

**TCVN**

**TIÊU CHUẨN QUỐC GIA**

**TCVN 7981 - 3 : 2009**

**ISO/TS 17369 - 3 : 2005**

Xuất bản lần 1

**TRAO ĐỔI SIÊU DỮ LIỆU VÀ DỮ LIỆU THỐNG KÊ  
PHẦN 3: LƯỢC ĐỒ VÀ TÀI LIỆU SDMX-ML**

*Statistical data and metadata exchange*

*Section 3: SDMX-ML: Schema and documentation*

**HÀ NỘI - 2009**

<b>Mục lục</b>	<b>Trang</b>
Lời nói đầu.....	5
1 Bối cảnh .....	7
1.1 XML trong các dự án nghiên cứu tình huống và trao đổi dữ liệu theo lô .....	7
1.2 Kết quả: Thiết kế XML.....	7
1.3 Thúc đẩy việc sử dụng tiêu chuẩn SDMX-ML.....	8
2 Tài liệu viện dẫn.....	8
3 Sự phù hợp.....	8
4 Tổng quan về thiết kế.....	8
4.1 Phạm vi áp dụng và các yêu cầu .....	8
4.2 Phương pháp thiết kế.....	10
4.3 Đóng gói SDMX-ML: Các môđun tên miền.....	11
5 Lược đồ (phi tập khóa) chung.....	12
5.1 Môđun tên miền SDMX .....	13
5.1.1 Các phần tử toàn cục.....	13
5.1.2 Các kiểu phức .....	14
5.1.3 Kiểu đơn giản .....	17
5.2 Môđun tên miền của cấu trúc SDMX.....	18
5.2.1 Kiểu phức .....	18
5.2.2 Kiểu đơn giản .....	25
5.3 Môđun tên miền dữ liệu SDMX chung.....	26
5.3.1 Phần tử chung.....	26
5.3.2 Kiểu phức .....	26
5.4 Môđun tên miền về truy vấn SDMX.....	28
5.4.1 Phần tử chung.....	28
5.4.2 Kiểu phức .....	28
5.4.3 Kiểu đơn giản .....	33
5.5 Môđun tên miền chung về SDMX.....	33
5.5.1 Kiểu phức .....	33
5.5.2 Kiểu đơn giản .....	34

## TCVN 7981-3 : 2009

5.6	Định dạng dữ liệu và mã hóa ký tự.....	34
6	Lược đồ tập khóa chuyên dụng: Cấu trúc chính và ánh xạ chuẩn.....	34
6.1	Cấu trúc chính của thông điệp Compact Data (dữ liệu cô đọng).....	35
6.1.1	Phần tử toàn cục.....	35
6.1.2	Kiểu phức.....	35
6.2	Cấu trúc chính của thông điệp Utility Data (dữ liệu tiện ích).....	35
6.2.1	Phần tử toàn cục.....	36
6.2.2	Kiểu phức.....	36
6.3	Cấu trúc chính của thông điệp Cross-Sectional Data (dữ liệu phân giao).....	36
6.3.1	Phần tử toàn cục.....	36
6.3.2	Kiểu phức.....	37
6.4	Ánh xạ tới các lược đồ về tập khóa.....	37
6.4.1	Các quy tắc chung:.....	37
6.4.2	Lược đồ Compact (cô đọng):.....	38
6.4.3	Lược đồ Cross-Sectional (phân giao).....	41
6.4.4	Lược đồ Utility (tiện ích).....	44
7	Phụ lục: Các thông điệp mẫu về SDMX - ML.....	47
7.1	CompactSample.xml (Tập mẫu về dữ liệu cô đọng).....	47
7.2	UtilitySample.xml (Tập mẫu về dữ liệu tiện ích).....	49
7.3	GenericSample.xml (Tập mẫu về dữ liệu chung).....	50
7.4	CrossSectionalSample.xml (Tập mẫu về dữ liệu phân giao).....	50

## Lời nói đầu

**TCVN 7981 - 3 : 2009** hoàn toàn tương đương với **ISO 17369 - 3 : 2005**

**TCVN 7981 - 3 : 2009** do Ban Kỹ thuật Tiêu chuẩn quốc gia TCVN/TC 154 "*Quá trình, các yếu tố dữ liệu và tài liệu trong thương mại, công nghiệp và hành chính*" biên soạn, Tổng cục Tiêu chuẩn Đo lường Chất lượng đề nghị, Bộ Khoa học và Công nghệ công bố.

Bộ tiêu chuẩn TCVN 7981 (ISO 17369) gồm các phần sau:

- TCVN 7981-1 : 2008 (ISO/TS 17369 - 1 : 2005); Phần 1: Khung tổng quát về các tiêu chuẩn SDMX.
- TCVN 7981-2 : 2008 (ISO/TS 17369 - 2 : 2005); Phần 2: Mô hình thông tin: Thiết kế khái niệm UML.
- TCVN 7981-3 : 2009 (ISO/TS 17369 - 3 : 2005); Phần 3: Lược đồ và tài liệu SDMX-ML.
- TCVN 7981-4 : 2009 (ISO/TS 17369 - 4 : 2005); Phần 4: Cú pháp và tài liệu SDMX-EDI.
- TCVN 7981-5 : 2009 (ISO/TS 17369 - 5 : 2005); Phần 5: Hướng dẫn thực thi tiêu chuẩn định dạng SDMX.
- TCVN 7981-6 : 2009 (ISO/TS 17369 - 6 : 2005); Phần 6: Hướng dẫn SDMX sử dụng các dịch vụ web.

## Trao đổi siêu dữ liệu và dữ liệu thống kê

### Phần 3: Lược đồ và tài liệu SDMX-ML

*Statistical data and metadata exchange*

*Section 3: SDMX-ML: Schema and documentation*

#### 1 Bối cảnh

##### 1.1 XML trong các dự án nghiên cứu tình huống và trao đổi dữ liệu theo lô

Theo tiến trình của các dự án trao đổi dữ liệu theo lô (Batch Data Exchange - BDE) và nghiên cứu tình huống, có hai lược đồ được xây dựng, cả hai lược đồ này đều dựa trên cơ sở mô hình thông tin theo đặc tả GESMES/TS. Vì vậy, chúng có nhiều khía cạnh giống nhau. Tuy nhiên, cũng có các đặc điểm khác biệt do sự khác nhau về yêu cầu kỹ thuật của hai dự án trên.

- XML trong BDE được tối ưu hóa để trao đổi các tập dữ liệu lớn theo lô. Nó hỗ trợ đúng kiểu trao đổi đã thiết kế cho GESMES/TS, không phải để hỗ trợ cú pháp XML.
- XML trong nghiên cứu tình huống được thiết kế và tối ưu hóa để hỗ trợ việc phổ biến web và cung cấp kiến trúc chia sẻ dữ liệu trên sổ đăng ký.

Rõ ràng, XML đơn thích hợp với nhiều phương pháp tiếp cận, điều đó thúc đẩy việc xây dựng tiêu chuẩn về SDMX-ML trong thời gian sớm nhất có thể.

Xem xét các yêu cầu kết hợp của tất cả các quá trình được hỗ trợ bởi công việc trước đây, thấy rằng việc một kiểu tài liệu đơn không phải là phương pháp tiếp cận tốt nhất. Tất cả các sản phẩm công nghệ SDMX (XML và các định dạng dữ liệu EDIFACT, sổ đăng ký, v.v) chia sẻ một mô hình thông tin, do đó mang cùng thông tin. Điều này đưa đến kết quả là một thiết kế XML ứng với năm hoặc sáu kiểu tài liệu biết trước.

##### 1.2 Kết quả: Thiết kế XML

Tất cả kiểu tài liệu này chia sẻ một "phong bì" chung ở mức thông điệp ("`SDMXMessage.xsd`"), cũng như một tập chung các thành phần ở mức thấp ("`SDMXCommon.xsd`"), do vậy, thông tin tiêu đề và cấu trúc cơ bản luôn giống nhau.

- Lược đồ mô tả cấu trúc tập khóa ("`SDMXStructure.xsd`")

## TCVN 7981-3 : 2009

- Lược đồ dữ liệu chung cho trao đổi dữ liệu chia sẻ ("SDMXGenericData.xsd")
- Lược đồ truy vấn chung để gọi các dịch vụ web ("SDMXQuery.xsd")
- Lược đồ tập khóa chuyên dụng đối với trao đổi song phương/cập nhật và sửa đổi("SDMXCompactData.xsd")
- Lược đồ tập khóa chuyên dụng cho việc xử lý cách trình bày và sử dụng bên trong ("SDMXUtilityData.xsd")
- Yêu cầu: Lược đồ tập khóa chuyên dụng về dữ liệu phần giao - được kết hợp với kiểu tài liệu cô đọng ("SDMXCrossSectionalData.xsd")

### 1.3 Thúc đẩy việc sử dụng tiêu chuẩn SDMX-ML

Ngoài các định dạng khác nhau này, các ánh xạ chuẩn và công cụ biến đổi tương ứng được xây dựng để tạo ra các lược đồ tập khóa chuyên dụng từ các mô tả cấu trúc và để biến đổi các trường hợp dữ liệu XML cụ thể từ một định dạng mô tả dữ liệu XML sang một định dạng khác và từ các định dạng này thành các thông điệp SDMX-ML tương ứng. Mức hỗ trợ các công cụ này thúc đẩy việc sử dụng SDMX, cho phép dữ liệu được sử dụng dễ dàng qua toàn bộ các quá trình và đáp ứng các yêu cầu phức tạp. Cuối cùng, tất cả định dạng chia sẻ Mô hình thông tin chung cho phép cách tiếp cận này đáp ứng một tập rộng các yêu cầu của SDMX.

## 2 Tài liệu viện dẫn

Các tài liệu viện dẫn sau là rất cần thiết cho việc áp dụng tiêu chuẩn. Đối với các tài liệu viện dẫn ghi năm ban hành thì áp dụng bản được nêu. Đối với các tài liệu viện dẫn không ghi năm ban hành thì áp dụng phiên bản mới nhất, bao gồm cả các sửa đổi.

Ngôn ngữ xác định lược đồ W3 XML, phiên bản 1.0 (URL:<http://www.w3c.org/XML/Schema#dev>),

Ngôn ngữ đánh dấu mở rộng W3C, phiên bản 1.0, xuất bản lần 3

(URL:<http://www.w3c.org/TR/2004/REC-xml-20040204/>)

## 3 Sự phù hợp

Các Điều 5 và 6 của tiêu chuẩn này mang tính quy định, cung cấp quy tắc tạo ra các trường hợp XML SDMX-ML cụ thể và các lược đồ XML W3C.

