetsとして記述される。データを提供するソースは、モデルからオリジナルデータフォーマットへのマッピングを扱う。

抽象モデルによって、受信者はそのネイティブの形式は何であるかを考慮しなくてもデータを用いて仕事ができる。DICOM、TIFF、JPEG、NIFTIまたはAnalyzeなどの様々な画像形式からのデータが、抽象的な画像モデルに含まれる。その後、受信者は、たとえ受信者がデータがどのようにネイティブで表されたか知らなくても、データを用いて仕事ができる。抽象モデルは多数のObjectDescriptorsの中で参照が付けられたデータに由来したかもしれない（例えば、単一ボリュームに結合された多数のCTスライス）。

抽象モデルは、ネイティブの表現で利用可能な十分なデータを一般に含んでいない。例えば、DICOMデータに通常由来する抽象的な画像モデルは、「処理された」画素データおよびその空間の構成への参照を通常含むが、しかしモダリティに特有の属性の多くを含んでいないかもしれない。受信者がそのように詳細にアクセスできるように、抽象モデルの供給者は、ObjectDescriptorsへの言及を、抽象モデルが由来してきたUUIDの形で提供できる。受信者は、ソースObjectDescriptorsを通じて、オリジナルデータフォーマットの属性へアクセスしてもよい。

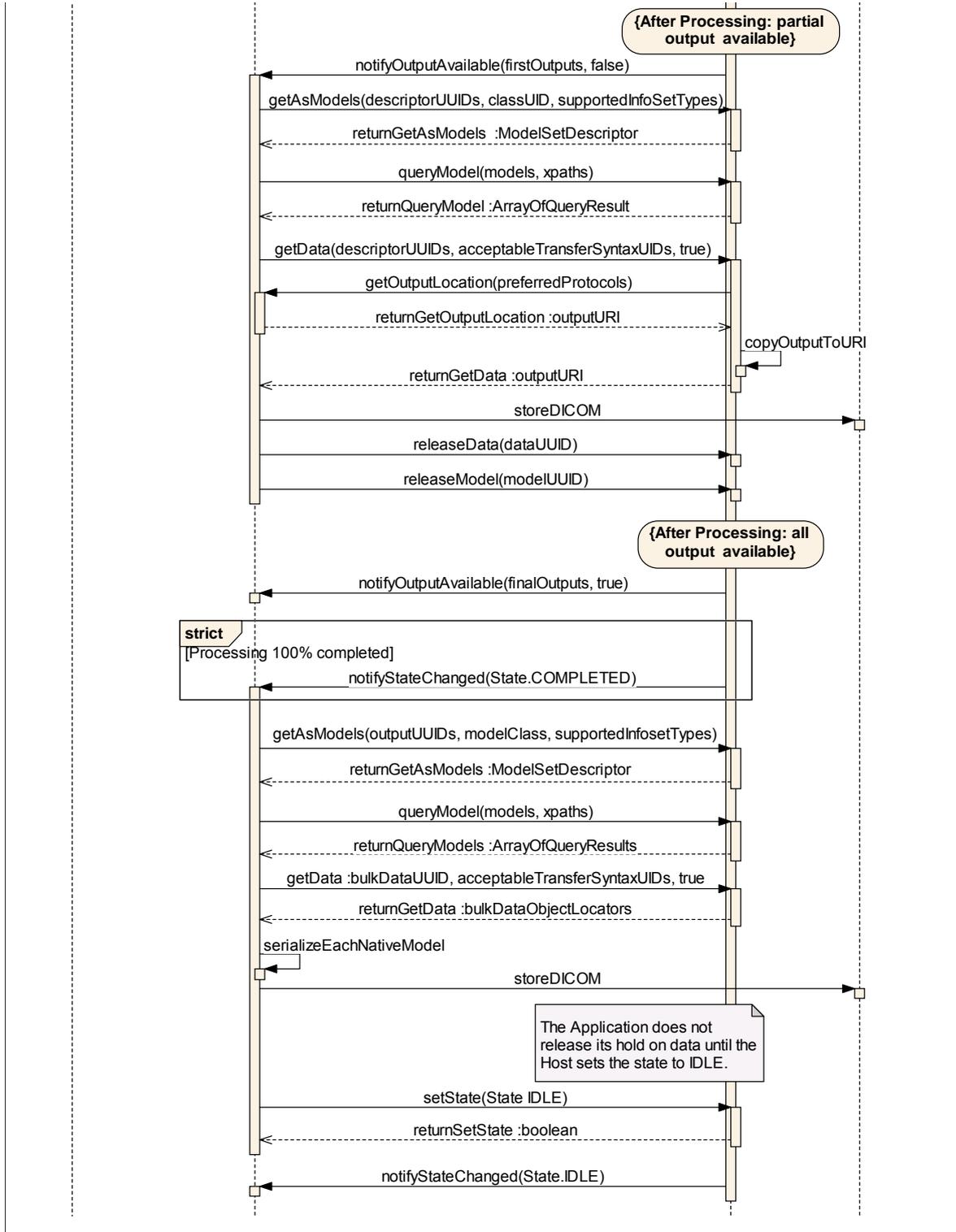
図8.3-2の中のシーケンス線図は、モデルに基いた方法を使用した潜在的な交換を例証する。さらに、それは、ホスト型アプリケーションが部分出力を返すのを例証する。これは、ホスト型アプリケーションがgetOutputLocation()方法を使用する一つの潜在的な方法であり、releaseModel()およびreleaseData()方法の潜在的な用途である。



(次のページに続く)

図8.3-2 モデルに基づいたデータ交換シーケンスの例

(前のページからの続き)



ホスティング・システムは、ファイルベースとモデルベースとの両方のインタフェースを、データソースとして、またデータ受信者として両方で、サポートしなければならない。

ホスト型アプリケーションは、ファイルベースかモデルベースか、少なくともいずれかのインタフェースをサポートしなければならない。データソースとしてかデータ受信者としてかは、ホスト型アプリケーションによる必要性に従う。ホスト型アプリケーションがモデルに基いたインタフェースをサポートする場合、それは附属書Aで定義されたモデルの少なくとも1つをサポートしなければならない。ホスト型アプリケーションは、ホスト型アプリケーションが実際に使用するインタフェースの部分だけを、実装することに決めてもよい；しかしながら、ホスティングシステムが呼び出すかもしれないインタフェース方法すべては、たとえホスト型アプリケーションが適切に返すだけだとしても、ホスティングシステムが呼び出せるように利用できなければならない。

次のセクションは、DataExchangeインタフェースの方法について記述する。

8.3.1 notifyDataAvailable(data : AvailableData, lastData : boolean) : ブール

データソースは、それが受信者に供給できる利用可能なデータの記述を用いてこの方法呼び出す。追加データが利用可能になるとデータソースが予期する場合、それはlastDataパラメータ中のFALSEを渡さなければならない。そうでなければ、lastDataパラメータ中のTRUEを渡さなければならない。また、IDLE状態を通じてもう一度移行した後まで、それ以上の呼出しをnotifyDataAvailableに対し行ってはならない。

データソースは、AvailableDataフレームで識別された形式でデータを提供できなければならない。

ホスティングシステムは、ホスト型アプリケーションが働くことが望ましい入力データをホスト型アプリケーションに通知するために、この方法を使用する。ホスト型アプリケーションは、ホスト型アプリケーションによって生産された出力をホスティングシステムに通知するために、この方法を使用する。

データの受信者が成功裡にAvailableDataリストを受取ったならば、この方法はTRUEを返す。そうでなければ、FALSEを返す。

注: この呼出しの受信者であるが、AvailableDataリストを受取ることに失敗したホスト型アプリケーションは、notifyStatus方法呼出しでの失敗の理由に報告するかもしれない。

データソースは、AvailableDataの中に、前の成功したnotifyDataAvailable呼出しの中で送られたデータへの言及を含んではならない(つまり方法呼出しがTRUEを返したときのもの)。

ホスト型アプリケーションは、FALSEのlastData指標を用いてnotifyDataAvailable()呼出しを受けたか又は送った場合、COMPLETED状態へ次の場合移行してはならない。

データソースは、空のデータ・リストを用いてnotifyDataAvailable()を呼ぶかもしれない。

注: 空のリストを備えたnotifyDataAvailable()を呼び出すことは、TRUEにlastData指標をセットするのに役立つ。

この方法が呼び出されないのは、ホスト型アプリケーションがINPROGRESS状態にある場合だけである。

8.3.2 getData(objectUIDs : ArrayOfUUID, acceptableTransferSyntaxUIDs : ArrayOfUID, includeBulkData : boolean) : ArrayOfObjectLocator

データの受信者は、データソースによって提供される2進法のデータへアクセスするちやめに、この方法を起動する。データソースは、データを検索するバイトストリームとして受信者が開くURIを提供する。

注: データの供給者は、2進法のデータの実際の準備を、受信者が実際にそれを要求するまで遅らせてもよい。

objectUIDs配列は、ソースが検索したい2進法のデータの**UUID**を提供する。その配列中の**UUID**の各々は、**AvailableData**構造(1つ以上の**notifyDataAvailable()**方法呼出しの中で受信者によって受取られたもの)の中で提供された**ObjectDescriptors**からか、または受信者によってアクセスされたモデル中のバルクデータポインタからのいずれかから取出される。

UUIDが**ObjectDescriptor**から来た場合、ソースは、各**UUID**に関連した**AvailableData**構造内の**ObjectDescriptor**中に列記された**MIME**コンテンツタイプおよび**classUID**を使用して、2進法のオブジェクトの**ObjectLocators**を返す。**UUID**がデータ交換モデルから来た場合、ソースはモデル内に記述された2進法のバルクデータを返す。

受信者は、バルクデータのための希望の転送構文を**acceptableTransferSyntaxUIDs**パラメータを介して列記する。受信者は、優先順に、**acceptableTransferSyntaxUIDs**パラメータ中に、**objectUIDs**によって表されるデータのためそれが受理する転送構文の**UID**を列記しなければならない。データの供給者は、それがサポートするリスト中の最初の転送構文を選択し使用しなければならない。**DICOM**データの場合、データの供給者は、最小限、明示的な**VR**リトルエンディアン転送構文をサポートしなければならない。**acceptableTransferSyntaxUIDs**は、転送構文が意味を持たない**MIME**コンテンツタイプに対し、空かもしれない。

UUIDによって識別された2進法のデータを**ObjectDescriptor**から検索する場合、もし受信者が**includeBulkData**フラグを**TRUE**に設定すれば、ソースはデータ・ストリーム内のバルクデータを供給しなければならない。そうでなければ、そのソースは画素データのようなバルクデータを省略しても、しなくともよい。

注: 例えば、**includeBulkData**フラグは、画素データをそれ自体検索するべきかどうか決定するために、受信者が2進法の**DICOM**形式中の画素データの記述を用いて仕事をしたい場合有用である。

この方法は、**objectUIDs**配列内の方法へ渡された各**UUID**に対し1つの**ObjectLocator**を返す。特にそのオブジェクトの**UUID**によって引用されたデータで受信者が読み込むことができる場合、**ObjectLocator**はファイルについて記述する。

受信者が**ObjectLocator URI**によつ参照されたデータを終了される場合、それらの**URI**を提供するため消費されている資源を解放するために、**releaseData()**方法呼び出すかもしれない。データの受信者によって明示的に解放されないデータ参照は、ホスト型アプリケーションが**IDLE**状態に入る場合、暗黙に解放される。

受信者は、与えられた**ObjectDescriptor**かバルクデータ**UUID**によって参照が付けられたデータを求めて、**getData()**を何度呼び出してもよい。**getData()**への呼出しは、**releaseData ()**への明示または暗黙の呼び出しのいずれかと一致しなければならない。

この方法が呼び出されなければならないのは、ホスト型アプリケーションが**INPROGRESS**または**COMPLETED**の状態にある場合だけである。ホスト型アプリケーションが**SUSPENDED**状態である場合も、ホスティングシステムがこの方法呼び出すかもしれない。

8.3.3 **getAsModels(objectUIDs : ArrayOfUUID, classUID : UID, supportedInfosetTypes : ArrayOfMimeType) : ModelSetDescriptor**

データの受信者は、データソースが、**objectUIDs**配列による参照データを、**classUID**による参照タイプのモデルとして提供することを求めるため、この方法を起動する。**objectUIDs**が取出されるのは、

ObjectDescriptorsからであり、それはデータの受信者に1つ以上の**notifyDataAvailable()**呼出しで渡されたものである。

データの受信者は、**supportedInfosetType**sの中に、優先順に、受信者が**Infosets**として処理できるMIME形式を列記しなければならない。データの受信者は「**text/xml**」MIME形式をサポートしなければならない。それは**supportedInfosetType**s配列に常に含まなければならない。データの供給者は、それがサポートするその配列中の最初のエントリを選択しなければならない。

この方法によって返された**ModelSetDescriptor**は、ソースによって提供されたモデルの**UUID**、並びに**objectUUIDs**配列(要求されたモデル中で表すことができない)によって引用されたデータオブジェクトの**UUID**を含んでいる。

受信者は、与えられた**UUID**による参照データを求めて**getAsModels()**を何回呼び出してもよい。成功した呼出しはそれぞれ異なる**ModelUUID**を返す。

受信者が1セットのモデルを終了した場合、それらのモデルを提供するために消費されている資源を解放するために、**releaseModels()**方法呼び出すかもしれない。データの受信者によって明示的に解放されないモデルは、ホスト型アプリケーションが**IDLE**状態に入るとき、暗黙に解放される。

この方法が呼び出されなければならないのは、ホスト型アプリケーションが**INPROGRESS**または**COMPLETED**の状態にある場合だけである。ホスト型アプリケーションが**SUSPENDED**状態である場合も、ホスティングシステムはこの方法呼び出すかもしれない。

8.3.4 queryModel(models : ArrayOfUUID, xpath : ArrayOfString) : ArrayOfQueryResult

データの受信者は、データソースが**XPath**問い合わせ文字列の各々で参照されたデータの部分集合を返すことを要求するため、この方法を起動する。その文字列は**xpath**パラメータ中で、モデル配列中で渡される**UUID**によって識別された各モデルから渡される。**XPath**問い合わせ文字列の各々は、モデル配列中で引用されたモデルの各々に適用される。

モデル配列中で渡された**UUID**は、1つ以上の**getAsModels()**方法呼出しによって返されたものから選ばなければならない。

問い合わせの結果が、方法によって**XML Infosets**として返される。それは文字列として返された**XML**の中で符号化される。特定の**ModelUUID**からの結果はそれぞれ、各**xpath**文字列の中で返された配列中の**QueryResult**要素として返される。言い換えれば、返された**QueryResults**の数は、モデル配列中の**UUID**の数×(掛ける)**xpath**配列中の**XPath**問い合わせ文字列の数である。

注: この方法が主に使用されるのは、**infosetType**が「**text/xml**」の場合である。

この方法が呼び出されなければならないのは、ホスト型アプリケーションが**INPROGRESS**または**COMPLETED**の状態にある場合だけである。ホスト型アプリケーションが**SUSPENDED**状態である場合も、ホスティングシステムがこの方法呼び出すかもしれない。

8.3.5 queryInfoSet(models : ArrayOfUUID, xpath : ArrayOfString) : ArrayOfQueryResultInfoSet

データの受信者は、データソースが、**XPath**問い合わせ文字列の各々で参照されたデータの部分集合を返すことを要求するためこの方法を起動する。その文字列は**xpath**パラメータ中で、モデル配列中で渡される**UUID**によって識別されたモデルの各々から渡される。**XPath**問い合わせ文字列の各々は、モデル配列中で引用されたモデルの各々に適用される。

モデル配列中で渡されたUUIDは、1つ以上のgetAsModels()方法呼出しによって返されたものから選ばなければならない。

問い合わせの結果が、その方法によってXML Infosetsとして返され、XMLの中で符号化され、getAsModel()呼び出しの間にネゴシエートされた形式で符号化されたバイト配列として返される。特定のモデルUUIDからの各結果は、各xpath文字列の中で返された配列中のQueryResultInfoset要素として返される。言い換えれば、返されたQueryResultsInfoset構造の数は、モデル配列中のUUIDの数×(掛け)xpath配列中のXPath問い合わせ文字列の数である。

注: この方法が主に使用されるのは、infosetタイプが文字列ベースでない場合、例えば「application/fastinfoset」の場合である。もし「text/xml」infosetタイプを使用して、モデル上で呼ばれた場合、バイト配列から文字列への転換が必要となる。

この方法が呼び出されなければならないのは、ホスト型アプリケーションがINPROGRESSまたはCOMPLETEDの状態にある場合である。ホスト型アプリケーションがSUSPENDED状態である場合も、ホスティングシステムがこの方法呼び出すかもしれない。

8.3.6 releaseData(objectUUIDs : ArrayOfUUID): void

データの受信者は、2進法のデータへのアクセスを解放するために、この方法を起動する。データは、getData()呼出しを通じてデータソースによって提供される。ArrayOfUUIDは、受信者が解放しているデータ・ストリームを識別する。この配列中のUUIDは、呼出しによってgetData()に返されたObjectLocatorsの中のロケータフィールドから取出されなければならない。

8.3.7 releaseModels(objectUUIDs : ArrayOfUUID): void

データの受信者は、データソースによって提供されるモデルへのアクセスを解放するために、この方法を起動する。ArrayOfUUIDは、受信者が解放しているモデルを識別する。この配列中のUUIDは、呼出しによってgetAsModels()に返されたModelSetDescriptorsの中のモデルフィールドから取出されなければならない。

9 データ型および構造

9.1 ArrayOf[Type]

特殊型式の配列のカプセル化を表すラッパーオブジェクト。クロスプラットフォームの互換性を可能にするために、API機能へのパラメータ、およびAPI機能からの戻り値の中で使用される。ラッパーは単一のフィールドを含んでいる。それは格納されているタイプの配列である。フィールド名は、タイプ名である。その最初の文字は小文字で、大文字ではない。

注: この構築は、たとえそれがJavaにおいて醜く見えてもMicrosoft®.NET言語結合をサポートするために必要であった。

9.2 AvailableData

受信者に利用可能なデータを通信するデータ構造。データは、階層的方法で組織される。通信するのは、患者、研究、シリーズおよび最後にObjectDescriptorsであって、利用可能なデータオブジェクトを識別するものである。データ構造の中のフィールドは次のとおりである:

- **ObjectDescriptors:** ObjectDescriptor []—ObjectDescriptorデータ構造の配列であって、多数の患者に適用するデータか、または患者/研究/シリーズ階層に合わないデータのいずれかのデータを列記する。
- **患者:** 患者 []—患者データ構造の配列。

9.2.1 ObjectDescriptor

次のフィールドをもったデータ構造:

- **DescriptorUUID:** UUID—この特定データオブジェクトを追跡するために、インタフェースが利用するUUID。
- **MimeType:** MimeType—この特定のデータオブジェクトのMIMEコンテンツタイプ。ソースから利用可能な最も自然な形のもの。最も自然な形とは典型的には、ソースがそのデータベース中のデータを維持する形である。例えばDICOMファイルである。
- **ClassUID:** UID—mimeTypeによって記述された形で表す、このデータオブジェクトのクラスを表すUID。オブジェクトのmimeTypeがデータ交換モデル、例えば附属書Aに定義されたものを指すオブジェクトについては、これはそのモデルのUIDである。オブジェクトのmimeTypeがapplication:dicomであるオブジェクトについては、これはDICOMオブジェクトのSOPクラスUIDである。オブジェクトのMIMEコンテンツタイプが追加クラスを持たないオブジェクトについては、これは空(カラ)かもしれない。
- **TransferSyntaxUID:** UID—mimeTypeによって記述された形で表す、このデータオブジェクトの転送構文を表すUID。転送構文が意味を持たないMIMEコンテンツタイプのオブジェクトについては、これは空かもしれない。
- **モダリティ:** 文字列—このデータの原点を最も良く表すモダリティ。標準値は定義用語から取出される。用語が列記されているのは、モダリティ(0008、0060)属性、PS3.3の一般的なシリーズモジュール(セクションC.7.3.1.1.1)の中である。

9.2.2 患者

特定の患者のためのデータを通信するデータ構造。データ構造の中のフィールドは次のとおりである:

- **名前:** 文字列— PS3.5にPN VRのために記述されるようにフォーマットされた患者の名前.. DICOM SOPインスタンスについては、これは患者の名前(0010、0010)属性の値である。
- **ID:** 文字列— PS3.5にLO VRのために記述されるようにフォーマットされた特定の患者の識別子として使用された文字列。DICOM SOPインスタンスについては、これは患者ID(0010、0020)属性の値である。
- **AssigningAuthority:** 文字列—患者にidを割り当てた組織であって、PS3.5にLO VRのために記述されるようにフォーマットされたもの。DICOM SOPインスタンスについては、これは、患者ID(0010、0021)属性の発行人の値である。
- **性別:** 文字列— 患者の性別。DICOM SOPインスタンスについては、これは患者の性別(0010、0040)属性の値である。他のすべての場合、それは、DICOM性別(0010、0040)属性に許容可能な値を使用しなければならない。

- 生年月日: 文字列 患者の生年月日であって、PS3.5にDA VRのために記述されるようにフォーマットされたもの。DICOM SOPインスタンスについては、これは、患者の生年月日(0010、0030)属性の値である。
- ObjectDescriptors: ObjectDescriptor []—ObjectDescriptorデータ構造の配列であって、列記するデータは、この患者に当てはまるが、しかしこの患者に関するどんな特定の研究にも当てはまらないデータである。
- 研究: 研究 []—研究データ構造の配列。

objectDescriptorsまたは研究の少なくとも1つは存在しなければならない。

9.2.3 研究

特定の研究のためのデータを通信するデータ構造。データ構造の中のフィールドは次のとおりである:

- StudyUID: UID— 研究のUID。DICOM SOPインスタンスについては、これは研究インスタンスUID(0020,000D)属性の値である。
- ObjectDescriptors: ObjectDescriptor []—ObjectDescriptorデータ構造の配列。列記するデータは、この研究(囲む患者内の)に当てはまるが、この研究内のどんな特定のシリーズにも当てはまらないデータである。
- シリーズ: シリーズ []— 多くのシリーズデータ構造。

9.2.4 シリーズ

特定のシリーズのためのデータを通信するデータ構造。データ構造の中のフィールドは次のとおりである:

- SeriesUID: UID— シリーズのUID。DICOM SOPインスタンスについては、これはシリーズインスタンスUID(0020,000E)属性の値である。
- ObjectDescriptors: ObjectDescriptor—ObjectDescriptorデータ構造の配列であって、列記するデータはこのシリーズにおいて既存のデータである(囲む研究内の、囲む患者内の)。

注: ほとんどのDICOM合成SOPインスタンスは、objectDescriptorsによってシリーズレベルにおいて識別されるであろう。

9.3 MimeType

その値が特定のMIMEコンテンツタイプを識別する定義用語であるデータ型。MIMEコンテンツタイプを定義する文字列のシンタックスはIETF RFC 2045に定義されている。トップレベルのMIMEコンテンツタイプはIETF RFC 2046に定義されている。MIMEコンテンツタイプが取出されるのは、インターネット番号割当て機関(IANA)によって管理されたリストからである。そのウェブサイトは<http://www.iana.org/assignments/media-types/>, でありIETF RFC 2048に記述されている。.