## 4 Tổng quan thiết kế

### 4.1 Phạm vi và yêu cầu

Để hiểu mối quan hệ giữa một số kiểu tài liệu, điều quan trọng là phải nắm rõ các yêu cầu mà chúng được thiết kế để hoàn thành công việc. Theo truyền thống, GESMES/TS (trước đây là GESMES/CB) được tạo ra để trao đổi lượng lớn dữ liệu giữa các bên đối tác. Việc sử dụng định dạng dữ liệu này thể hiện một số yêu cầu mà SDMX-ML chấp nhận chúng như chính yêu cầu của mình, đó là một trong các trường hợp sử dụng được yêu cầu để hỗ trợ:

- Lượng lớn dữ liệu phải đạt được trong một định dạng cô đọng hợp lý, bởi kích cỡ tiềm tàng của các cơ sở dữ liệu đang được trao đổi.

- GESMES/TS phải có khả năng gửi các cập nhật gia tăng, hơn là gửi toàn bộ và các cơ sở dữ liệu đầy đủ. Việc kiểm tra tính hợp lệ các trao đổi như vậy không đòi hỏi trao đổi toàn bộ tập dữ liệu, mà chỉ cần gửi thông tin đủ để đảm bảo các quá trình cập nhật và sửa đổi không chính xác.
- Thông tin có cấu trúc cũng như dữ liệu cần được truyền.
- Phải có một phép biến đổi đáng tin cậy sang và từ cú pháp GESMES/TS EDIFACT.
- Nên có khả năng trình bày thông tin trong ngôn ngữ tự nhiên bằng các ngôn ngữ tương đương hoặc đa ngôn ngữ.

Đây là một tập các yêu cầu mà định dạng XML trong BDE được thiết kế để đáp ứng chúng. Về bản chất, các kiểu trao đổi này có xu hướng trao đổi song phương (hoặc các trao đổi bằng "cổng nối", mà mức độ tiêu chuẩn hóa được đặt ở tập các trao đổi song phương). Trong các kiểu trao đổi này, cả hai bên đối tác đã thảo luận quá trình trao đổi và các tập khóa được sử dụng, để không gặp bất kỳ khó khăn nào.

SDMX-ML đối mặt với tập các yêu cầu lớn hơn. Tuy nhiên, Một trong các yêu cầu lớn nhất là hỗ trợ việc phổ biến web, trong đó, không có một bên đối tác nào, đúng hơn là không có nhà cung cấp dữ liệu và người sử dụng dữ liệu. Các vai trò này không có quan hệ cần thiết ngoài việc trao đổi dữ liệu đơn, đó đó, có thể ảnh hưởng đến việc hiểu quá trình phổ biến và các tập khóa được sử dụng, v.v. Ngoài ra, SDMX-ML được thiết kế để hỗ trợ việc sử dụng XML trong kiến trúc trung tâm sổ đăng ký, sử dụng công nghệ dịch vụ web. Các trường hợp sử dụng này đi cùng với các yêu cầu bổ sung cho các trường hợp sử dụng của trao đổi song phương và việc cập nhật các cơ sở dữ liệu:

- Để hỗ trợ các dịch vụ web và các phương pháp công nghệ tương tự, có một yêu cầu gửi các truy vấn đến nguồn thông tin cũng như dữ liệu và cấu trúc.
- Người sử dụng (và các dịch vụ sổ đăng ký) không biết rõ về tập khóa cụ thể, vì vậy họ phải điều khiển dữ liệu qua các tập khóa và thậm chí (dịch vụ so sánh) để đặt dữ liệu cấu trúc theo nhiều tập khóa trong trường hợp XML đơn.
- XML càng đơn giản càng tốt để cho phép người sở hữu web và người xây dựng web chưa quen với việc coi thống kê như một miền có thể sử dụng dễ dàng.
- Nên sử dụng XML càng "bình thường" càng tốt trong các công cụ XML tiêu chuẩn như môi trường phát triển web, phân tích cú pháp, công cụ soạn thảo v.v.
- Việc kiểm tra tính hợp lệ của các tập dữ liệu nên kiểm tra tính đầy đủ của tập dữ liệu đó - Hồ sơ kiểm tra tính hợp lệ cho các cập nhật gia tăng là không đầy đủ.
- Dữ liệu nên được cấu trúc không chỉ theo dữ liệu chuỗi thời gian, mà còn theo dữ liệu phân giao để đáp ứng các nhu cầu của người sử dụng khác nhau. Phải có khả năng lấy dữ liệu cấu trúc theo tập khóa đơn và biến đổi dữ liệu đó thành định dạng chuẩn cho phép một trong hai cách tối ưu hóa cấu trúc này.

- Các định dạng XML thúc đẩy việc tái sử dụng các ngữ nghĩa, khái niệm và danh sách mã chung trong phạm vi lớn nhất có thể, trong khi vẫn công nhận cơ quan duy trì tài nguyên cụ thể (danh sách mã, tập khóa, tập dữ liệu, v.v.).

Đây là một tập các yêu cầu rất rộng, trong quá trình khảo sát nghiên cứu, rõ ràng thấy rằng một số trong các yêu cầu đó rất dễ gây hiểu lầm. Hầu như không thể thiết kế một kiểu tài liệu XML đơn thỏa mãn tất cả các yêu cầu này. Đồng thời, mọi thiết kế được chấp nhận nên có quan hệ rõ ràng với Mô hình thông tin, để người sử dụng có thể dễ dàng hiểu ý tưởng của tập khóa và quan hệ của nó với dữ liệu thống kê.

#### **4.2 Phương pháp thiết kế**

Một trong các khía cạnh lớn nhất của hướng dẫn thực hiện GESMES/TS là mô hình dữ liệu của nó, cho phép thông điệp EDIFACT được sử dụng cho nhiều kiểu dữ liệu khác nhau. Thiết kế XML xây dựng dựa trên phương pháp này bằng cách mở rộng việc sử dụng của mô hình nhằm bao trùm không chỉ các kiểu dữ liệu thống kê được thể hiện như các tập khóa mà còn các cú pháp. Tập khóa là kết cấu siêu dữ liệu - nó được thể hiện trong nhiều cú pháp, nhưng không phụ thuộc vào chúng. Xem xét ý tưởng sử dụng mô hình thông tin SDMX (tập mô hình dữ liệu GESMES/TS) nhằm bao trùm các cú pháp, thấy rằng phương pháp tương tự có thể được sử dụng để bao trùm các định dạng XML về trường hợp sử dụng. Do chúng được dựa trên cùng một mô hình nên tính tương đương được bảo đảm. Đối với phép biến đổi đơn giản, dữ liệu bất kỳ được biểu diễn trong EDIFACT hoặc XML quá trình-cụ thể có thể được biến đổi thành thông tin ưa thích của bên nhận dữ liệu. Thêm nữa, từ mô tả có thể xử lý của tập khóa (mô tả XML), nó có khả năng tạo ra các mô tả định dạng, công cụ và cấu hình đặc trưng cho tập khóa đó.

Lí do chính chống lại phương pháp này là tính phức tạp của nó, đây là nhân tố tiêu cực khi ban bố các tiêu chuẩn quốc tế. Ngoài ra, nhìn vào các yêu cầu, nhận thấy rằng không chỉ cần các định dạng XML của tập khóa chuyên dụng mà còn cần các định dạng có thể cung cấp nhiều tập khóa mà không phải không thay đổi - đó là định dạng phi tập khóa chuyên dụng.

Kết quả của phân tích này là ý tưởng về quan điểm thỏa thuận. Ngay khi có thỏa thuận rằng; chỉ có một định dạng XML cho việc mô tả tập khóa - nhiều hơn là không cần thiết. Thì một yêu cầu tồn tại nhằm giúp cho các dịch vụ có thể sử dụng dữ liệu cấu trúc theo mọi tập khóa và đôi khi là trong tổ hợp của chúng. Điều này thể hiện nhu cầu về định dạng dữ liệu "chung". Việc truy vấn yêu cầu này đòi hỏi tạo ra một thông điệp Query (truy vấn) (trước đây, đã được thảo luận trong cộng đồng GESMES/TS, mặc dù chưa được hoàn thành). Ngoài ra, nhận thấy rằng, có ít nhất hai, có thể là ba kịch bản mâu thuẫn các yêu cầu về mặt thiết kế XML:

- Trao đổi, cập nhật và sửa đổi cơ sở dữ liệu;
- Sử dụng và xử lý XML "thông thường" đối với người sở hữu web, người phát triển và người sử dụng các công cụ XML điển hình;
- Trao đổi dữ liệu phân giao (có thể giống với kịch bản trao đổi cơ sở dữ liệu)

Để hỗ trợ tập rộng các yêu cầu, thì số lượng nhỏ các tài liệu tiêu chuẩn nên khớp với nhau để đáp



ứng các yêu cầu xử lý cụ thể. Điều này bao gồm ba kịch bản được mô tả ở trên và việc sử dụng kiểu tài liệu truy vấn, được yêu cầu cho việc phát triển các dịch vụ web và các ứng dụng tương tự liên quan đến việc tạo dữ liệu SDMX-ML trong thời gian chạy từ các cơ sở dữ liệu.

Trong thiết kế này vẫn duy trì ý tưởng tái sử dụng, bất cứ nơi nào có thể, các cấu trúc chung được tái sử dụng. Điều này dẫn đến một cấu trúc "thông điệp" chung, trong đó, có một tiêu đề được chia sẻ bởi tất cả tài liệu và một "phong bì" đơn (không nhầm lẫn với phong bì SOAP của các dịch vụ web, chứa toàn bộ các thông điệp SDMX-ML của mỗi loại). Hơn nữa, cấu trúc chính của kiểu tài liệu XML của tập khóa chuyên dụng giống với cấu trúc của kiểu tài liệu khác, nhằm mục đích mở rộng tối đa có thể. Để tăng tính nhất quán, nên sử dụng các định dạng XML với các cấu trúc xác định.

Kết quả cuối cùng là sự phân chia chủ yếu giữa các định dạng XML "chung" không gắn với các tập khóa riêng với tập các định dạng đặc trưng cho các tập khóa và kịch bản sử dụng riêng.

Các quyết định thiết kế này thực hiện khi đối tượng được trình bày giống như một phần tử hoặc thuộc tính XML được tạo ra dựa trên các yêu cầu cụ thể về mỗi định dạng XML. Đối với các định dạng mà tính cô đọng của dữ liệu là hết sức quan trọng, thì hầu hết được trình bày như các thuộc tính, bởi điều này dẫn tới việc trình bày dữ liệu cô đọng hơn. Trong các trường hợp khác - ví dụ trong các thông điệp UtilityData (dữ liệu tiện ích), các kiểu cấu trúc khác được sử dụng có cấu trúc dài dòng hơn, nhưng lại trình bày nhiều siêu dữ liệu hơn trong tập khóa (ví dụ: thứ tự khóa). Kiểu khác nhau trong các thân thiết kế này luôn xuất phát từ các yêu cầu về định dạng XML cụ thể đang được thiết kế.

### 4.3 Đóng gói SDMX-ML: Các mô đun tên miền

Trong thiết kế lược đồ XML đề xuất, có lược đồ đóng gói dựa trên ý tưởng rằng các tên miền XML có thể được sử dụng như "các mô đun", giúp người sử dụng hay ứng dụng bất kỳ chỉ cần quen với tập con của toàn bộ các thư viện có thể sử dụng nó. Phương pháp này rất phù hợp với thiết kế mô tả ở trên và thường được sử dụng chủ yếu trong các tiêu chuẩn XML về các miền khác.

Lợi ích chủ yếu của các tên miền - đặc biệt trong việc làm sáng tỏ yêu cầu rằng các cơ quan duy trì được tìm kiếm qua việc tái sử dụng các cấu trúc và dữ liệu họ duy trì - cho phép SDMX sở hữu các mô đun tên miền nào đó và cho phép các cơ quan duy trì khác sở hữu các tên miền đặc trưng cho các tập khóa họ duy trì.

Kết quả là một tập các gói tên miền nhất trí với phương pháp thiết kế mô tả ở trên. Mỗi mô đun là một trường hợp đơn lẻ của phần tử `schema` trong ngôn ngữ lược đồ XML của W3C. Ở đó, các mô đun này phụ thuộc một mô đun khác, chúng sử dụng cơ chế nhập lược đồ XML để đưa ra các kết cấu mô tả trong mô đun khác.

- Mô đun tên miền SDMX chứa các kết cấu thông điệp chung, bao gồm thông tin tiêu đề chung ("`SDMXMessage.xsd`") - được sử dụng với tất cả mô đun tên miền SDMX-ML khác
- Mô đun tên miền SDMX chứa các mô tả siêu dữ liệu có cấu trúc ví dụ như các tập khóa, khái niệm và danh sách mã ("`SDMXStructure.xsd`")

## TCVN 7981-3 : 2009

- Môđun tên miền SDMX chứa các kết cấu dùng chung qua tất cả kiểu thông điệp SDMX ("SDMXCommon.xsd") - cần thiết cho tất cả các môđun tên miền SDMX-ML khác (Để thuận tiện, cũng chứa tên miền XML ["xml.xsd"] được cung cấp bởi W3C bao gồm thuộc tính xml:lang trong các lược đồ).
- Môđun tên miền SDMX mô tả định dạng chung (phi tập khóa) về định dạng dữ liệu ("SDMXGenericData.xsd"),
- Môđun tên miền SDMX mô tả cấu trúc của thông điệp Query (truy vấn) chung ("SDMXQuery.xsd") - đối với người sử dụng và phát triển các dịch vụ web, v.v.
- Môđun tên miền SDMX cung cấp khung cơ cấu chung được sử dụng cho tất cả lược đồ tập khóa về Trao đổi Cơ sở dữ liệu, Cập nhật và Sửa đổi ("SDMXCompactData.xsd")- đối với việc sử dụng song phương.
- Tập các môđun tên miền được tạo ra và duy trì bởi người tạo ra các lược đồ "Compact" (cơ đợng) về tập khóa - không được duy trì bởi SDMX.
- Môđun tên miền SDMX cung cấp khung cơ cấu chung được sử dụng cho tất cả lược đồ tập khóa chuyên dụng về người sở hữu web và người phát triển sử dụng các công cụ XML tiêu chuẩn ("SDMXUtilityData.xsd") để xử lý và sử dụng sản phẩm công bố.
- Tập các môđun tên miền được tạo và duy trì bởi người tạo các lược đồ "Utility" (tiện ích) của tập khóa - không được duy trì bởi SDMX.
- Môđun tên miền SDMX cung cấp khung cơ cấu chung về tất cả lược đồ tập khóa chuyên dụng của dữ liệu phần giao ("SDMXCrossSectionalData.xsd") để sử dụng song phương và xử lý dữ liệu phần giao.
- Tập các môđun tên miền được tạo và duy trì bởi người tạo các lược đồ "CrossSectional"(phần giao) của tập khóa - không được duy trì bởi SDMX.

Các điều sau đây mô tả chi tiết các định dạng XML được đề xuất, các định dạng này nên được kiểm tra cùng với tài liệu cung cấp. Các lược đồ đề xuất này được chia thành các lược đồ chung trong đó, một tập các định nghĩa lược đồ hoàn chỉnh được cung cấp và các lược đồ về tập khóa trong đó, cấu trúc chính được cung cấp (cùng với mã lược đồ), cộng với hướng dẫn tập khóa cụ thể được ánh xạ tới cấu trúc chính.

### 5 Lược đồ (phi tập khóa) chung

Một vài lược đồ SDMX-ML là tương tự đối với tất cả các lược đồ tập khóa chuyên dụng. Các lược đồ này gồm:

- SDMXMessage.xsd, mô tả cấu trúc thông điệp cơ bản chung cho tất cả các thông điệp SDMX-ML;
- SDMXStructure.xsd, mô tả các tập khóa, danh sách mã và khái niệm;
- SDMXGenericData.xsd, mô tả dữ liệu qua các tập khóa về việc xử lý chung;

- `SDMXQuery.xsd`, đánh dấu các truy vấn dựa vào cơ sở dữ liệu và các dịch vụ web phù hợp với SDMX;
- `SDMXCommon.xsd`, mô tả các kết cấu chung được sử dụng trong các lược đồ khác

Trong các thông điệp này, chỉ có thông điệp `SDMXStructure` và `SDMXGenericData` được yêu cầu cho việc trao đổi dữ liệu chung. Tài liệu cho mỗi lược đồ trên được đưa ra dưới đây. (Bản thân các lược đồ được bổ sung riêng)

### 5.1 Mô đun tên miền của thông điệp SDMX

**Tên miền:** [http://www.SDMX.org/resources/SDMXML/schemas/v1\\_0/message](http://www.SDMX.org/resources/SDMXML/schemas/v1_0/message)

**Nhập:** [http://www.SDMX.org/resources/SDMXML/schemas/v1\\_0/structure](http://www.SDMX.org/resources/SDMXML/schemas/v1_0/structure)

(`SDMXStructure.xsd`)

**Nhập:** [http://www.SDMX.org/resources/SDMXML/schemas/v1\\_0/generic](http://www.SDMX.org/resources/SDMXML/schemas/v1_0/generic)

(`SDMXGenericData.xsd`)

**Nhập:** [http://www.SDMX.org/resources/SDMXML/schemas/v1\\_0/utility](http://www.SDMX.org/resources/SDMXML/schemas/v1_0/utility)

(`SDMXUtilityData.xsd`)

**Nhập:** [http://www.SDMX.org/resources/SDMXML/schemas/v1\\_0/compact](http://www.SDMX.org/resources/SDMXML/schemas/v1_0/compact)

(`SDMXCompactData.xsd`)

**Nhập:** [http://www.SDMX.org/resources/SDMXML/schemas/v1\\_0/cross](http://www.SDMX.org/resources/SDMXML/schemas/v1_0/cross)

(`SDMXCrossSectionalData.xsd`)

**Nhập:** [http://www.SDMX.org/resources/SDMXML/schemas/v1\\_0/query](http://www.SDMX.org/resources/SDMXML/schemas/v1_0/query)

(`SDMXQuery.xsd`)

**Nhập:** [http://www.SDMX.org/resources/SDMXML/schemas/v1\\_0/common](http://www.SDMX.org/resources/SDMXML/schemas/v1_0/common)

(`SDMXCommon.xsd`)

#### 5.1.1 Phần tử toàn cục

**Structure** (Cấu trúc) (**StructureType**/Kiểu cấu trúc): là thông điệp chứa tất cả siêu dữ liệu có cấu trúc về tập dữ liệu, có thể là các tập khóa, khái niệm hoặc danh sách mã.

**GenericData** (Dữ liệu chung) (**GenericDataType**/Kiểu dữ liệu chung): được sử dụng để truyền dữ liệu theo dạng tập khóa giao nhau.

**UtilityData** (Dữ liệu tiện ích) (**UtilityDataType**/Kiểu dữ liệu tiện ích): chứa dữ liệu dạng XML, đặc trưng cho mỗi tập khóa, theo các ánh xạ chuẩn và được tối ưu hóa để hỗ trợ công cụ soạn thảo và các ứng dụng khác đòi hỏi một lược đồ XML "điển hình". Định dạng này có thể được sử dụng để kiểm tra tính hợp lệ của dữ liệu theo một dạng tập khóa chuyên dụng giống như các lược đồ XML mong đợi và yêu cầu toàn bộ tập dữ liệu. Kiểu dữ liệu này không thể được sử dụng cho các cập nhật gia tăng.

## TCVN 7981-3 : 2009

**CompactData** (Dữ liệu cô đọng) (**CompactDataType**/Kiểu dữ liệu cô đọng): chứa dữ liệu theo định dạng XML được tối ưu hóa cho cập nhật gia tăng và việc truyền song phương các tập dữ liệu lớn. Nó đặc trưng cho mỗi tập khóa, theo các ánh xạ chuẩn. Cho phép các giá trị khóa được trình bày ở mức Group (nhóm).