9.4 ModelSetDescriptor

getAsModels()方法から返されたデータ構造であって次のフィールドを備えたもの:

- InfosetType: MimeType—infosetTypeのMIMEタイプであってデータソースによってリストから選ばれたもの。リストはgetAsModels()呼出しで受信者によってそれに渡されたリストである。

- **モデル: UUID []**— データオブジェクトのセットから作成されたモデルに言及するUUIDの配列である。オブジェクトは、`getAsModels()`呼出しへ渡されたUUIDの配列によって言及されている。
- **FailedSourceObjects: UUID []**— UUID配列に参照されたデータオブジェクトを指定するUUIDである。そのUUID配列は、要求されたモデルクラスで表すことができない`getAsModels()`呼出しへ渡されたものである。

注: 例えば、もし`getAsModels()`呼出しに渡されたUUID配列が、参照の同じ評価基準系(つまりボリュームスタック)からの100のCTスライス、プラス1つのGSPSオブジェクトを含めば、また、もし呼び出し元が抽象多次元画像モデルを要求すれば、そうすれば、`getAsModels()`によって返された**ModelSetDescriptor**は、1つのUUIDをモデル配列中に含む。識別するのは100のCTスライスから作成されたCTボリューム画像データ、および**failedSourceObjects**配列中の1つのUUIDである。そして GSPS オブジェクトのためにそのUUIDを指定する。

9.5 ObjectLocator

データオブジェクトの受信者がそのオブジェクトを取戻すことができる位置を表すデータ構造。それは次のフィールドから成る:

- **ロケータ: UUID**—この特定の**ObjectLocator**を追跡するためにインタフェースが利用するUUID。
- **ソース: UUID**—ソースのUUIDであって、この**ObjectLocator**のためデータを供給するもの。このUUIDは、データを自然な形(例えばファイルまたはバイトストリーム)で検索しようとする場合、**ObjectDescriptor**の中のUUIDと一致する。このUUIDがバルクデータポインタ中のUUIDと一致するのは、バルクデータをモデルから検索する場合である。
- **TransferSyntax: UID**—転送構文であって、その中でこのデータが符号化され、リストからデータソースによって選ばれる。リストは`getData()`呼出しの**acceptableTransferSyntaxUIDs**パラメータ中のデータの受信者によって渡される。転送構文が意味を持たないMIMEコンテンツタイプのオブジェクトについては、これは空かもしれない。
- **長さ: 長い**—UUIDによって参照されるデータオブジェクトの長さ
- **オフセット: 長い**—データオブジェクトが始まるファイルかバイトストリーム内のオフセット。
- **URI: URI**—受信者がデータオブジェクトを検索する資源を識別するURI。典型的には、そうである。しかしローカルのファイルシステム上のファイルに限定されない。受信者は、オブジェクト内のデータにアクセスできなければならない。それにはファイルIOまたはメモリマッピングを使用する。

9.6 QueryResult

モデルのXPath問い合わせからの結果を保持するデータ構造。それは次のフィールドから成る:

- モデル: UUID – この結果が由来したモデルのUUID。
- XPath: 文字列 – この結果に結びついたXPath問い合わせ文字列。
- 結果: XPathNode [] – 問い合わせ結果を保持するXPathNodesの配列。

9.7 QueryResultInfoSet

モデルのXPath問い合わせからの結果を保持するデータ構造。それは次のフィールドから成る:

- モデル: UUID – この結果が由来したモデルのUUID。
- XPath: 文字列 – この結果に結びついたXPath問い合わせ文字列。
- 結果: XPathNodeInfoSet [] – 問い合わせ結果を保持するXPathNodeInfoSet構造の配列。

9.8 長方形

表示スクリーン上の長方形の領域を定義するデータ構造。データ構造の中のフィールドは次のとおりである:

- RefPointX : int
- RefPointY : int

これが定義するのはスクリーン座標の中で領域の左上隅の位置である、そして

- 幅 : int
- 高さ : int

これが指定するのは、領域の範囲である。スクリーン座標はスクリーン左上隅の原点0.0から始まり、正方向の下へそして右に伸びる。

9.9 状態

状態は次の値をもつた列挙データ型である:

- IDLE
- INPROGRESS
- COMPLETED
- SUSPENDED
- CANCELED
- EXIT

数値の解釈はセクション7.2の状態に定義される。.

9.10 ステータス

次のフィールドをもったデータ構造：

- **StatusType : StatusType** – このステータスメッセージの厳しさレベル
- **CodingSchemeDesignator**：文字列 – **codeValues**が定義されるコード体系。**codeValue**の使用は、DICOMコード値(0008、0100)属性の使用と一致していなければならない。その属性はPS3.3の中で指定される。
- **CodeValue**：文字列 – このステータスメッセージの性質を表す指定のコード体系内の特定のコード値。メッセージの使用は、DICOMコード意味(0008、0104)属性の使用と一致していなければならない。その属性はPS3.3の中で指定される。
- **CodeMeaning**：文字列 – コード値用の表示できる文字列。メッセージの使用は、DICOMコード意味(0008、0104)属性の使用と一致していなければならない。その属性はPS3.3の中で指定される。
- 符号化された用語マクロからの他のフィールドであってセクション10.1の中で定義されたもの。

9.10.1 StatusType

次の値および定義をもった列挙データ型:

- **INFORMATION** – ステータスは情報目的だけである。
- **WARNING** – 示す条件は、ホスト型アプリケーションによって行われた仕事の速度が質に影響する条件である。しかしそれはホスト型アプリケーションがそのタスクを終えるのを妨げない。
- **ERROR** – 示す条件は、ホスト型アプリケーションが正確にそのタスクを終えるのを妨げる条件である。ホスト型アプリケーションは、継続しようと努力する。
- **FATALERROR** – 示す条件は、ホスト型アプリケーションがそのタスクを終えるのを妨げる条件である。ホスト型アプリケーションは、継続を試みず、自動的に**CANCELED**状態に移行する。

9.11 UID

終止符で分けた数字列。一意的識別子を表す(PS3.5を参照)。PS3.5にUI VRのために記述されるようにフォーマットされる。

9.12 UUID

ITUT勧告X.667に定義される普遍的な一意的識別子を表す文字列。16進法の表現形式を使用する。

9.13 XPathNode

次のフィールドをもったデータ構造。それはモデルのXPath問い合わせからの出力を表す。文字列ベースの表現で戻される。

- **NodeType : XPathNodeType**
- 値: 文字列

9.14 XPathNodeInfoSet

次のフィールドをもったデータ構造。それはモデルのXPath問い合わせからの出力を表す。バイト配列表現で戻される。

- NodeType : XPathNodeType
- InfoSetValue : byte[]

9.15 XPathNodeType

XPath問い合わせから戻る結果のタイプの一覧表。

注: この一覧表は、マイクロソフト.NETフレームワーク中で利用された同様の一覧表と互換性をもつ。

- ルート— 結果はXML InfoSetのトップレベルノードである。(つまり結果はXML InfoSet全体である)。
- 要素— 結果はXML InfoSet内のXML要素である。(つまり結果はXML InfoSetの部分集合である)。
- 属性— 結果は、XML InfoSet内のXML要素のXML属性である。
- テキスト— 結果は、XML InfoSet内のXML要素の本文のコンテンツである。文書オブジェクトモデル(DOM)テキストおよびCDATAノードタイプと等価である。少なくとも1文字を含んでいる。
- SignificantWhitespace— 結果はXML InfoSet内のXML要素のコンテンツである。そこではコンテンツは有意なスペースからだけ成る(例えばxml:spaceは、保存に設定された)。余白コードポイントは、ISO 10646(Unicode)のSPACE(U0020)、TAB(U0009)、CARRIAGE RETURN(U000D)またはLINE FEED(U000A)である。
- 余白— 結果はXML InfoSet内のXML要素のコンテンツである。そこでは、コンテンツは余白からだけ成る。余白コードポイントは、ISO 10646(Unicode)のSPACE(U0020)、TAB(U0009)、CARRIAGE RETURN(U000D)またはLINE FEED(U000A)である。
- コメント— 結果はXML InfoSet内のコメントである。
- ネーム空間— 結果はXML InfoSet内のネーム空間ディレクティブである。
- ProcessingInstruction— 結果はXML InfoSet内の処理命令である。
- すべて— 結果は、XPathNodeTypeに定義されたタイプの何かを含んでいるかもしれない。

10 データ交換モデル規約

モデルベースのDataExchangeインタフェース方法によって使用できるモデルは、附属書Aに定義される。これらのモデルが定義されるのは、XMLスキーマの形式であり、DSDLのRelax NGのコンパクトな形式で書かれ、ISO/IEC 19757によって指定される。

注: 作成者は、Relax NGのコンパクトな形式を他の形式へ翻訳し、実装内で使用してもよい。但し交換されたXML Infosetsが、規格によって指定されたスキーマに対して有効であることが条件である。例えば、

特定の実装は、強い確認規則を用いてスキーマを内部的に用いてもよい(例えば、ISO/IEC 19757で指定されるSchematron規則を使用して、または主張規則を備えたXSDLを使用して)。但し交換のために生産されたXMLが、規格に指定されたスキーマで読み取れること、かつ規格に指定されたスキーマで生産された形式の良いDICOMオブジェクトからのXMLが、実装の内部スキーマによって今までどおり利用されることが条件である。

モデルの実際のインスタンスはXML Infosetsである。附属書Aはモデルについて記述する際に次の規約に従う。

- 注:
1. モデルはXMLスキーマによって定義される。目的は、一般に利用可能なツールの使用によりモデルの操作とナビゲートを可能にすることである。例えば、XPathステートメントを用いて、特定のDICOM属性のようなモデル内の関心部分を識別し、かつそれを抽出できる。
 2. いくつかの実装は、入って来るオブジェクト(それはXMLの形でないかもしれない)から内部表現へ直接解析するかもしれない。例えば領域オブジェクトモデル(DOM)であり、オブジェクトをXMLに変換する必要がない。

各モデル記述内に、XML要素およびXML属性の表がある。これはそのモデルのXML Infosetについて記述するため使用される。これらの表は次の規約を利用する:

1. XML要素名(最初の欄で列記された)は、(英語の) CamelCase で表記され最初の文字は大文字である。
2. XML属性名(最初の欄で列記された)は、CamelCase で表記され最初の文字は小文字である。
3. XML要素およびXML属性の名は、それらの前に1組の「>」文字を付け、最初の先行XML要素内に入れ子にする。その名の前に1つ少い数の「>」文字を付ける。入れ子のXML属性は直近の囲むXML要素に関係している。
4. 「選択性」欄の記入は次のように解釈する:
 - 「R」は、XML要素かXML属性がすべてのモデル中で必要であることを示す。
 - 「C」は、もし記述中で述べた条件が合うならば、XML要素かXML属性がすべてのモデル中で必要であることを示す。
 - 「O」は、XML要素かXML属性がオプションであることを示す。
 - もしXML要素かXML属性が別のXML要素内で入れ子にされ、その囲むXML要素が存在しないならば、(つまりそれが「O」の選択性で定義され、XML Infosetの中に存在しないならば、またはそれが「RC」の選択性で定義され、条件が合わなかったのでXML Infoset中に含まれないならば)、入れ子のXML要素かXML属性は、XML Infoset中に含まれてはならない。これは選択性とは無関係である。
5. 「濃度(基数)」欄の記入は次のように解釈する:
 - 「A」はXML要素の代わりにXML属性としてこれが表されることを示す。したがって定義によって1の濃度を持っている。
 - 「1」は、このXML要素の単一のインスタンスだけが、直近の囲むXML要素の内部に含まれること、またはこのXML要素が別のXML要素内で入れ子にされない場合、トップレベルに含まれることを示す。

- 「0-n」は、このXML要素の0からnまでのインスタンスが、直近の囲むXML要素内に含まれること、またはこのXML要素が別のXML要素内で入れ子にされない場合、トップレベルに含まれることを示す。
 - 「1-n」は、このXML要素の1からnまでのインスタンスが、直近の囲むXML要素内に含まれること、またはこのXML要素が別のXML要素内で入れ子にされない場合、トップレベルに含まれることを示す。
6. モデル内にしばしば繰り返されるXML要素およびXML属性のセットは、マクロとして定義されてもよい。一般的に適用できるマクロはこのセクションで定義される。特定のモデルに特有のマクロは、そのモデルに固有の附属書で定義してもよい。マクロが表内に含まれている場合、それは、あたかもマクロの表のコンテンツが、マクロを参照する表内に挿入されるのと同じである。「>」文字のセットが指令の前にあり、マクロを表に含める。これはマクロの表に列記されたXML要素およびXML属性の名の最初に付けられる。

10.1 符号化された用語

モデルは符号化された用語を利用してもよい。DICOMの中の符号化された用語の表現はPS3.3に記述される。関心のある特定の用語は、PS3.16の中のコンテキストグループで組織され、次のマクロを使用して参照できる。

表10.1-1 符号化用語マクロ

名前	選択性	濃度	説明
基礎符号化エントリ属性			
CodeValue	R	1	参照が付けられた用語か概念を識別する特定のコード値。PS3.3セクション8.1を参照すること。
CodingSchemeDesignator	R	1	codeValueが定義されるコード体系を指定する。PS3.3セクション8.2を参照すること。
CodingSchemeVersion	C	1	PS3.3セクション8.2を参照すること。codingSchemeDesignatorがcodeValueを識別するには十分でない場合、要求される。
CodeMeaning	O	0-1	符号化された値の意味を人間に判読可能にした簡潔な記述。PS3.3セクション8.3を参照すること。
増強された符号化モード			
ContextIdentifier	O	0-1	写像する資源内に定義されたコンテキストグループを識別する。そこからcodeValueとcodeMeaningの値が選ばれたか、または個人のコンテキストグループ

			ブ拡張として加えられた。PS3.3セクション8.6および8.7を参照すること。
MappingResource	C	1	PS3.3セクション8.4を参照すること。 contextIdentifier XML属性が存在する場合、要求される。
ContextGroupVersion	C	1	PS3.3セクション8.5を参照すること。 contextIdentifier XML属性が存在する場合、要求される。
ContextGroupExtensionFlag	O	0-1	codeValue/codingScheme/codeMeaning が、 contextIdentifier XML属性中で識別されたコンテキストグループの個人拡張から選ばれるかどうかを示す。(PS3.3セクション8.7を参照。 数値: 「Y」、「N」
ContextGroupLocalVersion	C	1	PS3.3セクション8.7を参照すること。
ContextGroupExtensionCreatorUID	C	1	コンテキストグループに対する拡張を作成した人が構成を識別する。PS3.3セクション8.7を参照すること。 contextGroupExtensionFlag の値が「Y」である場合、要求される。

10.2 人名コンポーネント

人名コンポーネントは、DICOM規格のPS3.5の中で与えられた定義に従う。人名コンポーネントの使用法のPS3.5記述は、このマクロにも当てはまる。

表10.2-1 人名コンポーネントマクロ

名前	選択性	濃度	説明
FamilyName	O	0-1	人の姓(名字)。DICOM PS3.5でPN VRの記述を参照すること。
GivenName	O	0-1	人の(英語でいう)ファーストネーム。DICOM PS3.5でPN VRの記述を参照すること。
MiddleName	O	0-1	人のミドルネーム。DICOM PS3.5でPN VRの記述を参照すること。
NamePrefix	O	0-1	人の名前の接頭辞。DICOM PS3.5でPN VRの記述を参照すること。
NameSuffix	O	0-1	人の名前接尾辞。DICOM PS3.5でPN VRの記述を参照すること。

附属書A データ交換モデル

A.1 ネイティブのDICOMモデル

A.1.1 使用法

ネイティブのDICOMモデルは、2進法に符号化されたDICOM SOPインスタンスの表現をXML Infosetsとして定義する。これによりデータの受信者は、DICOMの2進法の符号化を理解するツールキットに依存する代わりにXMLに基いたツールを使用して、2進法のDICOMデータセットを通じてナビゲートできる。

注: この形式が他の用途の基礎として利用されることは、意図されていない。この形式は、データの純粋なXML表現で可能になる自己確認特長を利用していない。

パディングを例外として、ネイティブのDICOMモデルの新しいインスタンスを作成しているデータソース(例えばある分析アプリケーションからの結果)は、DICOM符号化規則に従って(例えば文字セットの取扱い)、DICOMのネイティブのモデルのインスタンス内のDicomAttributesに対する値を作成しなければならない。

データ受信者は、ネイティブのDICOMモデルのインスタンスからのデータを、2進法の符号化されたDICOMオブジェクトに変換し、必要に応じてパディングを調節して、DICOM PS3.5の中で指定された符号化規則に合わせなければならない。

A.1.2 識別

ネイティブのDICOMモデル用のObjectDescriptors MIMEコンテンツタイプは「application/x-dicom.native」でなければならない。