**CrossSectionalData** (Dữ liệu phần giao) (**CrossSectionalDataType**/Kiểu dữ liệu phần giao): chứa dữ liệu theo định dạng XML được tối ưu hóa đối với việc mô tả quan sát tại một điểm theo thời gian và đối với việc truyền song phương các tập dữ liệu. Nó đặc trưng cho mỗi tập khóa theo các ánh xạ chuẩn. Cho phép các giá trị khóa được trình bày từ mức Group (nhóm) xuống mức Observation (quan sát) và cho phép nhiều giá trị quan sát với các "đơn vị đo" khác nhau. Thời gian được gắn với mức DataSet (tập dữ liệu).

**QueryMessage** (Dữ liệu truy vấn) (**QueryMessageType**/Kiểu dữ liệu truy vấn): được sử dụng để truy vấn các cơ sở dữ liệu công bố trên web và gọi các dịch vụ web. Nó cho phép các truy vấn được tạo về dữ liệu và siêu dữ liệu có cấu trúc.

**MessageGroup** (Nhóm thông điệp) (**MessageGroupType**/Kiểu nhóm thông điệp): được sử dụng để cho phép nhiều thông điệp dữ liệu của một kiểu đơn được chứa trong đường truyền đơn. Phần tử này xuất hiện từ yêu cầu về các dịch vụ có khả năng trao đổi dữ liệu từ nhiều nguồn và được cấu trúc theo nhiều tập khóa.

**Header** (Tiêu đề) (**HeaderType**/Kiểu tiêu đề): được khai báo toàn cục để có thể hoạt động như phần tiêu đề của nhóm thay thế cho các lược đồ được sử dụng bên trong. Tuy đây là một ngoại lệ của toàn bộ các thiết kế SDMX-ML, nhưng nhiều người lại thích sử dụng kết cấu này. Chú ý rằng khi các thông điệp SDMX-ML được trao đổi bên ngoài tổ chức, thì nên sử dụng tiêu đề chuẩn - không tạo ra bất kỳ giả định nào về các trường bổ sung trong kiểu thay thế nếu không được nhất trí của các bên đối tác.

### 5.1.2 Kiểu phức

**MessageType** (Kiểu thông điệp): là kiểu trừu tượng được tất cả các thông điệp sử dụng, cho phép kế thừa các tính năng chung. Nó cũng cung cấp các quy định về tính duy nhất đối với các trường tiêu đề.

*Element Content (Type):*

Header (HeaderType)

**StructureType** (Kiểu cấu trúc): xác định các nội dung của thông điệp cấu trúc.

*Extends: MessageType*

*Element Content (Type):*

Agencies (structure:AgenciesType) - giá trị nhỏ nhất: 0

CodeLists (structure:CodeListsType) - giá trị nhỏ nhất: 0

Concepts (structure:ConceptsType) - giá trị nhỏ nhất: 0

KeyFamilies (structure:KeyFamiliesType) - giá trị nhỏ nhất: 0

**GenericDataType** (Kiểu dữ liệu chung): xác định các nội dung của thông điệp GenericData (dữ liệu chung).

*Extends: MessageType*

*Element Content (Type)*

DataSet (generic:DataSetType)

**UtilityDataType** (Kiểu dữ liệu tiện ích): xác định các nội dung của thông điệp UtilityData (dữ liệu tiện ích).

*Extends: MessageType*

*Element Content (Type):*

[Reference] (utility:DataSet)

**CompactDataType** (Kiểu dữ liệu cô đọng): xác định các nội dung của thông điệp CompactData (dữ liệu cô đọng).

*Extends: MessageType*

*Element Content (Type):*

[Reference] (compact:DataSet)

**CrossSectionalDataType** (Kiểu dữ liệu phần giao): xác định các nội dung của thông điệp CrossSectionalData (dữ liệu phần giao).

*Extends: MessageType*

*Element Content (Type):*

[Reference] (cross:DataSet)

**QueryMessageType** (Kiểu thông điệp truy vấn (Query)): xác định các nội dung của QueryMessage (thông điệp truy vấn (Query)).

*Extends: MessageType*

*Element Content (Type):*

Query (query:QueryType)

**MessageGroupType** (Kểu nhóm thông điệp): xác định các nội dung của thông điệp MessageGroup (nhóm thông điệp).

*Extends: MessageType*

*Choice:*

[Reference] (generic:DataSet) - giá trị lớn nhất: vô cùng

*Choice:*

## TCVN 7981-3 : 2009

Reference] (utility:DataSet) - giá trị lớn nhất: vô cùng

*Choice:*

[Reference] (compact:DataSet) - giá trị lớn nhất: vô cùng

*Choice:*

[Reference] (cross:DataSet) - giá trị lớn nhất: vô cùng

*Attribute:* id(xs:NMTOKEN) - tùy chọn

**HeaderType** (Kiểu tiêu đề): xác định các trường tiêu đề sử dụng cho tất cả các thông điệp. ID định danh định nghĩa luồng dữ liệu, khi được liên kết với thời gian, nó định danh duy nhất tập dữ liệu đó. Việc kiểm tra cho biết liệu thông điệp có dùng cho mục đích kiểm tra hay không. Trường Truncated (cắt bớt) được sử dụng trong các thông điệp dữ liệu đáp ứng các thông điệp Query (truy vấn) và đặt là true nếu đáp ứng đó bị cắt bớt để thỏa mãn các giới hạn về kích cỡ được đưa ra bởi thuộc tính defaultLimit (giới hạn mặc định) trong thông điệp Query (truy vấn). Trường Name (tên) cung cấp một tên để truyền. Trường Prepared (chuẩn bị) là ngày tháng chuẩn bị. Trường Sender (người gửi) là thông tin về người gửi và trường Receiver (người nhận) là thông tin về người nhận. Trường Agency (cơ quan) cung cấp mã định danh/từ viết tắt cho cơ quan duy trì tập dữ liệu. Id của tập dữ liệu cung cấp định danh cho tập dữ liệu. Mã Action (hoạt động) cung cấp mã để xác định xem liệu thông điệp bao là thông điệp Update (cập nhật) hay thông điệp Delete (xóa) (không được sử dụng với thông điệp UtilityData (dữ liệu tiện ích)). KeyFamilyRef (tham chiếu tập khóa) được sử dụng để tham chiếu tập khóa tới tập dữ liệu, sử dụng id của nó. (Thông tin này được yêu cầu ở mức DataSet (tập dữ liệu) cho một vài thông điệp, ngoài ra được cung cấp ở đây nhằm tạo thuận lợi cho các thông điệp không yêu cầu nó.) KeyFamilyAgency (cơ quan duy trì tập khóa) quy định cơ quan duy trì tập khóa sử dụng id được mã hóa của nó. Các trường tham chiếu đến tập dữ liệu không được sử dụng nếu thông điệp chứa truy vấn hoặc thông tin có cấu trúc - các thông điệp này cung cấp các trường cụ thể cho việc nắm bắt thông tin. Các thông điệp ở đây không được sử dụng như các mặc định. Trường Extracted (tách) là tem thời gian từ hệ thống trả về dữ liệu; ReportingBegin (báo cáo bắt đầu) và ReportingEnd (báo cáo kết thúc) cung cấp chu kỳ thời gian được chứa trong thông điệp. Trường Source (nguồn) cung cấp thông tin có thể đọc về nguồn dữ liệu đó.

*Element Content (Type):*

ID (xs:NCName)

Test (xs:boolean)

Truncated (xs:boolean)

Name (common:TextType) - giá trị nhỏ nhất: 0 - giá trị lớn nhất: vô cùng

Prepared (HeaderTimeType)

Sender (PartyType)

Receiver (PartyType) - giá trị nhỏ nhất: 0 - giá trị lớn nhất: vô cùng

KeyFamilyRef (xs:NMTOKEN) - giá trị nhỏ nhất: 0

KeyFamilyAgency (xs:NMTOKEN) - giá trị nhỏ nhất: 0

DataSetAgency (xs:NMTOKEN) - giá trị nhỏ nhất: 0

DataSetID (xs:NMTOKEN) - giá trị nhỏ nhất: 0

DataSetAction (common:ActionType) - giá trị nhỏ nhất: 0

Extracted (xs:dateTime) - giá trị nhỏ nhất: 0

ReportingBegin (HeaderTimeType) - giá trị nhỏ nhất: 0

ReportingEnd (HeaderTimeType) - giá trị nhỏ nhất: 0

Source (common:TextType) - giá trị nhỏ nhất: 0 - giá trị lớn nhất: vô cùng

**PartyType** (Kiểu bên tham gia): xác định thông tin được gửi về các bên tham gia khác nhau ví dụ như người gửi và người nhận thông điệp. Trường Name (tên) là ID của một bên và trường Contact (liên lạc) cung cấp các thông tin liên hệ chi tiết.

*Element Content (Type):*

Name (common:TextType) - giá trị nhỏ nhất: 0 - giá trị lớn nhất: vô cùng

Contact (ContactType) - giá trị nhỏ nhất: 0 - giá trị lớn nhất: vô cùng

*Attribute:* id (xs:NMTOKEN) - theo qui định

**ContactType** (kiểu liên hệ): xác định thông tin liên hệ của một bên tham gia. Trường Name (tên) cung cấp tên mà con người có thể đọc.

*Element Content (Type):*

Name (common:TextType) - giá trị nhỏ nhất: 0 - giá trị lớn nhất: vô cùng

Department (common:TextType) - giá trị nhỏ nhất: 0 - giá trị lớn nhất: vô cùng

Role (common:TextType) - giá trị nhỏ nhất: 0 - giá trị lớn nhất: vô cùng

*Choice:* giá trị nhỏ nhất: 0 - giá trị lớn nhất: vô cùng

Telephone (xs:string)

Fax (xs:string)

X400 (xs:string)

URI (xs:string)

Email (xs:string)

### 5.1.3 Kiểu đơn giản

**HeaderTimeType** (Kiểu tiêu đề thời gian): cung cấp kiểu liên kết xs:date và xs:dateTime cho các

## TCVN 7981-3 : 2009

trường tiêu đề trong thông điệp.

### 5.2 Môđun tên miền của cấu trúc SDMX

**Tên miền:** [http://www.SDMX.org/resources/SDMXML/schemas/v1\\_0/structure](http://www.SDMX.org/resources/SDMXML/schemas/v1_0/structure)

**Nhập:** [http://www.SDMX.org/resources/SDMXML/schemas/v1\\_0/common](http://www.SDMX.org/resources/SDMXML/schemas/v1_0/common)

(SDMXCommon.xsd)

#### 5.2.1 Kiểu phức

**AgenciesType** (Kiểu các cơ quan): bao gồm một hoặc nhiều cơ quan.

*Element Content (Type):*

Agency (AgencyType) - giá trị lớn nhất: vô cùng

**AgencyType** (Kiểu cơ quan): cung cấp cấu trúc để mô tả các cơ quan và thông tin liên hệ. Thuộc tính id mang mã định danh cơ quan. Thuộc tính version (phiên bản) chỉ ra phiên bản mô tả cơ quan đó. Thuộc tính uri cung cấp uri cho cách định danh luân phiên về thông tin cơ quan đó (điển hình là URL phân tích cơ quan được mô tả trong SDMX-ML). Phần tử Name (tên) là phần tử cung cấp tên có thể đọc về tổ chức. MaintenanceContact (liên hệ của duy trì) cung cấp thông tin liên hệ của cơ quan khi nó hoạt động như một MaintenanceAgency (cơ quan duy trì); CollectorContact (liên hệ của thu thập) hoạt động giống với MaintenanceAgency (cơ quan duy trì) khi cơ quan này hoạt động như người thu thập số liệu thống kê; DisseminatorContact (liên hệ của phổ biến) khi cơ quan hoạt động như người phổ biến số liệu thống kê; và ReporterContact (liên hệ của báo cáo) khi cơ quan hoạt động như người báo cáo số liệu thống kê. OtherContact (liên hệ khác) được sử dụng để mô tả vai trò khác. Chú ý rằng trường Role (vai trò) trong cấu trúc thông tin liên hệ được quy định cho OtherContact (liên hệ khác). Cho phép tham chiếu thông tin đầy đủ về Agency (cơ quan) bằng cách chỉ sử dụng (tối thiểu) các trường id, name (tên), uri, với trường uri trỏ tới một mô tả ngoại trú trong một thông điệp Structure (cấu trúc) SDMX-ML hợp lệ mà cung cấp thông tin đầy đủ hơn. (gọi là là "tham chiếu ngoại trú"). Nếu một tham chiếu ngoại trú được tạo, thì thuộc tính isExternalReference (tham chiếu ngoại trú) phải là "true".

*Element Content (Type):*

Name (common:TextType) - giá trị lớn nhất: vô cùng

MaintenanceContact (ContactType) - giá trị nhỏ nhất: 0

CollectorContact (ContactType) - giá trị nhỏ nhất: 0

DisseminatorContact (ContactType) - giá trị nhỏ nhất: 0

ReporterContact (ContactType) - giá trị nhỏ nhất: 0

OtherContact (ContactType) - giá trị nhỏ nhất: 0 - giá trị lớn nhất: vô cùng

*Attribute:* id (xs:NCName) - theo qui định

*Attribute:* version (xs:string) - tùy chọn



*Attribute:* uri (xs:anyURI) - tùy chọn

*Attribute:* isExternalReference (xs:Boolean) - tùy chọn

**ContactType** (kiểu liên hệ): xác định thông tin liên hệ của bên tham gia. Phần tử id mang thông tin id của người sử dụng đối với liên hệ đó, trong đó, trường Name (tên) cung cấp tên có thể đọc.

*Element Content (Type):*

Name (common:TextType) - giá trị nhỏ nhất: 0 - giá trị lớn nhất: vô cùng

id (xs:NMTOKEN) - giá trị nhỏ nhất: 0

Department (common:TextType) - giá trị nhỏ nhất: 0 - giá trị lớn nhất: vô cùng

Role (common:TextType) - giá trị nhỏ nhất: 0 - giá trị lớn nhất: vô cùng

*Choice:* giá trị nhỏ nhất: 0 - giá trị lớn nhất: vô cùng

Telephone (xs:string)

Fax (xs:string)

X400 (xs:string)

URI (xs:string)

Email (xs:string)

**CodeListsType** (Kiểu các danh sách mã): chứa một hoặc nhiều danh sách mã. Kiểu này cũng xác định các quy định tính duy nhất đối với các id của danh sách mã.

*Element Content (Type):*

CodeList (CodeListType) - giá trị nhỏ nhất: 0 - giá trị lớn nhất: vô cùng

**CodeListType** (Kiểu danh sách mã): xác định các nội dung của danh sách mã. Bao gồm ID, cơ quan duy trì danh sách mã, phiên bản của nó và URL đặt danh sách mã đó. Các phần tử được hỗ trợ cho việc cung cấp tên và các mã. Nó chấp nhận cung cấp id, tên và các trường uri ở mức tối thiểu, với uri chỉ tới thông điệp cấu trúc SDMX chứa các chi tiết đầy đủ về danh sách mã. (Nó được đặt tên là "tham chiếu ngoại trú") Nếu một tham chiếu ngoại trú được tạo ra thì thuộc tính isExternalReference phải là "true".

*Element Content (Type):*

Name (common:TextType) - giá trị lớn nhất: vô cùng

Code (CodeType) - giá trị nhỏ nhất: 0 - giá trị lớn nhất: vô cùng

Annotations (common:AnnotationsType) - giá trị nhỏ nhất: 0

*Attribute:* id (xs:NCName) - theo qui định

*Attribute:* agency (xs:NMTOKEN) - tùy chọn

*Attribute:* version (xs:string) - tùy chọn

## TCVN 7981-3 : 2009

*Attribute:* uri (xs:anyURI) - tùy chọn

*Attribute:* isExternalReference (xs:Boolean) - tùy chọn

**CodeType** (Kiểu mã): xác định cấu trúc của mã. Điều này chấp nhận các mô tả văn bản thuần túy như nội dung phần tử và mã hóa giá trị như thuộc tính giá trị. (Các mô tả ngắn hoặc thông tin thể hiện khác có thể được bổ sung sử dụng Annotation (ghi chú) với một trường chỉ dẫn [ví dụ, "ShortDescription"])

*Element Content (Type):*

Description (common:TextType) - giá trị lớn nhất: vô cùng

Annotations (common:AnnotationsType) - giá trị nhỏ nhất: 0

*Attribute:* value (xs:NMTOKEN) - theo qui định

**ConceptsType** (Kiểu các khái niệm): xác định cấu trúc của tập các khái niệm.

*Element Content (Type):*

Concept (ConceptType) - giá trị lớn nhất: vô cùng

**ConceptType** (Kiểu khái niệm): quy định thông tin được cung cấp về khái niệm đơn. Bao gồm tên, nội dung phần tử và ID. Có thể sử dụng trường uri để chỉ tới vị trí của thông điệp Cấu trúc SDMX-ML chứa nhiều phiên bản khái niệm chi tiết hơn. (Được đặt tên là "tham chiếu ngoại trú".) Nếu một tham chiếu ngoại trú được tạo ra, thì thuộc tính isExternalReference phải là "true".

*Element Content (Type):*

Name (common:TextType) - giá trị lớn nhất: vô cùng

Annotations (common:AnnotationsType) - giá trị nhỏ nhất: 0

*Attribute:* id (xs:NCName) - theo qui định

*Attribute:* agency (xs:NMTOKEN) - tùy chọn

*Attribute:* version (xs:string) - tùy chọn

*Attribute:* uri (xs:anyURI) - tùy chọn

*Attribute:* isExternalReference (xs:Boolean) - tùy chọn

**KeyFamiliesType** (Kiểu các tập khóa): xác định cấu trúc cho việc mô tả một hoặc nhiều tập khóa. Nó cũng cung cấp các quy định tính duy nhất đối với mỗi ID của tập khóa.

*Element Content (Type):*

KeyFamily (KeyFamilyType) - giá trị lớn nhất: vô cùng

**KeyFamilyType** (Kiểu tập khóa): xác định cấu trúc của mô tả tập khóa. Điều này bao gồm tên và tập các phần tử (thuộc tính và miền dữ liệu) giống với nội dung phần tử và ID, cơ quan, phiên bản và URL nơi định vị các thuộc tính.

*Element Content (Type):*

Name (common:TextType) - giá trị lớn nhất: vô cùng

Components (ComponentsType)

Annotations (common:AnnotationsType) - giá trị nhỏ nhất: 0

Attribute: id (xs:NCName) - theo qui định

Attribute: agency (xs:NMTOKEN) - tùy chọn

Attribute: version (xs:string) - tùy chọn

Attribute: uri (xs:anyURI) - tùy chọn

**ComponentsType** (Kiểu các thành phần): mô tả các miền dữ liệu, nhóm, thuộc tính và đơn vị đo của tập khóa. Nếu TimeDimension (miền dữ liệu thời gian) được chứa trong tập khóa - nếu các định dạng chuỗi thời gian về dữ liệu (các định dạng GenericData (dữ liệu chung), CompactData (dữ liệu cô đọng) và UtilityData (dữ liệu tiện ích)) được sử dụng thì phải có một miền dữ liệu tần suất.

*Element Content (Type):*

Dimension (DimensionType) - giá trị nhỏ nhất: 0 - giá trị lớn nhất: vô cùng

TimeDimension (TimeDimensionType) - giá trị nhỏ nhất: 0

PrimaryMeasure (PrimaryMeasureType)

CrossSectionalMeasure (CrossSectionalMeasureType) - giá trị nhỏ nhất: 0 - giá trị lớn nhất: vô cùng

Group (GroupType) - giá trị nhỏ nhất: 0 - giá trị lớn nhất: vô cùng

Attribute (AttributeType) - giá trị nhỏ nhất: 0 - giá trị lớn nhất: vô cùng

**DimensionType** (Kiểu miền dữ liệu): mô tả cấu trúc các miền dữ liệu phi thời gian. Thứ tự khai báo là rất quan trọng: nó được sử dụng để mô tả thứ tự khi chúng xuất hiện trong các định dạng dữ liệu mà các giá trị khóa được cung cấp theo một dạng có thứ tự (không kể miền dữ liệu thời gian, mà không được biểu diễn như một thành viên của khóa sắp xếp). Trong trường hợp các tập khóa được sử dụng cho dữ liệu phần giao cũng như dữ liệu chuỗi thời gian, mọi miền dữ liệu "đơn vị đo" phải có giá trị thuộc tính "isMeasureDimension" đặt là "true". Nếu một miền dữ liệu được khai báo là đơn vị đo, thì nó phải có đơn vị đo được khai báo ở trong tập khóa mà tương ứng với mỗi giá trị trong danh sách mã biểu diễn nó. Bất kỳ miền dữ liệu tương ứng với khái niệm tần suất phải có thuộc tính isFrequencyDimension đặt là "true". Chỉ có một miền dữ liệu này trong tập khóa. (Như đã quy ước trước đó, đó là miền dữ liệu đầu tiên trong tập các miền dữ liệu sắp xếp.) Nếu tập khóa mô tả dữ liệu phần giao, thì với mỗi miền dữ liệu phi thời gian, các thuộc tính crossSectionalAttachDataSet, crossSectionalAttachGroup, crossSectionalAttachSection và crossSectionalAttachObservation phải được đưa ra các giá trị. Giá trị "true" cho các thuộc tính này chỉ ra rằng miền dữ liệu có thể được cung cấp một giá trị ở mức chỉ báo trong cấu trúc phần giao.