ネイティブのDICOMモデル用のObjectDescriptorsクラスUIDは「1.2.840.10008.7.1.1」でなければならない。

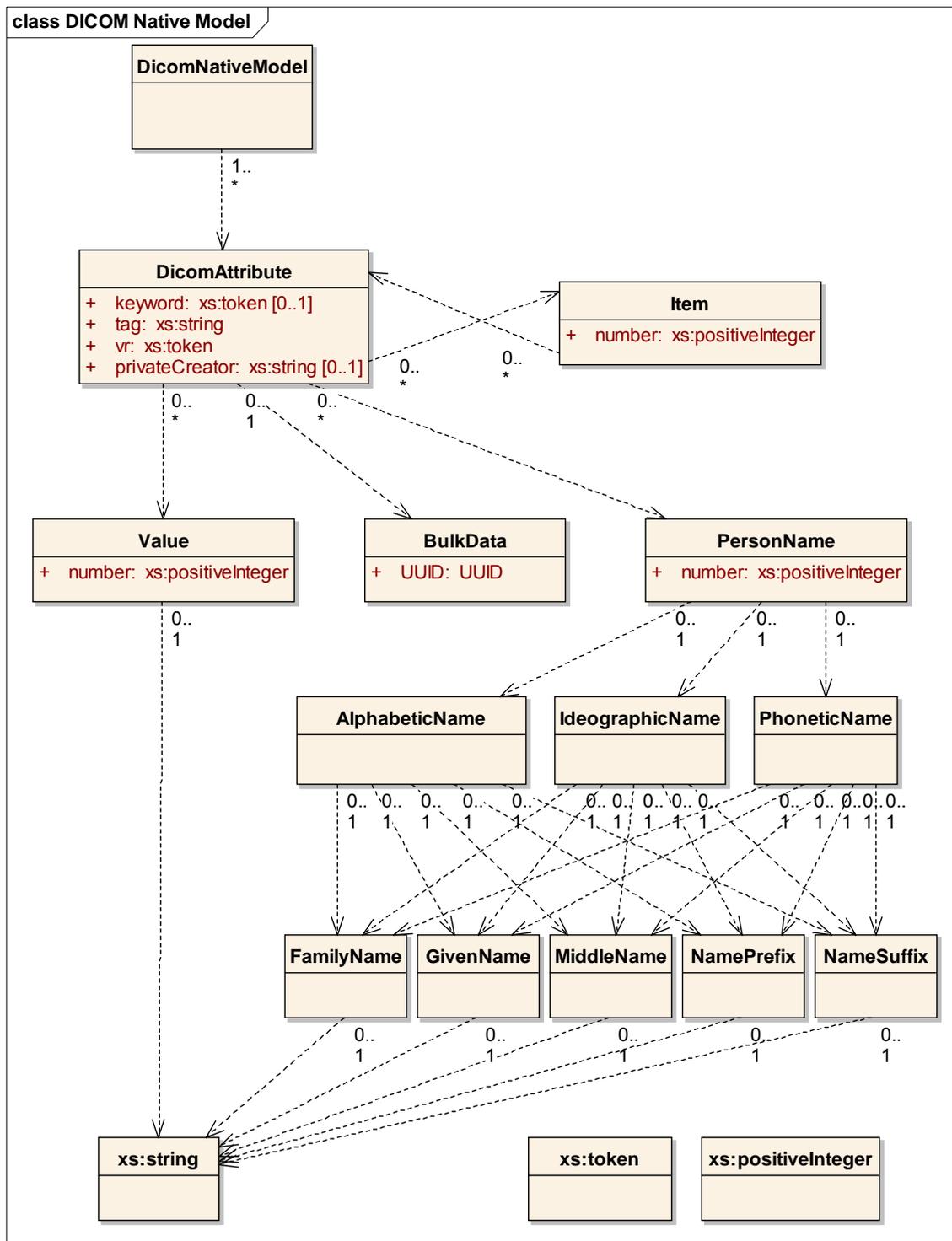
A.1.3 サポート

データソースおよびデータ受信者の両方としてのネイティブのDICOMモデルのサポートは、インタフェースを実装するすべてのホスティング・システムに対し要求されなければならない。

データソースまたはデータ受信者のいずれかとしてのネイティブのDICOMモデルのサポートは、インタフェースを実装するすべてのホスト型アプリケーションに対しオプションでなければならない。

A.1.4 情報モデル

ネイティブのDICOMモデルの図を図A.1.4-1に示す。



図A.1.4-1 ネイティブのDICOMモデル

A.1.5 説明

表A.1.5-1 ネイティブのDICOMモデル

名前	選択性	濃度	説明
NativeDicomModel	R	1	<p>Infoset (W3C勧告XML情報集合 http://www.w3.org/TR/xml-infoset/に定義) は DICOMデータセット(PS3.5に定義)を表し、下記のいずれかである：</p> <ul style="list-style-type: none"> - DICOM合成インスタンス全体のコンテンツ (PS3.3に定義)。ネイティブのモデル要求に応じたもの、または - DICOMの合成インスタンスの一部のコンテンツ。ネイティブのモデルについての問い合わせに応じたもの、または - シーケンスアイテムのコンテンツ (PS3.5に定義)。Infoset値要素内に再帰的に含まれたもの <p>ディレクティブml:space="preserve"が含まれていなければならない。</p>
「DICOMデータセットマクロ」表A.1.5-2を含んでいる			

表A.1.5-2 DICOMデータセットマクロ

名前	選択性	濃度	説明
DicomAttribute	O	0-n	各DICOM属性に対応するInfoset要素。
>keyword	C	A	<p>PS3.6に定義されるキーワード。</p> <p>DICOMデータ要素がホストに未知でないならば要求される。</p>
>tag	R	A	<p>データ要素タグのグループおよび要素の4桁零パディング16進法の値。デリミッターのない単一文字列として連結される。例えば、データ要素(0010、0020)は、「00100020」のタグをもつ。</p> <p>個人のデータ要素については、要素数の2つの最も有意な16進法の文字は00でなければならない。なぜなら個人のクリエイターは明示的に伝えられ、DICOM符号化の中で使用されるブロックが送られてはならないからである (つまり、個人のデータ要素は形式gggg00eeをもっている)。</p>

>vr	O	A	<p>この要素の値表現は、2文字の大文字の文字列として表され、PS3.5に定義され、PS3.6の中でこのデータ要素のために指定される。</p> <p>注: もし望まれば、実装は、データ値を有効にするために値表現を利用するかもしれない。</p>
>privateCreator	C	A	<p>この個人のデータ要素が使用されるブロックに対応する個人のクリエーターデータ要素の値。</p> <p>個人のデータ要素のために要求される。そうでなければ存在してはならない(つまりDICOM規格によって定義されたデータ要素のために)。</p>
>Value	C	1-n	<p>DICOMデータ要素の値フィールドからの値。各DICOM値またはシーケンスアイテムのために1つのInfoSet値要素がある。</p> <p>表されるDICOMデータ要素がゼロ長でないとき、そしてアイテム、PersonName、またはBulkData XML要素が存在しないときに要求される。囲む属性のVRがSQかPNのいずれかである場合、使用されてはならない。</p>
>>number	R	A	<p>値がDICOM値フィールド内に生じる順序。数は単調増加で1×1から始まる。</p> <p>注: 数XML属性はオリジナルの順序を保存するために使用される。</p>
>>plain character data	C	1	<p>平易な文字データとして符号化された単一のDICOM値。</p> <p>例えばDICOMの10進の文字列値フィールドがデリミッターで分離された2つの値を含むならば、例えば、「0.5\0.4」であれば、それは2つのInfoSet値要素として符号化される：</p> <p><値数=「1」>0.5</値> <値数=「2」>0.4</Value></p> <p>コード文字列値フィールドが、デリミッターに分離された3つの値を含み(それらの2番目がゼロ長ならば、例えば「MPG\XR3」ならば、次のように符号化</p>

			<p>される：</p> <p><値数=「1」>MPG</値> <値数=「2」></値> <値数=「3」>XR3</値></p> <p>後の例をゼロ長の値フィールドと対比すること。そうすれば、その場合には、InfoSet値要素は全くない。</p> <p>文字符号化は任意のDICOM特定文字セットにかかわらず、InfoSetのために宣言されたものである。DICOM特定文字セットからInfoSet文字符号化への必要な翻訳は行われていなければならない。</p> <p>注: この翻訳は、特にアジアの文字セットの場合、完全なロスなしではない。</p>
>Item	C	1-n	<p>DICOMシーケンスアイテム、言い換えれば入れ子のDICOMデータセット。</p> <p>表されたDICOMデータ要素がシーケンスであり(「SQ」のVRをもっている)、ゼロ長でない場合、要求される。そうでないと許可されない。</p>
>>number	R	A	<p>アイテムが、アイテムのシーケンス内で生じる順序。数は1×1から単調増加する。</p> <p>注: 数XML属性はオリジナルの順序を保存するために使用される。</p>
>>Include Include 'DICOM Data Set Macro' Table A.1.5-2	R	1	<p>シーケンスアイテムに対応するデータセットを再帰的に含んでいる。</p>
>PersonName	C	1-n	<p>名前(つまり、そのVRはPNである)を含んでいるDICOMデータ要素のXML中の解析された表現。</p> <p>注: 属性解析をPNのVRを用いて名前グループおよびコンポーネントのXML表現に対して行くと、XPathステートメントの生成が単純化され、名前の部分だけをDICOMデータから引く。</p> <p>表わされたDICOMデータ要素がPNのVRをもっており、ゼロ長でない場合、要求される。そうでないと許可されない。</p>

			DICOM PS3.5に定義された規則であつて、PNの値表現をもつDICOM属性内の名前コンポーネントのアルファベット、表意文字および音声のグループの使用法に関する規則が適用される。
>>number	R	A	<p>PersonNameがDICOM値フィールド内に生じる順序。数は1×1から単調増加する。</p> <p>注: 数XML属性はオリジナルの順序を保存するために使用される。</p>
>>Alphabetic	O	0-1	ローマ字で表される名前コンポーネントのグループ。(PNの値表現に関してはDICOM PS3.5での定義を参照。)
>>> Include 'Person Name Component Macro' Table 10.2-1			
>>Ideographic	O	0-1	表意文字で表される名前コンポーネントのグループ。(PNの値表現に関してはDICOM PS3.5での定義を参照。)
>>> Include 'Person Name Component Macro' Table 10.2-1			
>>Phonetic	O	0-1	表音文字で表される名前コンポーネントのグループ。(PNの値表現に関してはDICOM PS3.5での定義を参照。)
>>> Include 'Person Name Component Macro' Table 10.2-1			
>BulkData	C	1	<p>受信者がGetData()方法の使用を通じて検索するデータのブラブへの言及。</p> <p>表されたDICOMデータ要素がゼロ長ではない場合、およびXML Infoset値、アイテムまたはPersonName要素が存在しない場合、要求される。</p> <p>データの供給者は、大きなDICOM値フィールドをテキストとしてInfosetの中の値によって符号化するのを避けるためにBulkData参照を自由に使用してもよい。例えば、供給者は、大きな2進法の値、例えば画素データまたは参照表(それらは典型的にファイル内にある)をBulkData参照として含めてもよい。</p> <p>次のことに留意する。単一のBulkData Infoset要素が値フィールド全体を表す。値多重性が1を越す時、値当たり一つではない。例えばLUTが4096 16ビットのエン</p>

			<p>トリをもち、DICOMの中でOWの値表現で符号化され、8192のVLおよび1のVM、またはUS VRで8192のVLおよび4096のVMならば、両方とも単一のBulkData要素として表される。</p> <p>DICOM PS3.5の中のすべての規則(例えば、バイトオーダーリングおよびスワッピング)が適用される。</p> <p>注: 作成者は、OWとOFの値表現に関するPS3.5規定に特に注意するのがよい。</p> <p>BulkDataが文字列またはテキスト値表現をもっている場合、DICOMの特定文字セットデータ要素の値は、存在すれば、その符号化を決定するため必要かもしれない。</p>
>>UUID	R	A	<p>ITUT勧告X.667で定義された16進法の表現を使用する、UUIDとしてフォーマットされたこのバルクデータ参照の識別子。</p>

A.1.6 スキーマ

ネイティブのDICOMモデル用のXMLスキーマの規範バージョンは、次のとおりである：

```

default namespace="http://dicom.nema.org/PS3.19/models/NativeDICOM"

# This schema was created as an intermediary, a means of describing
# native binary encoded DICOM objects as XML Infosets, thus allowing
# one to manipulate binary DICOM objects using familiar XML tools.
# As such, the schema is designed to facilitate a simple, mechanical,
# bi-directional translation between binary encoded DICOM and XML-like
# constructs without constraints, and to simplify identifying portions
# of a DICOM object using XPath statements.
#
# Since this schema has minimal type checking, it is neither intended
# to be used for any operation that involves hand coding, nor to
# describe a definitive, fully validating encoding of DICOM concepts
# into XML, as what one might use, for example, in a robust XML
# database system or in XML-based forms, though it may be used
# as a means for translating binary DICOM Objects into such a form
# (e.g. through an XSLT script).

start = element NativeDicomModel { DicomDataSet }

# A DICOM Data Set is as defined in PS3.5. It does not appear
# as an XML Element, since it does not appear in the binary encoded
# DICOM objects. It exists here merely as a documentation aid.
DicomDataSet = DicomAttribute*

DicomAttribute = element DicomAttribute {
  Tag, VR, Keyword-, PrivateCreator-,

```

```

( BulkData | Value+ | Item+ | PersonName+ )-
}
BulkData = element BulkData{ UUID }
Value = element Value { Number, xsd:string }
Item = element Item { Number, DicomDataSet }
PersonName = element PersonName {
    Number,
    element SingleByte { NameComponents }-,
    element Ideographic { NameComponents }-,
    element Phonetic { NameComponents }-
}

NameComponents =
    element FamilyName {xsd:string}-,
    element GivenName {xsd:string}-,
    element MiddleName {xsd:string}-,
    element NamePrefix {xsd:string}-,
    element NameSuffix {xsd:string}-

# keyword is the attribute tag from PS3.6
# (derived from the DICOM Attribute's name)
Keyword = attribute keyword { xsd:token }
# canonical XML definition of Hex, with lowercase letters disallowed
Tag = attribute tag { xsd:string{ minLength="8" maxLength="8" pattern="[0-9A-F]{8}" } }
VR = attribute vr { "AE" | "AS" | "AT" | "CS" | "DA" | "DS" | "DT" | "FL" | "FD"
    | "IS" | "LO" | "LT" | "OB" | "OF" | "OW" | "PN" | "SH" | "SL"
    | "SQ" | "SS" | "ST" | "TM" | "UI" | "UL" | "UN" | "US" | "UT" }
PrivateCreator = attribute privateCreator{ xsd:string }
UUID = attribute uuid { xsd:string }
Number = attribute number { xsd:positiveInteger }

```

A.1.7 例

この例は、XPath問い合わせであり、ビューコードシーケンスの最初のアイテムのコード意味を抽出する例である：

```

/DicomNativeModel/DicomAttribute[@keyword="ViewCodeSequence"]/Item[@number=1]/
/DicomAttribute[@keyword="CodeMeaning"]/Value[@number=1]

```

A.2 抽象の多次元画像モデル

A.2.1 使用法

抽象多次元画像モデルを使用して、離散サンプリングされた多次元の画像データを参照できる。標本値は1価(スカラー)または値ベクトル(ベクトル)のいずれかである。例えば、3次元画像の時間変化画像であって、多数のエネルギーレベルで設定されたものである。抽象多次元画像モデルは、DICOMオブジェクトの増強多重フレームファミリにならって作られる。数学的な用語では、これは関数 $I(x, y, z, t, \dots)$ により定義されたデータセットであり、そこで (x, y, z, t, \dots) は次元である。そして標本値 I はコンポーネントのベクトルまたはスカラー(つまり単一のコンポーネント)のいずれかである。このモデルの主目的は、アプリケーションが、データの基礎フォーマットに無関係に、画像データを処理できるようにすることである。

DICOM SOPインスタンスを抽象多次元画像モデルに変換する場合、データの供給者はこれらの規則に出来るだけ従わなければならない:

注: DICOM SOPインスタンスと抽象多次元画像モデルとの間の転換をする場合、決定論的な振舞いは期待されていないし保証されていない。例えば、同じDICOM SOPインスタンスを与えられたとき、異なるホスティング・システムは、ある詳細(例えばコンポーネント値の単位)または次元において異なる抽象多次元画像モデルを作成してもよい。

1. 多重 DICOM SOP インスタンスであって、評価基準系 UID が同じものは、抽象多次元画像モデルの単一インスタンスへ組合せられなければならない。多重シリーズの DICOM SOP インスタンスであって、評価基準系 UUID が同じものは、抽象多次元画像モデルの単一インスタンスへ適宜組合せられてもよい。
2. 単一の DICOM SOP インスタンスは、抽象多次元画像モデルの多重インスタンスに分割されてはならない。
3. 抽象多次元画像モデル内に利用された座標系は、抽象多次元画像モデルインスタンスの生成に利用された DICOM オブジェクトによって定義された座標系を適宜使用しなければならない。実際は、抽象多次元画像モデル内に利用された座標系および次元定義は、原始データを抽象多次元画像モデルに変換するための内挿が不要になるように選ばなければならない。

注: 内挿が必要なのは、原始データがフレームに基いたデカルト座標グリッドに配置されない場合かもしれない。

4. 空間の座標、例えば画像位置(患者)(0020, 0032)は、抽象多次元画像モデルインスタンス内で利用される座標系に変換されなければならない。
5. 画素値は、画素データが入っている抽象多次元画像モデルの意味論および次元の単位と一致するために、必要に応じ空間的に変換されなければならない。
6. 画素データ(7FE0, 0010)属性内の埋込みオーバーレイは、画素値が入っている抽象多次元画像モデルのコンポーネントのデータ型と一致するために、画素値をはずされ、画素値符号は、必要に応じ拡張・変換されなければならない。
7. 画素データの画素値は、画素値が入っている抽象多次元画像モデルのコンポーネントの意味論および単位と一致するために変換されなければならない。

注: 典型的に、VOIおよびプレゼンテーションのLUTのようなプレゼンテーション設定は、DICOM SOPインスタンスから抽象多次元画像モデルを作成するのに使用されない。例外は、コンポーネントの意味論お

よび単位と一致するために、そのようなLUTのアプリケーションが必要なときである。モダリティLUTまたはリスケールスロープが、抽象多次元画像モデルの意味論および単位と一致するため適用されなければならないことが多い。

8. 画素パディング値に相当する画素値は、ストリップされ(つまり零または適切な代替値に設定され)なければならない。`PixelMapOfValidData` 中の空間対応値は、`outValue` に設定されるかまたは `inValue` 以外に適宜設定されなければならない。

抽象多次元画像モデルのインスタンス内のデータをDICOM SOPインスタンスに変換する場合、抽象モデルインスタンスの受信者は、画素データを、作成されているDICOM SOPインスタンスのネイティブの形式と互換性をもつ値に変換しなければならない。この変換は、モダリティLUT情報の再構成、画素パディング値の挿入、画素間隔および原点の変換などを含むかもしれない。これをデータの変換先のSOPクラスによって指定されるように行う。単一の抽象多次元画像モデルを多重のDICOM SOPインスタンスに変換する場合、SOPクラスによって許容されたならば、DICOM SOPインスタンスはすべて同じ評価基準系UID(0020、0052)をもたなければならない。

A.2.2 識別

抽象多次元画像モデルのためのObjectDescriptors MIME コンテンツタイプは"application/x-dicom.abstract"である。

注: これは実験的なMIME形式である。正式なMIME形式が適用される。実装は、実験的なMIME形式と正式なMIME形式の両方をサポートすると予想され、インタフェースに対するバージョン変更はない。