## TCVN 7981-3 : 2009

Chú ý rằng các thuộc tính này không được cung cấp cho bất kỳ miền dữ liệu nào với isFrequencyDimension đặt là "true", khi các miền dữ liệu này được đính kèm ở mức Group (nhóm) như thời gian. Tập khóa được thiết kế cho việc sử dụng phần giao phải được cấu trúc để mọi khóa của tập quan sát có thể được mô tả rõ ràng bằng cách lấy mỗi giá trị miền dữ liệu từ mức quan sát, mức phần, mức nhóm và mức tập dữ liệu, đúng với trình tự đưa ra trong tập khóa.

### *Element Content (Type):*

Annotations (common:AnnotationsType) - giá trị nhỏ nhất: 0

Attribute: concept (xs:NMTOKEN) - theo qui định

Attribute: codelist (xs:NMTOKEN) - theo qui định

Attribute: isMeasureDimension (xs:boolean) - default: false

Attribute: isFrequencyDimension (xs:boolean) - default: false

Attribute: crossSectionalAttachDataSet (xs:boolean) - tùy chọn

Attribute: crossSectionalAttachGroup (xs:boolean) - tùy chọn

Attribute: crossSectionalAttachSection (xs:boolean) - tùy chọn

Attribute: crossSectionalAttachObservation (xs:boolean) - tùy chọn

**TimeDimensionType** (Kiểu miền dữ liệu thời gian): mô tả miền dữ liệu thời gian đặc biệt. Mọi miền dữ liệu được sử dụng cho các định dạng chuỗi thời gian (GenericData (dữ liệu chung), CompactData (dữ liệu cô đọng) và UtilityData (dữ liệu tiện ích)) phải bao hàm miền dữ liệu thời gian. Mọi tập khóa sử dụng miền dữ liệu thời gian phải khai báo miền dữ liệu thời gian, miền dữ liệu đầu tiên trong tập khóa như đã quy ước (tập các miền dữ liệu phi thời gian). Phần tử TextFormat (định dạng văn bản) có thể được bao hàm để chỉ ra việc biểu diễn thời gian trong một vài định dạng dữ liệu phi XML. Thuộc tính khái niệm phải chứa tên của khái niệm thời gian. Thuộc tính danh sách mã có thể cung cấp giá trị về tên của danh sách mã nếu cần.

### *Element Content (Type):*

TextFormat (TextFormatType) - giá trị nhỏ nhất: 0

Annotations (common:AnnotationsType) - giá trị nhỏ nhất: 0

Attribute: concept (xs:NMTOKEN) - theo qui định

Attribute: codelist (xs:NMTOKEN) - tùy chọn

**GroupType** (Kiểu nhóm): khai báo các nhóm dữ liệu hữu ích, dựa trên việc lựa chọn các miền dữ liệu (phi thời gian) được khai báo tạo thành các khóa từng phần mà các thuộc tính được đính kèm. Giá trị của phần tử DimensionRef (tham chiếu miền dữ liệu) là khái niệm của miền dữ liệu - đó là giá trị của thuộc tính khái niệm của miền dữ liệu. Do đó, nếu dữ liệu được thể hiện như tập chuỗi thời gian trong đó tập chuỗi này thay đổi theo các tần suất khác nhau thì "nhóm quan hệ" được khai báo, với tất cả miền dữ liệu ngoại trừ miền dữ liệu tần suất trong đó. Nếu dữ liệu được

nhóm với nhau như một tập gồm các quốc gia, thì “nhóm quốc gia” có thể được khai báo, với tất cả miền dữ liệu ngoại trừ miền dữ liệu quốc gia tạo thành một phần của khóa từng phần. Nếu mọi miền dữ liệu không phải là một phần của nhóm thì đều có giá trị biến đổi ở mức chuỗi (cho các định dạng chuỗi thời gian). Chỉ có miền dữ liệu đơn bị loại khỏi khóa từng phần - nó có thể là mọi tập con của các miền dữ liệu sắp xếp (đó là toàn bộ các miền dữ liệu ngoại trừ miền dữ liệu của chuỗi thời gian mà có thể không bao giờ được khai báo khi ở trong khóa từng phần theo nhóm). Tất cả các nhóm khai báo trong tập khóa phải là duy nhất - tức là, không thể sao chép lại các khóa từng phần. Tất cả các nhóm cũng phải được gán cho một tên duy nhất (các thuộc tính id). Mặc dù theo quy ước phải khai báo các miền dữ liệu theo cùng thứ tự khi chúng được khai báo trong khóa sắp xếp, nhưng không có yêu cầu nào bắt buộc phải làm như vậy - thứ tự các giá trị của tập khóa được lấy theo thứ tự mà các miền dữ liệu được khai báo. Phần tử Description (mô tả) cung cấp mô tả mà con người có thể hiểu được (trong nhiều ngôn ngữ hoặc song ngữ) của nhóm. Chú ý rằng với các định dạng phần giao, cơ chế nhóm có tên không được sử dụng, mà thay được thay thế bằng nhóm chung có thời gian và các giá trị tần suất, chấp nhận các thuộc tính mức nhóm có sẵn được quy định nếu cần.

*Element Content (Type):*

DimensionRef (xs:NMTOKEN) - giá trị lớn nhất: vô cùng

Description (common:TextType) - giá trị nhỏ nhất: 0 - giá trị lớn nhất: vô cùng

Annotations (common:AnnotationsType) - giá trị nhỏ nhất: 0

*Attribute: name* (xs:NMTOKEN) - theo qui định

**AttributeType** (Kiểu thuộc tính): mô tả cấu trúc của các thuộc tính được khai báo trong tập khóa. Nếu thuộc tính danh sách mã không được sử dụng, thì thuộc tính không được mã hóa. Bạn có thể sử dụng phần tử TextFormat (định dạng văn bản) để quy định giá trị của thuộc tính không được mã hóa. Thuộc tính khái niệm chứa tên của khái niệm. Thuộc tính danh sách mã cung cấp giá trị id của danh sách mã. Thuộc tính attachmentLevel chỉ ra mức mà thuộc tính được đính kèm trong các định dạng chuỗi thời gian (các định dạng GenericData (dữ liệu chung), CompactData (dữ liệu cô đọng) và UtilityData (dữ liệu tiện ích) ). Thuộc tính assignmentStatus cho biết liệu giá trị được cung cấp cho thuộc tính hay không khi gửi tài liệu cùng với dữ liệu. Phần tử AttachmentGroup (nhóm đính kèm) chỉ được bao hàm khi thuộc tính được đính kèm ở mức Group (nhóm), nhằm chỉ ra nhóm hay các nhóm khai báo thuộc tính có thể đính kèm. Với mỗi nhóm này, phần tử AttachmentGroup xuất hiện với nội dung của phần tử là tên của nhóm. Phần tử AttachmentMeasure (đơn vị đo kèm theo) cũng tương tự như vậy, chỉ ra các định dạng phần giao mà đơn vị đo hoặc đơn vị đo khai báo thuộc tính được đính kèm tới mức quan sát. Thuộc tính isTimeFormat cho biết thuộc tính biểu diễn khái niệm của định dạng thời gian (đặc biệt thuộc tính mức chuỗi bắt buộc với việc biểu diễn danh sách mã được lấy từ tiêu chuẩn ISO 8601). Với các tập khóa không được sử dụng để cấu trúc các định dạng phần giao, phần tử này có thể bị loại bỏ. Mỗi phần tử này đều chứa tên của đơn vị đo được khai báo. Các thuộc tính crossSectionalAttachDataSet, crossSectionalAttachGroup, crossSectionalAttachSection và

## TCVN 7981-3 : 2009

crossSectionalAttachObservation chỉ ra mức độ hoặc các mức độ đính kèm cho các định dạng dữ liệu phân giao và có thể bị loại bỏ nếu tập khóa không được sử dụng để cấu trúc chúng. Giá trị "true" cho biết nó được phép cung cấp giá trị về thuộc tính ở mức quy định trong cấu trúc. Chú ý rằng tất cả các nhóm trong các định dạng phân giao được thay thế bởi nhóm chung có các giá trị về thời gian và tần suất, cho phép bất kỳ thuộc tính mức nhóm được đính kèm với nó.

### *Element Content (Type):*

*TextFormat* (TextFormatType) - giá trị nhỏ nhất: 0

*AttachmentGroup* (xs:NMTOKEN) - giá trị nhỏ nhất: 0 - giá trị lớn nhất: vô cùng

*AttachmentMeasure* (xs:NMTOKEN) - giá trị nhỏ nhất: 0 - giá trị lớn nhất: vô cùng

*Annotations* (common:AnnotationsType) - giá trị nhỏ nhất: 0

*Attribute: concept* (xs:NMTOKEN) - theo qui định

*Attribute: codelist* (xs:NMTOKEN) - tùy chọn

*Attribute: attachmentLevel* (structure:AttachmentLevelType) - theo qui định

*Attribute: assignmentStatus* (structure:AssignmentStatusType) - theo qui định

*Attribute: isTimeFormat* (xs:boolean) - *default: false*

*Attribute: crossSectionalAttachDataSet* (xs:boolean) - tùy chọn

*Attribute: crossSectionalAttachGroup* (xs:boolean) - tùy chọn

*Attribute: crossSectionalAttachSection* (xs:boolean) - tùy chọn

*Attribute: crossSectionalAttachObservation* (xs:boolean) - tùy chọn

**TextFormatType** (Kiểu định dạng văn bản): xác định thông tin về việc mô tả định dạng văn bản. Nếu thuộc tính TextType (kiểu văn bản) không được quy định, thì các ký tự hợp lệ có thể được chứa trong trường văn bản. (tương ứng với kiểu dữ liệu xs:string của Lược đồ XML của W3C) Trong trường hợp này, thuộc tính Length (độ dài) được hiểu là độ dài lớn nhất. Nói cách khác, độ dài cung cấp giá trị cực đại hoặc các độ dài chuỗi ký tự như mỗi giá trị thuộc tính TextType (kiểu văn bản). Thuộc tính thập phân cung cấp độ chính xác (số các vị trí thập phân) mà dữ liệu số phải sử dụng. Đây là một số nguyên chỉ ra số lượng các chữ số xuất hiện sau dấu phân cách thập phân ("."). Nếu được sử dụng, chữ số vắng mặt trong dữ liệu số được hiển thị là 0. Nếu không được sử dụng thì không có giới hạn nào trên các chữ số được cung cấp trong dữ liệu tồn tại cho các mục định trao đổi.

*Attribute: length* (xs:integer) - tùy chọn

*Attribute: decimals* (xs:integer) - tùy chọn

*Attribute: TextType* (TextTypeType) - tùy chọn

**PrimaryMeasureType** (Kiểu đơn vị đo chính): mô tả các giá trị quan sát về tất cả các cách thể hiện của dữ liệu, ngoại trừ các định dạng phân giao có nhiều đơn vị đo (một tập các đơn vị đo

phần giao được sử dụng thay thế). Thuộc tính khái niệm chỉ tới khái niệm duy nhất được biểu diễn bởi đơn vị đo. **PrimaryMeasure** (đơn vị đo chính) được liên kết với khái niệm OBS-VALUE.

*Element Content (Type):*

Annotations (common:AnnotationsType) - giá trị nhỏ nhất: 0

Attribute: concept (xs:NMTOKEN) - theo qui định

**CrossSectionalMeasureType** (Kiểu đơn vị đo phần giao): mô tả các giá trị quan sát về các định dạng dữ liệu phần giao đơn vị đo. Với các tập khóa phi phần giao, không cần thiết quy định các đơn vị đo phần giao. Thuộc tính khái niệm chỉ tới khái niệm duy nhất được biểu diễn bởi phép đo. Thuộc tính **measureDimension** chứa tên khái niệm của miền dữ liệu đơn vị đo. Thuộc tính mã chứa giá trị của mã tương ứng trong danh sách mã sử dụng để biểu diễn miền dữ liệu đơn vị đo. **CrossSectionalMeasure** (Đơn vị đo phần giao) phải được khai báo cho mỗi mã trong danh sách mã sử dụng để biểu diễn miền dữ liệu đơn vị đo - Chúng thay thế đơn vị đo chính về các định dạng dữ liệu phần giao nhiều đơn vị đo.

*Element Content (Type):*

Annotations (common:AnnotationsType) - giá trị nhỏ nhất: 0

Attribute: concept (xs:NMTOKEN) - theo qui định

Attribute: measureDimension (xs:NMTOKEN) - theo qui định

Attribute: code (xs:NMTOKEN) - theo qui định

### 5.2.2 Kiểu đơn giản

**AttachmentLevelType** (Kiểu mức đính kèm):

*Restricts xs:NMTOKEN*

Code: DataSet - Data set level

Code: Group - Group level

Code: Series - Series level

Code: Observation - Observation level

**AssignmentStatusType** (Kiểu trạng thái gán):

*Restricts xs:NMTOKEN*

Code: Mandatory - Việc cung cấp giá trị thuộc tính là bắt buộc

Code: Conditional - Việc cung cấp giá trị thuộc tính là tùy chọn

**TextTypeType** (Kiểu văn bản): cung cấp danh sách các kiểu ký tự được cho phép trong **TextFormat** field.

*Restricts xs:NMTOKEN*

## TCVN 7981-3 : 2009

Code: Alpha - Cho phép các ký tự không phải là số được sử dụng trong chuỗi ký tự, với độ dài lớn nhất được quy định trong thuộc tính độ dài.

Code: AlphaFixed - Cho phép các ký tự không phải là số được sử dụng trong chuỗi ký tự, với độ dài (set length) được quy định trong thuộc tính độ dài.

Code: Num - Cho phép các ký tự số (0 - 9) được sử dụng trong chuỗi ký tự, với độ dài lớn nhất được quy định trong thuộc tính độ dài.

Code: NumFixed - Cho phép các ký tự số (0 - 9) được sử dụng trong chuỗi ký tự với độ dài được quy định trong thuộc tính độ dài.

Code: AlphaNum - Cho phép các ký tự số và không phải là số được sử dụng trong chuỗi ký tự, với độ dài lớn nhất được quy định trong thuộc tính độ dài.

Code: AlphaNumFixed - Cho phép các ký tự số và không phải là số được sử dụng trong chuỗi ký tự, với độ dài được quy định trong thuộc tính độ dài.

### 5.3 Môđun tên miền dữ liệu SDMX chung

**Tên miền:** [http://www.SDMX.org/resources/SDMXML/schemas/v1\\_0/generic](http://www.SDMX.org/resources/SDMXML/schemas/v1_0/generic)

**Nhập:** [http://www.SDMX.org/resources/SDMXML/schemas/v1\\_0/common](http://www.SDMX.org/resources/SDMXML/schemas/v1_0/common)

(SDMXCommon.xsd)

#### 5.3.1 Phần tử chung

**DataSet (DataSetType)** (Tập dữ liệu (Kiểu tập dữ liệu)): Phần tử DataSet (tập dữ liệu) chứa một hoặc nhiều nhóm tạo thành tập dữ liệu.

#### 5.3.2 Kiểu phức

**DataSetType** (Kiểu tập dữ liệu): xác định cấu trúc của tập dữ liệu. Nó bao gồm tham chiếu tập khóa chứa ID của tập khóa và các giá trị thuộc tính đính kèm ở mức tập dữ liệu. DataSet (tập dữ liệu) có thể được sử dụng để truyền tài liệu (chỉ các giá trị thuộc tính), dữ liệu hoặc kết hợp cả hai. Nếu chỉ cung cấp tài liệu thì không cần gửi tập các thuộc tính hoàn thiện. Nếu chỉ truyền dữ liệu thì Group có thể bị loại bỏ nếu muốn. Các quy định về tính duy nhất được định nghĩa cho các thuộc tính của tập dữ liệu. Nếu các thuộc tính mức tập dữ liệu được gửi đi trong thông điệp xóa, thì giá trị thuộc tính hợp lệ sẽ chỉ ra rằng nên xóa giá trị hiện thời. Thuộc tính keyFamilyURI cho phép URI (điển hình là URL) được cung cấp, chỉ tới thông điệp Cấu trúc SDMX-ML mô tả tập khóa.

*Attribute:* keyFamilyURI (xs:anyURI) - tùy chọn

*Element Content (Type):*

KeyFamilyRef (xs:NCName)

Attributes (ValueType) - giá trị nhỏ nhất: 0

*Choice:* - giá trị nhỏ nhất: 0 - giá trị lớn nhất: vô cùng



Group (GroupType) - giá trị nhỏ nhất: 0 - giá trị lớn nhất: vô cùng

Series (SeriesType) - giá trị nhỏ nhất: 0 - giá trị lớn nhất: vô cùng

Annotations (common:AnnotationsType) - giá trị nhỏ nhất: 0

**GroupType** (Kiểu nhóm): Các giá trị khóa ở mức nhóm có thể được tuyên bố rõ ràng và tất cả giá trị không phải là ký tự thay thế được liệt kê trong GroupKey (khóa mật mã) - chúng cũng phải được đưa ra giá trị ở mức chuỗi. Không cần thiết phải quy định khóa mật mã, vì nó được suy luận từ các giá trị lặp đi lặp lại ở mức chuỗi. Nếu chỉ tài liệu (các thuộc tính mức nhóm) được truyền thì GroupKey (khóa mật mã) không bị loại bỏ. Thuộc tính kiểu chứa tên của nhóm được khái báo trong tập khóa. Nếu các thuộc tính mức nhóm được quy định trong thông điệp xóa thì giá trị hợp lệ được cung cấp cho thuộc tính chỉ ra rằng giá trị thuộc tính hiện tại nên được xóa bỏ thay cho thuộc tính quy định.

*Attribute:* type (xs:NMTOKEN) - theo qui định

*Element Content (Type):*

GroupKey (ValuesType) - giá trị nhỏ nhất: 0

Attributes(ValuesType) - giá trị nhỏ nhất: 0

Series (SeriesType) - giá trị lớn nhất: vô cùng

Annotations (AnnotationsType) - giá trị nhỏ nhất: 0

**SeriesType** (Kiểu chuỗi): quy định cấu trúc của chuỗi. Điều này bao gồm tất cả các giá trị khóa, giá trị về các thuộc tính và tập các quan sát tạo ra nội dung chuỗi. Các thông điệp có thể truyền các thuộc tính, dữ liệu hoặc cả hai. Bất luận trường hợp nào khóa của chuỗi thời gian đều được yêu cầu. Các giá trị khóa xuất hiện tại mức Series (chuỗi) ở trình tự tương ứng với trình tự khóa trong tập khóa. Chuỗi trong thông điệp xóa không cần cung cấp nhiều khóa, chỉ ra rằng toàn bộ các chuỗi được định danh bằng khóa đó nên được xóa. Nếu các thuộc tính chuỗi được gửi đi trong thông điệp xóa và giá trị hợp lệ được quy định cho thuộc tính thì thuộc tính nên được xóa bỏ.

*Element Content (Type):*

SeriesKey (SeriesKeyType)

Attributes (ValuesType) - giá trị nhỏ nhất: 0

Obs (ObsType) - giá trị nhỏ nhất: 0 - giá trị lớn nhất: vô cùng

Annotations (common:AnnotationsType) - giá trị nhỏ nhất: 0

**SeriesKeyType** (Kiểu khóa của chuỗi thời gian): xác định nội dung của khóa chuỗi. Mỗi miền dữ liệu phi thời gian phải có giá trị cung cấp cho nó, theo thứ tự mà các miền dữ liệu được quy định trong tập khóa.