抽象多次元画像モデル用のObjectDescriptorsクラスUIDは、「1.2.840.10008.7.1.2」である。

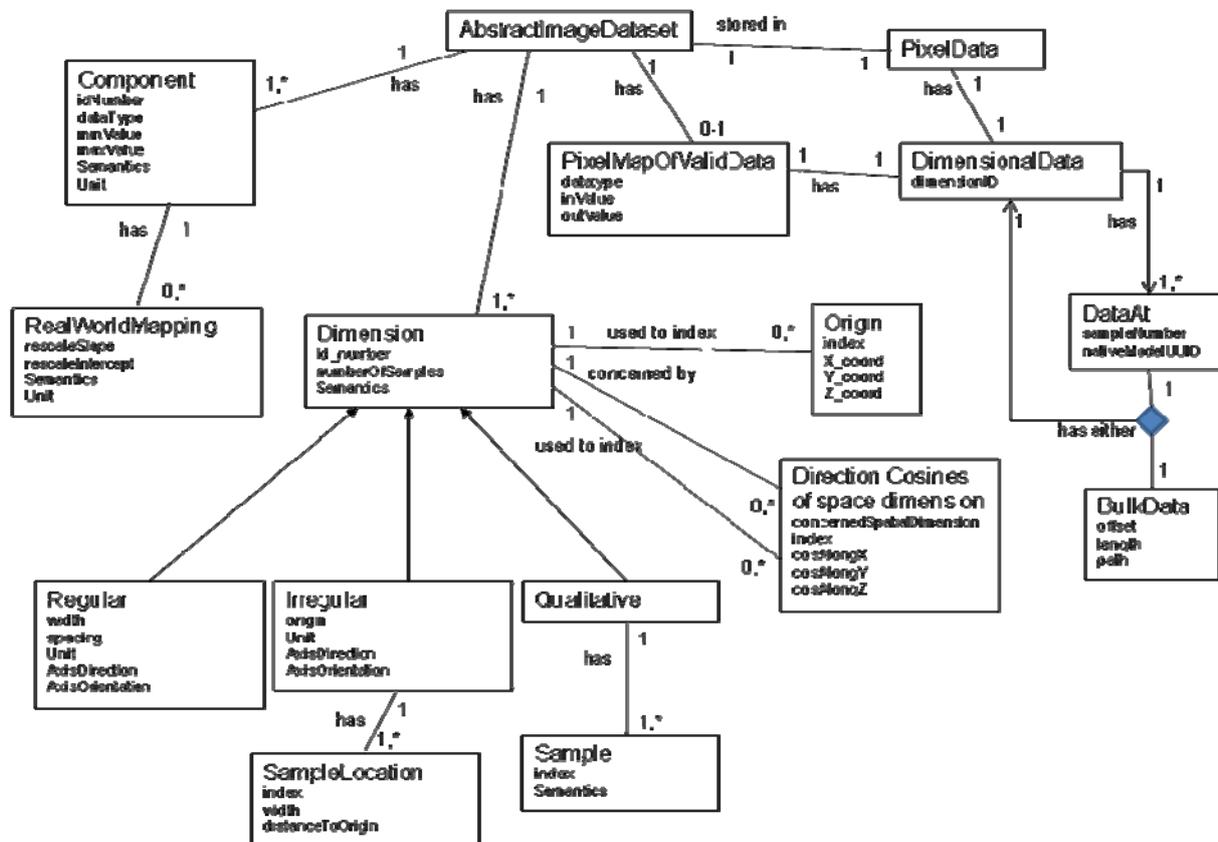
A.2.3 サポート

データソースおよびデータ受信者の両方として抽象多次元画像モデルをサポートすることは、インタフェースを実装するすべてのホスティング・システムに要求される。

データソースまたはデータ受信者のいずれかとして抽象多次元画像モデルをサポートすることは、インタフェースを実装するすべてのホスト型アプリケーションにとってはオプションである。

A.2.4 情報モデル

抽象多次元画像モデルの図を図A.2.4-1に示す。



図A.2.4-1 抽象多次元画像モデル

A.2.5 説明

表A.2.5-1 抽象的な画像モデル

名前	選択性	濃度	説明
AbstractImageDataSet	R	1	すべての抽象的な画像モデルに要求されるトップレベル要素。抽象的な画像データセット全体を保持する。
>Component	R	1-n	関数出力のコンポーネントについて記述する。出力がスカラーである場合、1つのコンポーネントだけがある。ベクトル出力は、ベクトル中の各位置のコンポーネントを要求する。多数のコンポーネントがある場合、コンポーネントはそれぞれのidNumbersによって定義された順序で各々の値に現れる。
>>idNumber	R	A	この特定コンポーネントを識別する。数は1から単調増加する。

>>datatype	R	A	このコンポーネントの値がどのように表されるか説明する。数値は次のとおりである: SIGNED_INT8 SIGNED_INT16 SIGNED_INT32 UNSIGNED_INT8 UNSIGNED_INT16 UNSIGNED_INT32 FLOAT32 FLOAT64
>>minValue	O	A	このコンポーネントが示す最小値。このXML属性が見当たらない場合、これはデータ型によって表すことができる最小値である。
>>maxValue	O	A	このコンポーネントが示す極大値。このXML属性が見当たらない場合、これはデータ型によって表すことができる最大値である。
>>Semantics	R	1	このコンポーネントが何を表すか説明する符号化された値。
>>> <i>Include 'Coded Terminology Macro' Table 10.1-1</i>			定義されたコンテキストIDは7180である。
>>Unit	R	1	この次元が何ユニットであるか説明する、符号化された値。
>>> <i>Include 'Coded Terminology Macro' Table 10.1-1</i>			定義されたコンテキストIDは7181である。
>Dimension	R	1-n	次元について記述する。
>>idNumber	R	A	この特定の次元を識別する。数は1から始まる。より低いidNumberを備えた次元は、より高いidNumberを備えたものより速く変る。
>>numberOfSamples	R	A	例えばこの次元で標本数 X軸に沿ったコラム数 Y軸に沿ったロウ(row)の数 Z軸に沿ったスライスの数 質の記述の数。
>>Semantics	R	1	この次元が何を表すか説明する符号化された値。

>>> Include 'Coded Terminology Macro' Table 10.1-1			定義されたコンテキストIDは7182である。
>> Regular	C	1	この次元に規則的に間隔を置かれた標本について記述するため使用される。 Irregular も Qualitative も存在しない場合要求される。そうでなければ存在してはならない。
>>>width	R	A	標本幅。
>>>spacing	R	A	標本間隔。
>>>Unit	R	1	標本幅および間隔の単位が何であるか説明する符号化された値。
>>>> Include 'Coded Terminology Macro' Table 10.1-1			定義されたコンテキストIDは7183である。
>>>AxisDirection	O	1	この次元の軸の方向。 注: このXML要素が適用できるのは、空間の次元、例えば線形の変位だけである。典型的に、これは患者との関係にある。
>>>> Include 'Coded Terminology Macro' Table 10.1-1			定義されたコンテキストIDは7184である。
>>>AxisOrientation	O	1	値が増加しているこの次元の軸のオリエンテーション。 注: このXML要素は、単に、線形の置換に対処するもののような空間の次元にアプリケーション可能かもしれない。典型的に、これは患者との関係にある。
>>>> Include 'Coded Terminology Macro' Table 10.1-1			定義されたコンテキストIDは7185である。
>>Irregular	C	1	この次元に規則的に間隔を置かれた標本について記述するため使用される。要求されるのは、 Irregular も Qualitative も存在しない場合である。そうでなければ存在してはならない。
>>>origin	R	A	標本位置の各々が測定される参照位置。
>>>SampleLocation	R	1-n	各標本の位置を原点からのオフセットで記述する。このシーケンスで numberOfSamples SampleLocation XML

			要素がなければならない。
>>>>index	R	A	この標本位置の指標値であり、数は1から始まり numberOfSamples まで増える。
>>>>width	R	A	標本幅。
>>>>distanceToOrigin	R	A	原点位置からのこの標本位置の距離。
>>>Unit	R	1	標本幅および位置の単位が何であるか説明する符号化された値。
>>>> <i>Include 'Coded Terminology Macro' Table 10.1-1</i>			定義されたコンテキストIDは 7183 である。
>>>AxisDirection	O	1	この次元の軸の方向。 注: このXML要素は、単に、線形の置換に対処するもののような空間の次元にアプリケーション可能かもしれない。典型的に、これは患者との関係にある。
>>>> <i>Include 'Coded Terminology Macro' Table 10.1-1</i>			定義されたコンテキストIDは 7184 である。
>>>AxisOrientation	O	1	値が増加しているこの次元の軸のオリエンテーション。 注: このXML要素は、単に、線形の置換に対処するもののような空間の次元にアプリケーション可能かもしれない。典型的に、これは患者との関係にある。
>>>> <i>Include 'Coded Terminology Macro' Table 10.1-1</i>			定義されたコンテキストIDは 7185 である。
>>Qualitative	C	1	この次元に規則的に間隔を置かれた標本について記述するため使用される。 Irregular も Qualitative も存在しない場合、要求される。そうでなければ存在してはならない。
>>>Sample	R	1-n	この次元に沿った標本がそれぞれ表すものの記述。このシーケンスで numberOfSamples 標本XML要素がなければならない。
>>>>index	R	A	この標本の指標値。数は1から始まり、 numberOfSamples まで増加する。

>>>>Semantics	R	1	この標本が何を表すか説明する符号化された値。
>>>> <i>Include 'Coded Terminology Macro' Table 10.1-1</i>			定義されたコンテキストIDは7186である。
>>Origin	O	0-n	画像データ値の空間フレームまたはボリュームの、抽象多次元画像モデルの座標系中の空間の位置を指定する。異なるフレームまたはボリュームは原点を共有するか、または各フレームまたはボリュームに対し異なる原点があるかもしれない。この次元内に単一の原点XML要素だけがある場合、この原点はこの次元に沿ったすべての標本に当てはまる。そうでなければ、 numberOfSamples 原点XML要素がなければならない、この次元に沿った各標本に対するものである。次元に対する標本指標値は、その idNumbers がこの次元の idNumber 未満である場合、すべて1に等しい。
>>>index	R	A	標本の指標、それに対しこの原点。これがこの次元に沿ったすべての標本に当てはまる単一の原点である場合、指標は省かれるか、または「0」(零)の値を与えられなければならない。そうでなければ、値は1と numberOfSamles の間の適切な数でなければならない。
>>>xCoord	R	A	抽象多次元画像モデルの座標系中のこの原点のXの位置。
>>>yCoord	R	A	抽象多次元画像モデルの座標系中のこの原点のY位置。
>>>zCoord	R	A	抽象多次元画像モデルの座標系中のこの原点のZ位置。
>>DirectionCosines	O	0-n	次元の抽象多次元画像モデルの座標系中の方向を指定する。その idNumber が concernedSpatialDimension の中で与えられる。 concernedSpatialDimension の idNumber はこの次元の idNumber 未満でなければならない。この次元XML要素内に特定の concernedSpatialDimension をもった単一の DirectionCosines XML要素だけがある場合、この方向余弦はこの次元に沿ったすべての標本に当てはまる。そう

			でなければ、numberOfSamples DirectionCosines XML要素が、この特定のconcernedSpatialDimensionをもっていなければならない。この次元に沿った各標本に対するものである。
>>>concernedSpatialDimension	R	A	この特定の次元のidNumberであって、それに対しDirectionCosines XML要素が適用する。concernedSpatialDimensionの値はこの次元のidNumber未満でなければならない。
>>>index	C	A	数は1から始まる、この方向仕様の指標。この次元に沿ったすべての標本に当てはまるのが単一の値のDirectionCosinesである場合、指標は省かれるか、または「0」(零)の値を与えられなければならない。そうでなければ、指標の値は、この次元に沿った特定の標本値のDirectionCosinesを参照する。
>>>cosAlongX	R	A	このconcernedSpatialDimensionのための抽象多次元画像モデルの座標系のX軸に沿った方向余弦。
>>>cosAlongY	R	A	このconcernedSpatialDimensionのための抽象多次元画像モデルの座標系のY軸に沿った方向余弦。
>>>cosAlongZ	R	A	このconcernedSpatialDimensionのための抽象多次元画像モデルの座標系のZ軸に沿った方向余弦。
>PixelData	R	1	画素データがどこに位置するか定義する構造。次元のラインに沿って組織される。
>>Include 'Dimensional Data Macro' Table A.2.5-2			
>PixelMapOfValidData	O	0-1	どの画素がデータセットに属するか否か識別する画素マップ。画素マップの次元は、画像データの次元と一致する、つまり、画像データ中の標本と画素マップ中の標本の間の一対一の対応がある。画素マップデータへのポインタは次元XML要素の一つに含まれている。
>>datatype	R	A	画素マップ中の標本がどのように符号化されるかが説明される。数値は以下のと

			<p>おりである :</p> <p>BIT1 UNSIGNED_INT8For BIT1,</p> <p>BIT1については、ビット順は、LSBから始まりMSBに向かってUNSIGNED_INT8 (つまり8ビット) バイト内に行く。ビットはパディングが零で、フル8ビットバイトが最も急速に変る次元の終わりになる (つまりそのidNumberが1である次元)。</p>
>>inValue	C	A	<p>画素マップ内の値であって、この標本がデータセットの一部と見なされなければならないことを示す。すべての標本であって、その画素マップ値がinValueと一致しないものは、データセットの一部として考えてはならない。要求されるのは、outValueが存在しない場合である。outValueが存在する場合、存在してはならない。</p>
>>outValue	C	A	<p>画素マップ内の値。この標本がデータセットの一部と見なされてはならないことを示す。すべての標本であって、その画素マップ値がoutValueと一致しないものは、データセットの一部と見なされなければならない。要求されるのはinValueが存在しない場合である。inValueが存在する場合、存在してはならない。</p>
>> 「次元データマクロ」表A.2.5-2を含んでいる			

表A.2.5-2 次元データマクロ

DimensionalData	R	1	<p>各次元のデータを組織する入れ子のツリー構造。ツリー構造のトップレベルは、最も高いidNumberを備えた次元を参照するのがよい。</p>
>dimensionID	R	A	<p>このAbstractImageDataSetの中の次元のidNumber。それに対しこのDimensionalDataが参照する</p>
>DataAt	O	1-n	<p>画像データが発見される場所への言及。このAbstractImageDataSet内の一つの次元XML要素だけが、バルク画素データに対しUUIDをもたなければならない (つまり、すべてのバルクデータ参照は同じ次</p>

			元レベルにある)。 注: もしデータソースが、モデル準備の部分として、多数のより小さなネイティブオブジェクトから画素データの単一ファイルを作成するならば、 descriptorUUID XML属性を提供するため、ソースは、多数の bulkDataUUIDs を作成する必要があるかもしれない。それはその単一の画素データファイル内の異なるオフセットを参照するものである。
>>indexWithinDimension	R	A	このレベルのデータの配列中のこの標本点の順序の位置(例えば指数)。数は1から始まる。
>>descriptorUUID	C	A	ObjectDescriptor を参照するUUID。そこからこのデータが取出され、フォーマットが、ITU勧告X.667によって定義された16進法の表現で行われる。 入れ子のツリー構造のレベルで要求される。そこではソースが descriptorUUID からのデータを、抽象多次元画像モデルへ加えた。
>>bulkDataUUID	C	A	UUIDとしてフォーマットされたバルク画素データへアクセスするために、データの受信者が getData() 要求の中で使用する識別子。それにはITU-T勧告X.667に定義された16進法の表現を使用する。 次元データマクロが入れ子ツリー構造のこのレベルに存在しない場合、要求される。そうでなければ存在してはならない。
>>条件付で次元データマクロ表A.2.5-2 を含んでいる			bulkDataUUID または次元データのうちの1つだけが各レベルに含まれていなければならない。次元データが含まれている場合、入れ子のツリー構造の1段低いレベルでなければならない。つまり次元は、 idNumber が、囲む次元データにより参照される次元より1つ少ないものである。

A.2.6 スキーマ

抽象多次元画像モデル用のRelax NGコンパクトスキーマは次のとおりである：

default namespace = "http://dicom.nema.org/PS3.19/models/AbstractImage"

start = AbstractImageDataSet
AbstractImageDataSet =

```
element AbstractImageDataSet {
  element Component{
    attribute idNumber { xsd:positiveInteger },
    attribute datatype { ComponentDatatype },
    attribute minValue { xsd:double }-,
    attribute maxValue { xsd:double }-,
    element Semantics { CodedTerm },
    element Unit { CodedTerm },
    element RealWordMapping {
      attribute rescaleSlope { xsd:double },
      attribute rescaleIntercept { xsd:double },
      element Unit { CodedTerm },
      element Semantics { CodedTerm }
    }*
  }+,
  element Dimension {
    attribute idNumber { xsd:positiveInteger },
    attribute numberOfSamples { xsd:positiveInteger },
    element Semantics { CodedTerm },
    (element Regular {
      attribute width { xsd:double },
      attribute spacing { xsd:double },
      element Unit { CodedTerm },
      element AxisDirection { CodedTerm }-,
      element AxisOrientation { CodedTerm }-
    }
  | element Irregular {
    element origin { xsd:double },
    element SampleLocation {
      attribute index { xsd:positiveInteger },
      attribute width { xsd:double },
      attribute distanceToOrigin { xsd:double }
    }+,
    element Unit { CodedTerm },
    element AxisDirection { CodedTerm }-,
    element AxisOrientation { CodedTerm }-
  }
  | element Qualitative {
    element Sample {
      attribute index { xsd:positiveInteger },
      element Semantics { CodedTerm }
    }+
  }+
  }+),
  element Origin {
    attribute index { xsd:positiveInteger }-,
    attribute xCoord { xsd:double },
    attribute yCoord { xsd:double },
    attribute zCoord { xsd:double }
  }*,
  element DirectionCosines {
    attribute concernedSpatialDimension { xsd:positiveInteger },
    attribute index { xsd:positiveInteger }-,
    attribute cosAlongX { xsd:double },
    attribute cosAlongY { xsd:double },
    attribute cosAlongZ { xsd:double }
  }
```

```

    }*
  }+,
  element PixelData { DimensionalData },
  element PixelMapOfValidData {
    attribute datatype { PixelMapDatatype },
    (
      attribute inValue { xsd:positiveInteger }
      | attribute outValue { xsd:positiveInteger }
    ),
    DimensionalData
  }-
}

ComponentDatatype =
  "SIGNED_INT8"
  | "SIGNED_INT16"
  | "SIGNED_INT32"
  | "UNSIGNED_CHAR8"
  | "UNSIGNED_INT16"
  | "UNSIGNED_INT32"
  | "FLOAT32"
  | "FLOAT64"

PixelMapDatatype =
  "BIT1"
  | "UNSIGNED_INT8"

DimensionalData =
  element DimensionalData {
    attribute dimensionID { xsd:positiveInteger },
    element DataAt
    {
      attribute sampleNumber { xsd:positiveInteger },
      attribute descriptorUUID { xsd:string }-,
      ( DimensionalData | BulkDataPointer )
    }+
  }

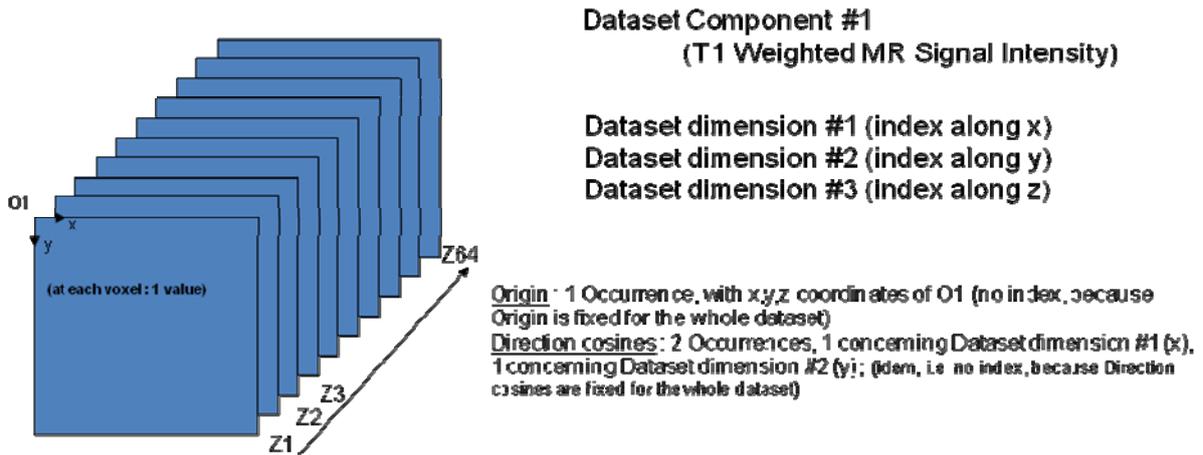
BulkDataPointer =
  attribute UUID { xsd:string }

CodedTerm =
  element CodeValue { xsd:string },
  element CodingSchemeDesignator { xsd:string },
  element CodingSchemeVersion { xsd:string }-,
  element CodeMeaning { xsd:string }-,
  (
    element ContextIdentifier { xsd:string },
    element MappingResource { xsd:string },
    element ContextGroupVersion { xsd:string }
  )-,
  (
    element ContextGroupExtensionFlag { xsd:string },
    element ContextGroupLocalVersion { xsd:string }-,
    element ContextGroupExtensionCreatorUID { xsd:string }-
  )-

```

A.2.7 例
A.2.7.1 単純な3Dボリューム

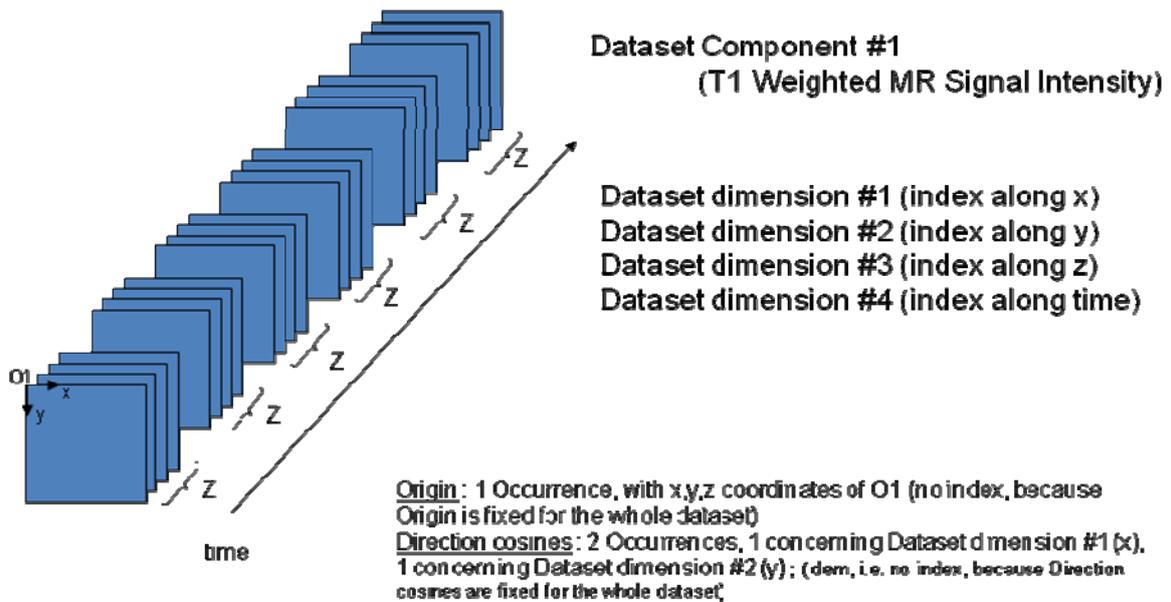
Example of 3D Gray Scale Volume



図A.2.7.1-1 単純な3Dボリュームの例

A.2.7.2 単純な4Dボリューム

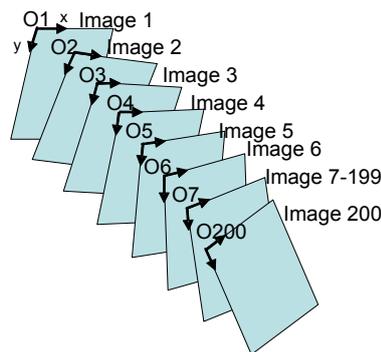
Example of 4D data



図A.2.7.2-1 単純な4Dボリュームの例

A.2.7.3 2D超音波

超音波映像の2Dシリーズの例(平行ではない)



Dataset component #1 (echo)

Dataset dimension #1 (index along x)
Dataset dimension #2 (index along y)
Dataset dimension #3 (index along t)

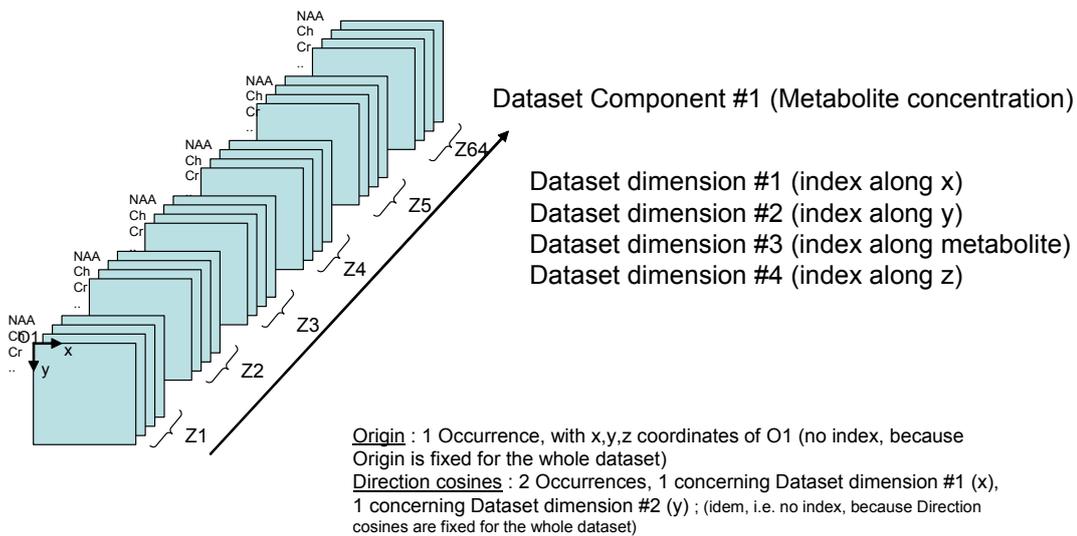
Origin: 200 Occurrences with x,y,z coordinates of O1,O2,..., O200, indexed by Dataset dimension #3 (corresponding to t1, t2, ..., t200)
Direction cosines: 400 Occurrences with 200 concerning Dataset dimension #1 (x) indexed by Dataset dimension #3 (idem t1, t2, ..., t200) and 200 concerning Dataset dimension #2 (y) indexed by Dataset dimension #3 (idem t1, t2, ..., t200)

図 A.2.7.3-1 2D 超音波の例

- この特定の場合、我々は、3つの次元：Xに沿った変位の#1、Yに沿った変位の#2、時系列に指標を付けるための#3をもつ。我々が時間に沿った200の画像をもっている場合(つまり numberOfSamples XML 属性が200に設定される場合)、我々は、DirectionCosines XML 要素の400の発生を次元 XML 要素内にもつ。その idNumber XML 属性は#3に設定される(時間を参照する次元)。200の最初の発生は、値#1を備えた XML 属性 concernedSpatialDimension をもち(X軸に沿った方向余弦を指定する)、かつ XML 属性指標によって指標付けられる。それは1~200で時間に沿った200の画像に対応して変わる。200の次の発生は、値#2を備えた XML 属性 concernedSpatialDimension をもち(Y軸に沿った方向余弦を指定する)、XML 属性指標によって指標付けられる。それは1~200で変わる。
- 同様に、この例において、我々は Origin XML 要素の200の発生を、次元 XML 要素内にもつ。それは値3に設定された idNumber XML 属性をもち、もちろん1~200で変わる XML 属性指標による。

A.2.7.4 3D MRメタボライトマップー 単一のコンポーネント

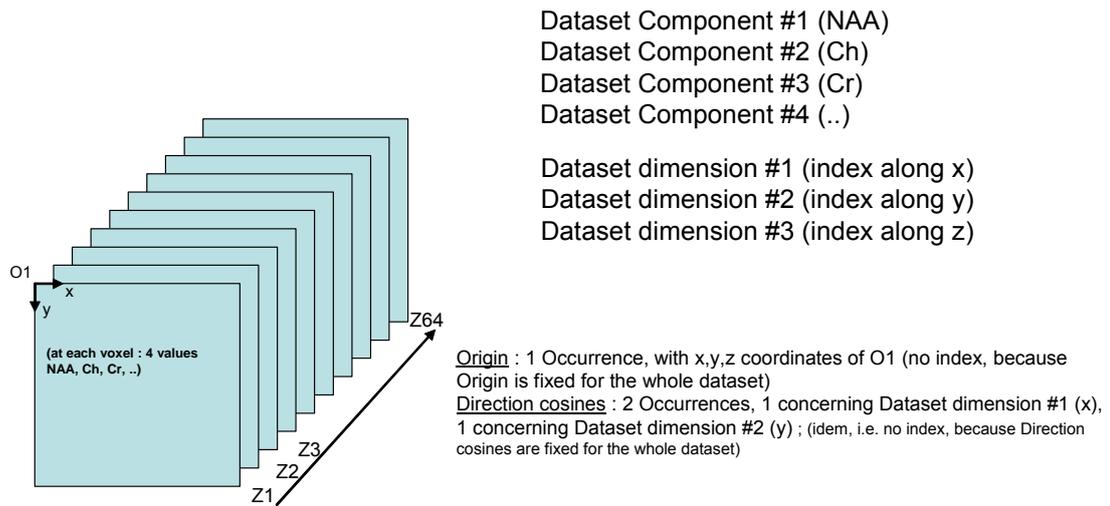
Example of 3D MR spectro data (using 1 single component)



図A.2.7.4-1 単一コンポーネント3D MRメタボライトの例

A.2.7.5 3D MRメタボライトマップー 複合のコンポーネント

Example of 3D MR spectro data (using 4 components)



図A.2.7.5-1 複合コンポーネント3Dメタボライトマップの例

附属書B インタフェース定義

B.1 アプリケーションインタフェースー VERSION 20100825

B.1.1 インタフェースのWSDL定義

下記はApplicationService-20100825.wsdlのコンテンツである。 :

```
<-xml version="1.0" encoding="utf-8"->
<wsdl:definitions name="ApplicationService-20100825"
  targetNamespace="http://dicom.nema.org/PS3.19/ApplicationService-20100825"
  xmlns:tns="http://dicom.nema.org/PS3.19/ApplicationService-20100825"
  xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
  xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-
utility-1.0.xsd"
  xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
  xmlns:wsam="http://www.w3.org/2007/05/addressing/metadata"
  xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing"
  xmlns:wsp="http://schemas.xmlsoap.org/ws/2004/09/policy"
  xmlns:wsap="http://schemas.xmlsoap.org/ws/2004/08/addressing/policy"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:microsoft="http://schemas.microsoft.com/ws/2005/12/wsdl/contract"
  xmlns:wsaw="http://www.w3.org/2006/05/addressing/wsdl"
  xmlns:soap12="http://schemas.xmlsoap.org/wsdl/soap12/"
  xmlns:wsa10="http://www.w3.org/2005/08/addressing"
  xmlns:wsx="http://schemas.xmlsoap.org/ws/2004/09/mex"
  xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">
  <wsdl:types>
    <xsd:schema
targetNamespace="http://dicom.nema.org/PS3.19/Imports/ApplicationService-20100825">
      <xsd:import namespace="http://dicom.nema.org/PS3.19/ApplicationService-20100825"
        schemaLocation="./ApplicationService-20100825.xsd"/>
      <xsd:import namespace="http://schemas.microsoft.com/2003/10/Serialization/"
        schemaLocation="./Types.xsd" />
      <xsd:import namespace="http://schemas.microsoft.com/2003/10/Serialization/Arrays"
        schemaLocation="./ArrayOfString.xsd" />
      <xsd:import namespace="http://schemas.datacontract.org/2004/07/System.Xml.XPath"
        schemaLocation="./XPathNodeType.xsd" />
    </xsd:schema>
  </wsdl:types>
  <wsdl:message name="IApplicationService_GetState_InputMessage">
    <wsdl:part name="parameters" element="tns:GetState"/>
  </wsdl:message>
  <wsdl:message name="IApplicationService_GetState_OutputMessage">
    <wsdl:part name="parameters" element="tns:GetStateResponse"/>
  </wsdl:message>
  <wsdl:message name="IApplicationService_SetState_InputMessage">
    <wsdl:part name="parameters" element="tns:SetState"/>
  </wsdl:message>
  <wsdl:message name="IApplicationService_SetState_OutputMessage">
    <wsdl:part name="parameters" element="tns:SetStateResponse"/>
  </wsdl:message>
  <wsdl:message name="IApplicationService_BringToFront_InputMessage">
    <wsdl:part name="parameters" element="tns:BringToFront"/>
  </wsdl:message>
  <wsdl:message name="IApplicationService_BringToFront_OutputMessage">
    <wsdl:part name="parameters" element="tns:BringToFrontResponse"/>
  </wsdl:message>
```

```

</wsdl:message>
<wsdl:message name="IApplicationService_NotifyDataAvailable_InputMessage">
  <wsdl:part name="parameters" element="tns:NotifyDataAvailable"/>
</wsdl:message>
<wsdl:message name="IApplicationService_NotifyDataAvailable_OutputMessage">
  <wsdl:part name="parameters" element="tns:NotifyDataAvailableResponse"/>
</wsdl:message>
<wsdl:message name="IApplicationService_GetData_InputMessage">
  <wsdl:part name="parameters" element="tns:GetData"/>
</wsdl:message>
<wsdl:message name="IApplicationService_GetData_OutputMessage">
  <wsdl:part name="parameters" element="tns:GetDataResponse"/>
</wsdl:message>
<wsdl:message name="IApplicationService_ReleaseData_InputMessage">
  <wsdl:part name="parameters" element="tns:ReleaseData"/>
</wsdl:message>
<wsdl:message name="IApplicationService_ReleaseData_OutputMessage">
  <wsdl:part name="parameters" element="tns:ReleaseDataResponse"/>
</wsdl:message>
<wsdl:message name="IApplicationService_GetAsModels_InputMessage">
  <wsdl:part name="parameters" element="tns:GetAsModels"/>
</wsdl:message>
<wsdl:message name="IApplicationService_GetAsModels_OutputMessage">
  <wsdl:part name="parameters" element="tns:GetAsModelsResponse"/>
</wsdl:message>
<wsdl:message name="IApplicationService_ReleaseModels_InputMessage">
  <wsdl:part name="parameters" element="tns:ReleaseModels"/>
</wsdl:message>
<wsdl:message name="IApplicationService_ReleaseModels_OutputMessage">
  <wsdl:part name="parameters" element="tns:ReleaseModelsResponse"/>
</wsdl:message>
<wsdl:message name="IApplicationService_QueryModel_InputMessage">
  <wsdl:part name="parameters" element="tns:QueryModel"/>
</wsdl:message>
<wsdl:message name="IApplicationService_QueryModel_OutputMessage">
  <wsdl:part name="parameters" element="tns:QueryModelResponse"/>
</wsdl:message>
<wsdl:message name="IApplicationService_QueryInfoSet_InputMessage">
  <wsdl:part name="parameters" element="tns:QueryInfoSet"/>
</wsdl:message>
<wsdl:message name="IApplicationService_QueryInfoSet_OutputMessage">
  <wsdl:part name="parameters" element="tns:QueryInfoSetResponse"/>
</wsdl:message>
<wsdl:portType name="IApplicationService-20100825">
  <wsdl:operation name="GetState">
    <wsdl:input
wsaw:Action="http://dicom.nema.org/PS3.19/IApplicationService/GetState"
  message="tns:IApplicationService_GetState_InputMessage"/>
    <wsdl:output
wsaw:Action="http://dicom.nema.org/PS3.19/IApplicationService/GetStateResponse"
  message="tns:IApplicationService_GetState_OutputMessage"/>
    </wsdl:operation>
    <wsdl:operation name="SetState">
    <wsdl:input
wsaw:Action="http://dicom.nema.org/PS3.19/IApplicationService/SetState"
  message="tns:IApplicationService_SetState_InputMessage"/>
    <wsdl:output
wsaw:Action="http://dicom.nema.org/PS3.19/IApplicationService/SetStateResponse"
  message="tns:IApplicationService_SetState_OutputMessage"/>
    </wsdl:operation>
    <wsdl:operation name="BringToFront">

```

```
<wsdl:input
wsaw:Action="http://dicom.nema.org/PS3.19/IApplicationService/BringToFront"
  message="tns:IApplicationService_BringToFront_InputMessage" />
<wsdl:output

wsaw:Action="http://dicom.nema.org/PS3.19/IApplicationService/BringToFrontResponse"
  message="tns:IApplicationService_BringToFront_OutputMessage" />
</wsdl:operation>
<wsdl:operation name="NotifyDataAvailable">
  <wsdl:input
wsaw:Action="http://dicom.nema.org/PS3.19/IApplicationService/NotifyDataAvailable"
  message="tns:IApplicationService_NotifyDataAvailable_InputMessage" />
  <wsdl:output

wsaw:Action="http://dicom.nema.org/PS3.19/IApplicationService/NotifyDataAvailableRespon
se"
  message="tns:IApplicationService_NotifyDataAvailable_OutputMessage" />
</wsdl:operation>
<wsdl:operation name="GetData">
  <wsdl:input wsaw:Action="http://dicom.nema.org/PS3.19/IApplicationService/GetData"
  message="tns:IApplicationService_GetData_InputMessage" />
  <wsdl:output
wsaw:Action="http://dicom.nema.org/PS3.19/IApplicationService/GetDataResponse"
  message="tns:IApplicationService_GetData_OutputMessage" />
</wsdl:operation>
<wsdl:operation name="ReleaseData">
  <wsdl:input
wsaw:Action="http://dicom.nema.org/PS3.19/IApplicationService/ReleaseData"
  message="tns:IApplicationService_ReleaseData_InputMessage" />
  <wsdl:output

wsaw:Action="http://dicom.nema.org/PS3.19/IApplicationService/ReleaseDataResponse"
  message="tns:IApplicationService_ReleaseData_OutputMessage" />
</wsdl:operation>
<wsdl:operation name="GetAsModels">
  <wsdl:input
wsaw:Action="http://dicom.nema.org/PS3.19/IApplicationService/GetAsModels"
  message="tns:IApplicationService_GetAsModels_InputMessage" />
  <wsdl:output

wsaw:Action="http://dicom.nema.org/PS3.19/IApplicationService/GetAsModelsResponse"
  message="tns:IApplicationService_GetAsModels_OutputMessage" />
</wsdl:operation>
<wsdl:operation name="ReleaseModels">
  <wsdl:input
wsaw:Action="http://dicom.nema.org/PS3.19/IApplicationService/ReleaseModels"
  message="tns:IApplicationService_ReleaseModels_InputMessage" />
  <wsdl:output

wsaw:Action="http://dicom.nema.org/PS3.19/IApplicationService/ReleaseModelsResponse"
  message="tns:IApplicationService_ReleaseModels_OutputMessage" />
</wsdl:operation>
<wsdl:operation name="QueryModel">
  <wsdl:input
wsaw:Action="http://dicom.nema.org/PS3.19/IApplicationService/QueryModel"
  message="tns:IApplicationService_QueryModel_InputMessage" />
  <wsdl:output
wsaw:Action="http://dicom.nema.org/PS3.19/IApplicationService/QueryModelResponse"
  message="tns:IApplicationService_QueryModel_OutputMessage" />
</wsdl:operation>
<wsdl:operation name="QueryInfoSet">
```

```
<wsdl:input
wsaw:Action="http://dicom.nema.org/PS3.19/IApplicationService/QueryInfoSet"
  message="tns:IApplicationService_QueryInfoSet_InputMessage"/>
<wsdl:output

wsaw:Action="http://dicom.nema.org/PS3.19/IApplicationService/QueryInfoSetResponse"
  message="tns:IApplicationService_QueryInfoSet_OutputMessage"/>
</wsdl:operation>
</wsdl:portType>
<wsdl:binding name="ApplicationService-20100825Binding"
type="tns:IApplicationService-20100825">
  <soap:binding transport="http://schemas.xmlsoap.org/soap/http"/>
  <wsdl:operation name="GetState">
    <soap:operation
soapAction="http://dicom.nema.org/PS3.19/IApplicationService/GetState"
  style="document"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="SetState">
    <soap:operation
soapAction="http://dicom.nema.org/PS3.19/IApplicationService/SetState"
  style="document"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="BringToFront">
    <soap:operation
soapAction="http://dicom.nema.org/PS3.19/IApplicationService/BringToFront"
  style="document"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="NotifyDataAvailable">
    <soap:operation

soapAction="http://dicom.nema.org/PS3.19/IApplicationService/NotifyDataAvailable"
  style="document"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="GetData">
    <soap:operation
soapAction="http://dicom.nema.org/PS3.19/IApplicationService/GetData"
  style="document"/>
```

```
<wsdl:input>
  <soap:body use="literal"/>
</wsdl:input>
<wsdl:output>
  <soap:body use="literal"/>
</wsdl:output>
</wsdl:operation>
<wsdl:operation name="ReleaseData">
  <soap:operation
soapAction="http://dicom.nema.org/PS3.19/IApplicationService/ReleaseData"
  style="document"/>
  <wsdl:input>
    <soap:body use="literal"/>
  </wsdl:input>
  <wsdl:output>
    <soap:body use="literal"/>
  </wsdl:output>
</wsdl:operation>
<wsdl:operation name="GetAsModels">
  <soap:operation
soapAction="http://dicom.nema.org/PS3.19/IApplicationService/GetAsModels"
  style="document"/>
  <wsdl:input>
    <soap:body use="literal"/>
  </wsdl:input>
  <wsdl:output>
    <soap:body use="literal"/>
  </wsdl:output>
</wsdl:operation>
<wsdl:operation name="ReleaseModels">
  <soap:operation
soapAction="http://dicom.nema.org/PS3.19/IApplicationService/ReleaseModels"
  style="document"/>
  <wsdl:input>
    <soap:body use="literal"/>
  </wsdl:input>
  <wsdl:output>
    <soap:body use="literal"/>
  </wsdl:output>
</wsdl:operation>
<wsdl:operation name="QueryModel">
  <soap:operation
soapAction="http://dicom.nema.org/PS3.19/IApplicationService/QueryModel"
  style="document"/>
  <wsdl:input>
    <soap:body use="literal"/>
  </wsdl:input>
  <wsdl:output>
    <soap:body use="literal"/>
  </wsdl:output>
</wsdl:operation>
<wsdl:operation name="QueryInfoSet">
  <soap:operation
soapAction="http://dicom.nema.org/PS3.19/IApplicationService/QueryInfoSet"
  style="document"/>
  <wsdl:input>
    <soap:body use="literal"/>
  </wsdl:input>
  <wsdl:output>
    <soap:body use="literal"/>
  </wsdl:output>
```

```

    </wsdl:operation>
  </wsdl:binding>
  <wsdl:service name="ApplicationService-20100825">
    <wsdl:port name="ApplicationServiceBinding" binding="tns:ApplicationService-
20100825Binding">
      <soap:address location="http://localhost/Service"/>
    </wsdl:port>
  </wsdl:service>
</wsdl:definitions>

```

B.1.2 使用されたデータフレームの定義

B.1.2.1 主要な定義

下記はApplicationService-20100825.xsdのコンテンツである。

```

<?xml version="1.0" encoding="utf-8" ?>
<xs:schema xmlns:tns="http://dicom.nema.org/PS3.19/ApplicationService-20100825"
elementFormDefault="qualified"
targetNamespace="http://dicom.nema.org/PS3.19/ApplicationService-20100825"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:import namespace="http://schemas.microsoft.com/2003/10/Serialization/Arrays"/>
  <xs:import namespace="http://schemas.datacontract.org/2004/07/System.Xml.XPath"/>
  <xs:element name="GetState">
    <xs:complexType>
      <xs:sequence/>
    </xs:complexType>
  </xs:element>
  <xs:element name="GetStateResponse">
    <xs:complexType>
      <xs:sequence>
        <xs:element minOccurs="0" name="GetStateResult" type="tns:State"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:simpleType name="State">
    <xs:restriction base="xs:string">
      <xs:enumeration value="IDLE"/>
      <xs:enumeration value="INPROGRESS"/>
      <xs:enumeration value="SUSPENDED"/>
      <xs:enumeration value="COMPLETED"/>
      <xs:enumeration value="CANCELED"/>
      <xs:enumeration value="EXIT"/>
    </xs:restriction>
  </xs:simpleType>
  <xs:element name="State" nillable="true" type="tns:State"/>
  <xs:element name="SetState">
    <xs:complexType>
      <xs:sequence>
        <xs:element minOccurs="0" name="state" type="tns:State"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="SetStateResponse">
    <xs:complexType>
      <xs:sequence>
        <xs:element minOccurs="0" name="SetStateResult" type="xs:boolean"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>

```

```
<xs:element name="BringToFront">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" name="location" nillable="true" type="tns:Rectangle"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:complexType name="Rectangle">
  <xs:sequence>
    <xs:element minOccurs="0" name="Height" type="xs:int"/>
    <xs:element minOccurs="0" name="Width" type="xs:int"/>
    <xs:element minOccurs="0" name="RefPointX" type="xs:int"/>
    <xs:element minOccurs="0" name="RefPointY" type="xs:int"/>
  </xs:sequence>
</xs:complexType>
<xs:element name="Rectangle" nillable="true" type="tns:Rectangle"/>
<xs:element name="BringToFrontResponse">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" name="BringToFrontResult" type="xs:boolean"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="NotifyDataAvailable">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" name="data" nillable="true" type="tns:AvailableData"/>
      <xs:element minOccurs="0" name="lastData" type="xs:boolean"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:complexType name="AvailableData">
  <xs:sequence>
    <xs:element minOccurs="0" name="ObjectDescriptors" nillable="true"
      type="tns:ArrayOfObjectDescriptor"/>
    <xs:element minOccurs="0" name="Patients" nillable="true"
      type="tns:ArrayOfPatient"/>
  </xs:sequence>
</xs:complexType>
<xs:element name="AvailableData" nillable="true" type="tns:AvailableData"/>
<xs:complexType name="ArrayOfObjectDescriptor">
  <xs:sequence>
    <xs:element minOccurs="0" maxOccurs="unbounded" name="ObjectDescriptor"
      nillable="true"
      type="tns:ObjectDescriptor"/>
  </xs:sequence>
</xs:complexType>
<xs:element name="ArrayOfObjectDescriptor" nillable="true"
  type="tns:ArrayOfObjectDescriptor"/>
<xs:complexType name="ObjectDescriptor">
  <xs:sequence>
    <xs:element minOccurs="0" name="ClassUID" nillable="true" type="tns:UID"/>
    <xs:element minOccurs="0" name="MimeType" nillable="true" type="tns:MimeType"/>
    <xs:element minOccurs="0" name="Modality" nillable="true" type="tns:Modality"/>
    <xs:element minOccurs="0" name="TransferSyntaxUID" nillable="true"
      type="tns:UID"/>
    <xs:element minOccurs="0" name="DescriptorUuid" nillable="true" type="tns:UUID"/>
  </xs:sequence>
</xs:complexType>
<xs:element name="ObjectDescriptor" nillable="true" type="tns:ObjectDescriptor"/>
<xs:complexType name="UID">
```

```

    <xs:sequence>
      <xs:element minOccurs="0" name="Uid" nillable="true" type="xs:string"/>
    </xs:sequence>
  </xs:complexType>
  <xs:element name="UID" nillable="true" type="tns:UID"/>
  <xs:complexType name="MimeType">
    <xs:sequence>
      <xs:element minOccurs="0" name="Type" nillable="true" type="xs:string"/>
    </xs:sequence>
  </xs:complexType>
  <xs:element name="MimeType" nillable="true" type="tns:MimeType"/>
  <xs:complexType name="Modality">
    <xs:sequence>
      <xs:element minOccurs="0" name="Modality" nillable="true" type="xs:string"/>
    </xs:sequence>
  </xs:complexType>
  <xs:element name="Modality" nillable="true" type="tns:Modality"/>
  <xs:complexType name="UUID">
    <xs:sequence>
      <xs:element minOccurs="0" name="Uuid" nillable="true" type="xs:string"/>
    </xs:sequence>
  </xs:complexType>
  <xs:element name="UUID" nillable="true" type="tns:UUID"/>
  <xs:complexType name="ArrayOfPatient">
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="unbounded" name="Patient" nillable="true"
        type="tns:Patient"/>
    </xs:sequence>
  </xs:complexType>
  <xs:element name="ArrayOfPatient" nillable="true" type="tns:ArrayOfPatient"/>
  <xs:complexType name="Patient">
    <xs:sequence>
      <xs:element minOccurs="0" name="AssigningAuthority" nillable="true"
type="xs:string"/>
      <xs:element minOccurs="0" name="DateOfBirth" type="xs:dateTime"/>
      <xs:element minOccurs="0" name="ID" nillable="true" type="xs:string"/>
      <xs:element minOccurs="0" name="Name" nillable="true" type="xs:string"/>
      <xs:element minOccurs="0" name="ObjectDescriptors" nillable="true"
        type="tns:ArrayOfObjectDescriptor"/>
      <xs:element minOccurs="0" name="Sex" nillable="true" type="xs:string"/>
      <xs:element minOccurs="0" name="Studies" nillable="true" type="tns:ArrayOfStudy"/>
    </xs:sequence>
  </xs:complexType>
  <xs:element name="Patient" nillable="true" type="tns:Patient"/>
  <xs:complexType name="ArrayOfStudy">
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="unbounded" name="Study" nillable="true"
type="tns:Study"
        />
    </xs:sequence>
  </xs:complexType>
  <xs:element name="ArrayOfStudy" nillable="true" type="tns:ArrayOfStudy"/>
  <xs:complexType name="Study">
    <xs:sequence>
      <xs:element minOccurs="0" name="ObjectDescriptors" nillable="true"
        type="tns:ArrayOfObjectDescriptor"/>
      <xs:element minOccurs="0" name="Series" nillable="true" type="tns:ArrayOfSeries"/>
      <xs:element minOccurs="0" name="StudyUID" nillable="true" type="tns:UID"/>
    </xs:sequence>
  </xs:complexType>
  <xs:element name="Study" nillable="true" type="tns:Study"/>

```

```
<xs:complexType name="ArrayOfSeries">
  <xs:sequence>
    <xs:element minOccurs="0" maxOccurs="unbounded" name="Series" nillable="true"
      type="tns:Series"/>
  </xs:sequence>
</xs:complexType>
<xs:element name="ArrayOfSeries" nillable="true" type="tns:ArrayOfSeries"/>
<xs:complexType name="Series">
  <xs:sequence>
    <xs:element minOccurs="0" name="ObjectDescriptors" nillable="true"
      type="tns:ArrayOfObjectDescriptor"/>
    <xs:element minOccurs="0" name="SeriesUID" nillable="true" type="tns:UID"/>
  </xs:sequence>
</xs:complexType>
<xs:element name="Series" nillable="true" type="tns:Series"/>
<xs:element name="NotifyDataAvailableResponse">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" name="NotifyDataAvailableResult" type="xs:boolean"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="GetData">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" name="objects" nillable="true"
type="tns:ArrayOfUUID"/>
      <xs:element minOccurs="0" name="acceptableTransferSyntaxes" nillable="true"
      type="tns:ArrayOfUID"/>
      <xs:element minOccurs="0" name="includeBulkData" type="xs:boolean"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:complexType name="ArrayOfUUID">
  <xs:sequence>
    <xs:element minOccurs="0" maxOccurs="unbounded" name="UUID" nillable="true"
type="tns:UID"/>
  </xs:sequence>
</xs:complexType>
<xs:element name="ArrayOfUID" nillable="true" type="tns:ArrayOfUID"/>
<xs:complexType name="ArrayOfUID">
  <xs:sequence>
    <xs:element minOccurs="0" maxOccurs="unbounded" name="UID" nillable="true"
type="tns:UID"/>
  </xs:sequence>
</xs:complexType>
<xs:element name="ArrayOfUID" nillable="true" type="tns:ArrayOfUID"/>
<xs:element name="GetDataResponse">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" name="GetDataResult" nillable="true"
      type="tns:ArrayOfObjectLocator"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:complexType name="ArrayOfObjectLocator">
  <xs:sequence>
    <xs:element minOccurs="0" maxOccurs="unbounded" name="ObjectLocator"
nillable="true"
      type="tns:ObjectLocator"/>
  </xs:sequence>
</xs:complexType>
```

```

</xs:complexType>
<xs:element name="ArrayOfObjectLocator" nillable="true"
type="tns:ArrayOfObjectLocator"/>
<xs:complexType name="ObjectLocator">
  <xs:sequence>
    <xs:element minOccurs="0" name="Length" type="xs:long"/>
    <xs:element minOccurs="0" name="Offset" type="xs:long"/>
    <xs:element minOccurs="0" name="TransferSyntax" nillable="true" type="tns:UID"/>
    <xs:element minOccurs="0" name="URI" nillable="true" type="xs:anyURI"/>
    <xs:element minOccurs="0" name="Locator" nillable="true" type="tns:UUID"/>
    <xs:element minOccurs="0" name="Source" nillable="true" type="tns:UUID"/>
  </xs:sequence>
</xs:complexType>
<xs:element name="ObjectLocator" nillable="true" type="tns:ObjectLocator"/>
<xs:element name="ReleaseData">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" name="objects" nillable="true"
type="tns:ArrayOfUUID"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="ReleaseDataResponse">
  <xs:complexType>
    <xs:sequence/>
  </xs:complexType>
</xs:element>
<xs:element name="GetAsModels">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" name="objects" nillable="true"
type="tns:ArrayOfUUID"/>
      <xs:element minOccurs="0" name="classUID" nillable="true" type="tns:UID"/>
      <xs:element minOccurs="0" name="supportedInfoSetTypes" nillable="true"
type="tns:ArrayOfMimeType"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:complexType name="ArrayOfMimeType">
  <xs:sequence>
    <xs:element minOccurs="0" maxOccurs="unbounded" name="MimeType" nillable="true"
type="tns:MimeType"/>
  </xs:sequence>
</xs:complexType>
<xs:element name="ArrayOfMimeType" nillable="true" type="tns:ArrayOfMimeType"/>
<xs:element name="GetAsModelsResponse">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" name="GetAsModelsResult" nillable="true"
type="tns:ModelSetDescriptor"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:complexType name="ModelSetDescriptor">
  <xs:sequence>
    <xs:element minOccurs="0" name="FailedSourceObjects" nillable="true"
type="tns:ArrayOfUUID"/>
    <xs:element minOccurs="0" name="InfosetType" nillable="true" type="tns:MimeType"/>
    <xs:element minOccurs="0" name="Models" nillable="true" type="tns:ArrayOfUUID"/>
  </xs:sequence>
</xs:complexType>

```

```
<xs:element name="ModelSetDescriptor" nillable="true" type="tns:ModelSetDescriptor"/>
<xs:element name="ReleaseModels">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" name="models" nillable="true" type="tns:ArrayOfUUID"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="ReleaseModelsResponse">
  <xs:complexType>
    <xs:sequence/>
  </xs:complexType>
</xs:element>
<xs:element name="QueryModel">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" name="models" nillable="true" type="tns:ArrayOfUUID"/>
      <xs:element minOccurs="0" name="xPaths" nillable="true"
        xmlns:q1="http://schemas.microsoft.com/2003/10/Serialization/Arrays"
        type="q1:ArrayOfstring"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="QueryModelResponse">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" name="QueryModelResult" nillable="true"
        type="tns:ArrayOfQueryResult"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:complexType name="ArrayOfQueryResult">
  <xs:sequence>
    <xs:element minOccurs="0" maxOccurs="unbounded" name="QueryResult" nillable="true"
      type="tns:QueryResult"/>
  </xs:sequence>
</xs:complexType>
<xs:element name="ArrayOfQueryResult" nillable="true" type="tns:ArrayOfQueryResult"/>
<xs:complexType name="QueryResult">
  <xs:sequence>
    <xs:element minOccurs="0" name="Model" nillable="true" type="tns:UUID"/>
    <xs:element minOccurs="0" name="Result" nillable="true"
      type="tns:ArrayOfXPathNode"/>
    <xs:element minOccurs="0" name="XPath" nillable="true" type="xs:string"/>
  </xs:sequence>
</xs:complexType>
<xs:element name="QueryResult" nillable="true" type="tns:QueryResult"/>
<xs:complexType name="ArrayOfXPathNode">
  <xs:sequence>
    <xs:element minOccurs="0" maxOccurs="unbounded" name="XPathNode" nillable="true"
      type="tns:XPathNode"/>
  </xs:sequence>
</xs:complexType>
<xs:element name="ArrayOfXPathNode" nillable="true" type="tns:ArrayOfXPathNode"/>
<xs:complexType name="XPathNode">
  <xs:sequence>
    <xs:element minOccurs="0" name="NodeType"
      xmlns:q2="http://schemas.datacontract.org/2004/07/System.Xml.XPath"
      type="q2:XPathNodeType"/>
    <xs:element minOccurs="0" name="Value" nillable="true" type="xs:string"/>
  </xs:sequence>
</xs:complexType>
</xs:element>
```

```

</xs:complexType>
<xs:element name="XPathNode" nillable="true" type="tns:XPathNode"/>
<xs:element name="QueryInfoSet">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" name="models" nillable="true" type="tns:ArrayOfUUID"/>
      <xs:element minOccurs="0" name="xPaths" nillable="true"
        xmlns:q3="http://schemas.microsoft.com/2003/10/Serialization/Arrays"
        type="q3:ArrayOfstring"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="QueryInfoSetResponse">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" name="QueryInfoSetResult" nillable="true"
        type="tns:ArrayOfQueryResultInfoSet"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:complexType name="ArrayOfQueryResultInfoSet">
  <xs:sequence>
    <xs:element minOccurs="0" maxOccurs="unbounded" name="QueryResultInfoSet"
nillable="true"
      type="tns:QueryResultInfoSet"/>
  </xs:sequence>
</xs:complexType>
<xs:element name="ArrayOfQueryResultInfoSet" nillable="true"
type="tns:ArrayOfQueryResultInfoSet"/>
<xs:complexType name="QueryResultInfoSet">
  <xs:sequence>
    <xs:element minOccurs="0" name="Model" nillable="true" type="tns:UUID"/>
    <xs:element minOccurs="0" name="Result" nillable="true"
type="tns:ArrayOfXPathNodeInfoSet"/>
    <xs:element minOccurs="0" name="XPath" nillable="true" type="xs:string"/>
  </xs:sequence>
</xs:complexType>
<xs:element name="QueryResultInfoSet" nillable="true" type="tns:QueryResultInfoSet"/>
<xs:complexType name="ArrayOfXPathNodeInfoSet">
  <xs:sequence>
    <xs:element minOccurs="0" maxOccurs="unbounded" name="XPathNodeInfoSet"
nillable="true"
      type="tns:XPathNodeInfoSet"/>
  </xs:sequence>
</xs:complexType>
<xs:element name="ArrayOfXPathNodeInfoSet" nillable="true"
type="tns:ArrayOfXPathNodeInfoSet"/>
<xs:complexType name="XPathNodeInfoSet">
  <xs:sequence>
    <xs:element minOccurs="0" name="InfoSetValue" nillable="true"
type="xs:base64Binary"/>
    <xs:element minOccurs="0" name="NodeType"
      xmlns:q4="http://schemas.datacontract.org/2004/07/System.Xml.XPath"
type="q4:XPathNodeType"
      />
  </xs:sequence>
</xs:complexType>
<xs:element name="XPathNodeInfoSet" nillable="true" type="tns:XPathNodeInfoSet"/>
</xs:schema>

```

B.1.2.2 参照が付けられた定義

下記はXPathNodeType.xsdのコンテンツである :

```
<-xml version="1.0" encoding="utf-8"->
<xs:schema xmlns:tns="http://schemas.datacontract.org/2004/07/System.Xml.XPath"
elementFormDefault="qualified"
targetNamespace="http://schemas.datacontract.org/2004/07/System.Xml.XPath"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:simpleType name="XPathNodeType">
    <xs:restriction base="xs:string">
      <xs:enumeration value="Root" />
      <xs:enumeration value="Element" />
      <xs:enumeration value="Attribute" />
      <xs:enumeration value="Namespace" />
      <xs:enumeration value="Text" />
      <xs:enumeration value="SignificantWhitespace" />
      <xs:enumeration value="Whitespace" />
      <xs:enumeration value="ProcessingInstruction" />
      <xs:enumeration value="Comment" />
      <xs:enumeration value="All" />
    </xs:restriction>
  </xs:simpleType>
  <xs:element name="XPathNodeType" nillable="true" type="tns:XPathNodeType" />
</xs:schema>
```

下記はTypes.xsdのコンテンツである :

```
<-xml version="1.0" encoding="utf-8"->
<xs:schema xmlns:tns="http://schemas.microsoft.com/2003/10/Serialization/"
attributeFormDefault="qualified" elementFormDefault="qualified"
targetNamespace="http://schemas.microsoft.com/2003/10/Serialization/"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="anyType" nillable="true" type="xs:anyType" />
  <xs:element name="anyURI" nillable="true" type="xs:anyURI" />
  <xs:element name="base64Binary" nillable="true" type="xs:base64Binary" />
  <xs:element name="boolean" nillable="true" type="xs:boolean" />
  <xs:element name="byte" nillable="true" type="xs:byte" />
  <xs:element name="dateTime" nillable="true" type="xs:dateTime" />
  <xs:element name="decimal" nillable="true" type="xs:decimal" />
  <xs:element name="double" nillable="true" type="xs:double" />
  <xs:element name="float" nillable="true" type="xs:float" />
  <xs:element name="int" nillable="true" type="xs:int" />
  <xs:element name="long" nillable="true" type="xs:long" />
  <xs:element name="QName" nillable="true" type="xs:QName" />
  <xs:element name="short" nillable="true" type="xs:short" />
  <xs:element name="string" nillable="true" type="xs:string" />
  <xs:element name="unsignedByte" nillable="true" type="xs:unsignedByte" />
  <xs:element name="unsignedInt" nillable="true" type="xs:unsignedInt" />
  <xs:element name="unsignedLong" nillable="true" type="xs:unsignedLong" />
  <xs:element name="unsignedShort" nillable="true" type="xs:unsignedShort" />
  <xs:element name="char" nillable="true" type="tns:char" />
  <xs:simpleType name="char">
    <xs:restriction base="xs:int" />
  </xs:simpleType>
  <xs:element name="duration" nillable="true" type="tns:duration" />
  <xs:simpleType name="duration">
    <xs:restriction base="xs:duration">
      <xs:pattern value="\-P(\d*D)-(T(\d*H)-(\d*M)-(\d*(\.\d*)-S)-)-" />
      <xs:minInclusive value="-P10675199DT2H48M5.4775808S" />
    </xs:restriction>
  </xs:simpleType>
</xs:schema>
```

```

        <xs:maxInclusive value="P10675199DT2H48M5.4775807S" />
    </xs:restriction>
</xs:simpleType>
<xs:element name="guid" nillable="true" type="tns:guid" />
<xs:simpleType name="guid">
    <xs:restriction base="xs:string">
        <xs:pattern value="[\da-fA-F]{8}-[\da-fA-F]{4}-[\da-fA-F]{4}-[\da-fA-F]{4}-[\da-fA-F]{12}" />
    </xs:restriction>
</xs:simpleType>
<xs:attribute name="FactoryType" type="xs:QName" />
<xs:attribute name="Id" type="xs:ID" />
<xs:attribute name="Ref" type="xs:IDREF" />
</xs:schema>

```

下記はArrayOfString.xsdのコンテンツである。

```

<-xml version="1.0" encoding="utf-8"->
<xs:schema xmlns:tns="http://schemas.microsoft.com/2003/10/Serialization/Arrays"
elementFormDefault="qualified"
targetNamespace="http://schemas.microsoft.com/2003/10/Serialization/Arrays"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
    <xs:complexType name="ArrayOfstring">
        <xs:sequence>
            <xs:element minOccurs="0" maxOccurs="unbounded" name="string" nillable="true"
type="xs:string" />
        </xs:sequence>
    </xs:complexType>
    <xs:element name="ArrayOfstring" nillable="true" type="tns:ArrayOfstring" />
</xs:schema>

```

B.2 ホストインタフェースー VERSION 20100825

B.2.1 インタフェースのWSDL定義

下記はHostService-20100825.wsdlのコンテンツである：

```

<-xml version="1.0" encoding="utf-8"->
<wsdl:definitions name="HostService-20100825"
targetNamespace="http://dicom.nema.org/PS3.19/HostService-20100825"
xmlns:tns="http://dicom.nema.org/PS3.19/HostService-20100825"
xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-
utility-1.0.xsd"
xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:wsam="http://www.w3.org/2007/05/addressing/metadata"
xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing"
xmlns:wsp="http://schemas.xmlsoap.org/ws/2004/09/policy"
xmlns:wsap="http://schemas.xmlsoap.org/ws/2004/08/addressing/policy"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:microsoft="http://schemas.microsoft.com/ws/2005/12/wsdl/contract"
xmlns:wsaw="http://www.w3.org/2006/05/addressing/wsdl"
xmlns:soap12="http://schemas.xmlsoap.org/wsdl/soap12/"
xmlns:wsa10="http://www.w3.org/2005/08/addressing"
xmlns:wsx="http://schemas.xmlsoap.org/ws/2004/09/mex"
xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">
    <wsdl:types>
        <xsd:schema targetNamespace="http://dicom.nema.org/PS3.19/Imports/HostService-
20100825">

```

```
<xsd:import namespace="http://dicom.nema.org/PS3.19/HostService-20100825"
  schemaLocation="./HostService-20100825.xsd"/>
<xsd:import namespace="http://schemas.microsoft.com/2003/10/Serialization/"
  schemaLocation="./Types.xsd" />
<xsd:import namespace="http://schemas.microsoft.com/2003/10/Serialization/Arrays"
  schemaLocation="./ArrayOfString.xsd" />
<xsd:import namespace="http://schemas.datacontract.org/2004/07/System.Xml.XPath"
  schemaLocation="./XPathNodeType.xsd"/>
</xsd:schema>
</wsdl:types>
<wsdl:message name="IHostService_GenerateUID_InputMessage">
  <wsdl:part name="parameters" element="tns:GenerateUID"/>
</wsdl:message>
<wsdl:message name="IHostService_GenerateUID_OutputMessage">
  <wsdl:part name="parameters" element="tns:GenerateUIDResponse"/>
</wsdl:message>
<wsdl:message name="IHostService_GetAvailableScreen_InputMessage">
  <wsdl:part name="parameters" element="tns:GetAvailableScreen"/>
</wsdl:message>
<wsdl:message name="IHostService_GetAvailableScreen_OutputMessage">
  <wsdl:part name="parameters" element="tns:GetAvailableScreenResponse"/>
</wsdl:message>
<wsdl:message name="IHostService_GetOutputLocation_InputMessage">
  <wsdl:part name="parameters" element="tns:GetOutputLocation"/>
</wsdl:message>
<wsdl:message name="IHostService_GetOutputLocation_OutputMessage">
  <wsdl:part name="parameters" element="tns:GetOutputLocationResponse"/>
</wsdl:message>
<wsdl:message name="IHostService_NotifyStateChanged_InputMessage">
  <wsdl:part name="parameters" element="tns:NotifyStateChanged"/>
</wsdl:message>
<wsdl:message name="IHostService_NotifyStateChanged_OutputMessage">
  <wsdl:part name="parameters" element="tns:NotifyStateChangedResponse"/>
</wsdl:message>
<wsdl:message name="IHostService_NotifyStatus_InputMessage">
  <wsdl:part name="parameters" element="tns:NotifyStatus"/>
</wsdl:message>
<wsdl:message name="IHostService_NotifyStatus_OutputMessage">
  <wsdl:part name="parameters" element="tns:NotifyStatusResponse"/>
</wsdl:message>
<wsdl:message name="IHostService_NotifyDataAvailable_InputMessage">
  <wsdl:part name="parameters" element="tns:NotifyDataAvailable"/>
</wsdl:message>
<wsdl:message name="IHostService_NotifyDataAvailable_OutputMessage">
  <wsdl:part name="parameters" element="tns:NotifyDataAvailableResponse"/>
</wsdl:message>
<wsdl:message name="IHostService_GetData_InputMessage">
  <wsdl:part name="parameters" element="tns:GetData"/>
</wsdl:message>
<wsdl:message name="IHostService_GetData_OutputMessage">
  <wsdl:part name="parameters" element="tns:GetDataResponse"/>
</wsdl:message>
<wsdl:message name="IHostService_ReleaseData_InputMessage">
  <wsdl:part name="parameters" element="tns:ReleaseData"/>
</wsdl:message>
<wsdl:message name="IHostService_ReleaseData_OutputMessage">
  <wsdl:part name="parameters" element="tns:ReleaseDataResponse"/>
</wsdl:message>
<wsdl:message name="IHostService_GetAsModels_InputMessage">
  <wsdl:part name="parameters" element="tns:GetAsModels"/>
</wsdl:message>
```

```
<wsdl:message name="IHostService_GetAsModels_OutputMessage">
  <wsdl:part name="parameters" element="tns:GetAsModelsResponse"/>
</wsdl:message>
<wsdl:message name="IHostService_ReleaseModels_InputMessage">
  <wsdl:part name="parameters" element="tns:ReleaseModels"/>
</wsdl:message>
<wsdl:message name="IHostService_ReleaseModels_OutputMessage">
  <wsdl:part name="parameters" element="tns:ReleaseModelsResponse"/>
</wsdl:message>
<wsdl:message name="IHostService_QueryModel_InputMessage">
  <wsdl:part name="parameters" element="tns:QueryModel"/>
</wsdl:message>
<wsdl:message name="IHostService_QueryModel_OutputMessage">
  <wsdl:part name="parameters" element="tns:QueryModelResponse"/>
</wsdl:message>
<wsdl:message name="IHostService_QueryInfoSet_InputMessage">
  <wsdl:part name="parameters" element="tns:QueryInfoSet"/>
</wsdl:message>
<wsdl:message name="IHostService_QueryInfoSet_OutputMessage">
  <wsdl:part name="parameters" element="tns:QueryInfoSetResponse"/>
</wsdl:message>
<wsdl:portType name="IHostService-20100825">
  <wsdl:operation name="GenerateUID">
    <wsdl:input wsaw:Action="http://dicom.nema.org/PS3.19/IHostService/GenerateUID"
      message="tns:IHostService_GenerateUID_InputMessage"/>
    <wsdl:output
wsaw:Action="http://dicom.nema.org/PS3.19/IHostService/GenerateUIDResponse"
      message="tns:IHostService_GenerateUID_OutputMessage"/>
    </wsdl:operation>
    <wsdl:operation name="GetAvailableScreen">
      <wsdl:input
wsaw:Action="http://dicom.nema.org/PS3.19/IHostService/GetAvailableScreen"
      message="tns:IHostService_GetAvailableScreen_InputMessage"/>
      <wsdl:output
wsaw:Action="http://dicom.nema.org/PS3.19/IHostService/GetAvailableScreenResponse"
      message="tns:IHostService_GetAvailableScreen_OutputMessage"/>
    </wsdl:operation>
    <wsdl:operation name="GetOutputLocation">
      <wsdl:input
wsaw:Action="http://dicom.nema.org/PS3.19/IHostService/GetOutputLocation"
      message="tns:IHostService_GetOutputLocation_InputMessage"/>
      <wsdl:output
wsaw:Action="http://dicom.nema.org/PS3.19/IHostService/GetOutputLocationResponse"
      message="tns:IHostService_GetOutputLocation_OutputMessage"/>
    </wsdl:operation>
    <wsdl:operation name="NotifyStateChanged">
      <wsdl:input
wsaw:Action="http://dicom.nema.org/PS3.19/IHostService/NotifyStateChanged"
      message="tns:IHostService_NotifyStateChanged_InputMessage"/>
      <wsdl:output
wsaw:Action="http://dicom.nema.org/PS3.19/IHostService/NotifyStateChangedResponse"
      message="tns:IHostService_NotifyStateChanged_OutputMessage"/>
    </wsdl:operation>
    <wsdl:operation name="NotifyStatus">
      <wsdl:input wsaw:Action="http://dicom.nema.org/PS3.19/IHostService/NotifyStatus"
      message="tns:IHostService_NotifyStatus_InputMessage"/>
      <wsdl:output
wsaw:Action="http://dicom.nema.org/PS3.19/IHostService/NotifyStatusResponse"
      message="tns:IHostService_NotifyStatus_OutputMessage"/>
    </wsdl:operation>
  </wsdl:portType>

```

```
</wsdl:operation>
<wsdl:operation name="NotifyDataAvailable">
  <wsdl:input
wsaw:Action="http://dicom.nema.org/PS3.19/IHostService/NotifyDataAvailable"
  message="tns:IHostService_NotifyDataAvailable_InputMessage" />
  <wsdl:output

wsaw:Action="http://dicom.nema.org/PS3.19/IHostService/NotifyDataAvailableResponse"
  message="tns:IHostService_NotifyDataAvailable_OutputMessage" />
</wsdl:operation>
<wsdl:operation name="GetData">
  <wsdl:input wsaw:Action="http://dicom.nema.org/PS3.19/IHostService/GetData"
  message="tns:IHostService_GetData_InputMessage" />
  <wsdl:output
wsaw:Action="http://dicom.nema.org/PS3.19/IHostService/GetDataResponse"
  message="tns:IHostService_GetData_OutputMessage" />
</wsdl:operation>
<wsdl:operation name="ReleaseData">
  <wsdl:input wsaw:Action="http://dicom.nema.org/PS3.19/IHostService/ReleaseData"
  message="tns:IHostService_ReleaseData_InputMessage" />
  <wsdl:output
wsaw:Action="http://dicom.nema.org/PS3.19/IHostService/ReleaseDataResponse"
  message="tns:IHostService_ReleaseData_OutputMessage" />
</wsdl:operation>
<wsdl:operation name="GetAsModels">
  <wsdl:input wsaw:Action="http://dicom.nema.org/PS3.19/IHostService/GetAsModels"
  message="tns:IHostService_GetAsModels_InputMessage" />
  <wsdl:output
wsaw:Action="http://dicom.nema.org/PS3.19/IHostService/GetAsModelsResponse"
  message="tns:IHostService_GetAsModels_OutputMessage" />
</wsdl:operation>
<wsdl:operation name="ReleaseModels">
  <wsdl:input wsaw:Action="http://dicom.nema.org/PS3.19/IHostService/ReleaseModels"
  message="tns:IHostService_ReleaseModels_InputMessage" />
  <wsdl:output
wsaw:Action="http://dicom.nema.org/PS3.19/IHostService/ReleaseModelsResponse"
  message="tns:IHostService_ReleaseModels_OutputMessage" />
</wsdl:operation>
<wsdl:operation name="QueryModel">
  <wsdl:input wsaw:Action="http://dicom.nema.org/PS3.19/IHostService/QueryModel"
  message="tns:IHostService_QueryModel_InputMessage" />
  <wsdl:output
wsaw:Action="http://dicom.nema.org/PS3.19/IHostService/QueryModelResponse"
  message="tns:IHostService_QueryModel_OutputMessage" />
</wsdl:operation>
<wsdl:operation name="QueryInfoSet">
  <wsdl:input wsaw:Action="http://dicom.nema.org/PS3.19/IHostService/QueryInfoSet"
  message="tns:IHostService_QueryInfoSet_InputMessage" />
  <wsdl:output
wsaw:Action="http://dicom.nema.org/PS3.19/IHostService/QueryInfoSetResponse"
  message="tns:IHostService_QueryInfoSet_OutputMessage" />
</wsdl:operation>
</wsdl:portType>
<wsdl:binding name="HostService-YYYYNNDDBinding" type="tns:IHostService-20100825">
  <soap:binding transport="http://schemas.xmlsoap.org/soap/http" />
  <wsdl:operation name="GenerateUID">
    <soap:operation soapAction="http://dicom.nema.org/PS3.19/IHostService/GenerateUID"
      style="document" />
    <wsdl:input>
      <soap:body use="literal" />
    </wsdl:input>
  </wsdl:operation>
</wsdl:binding>
</wsdl:service>
```

```
<wsdl:output>
  <soap:body use="literal"/>
</wsdl:output>
</wsdl:operation>
<wsdl:operation name="GetAvailableScreen">
  <soap:operation
soapAction="http://dicom.nema.org/PS3.19/IHostService/GetAvailableScreen"
  style="document"/>
  <wsdl:input>
    <soap:body use="literal"/>
  </wsdl:input>
  <wsdl:output>
    <soap:body use="literal"/>
  </wsdl:output>
</wsdl:operation>
<wsdl:operation name="GetOutputLocation">
  <soap:operation
soapAction="http://dicom.nema.org/PS3.19/IHostService/GetOutputLocation"
  style="document"/>
  <wsdl:input>
    <soap:body use="literal"/>
  </wsdl:input>
  <wsdl:output>
    <soap:body use="literal"/>
  </wsdl:output>
</wsdl:operation>
<wsdl:operation name="NotifyStateChanged">
  <soap:operation
soapAction="http://dicom.nema.org/PS3.19/IHostService/NotifyStateChanged"
  style="document"/>
  <wsdl:input>
    <soap:body use="literal"/>
  </wsdl:input>
  <wsdl:output>
    <soap:body use="literal"/>
  </wsdl:output>
</wsdl:operation>
<wsdl:operation name="NotifyStatus">
  <soap:operation
soapAction="http://dicom.nema.org/PS3.19/IHostService/NotifyStatus"
  style="document"/>
  <wsdl:input>
    <soap:body use="literal"/>
  </wsdl:input>
  <wsdl:output>
    <soap:body use="literal"/>
  </wsdl:output>
</wsdl:operation>
<wsdl:operation name="NotifyDataAvailable">
  <soap:operation
soapAction="http://dicom.nema.org/PS3.19/IHostService/NotifyDataAvailable"
  style="document"/>
  <wsdl:input>
    <soap:body use="literal"/>
  </wsdl:input>
  <wsdl:output>
    <soap:body use="literal"/>
  </wsdl:output>
</wsdl:operation>
<wsdl:operation name="GetData">
  <soap:operation soapAction="http://dicom.nema.org/PS3.19/IHostService/GetData"
```

```
        style="document" />
    <wsdl:input>
        <soap:body use="literal" />
    </wsdl:input>
    <wsdl:output>
        <soap:body use="literal" />
    </wsdl:output>
</wsdl:operation>
<wsdl:operation name="ReleaseData">
    <soap:operation soapAction="http://dicom.nema.org/PS3.19/IHostService/ReleaseData"
        style="document" />
    <wsdl:input>
        <soap:body use="literal" />
    </wsdl:input>
    <wsdl:output>
        <soap:body use="literal" />
    </wsdl:output>
</wsdl:operation>
<wsdl:operation name="GetAsModels">
    <soap:operation soapAction="http://dicom.nema.org/PS3.19/IHostService/GetAsModels"
        style="document" />
    <wsdl:input>
        <soap:body use="literal" />
    </wsdl:input>
    <wsdl:output>
        <soap:body use="literal" />
    </wsdl:output>
</wsdl:operation>
<wsdl:operation name="ReleaseModels">
    <soap:operation
soapAction="http://dicom.nema.org/PS3.19/IHostService/ReleaseModels"
        style="document" />
    <wsdl:input>
        <soap:body use="literal" />
    </wsdl:input>
    <wsdl:output>
        <soap:body use="literal" />
    </wsdl:output>
</wsdl:operation>
<wsdl:operation name="QueryModel">
    <soap:operation soapAction="http://dicom.nema.org/PS3.19/IHostService/QueryModel"
        style="document" />
    <wsdl:input>
        <soap:body use="literal" />
    </wsdl:input>
    <wsdl:output>
        <soap:body use="literal" />
    </wsdl:output>
</wsdl:operation>
<wsdl:operation name="QueryInfoSet">
    <soap:operation
soapAction="http://dicom.nema.org/PS3.19/IHostService/QueryInfoSet"
        style="document" />
    <wsdl:input>
        <soap:body use="literal" />
    </wsdl:input>
    <wsdl:output>
        <soap:body use="literal" />
    </wsdl:output>
</wsdl:operation>
</wsdl:binding>
```

```
<wsdl:service name="HostService-20100825">
  <wsdl:port name="HostServiceBinding" binding="tns:HostService-YYYYNNDDDBinding">
    <soap:address location="http://localhost/Service"/>
  </wsdl:port>
</wsdl:service>
</wsdl:definitions>
```

B.2.2 使用されたデータ構造の定義

B.2.2.1 主要な定義

下記はHostService-20100825.xsdのコンテンツである：

```
<?xml version="1.0" encoding="utf-8" ?>
<xs:schema xmlns:tns="http://dicom.nema.org/PS3.19/HostService-20100825"
  elementFormDefault="qualified"
  targetNamespace="http://dicom.nema.org/PS3.19/HostService-20100825"
  xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:import namespace="http://schemas.microsoft.com/2003/10/Serialization/Arrays"/>
  <xs:import namespace="http://schemas.datacontract.org/2004/07/System.Xml.XPath"/>
  <xs:element name="GenerateUID">
    <xs:complexType>
      <xs:sequence/>
    </xs:complexType>
  </xs:element>
  <xs:element name="GenerateUIDResponse">
    <xs:complexType>
      <xs:sequence>
        <xs:element minOccurs="0" name="GenerateUIDResult" nillable="true"
type="tns:UID"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:complexType name="UID">
    <xs:sequence>
      <xs:element minOccurs="0" name="Uid" nillable="true" type="xs:string"/>
    </xs:sequence>
  </xs:complexType>
  <xs:element name="UID" nillable="true" type="tns:UID"/>
  <xs:element name="GetAvailableScreen">
    <xs:complexType>
      <xs:sequence>
        <xs:element minOccurs="0" name="preferredScreen" nillable="true"
type="tns:Rectangle"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:complexType name="Rectangle">
    <xs:sequence>
      <xs:element minOccurs="0" name="Height" type="xs:int"/>
      <xs:element minOccurs="0" name="Width" type="xs:int"/>
      <xs:element minOccurs="0" name="RefPointX" type="xs:int"/>
      <xs:element minOccurs="0" name="RefPointY" type="xs:int"/>
    </xs:sequence>
  </xs:complexType>
  <xs:element name="Rectangle" nillable="true" type="tns:Rectangle"/>
  <xs:element name="GetAvailableScreenResponse">
    <xs:complexType>
      <xs:sequence>
        <xs:element minOccurs="0" name="GetAvailableScreenResult" nillable="true"
type="tns:Rectangle"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
```



```

    <xs:element minOccurs="0" name="ContextGroupVersion" nillable="true"
type="xs:string"/>
    <xs:element minOccurs="0" name="ContextGroupExtensionFlag" nillable="true"
type="xs:string"/>
    <xs:element minOccurs="0" name="ContextGroupLocalVersion" nillable="true"
type="xs:string"/>
    <xs:element minOccurs="0" name="ContextGroupExtensionCreatorUID" nillable="true"
type="xs:string"/>
  </xs:sequence>
</xs:complexType>
<xs:element name="Status" nillable="true" type="tns:Status"/>
<xs:simpleType name="StatusType">
  <xs:restriction base="xs:string">
    <xs:enumeration value="INFORMATION"/>
    <xs:enumeration value="WARNING"/>
    <xs:enumeration value="ERROR"/>
    <xs:enumeration value="FATALERROR"/>
  </xs:restriction>
</xs:simpleType>
<xs:element name="StatusType" nillable="true" type="tns:StatusType"/>
<xs:element name="NotifyStatusResponse">
  <xs:complexType>
    <xs:sequence/>
  </xs:complexType>
</xs:element>
<xs:element name="NotifyDataAvailable">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" name="data" nillable="true" type="tns:AvailableData"/>
      <xs:element minOccurs="0" name="lastData" type="xs:boolean"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:complexType name="AvailableData">
  <xs:sequence>
    <xs:element minOccurs="0" name="ObjectDescriptors" nillable="true"
      type="tns:ArrayOfObjectDescriptor"/>
    <xs:element minOccurs="0" name="Patients" nillable="true"
type="tns:ArrayOfPatient"/>
  </xs:sequence>
</xs:complexType>
<xs:element name="AvailableData" nillable="true" type="tns:AvailableData"/>
<xs:complexType name="ArrayOfObjectDescriptor">
  <xs:sequence>
    <xs:element minOccurs="0" maxOccurs="unbounded" name="ObjectDescriptor"
nillable="true"
      type="tns:ObjectDescriptor"/>
  </xs:sequence>
</xs:complexType>
<xs:element name="ArrayOfObjectDescriptor" nillable="true"
type="tns:ArrayOfObjectDescriptor"/>
<xs:complexType name="ObjectDescriptor">
  <xs:sequence>
    <xs:element minOccurs="0" name="ClassUID" nillable="true" type="tns:UID"/>
    <xs:element minOccurs="0" name="MimeType" nillable="true" type="tns:MimeType"/>
    <xs:element minOccurs="0" name="Modality" nillable="true" type="tns:Modality"/>
    <xs:element minOccurs="0" name="TransferSyntaxUID" nillable="true"
type="tns:UID"/>
    <xs:element minOccurs="0" name="DescriptorUuid" nillable="true" type="tns:UUID"/>
  </xs:sequence>
</xs:complexType>

```

```
<xs:element name="ObjectDescriptor" nillable="true" type="tns:ObjectDescriptor"/>
<xs:complexType name="MimeType">
  <xs:sequence>
    <xs:element minOccurs="0" name="Type" nillable="true" type="xs:string"/>
  </xs:sequence>
</xs:complexType>
<xs:element name="MimeType" nillable="true" type="tns:MimeType"/>
<xs:complexType name="Modality">
  <xs:sequence>
    <xs:element minOccurs="0" name="Modality" nillable="true" type="xs:string"/>
  </xs:sequence>
</xs:complexType>
<xs:element name="Modality" nillable="true" type="tns:Modality"/>
<xs:complexType name="UUID">
  <xs:sequence>
    <xs:element minOccurs="0" name="Uuid" nillable="true" type="xs:string"/>
  </xs:sequence>
</xs:complexType>
<xs:element name="UUID" nillable="true" type="tns:UUID"/>
<xs:complexType name="ArrayOfPatient">
  <xs:sequence>
    <xs:element minOccurs="0" maxOccurs="unbounded" name="Patient" nillable="true"
      type="tns:Patient"/>
  </xs:sequence>
</xs:complexType>
<xs:element name="ArrayOfPatient" nillable="true" type="tns:ArrayOfPatient"/>
<xs:complexType name="Patient">
  <xs:sequence>
    <xs:element minOccurs="0" name="AssigningAuthority" nillable="true"
type="xs:string"/>
    <xs:element minOccurs="0" name="DateOfBirth" type="xs:dateTime"/>
    <xs:element minOccurs="0" name="ID" nillable="true" type="xs:string"/>
    <xs:element minOccurs="0" name="Name" nillable="true" type="xs:string"/>
    <xs:element minOccurs="0" name="ObjectDescriptors" nillable="true"
      type="tns:ArrayOfObjectDescriptor"/>
    <xs:element minOccurs="0" name="Sex" nillable="true" type="xs:string"/>
    <xs:element minOccurs="0" name="Studies" nillable="true" type="tns:ArrayOfStudy"/>
  </xs:sequence>
</xs:complexType>
<xs:element name="Patient" nillable="true" type="tns:Patient"/>
<xs:complexType name="ArrayOfStudy">
  <xs:sequence>
    <xs:element minOccurs="0" maxOccurs="unbounded" name="Study" nillable="true"
type="tns:Study"
/>
  </xs:sequence>
</xs:complexType>
<xs:element name="ArrayOfStudy" nillable="true" type="tns:ArrayOfStudy"/>
<xs:complexType name="Study">
  <xs:sequence>
    <xs:element minOccurs="0" name="ObjectDescriptors" nillable="true"
      type="tns:ArrayOfObjectDescriptor"/>
    <xs:element minOccurs="0" name="Series" nillable="true" type="tns:ArrayOfSeries"/>
    <xs:element minOccurs="0" name="StudyUID" nillable="true" type="tns:UID"/>
  </xs:sequence>
</xs:complexType>
<xs:element name="Study" nillable="true" type="tns:Study"/>
<xs:complexType name="ArrayOfSeries">
  <xs:sequence>
    <xs:element minOccurs="0" maxOccurs="unbounded" name="Series" nillable="true"
      type="tns:Series"/>
  </xs:sequence>
</xs:complexType>
```

```

    </xs:sequence>
</xs:complexType>
<xs:element name="ArrayOfSeries" nillable="true" type="tns:ArrayOfSeries"/>
<xs:complexType name="Series">
  <xs:sequence>
    <xs:element minOccurs="0" name="ObjectDescriptors" nillable="true"
      type="tns:ArrayOfObjectDescriptor"/>
    <xs:element minOccurs="0" name="SeriesUID" nillable="true" type="tns:UID"/>
  </xs:sequence>
</xs:complexType>
<xs:element name="Series" nillable="true" type="tns:Series"/>
<xs:element name="NotifyDataAvailableResponse">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" name="NotifyDataAvailableResult" type="xs:boolean"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="GetData">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" name="objects" nillable="true"
type="tns:ArrayOfUUID"/>
      <xs:element minOccurs="0" name="acceptableTransferSyntaxes" nillable="true"
      type="tns:ArrayOfUID"/>
      <xs:element minOccurs="0" name="includeBulkData" type="xs:boolean"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:complexType name="ArrayOfUUID">
  <xs:sequence>
    <xs:element minOccurs="0" maxOccurs="unbounded" name="UUID" nillable="true"
type="tns:UUID"/>
  </xs:sequence>
</xs:complexType>
<xs:element name="ArrayOfUUID" nillable="true" type="tns:ArrayOfUUID"/>
<xs:complexType name="ArrayOfUID">
  <xs:sequence>
    <xs:element minOccurs="0" maxOccurs="unbounded" name="UID" nillable="true"
type="tns:UID"/>
  </xs:sequence>
</xs:complexType>
<xs:element name="ArrayOfUID" nillable="true" type="tns:ArrayOfUID"/>
<xs:element name="GetDataResponse">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" name="GetDataResult" nillable="true"
      type="tns:ArrayOfObjectLocator"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:complexType name="ArrayOfObjectLocator">
  <xs:sequence>
    <xs:element minOccurs="0" maxOccurs="unbounded" name="ObjectLocator"
nillable="true"
      type="tns:ObjectLocator"/>
  </xs:sequence>
</xs:complexType>
<xs:element name="ArrayOfObjectLocator" nillable="true"
type="tns:ArrayOfObjectLocator"/>
<xs:complexType name="ObjectLocator">

```

```
<xs:sequence>
  <xs:element minOccurs="0" name="Length" type="xs:long"/>
  <xs:element minOccurs="0" name="Offset" type="xs:long"/>
  <xs:element minOccurs="0" name="TransferSyntax" nillable="true" type="tns:UID"/>
  <xs:element minOccurs="0" name="URI" nillable="true" type="xs:anyURI"/>
  <xs:element minOccurs="0" name="Locator" nillable="true" type="tns:UUID"/>
  <xs:element minOccurs="0" name="Source" nillable="true" type="tns:UUID"/>
</xs:sequence>
</xs:complexType>
<xs:element name="ObjectLocator" nillable="true" type="tns:ObjectLocator"/>
<xs:element name="ReleaseData">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" name="objects" nillable="true"
type="tns:ArrayOfUUID" />
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="ReleaseDataResponse">
  <xs:complexType>
    <xs:sequence/>
  </xs:complexType>
</xs:element>
<xs:element name="GetAsModels">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" name="objects" nillable="true"
type="tns:ArrayOfUUID" />
      <xs:element minOccurs="0" name="classUID" nillable="true" type="tns:UID"/>
      <xs:element minOccurs="0" name="supportedInfoSetTypes" nillable="true"
type="tns:ArrayOfMimeType" />
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:complexType name="ArrayOfMimeType">
  <xs:sequence>
    <xs:element minOccurs="0" maxOccurs="unbounded" name="MimeType" nillable="true"
type="tns:MimeType" />
  </xs:sequence>
</xs:complexType>
<xs:element name="ArrayOfMimeType" nillable="true" type="tns:ArrayOfMimeType"/>
<xs:element name="GetAsModelsResponse">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" name="GetAsModelsResult" nillable="true"
type="tns:ModelSetDescriptor" />
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:complexType name="ModelSetDescriptor">
  <xs:sequence>
    <xs:element minOccurs="0" name="FailedSourceObjects" nillable="true"
type="tns:ArrayOfUUID" />
    <xs:element minOccurs="0" name="InfosetType" nillable="true" type="tns:MimeType"/>
    <xs:element minOccurs="0" name="Models" nillable="true" type="tns:ArrayOfUUID"/>
  </xs:sequence>
</xs:complexType>
<xs:element name="ModelSetDescriptor" nillable="true" type="tns:ModelSetDescriptor"/>
<xs:element name="ReleaseModels">
  <xs:complexType>
    <xs:sequence>
```

```

        <xs:element minOccurs="0" name="models" nillable="true" type="tns:ArrayOfUUID"/>
    </xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="ReleaseModelsResponse">
    <xs:complexType>
        <xs:sequence/>
    </xs:complexType>
</xs:element>
<xs:element name="QueryModel">
    <xs:complexType>
        <xs:sequence>
            <xs:element minOccurs="0" name="models" nillable="true" type="tns:ArrayOfUUID"/>
            <xs:element minOccurs="0" name="xPaths" nillable="true"
                xmlns:q2="http://schemas.microsoft.com/2003/10/Serialization/Arrays"
                type="q2:ArrayOfstring"/>
        </xs:sequence>
    </xs:complexType>
</xs:element>
<xs:element name="QueryModelResponse">
    <xs:complexType>
        <xs:sequence>
            <xs:element minOccurs="0" name="QueryModelResult" nillable="true"
                type="tns:ArrayOfQueryResult"/>
        </xs:sequence>
    </xs:complexType>
</xs:element>
<xs:complexType name="ArrayOfQueryResult">
    <xs:sequence>
        <xs:element minOccurs="0" maxOccurs="unbounded" name="QueryResult" nillable="true"
            type="tns:QueryResult"/>
    </xs:sequence>
</xs:complexType>
<xs:element name="ArrayOfQueryResult" nillable="true" type="tns:ArrayOfQueryResult"/>
<xs:complexType name="QueryResult">
    <xs:sequence>
        <xs:element minOccurs="0" name="Model" nillable="true" type="tns:UUID"/>
        <xs:element minOccurs="0" name="Result" nillable="true"
            type="tns:ArrayOfXPathNode"/>
        <xs:element minOccurs="0" name="XPath" nillable="true" type="xs:string"/>
    </xs:sequence>
</xs:complexType>
<xs:element name="QueryResult" nillable="true" type="tns:QueryResult"/>
<xs:complexType name="ArrayOfXPathNode">
    <xs:sequence>
        <xs:element minOccurs="0" maxOccurs="unbounded" name="XPathNode" nillable="true"
            type="tns:XPathNode"/>
    </xs:sequence>
</xs:complexType>
<xs:element name="ArrayOfXPathNode" nillable="true" type="tns:ArrayOfXPathNode"/>
<xs:complexType name="XPathNode">
    <xs:sequence>
        <xs:element minOccurs="0" name="NodeType"
            xmlns:q3="http://schemas.datacontract.org/2004/07/System.Xml.XPath"
            type="q3:XPathNodeType"/>
        <xs:element minOccurs="0" name="Value" nillable="true" type="xs:string"/>
    </xs:sequence>
</xs:complexType>
<xs:element name="XPathNode" nillable="true" type="tns:XPathNode"/>
<xs:element name="QueryInfoSet">
    <xs:complexType>

```

```
<xs:sequence>
  <xs:element minOccurs="0" name="models" nillable="true" type="tns:ArrayOfUUID"/>
  <xs:element minOccurs="0" name="xPaths" nillable="true"
    xmlns:q4="http://schemas.microsoft.com/2003/10/Serialization/Arrays"
    type="q4:ArrayOfstring"/>
</xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="QueryInfoSetResponse">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" name="QueryInfoSetResult" nillable="true"
        type="tns:ArrayOfQueryResultInfoSet"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:complexType name="ArrayOfQueryResultInfoSet">
  <xs:sequence>
    <xs:element minOccurs="0" maxOccurs="unbounded" name="QueryResultInfoSet"
nillable="true"
      type="tns:QueryResultInfoSet"/>
  </xs:sequence>
</xs:complexType>
<xs:element name="ArrayOfQueryResultInfoSet" nillable="true"
type="tns:ArrayOfQueryResultInfoSet"/>
<xs:complexType name="QueryResultInfoSet">
  <xs:sequence>
    <xs:element minOccurs="0" name="Model" nillable="true" type="tns:UUID"/>
    <xs:element minOccurs="0" name="Result" nillable="true"
type="tns:ArrayOfXPathNodeInfoSet"/>
    <xs:element minOccurs="0" name="XPath" nillable="true" type="xs:string"/>
  </xs:sequence>
</xs:complexType>
<xs:element name="QueryResultInfoSet" nillable="true" type="tns:QueryResultInfoSet"/>
<xs:complexType name="ArrayOfXPathNodeInfoSet">
  <xs:sequence>
    <xs:element minOccurs="0" maxOccurs="unbounded" name="XPathNodeInfoSet"
nillable="true"
      type="tns:XPathNodeInfoSet"/>
  </xs:sequence>
</xs:complexType>
<xs:element name="ArrayOfXPathNodeInfoSet" nillable="true"
type="tns:ArrayOfXPathNodeInfoSet"/>
<xs:complexType name="XPathNodeInfoSet">
  <xs:sequence>
    <xs:element minOccurs="0" name="InfoSetValue" nillable="true"
type="xs:base64Binary"/>
    <xs:element minOccurs="0" name="NodeType"
      xmlns:q5="http://schemas.datacontract.org/2004/07/System.Xml.XPath"
type="q5:XPathNodeType"
      />
  </xs:sequence>
</xs:complexType>
<xs:element name="XPathNodeInfoSet" nillable="true" type="tns:XPathNodeInfoSet"/>
</xs:schema>
```

B.2.2.2 参照が付けられた定義

下記はXPathNodeType.xsdのコンテンツである：

```
<!--xml version="1.0" encoding="utf-8"-->
<xs:schema xmlns:tns="http://schemas.datacontract.org/2004/07/System.Xml.XPath"
elementFormDefault="qualified"
targetNamespace="http://schemas.datacontract.org/2004/07/System.Xml.XPath"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:simpleType name="XPathNodeType">
    <xs:restriction base="xs:string">
      <xs:enumeration value="Root" />
      <xs:enumeration value="Element" />
      <xs:enumeration value="Attribute" />
      <xs:enumeration value="Namespace" />
      <xs:enumeration value="Text" />
      <xs:enumeration value="SignificantWhitespace" />
      <xs:enumeration value="Whitespace" />
      <xs:enumeration value="ProcessingInstruction" />
      <xs:enumeration value="Comment" />
      <xs:enumeration value="All" />
    </xs:restriction>
  </xs:simpleType>
  <xs:element name="XPathNodeType" nillable="true" type="tns:XPathNodeType" />
</xs:schema>
```

下記はTypes.xsdのコンテンツである：

```
<!--xml version="1.0" encoding="utf-8"-->
<xs:schema xmlns:tns="http://schemas.microsoft.com/2003/10/Serialization/"
attributeFormDefault="qualified" elementFormDefault="qualified"
targetNamespace="http://schemas.microsoft.com/2003/10/Serialization/"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="anyType" nillable="true" type="xs:anyType" />
  <xs:element name="anyURI" nillable="true" type="xs:anyURI" />
  <xs:element name="base64Binary" nillable="true" type="xs:base64Binary" />
  <xs:element name="boolean" nillable="true" type="xs:boolean" />
  <xs:element name="byte" nillable="true" type="xs:byte" />
  <xs:element name="dateTime" nillable="true" type="xs:dateTime" />
  <xs:element name="decimal" nillable="true" type="xs:decimal" />
  <xs:element name="double" nillable="true" type="xs:double" />
  <xs:element name="float" nillable="true" type="xs:float" />
  <xs:element name="int" nillable="true" type="xs:int" />
  <xs:element name="long" nillable="true" type="xs:long" />
  <xs:element name="QName" nillable="true" type="xs:QName" />
  <xs:element name="short" nillable="true" type="xs:short" />
  <xs:element name="string" nillable="true" type="xs:string" />
  <xs:element name="unsignedByte" nillable="true" type="xs:unsignedByte" />
  <xs:element name="unsignedInt" nillable="true" type="xs:unsignedInt" />
  <xs:element name="unsignedLong" nillable="true" type="xs:unsignedLong" />
  <xs:element name="unsignedShort" nillable="true" type="xs:unsignedShort" />
  <xs:element name="char" nillable="true" type="tns:char" />
  <xs:simpleType name="char">
    <xs:restriction base="xs:int" />
  </xs:simpleType>
  <xs:element name="duration" nillable="true" type="tns:duration" />
  <xs:simpleType name="duration">
    <xs:restriction base="xs:duration">
      <xs:pattern value="--P(\d*D)-(T(\d*H)-(\d*M)-(\d*(\.\d*)-S)-)" />
      <xs:minInclusive value="-P10675199DT2H48M5.4775808S" />
      <xs:maxInclusive value="P10675199DT2H48M5.4775807S" />
    </xs:restriction>
  </xs:simpleType>
  <xs:element name="guid" nillable="true" type="tns:guid" />
```

```
<xs:simpleType name="guid">
  <xs:restriction base="xs:string">
    <xs:pattern value="\da-fA-F}{8}-[\da-fA-F]{4}-[\da-fA-F]{4}-[\da-fA-F]{4}-[\da-fA-F]{12}" />
  </xs:restriction>
</xs:simpleType>
<xs:attribute name="FactoryType" type="xs:QName" />
<xs:attribute name="Id" type="xs:ID" />
<xs:attribute name="Ref" type="xs:IDREF" />
</xs:schema>
```

下記はArrayOfString.xsdのコンテンツである :

```
<-xml version="1.0" encoding="utf-8"->
<xs:schema xmlns:tns="http://schemas.microsoft.com/2003/10/Serialization/Arrays"
  elementFormDefault="qualified"
  targetNamespace="http://schemas.microsoft.com/2003/10/Serialization/Arrays"
  xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:complexType name="ArrayOfstring">
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="unbounded" name="string" nillable="true"
        type="xs:string" />
    </xs:sequence>
  </xs:complexType>
  <xs:element name="ArrayOfstring" nillable="true" type="tns:ArrayOfstring" />
</xs:schema>
```