*Element Content (Type):*

## TCVN 7981-3 : 2009

Value (ValueType) - giá trị lớn nhất: vô cùng

**ObsType** (Kiểu quan sát): xác định cấu trúc của một quan sát. Điều này bao gồm giá trị thời gian và giá trị quan sát cũng như các giá trị về mỗi thuộc tính được gán ở mức quan sát bởi tập khóa. Trong thông điệp xóa, chỉ thời gian cần được đưa ra, nó cho biết rằng quan sát được định danh bởi khóa và thời gian nên được xóa bỏ. Với thông điệp cập nhật, cả giá trị thời gian và giá trị quan sát được yêu cầu. Nếu các thuộc tính xuất hiện trong thông điệp xóa thì giá trị hợp lệ cung cấp cho thuộc tính cho biết giá trị hiện tại nên được xóa bỏ.

*Element Content (Type):*

Time (common:TimePeriodType)

ObsValue (ObsValueType) - giá trị nhỏ nhất: 0

Attributes (ValuesType) - min

Annotations (common:AnnotationsType) - giá trị nhỏ nhất: 0

**ValuesType** (Kiểu các giá trị):

*Element Content (Type):*

Value (ValueType) - giá trị lớn nhất: vô cùng

**ValueType** (Kiểu giá trị): được sử dụng để gán giá trị đơn vào khái niệm, như cho các giá trị thuộc tính và các giá trị khóa. Kiểu này không có nội dung phần tử.

*Attribute: concept* (xs:NCName)

*Attribute: value* (xs:string)

**ObsValueType** (Kiểu giá trị quan sát): mô tả tập thông tin về giá trị quan sát. Điều này được liên kết với khái niệm đơn vị đo chính được khai báo trong tập khóa.

*Attribute: value* (xs:double)

### 5.4 Mô đun tên miền về truy vấn SDMX

**Tên miền:** [http://www.SDMX.org/resources/SDMXML/schemas/v1\\_0/query](http://www.SDMX.org/resources/SDMXML/schemas/v1_0/query)

**Nhập:** [http://www.SDMX.org/resources/SDMXML/schemas/v1\\_0/common](http://www.SDMX.org/resources/SDMXML/schemas/v1_0/common)

(SDMXCommon.xsd)

#### 5.4.1 Phần tử chung

**Query(QueryType)** (Truy vấn (Kiểu truy vấn)): Thông điệp Query (truy vấn) cho phép truy vấn các cơ sở dữ liệu và dịch vụ web theo SDMX. Nó cho phép các truy vấn lấy lại dữ liệu, các tập khóa, các danh sách mã và các khái niệm.

#### 5.4.2 Kiểu phức

**QueryType** (Kiểu truy vấn): Phần tử Query (truy vấn) là phần tử ở mức trên cùng của tên miền này, nó được tham chiếu bởi phong bì thông điệp SDMX hoặc được đặt bên trong phong bì

khác, ví dụ như SOAP. Nó bao gồm một truy vấn. Thuộc tính defaultLimit (giới hạn mặc định) là kích cỡ hồi đáp lớn nhất được đề nghị tính bằng kilobyte.

*Element Content (Type):*

DataWhere (DataWhereType) - giá trị nhỏ nhất: 0 - giá trị lớn nhất: vô cùng

KeyFamilyWhere (KeyFamilyWhereType) - giá trị nhỏ nhất: 0 - giá trị lớn nhất: vô cùng

CodelistWhere (CodelistWhereType) - giá trị nhỏ nhất: 0 - giá trị lớn nhất: vô cùng

ConceptWhere (ConceptWhereType) - giá trị nhỏ nhất: 0 - giá trị lớn nhất: vô cùng

AgencyWhere (AgencyWhereType) - giá trị nhỏ nhất: 0 - giá trị lớn nhất: vô cùng

*Attribute:* defaultLimit (giới hạn mặc định) (xs:integer) - tùy chọn

**DataWhereType:** Phần tử DataWhere biểu diễn truy vấn về dữ liệu. Nó chứa tất cả các mệnh đề trong truy vấn đó, được biểu diễn bởi các phần tử con của nó.

*Element Content (Type):*

(Choice)

DataSet (xs:string)

KeyFamily (xs:string)

Dimension (DimensionType)

Attribute (AttributeType)

Codelist (CodelistType)

Time (TimeType)

Category (CategoryType)

Concept (xs:string)

Agency (xs:string)

Or (OrType)

And (AndType)

**AndType** (Kiểu And): Với phần tử And, mỗi thành phần con tức thời biểu diễn các mệnh đề, trong đó tất cả các mệnh đề biểu diễn phải được thỏa mãn. Nếu phần tử con là A, B và C thì mọi câu trả lời đúng cần đáp ứng tất cả các điều kiện A, B và C.

*Element Content (Type):*

DataSet (xs:string) - giá trị nhỏ nhất: 0 - giá trị lớn nhất: vô cùng

KeyFamily (xs:string) - giá trị nhỏ nhất: 0 - giá trị lớn nhất: vô cùng

Dimension (DimensionType) - giá trị nhỏ nhất: 0 - giá trị lớn nhất: vô cùng

**TCVN 7981-3 : 2009**

Attribute (AttributeType) - giá trị nhỏ nhất: 0 - giá trị lớn nhất: vô cùng

Codelist (CodelistType) - giá trị nhỏ nhất: 0 - giá trị lớn nhất: vô cùng

Time (TimeType) - giá trị nhỏ nhất: 0 - giá trị lớn nhất: vô cùng

Category (CategoryType) - giá trị nhỏ nhất: 0 - giá trị lớn nhất: vô cùng

Concept (xs:string) - giá trị nhỏ nhất: 0 - giá trị lớn nhất: vô cùng

Agency (xs:string) - giá trị nhỏ nhất: 0 - giá trị lớn nhất: vô cùng

Or (OrType) - giá trị nhỏ nhất: 0 - giá trị lớn nhất: vô cùng

And (AndType) - giá trị nhỏ nhất: 0 - giá trị lớn nhất: vô cùng

**OrType** (Kiểu Or): Phần tử con tức thời của phần tử Or biểu diễn các mệnh đề trong truy vấn, mà chỉ phần tử trong đó mới đáp ứng đủ điều kiện truy vấn. Nếu các phần tử con này là A, B và C thì bất kỳ kết quả đáp ứng điều kiện hoặc A hoặc B hoặc C cũng đều đáp ứng truy vấn đó.

*Element Content (Type):*

DataSet (xs:string) - giá trị nhỏ nhất: 0 - giá trị lớn nhất: vô cùng

KeyFamily (xs:string) - giá trị nhỏ nhất: 0 - giá trị lớn nhất: vô cùng

Dimension (DimensionType) - giá trị nhỏ nhất: 0 - giá trị lớn nhất: vô cùng

Attribute (AttributeType) - giá trị nhỏ nhất: 0 - giá trị lớn nhất: vô cùng

Codelist (CodelistType) - giá trị nhỏ nhất: 0 - giá trị lớn nhất: vô cùng

Time (TimeType) - giá trị nhỏ nhất: 0 - giá trị lớn nhất: vô cùng

Category (CategoryType) - giá trị nhỏ nhất: 0 - giá trị lớn nhất: vô cùng

Concept (xs:string) - giá trị nhỏ nhất: 0 - giá trị lớn nhất: vô cùng

Agency (xs:string) - giá trị nhỏ nhất: 0 - giá trị lớn nhất: vô cùng

Or (OrType) - giá trị nhỏ nhất: 0 - giá trị lớn nhất: vô cùng

And (AndType) - giá trị nhỏ nhất: 0 - giá trị lớn nhất: vô cùng

**DimensionType** (Kiểu miền dữ liệu): Các phần tử Dimension (miền dữ liệu) chứa giá trị (đơn) được tìm kiếm trong khóa của tập dữ liệu. Thuộc tính tên nắm giữ ID cơ quan duy trì của miền dữ liệu. Nếu nội dung là rỗng thì truy vấn dùng miền dữ liệu với tên đã cho. Nếu thuộc tính tên không được cung cấp thì truy vấn dùng cho giá trị khóa cho trước trong mọi miền dữ liệu.

[data] (xs:string)

**AttributeType** (Kiểu thuộc tính): Các phần tử Attribute (thuộc tính) chứa giá trị (đơn) của thuộc tính được truy vấn. Thuộc tính tên chứa tên đủ điều kiện của thuộc tính. Thuộc tính attachmentLevel quy định mức đính kèm của thuộc tính. Nếu nội dung của Attribute (thuộc tính) là rỗng thì việc tìm kiếm sẽ thay cho thuộc tính quy định (và mức đính kèm). Nếu thuộc tính tên

không được quy định thì truy vấn sẽ thay cho thuộc tính ở mức đính kèm, khi các mặc định về giá trị là "Any"

[data] (xs:string)

**CodelistType** (Kiểu danh sách mã): Phần tử Codelist (danh sách mã) cho phép các truy vấn quy định giá trị (đơn) được tìm trong danh sách mã như nội dung phần tử và tên đủ điều kiện của danh sách mã được truy vấn trong thuộc tính tên. Nếu không có nội dung nào được cung cấp thì truy vấn sẽ thay cho danh sách mã được đặt tên. Nếu thuộc tính tên là rỗng thì giá trị được tìm kiếm trong bất kỳ danh sách mã.

[data] (xs:string)

**CategoryType** (Kiểu phân loại): Phần tử Category (phân loại) chấp nhận một tìm kiếm được tạo trên các giá trị trong phân loại cụ thể, được quy định với thuộc tính tên. Nếu không có nội dung phần tử thì việc tìm kiếm được thực hiện đối với phân loại được đặt tên; nếu tên không được cung cấp thì giá trị phân loại sẽ được cung cấp như nội dung được tìm thấy trong các phân loại có sẵn.

[data] (xs:string)

**KeyFamilyWhereType**: Phần tử KeyFamilyWhere biểu diễn một truy vấn về một tập khóa hoặc các tập khóa. Nó chứa tất cả mệnh đề trong truy vấn đó, được biểu diễn bởi các thành phần con.

*Element Content (Type):*

(Choice)

KeyFamily (xs:string)

Dimension (DimensionType)

Attribute (AttributeType)

Codelist (CodelistType)

Category (CategoryType)

Concept (xs:string)

Agency (xs:string)

Or (OrType)

And (AndType)

**CodelistWhereType**: Phần tử CodelistWhere biểu diễn truy vấn cho một danh sách mã hoặc nhiều danh sách mã. Phần tử này chứa tất cả mệnh đề trong truy vấn đó, được biểu diễn bởi các phần tử con.

*Element Content (Type):*

(Choice)

Codelist (CodelistType)

## TCVN 7981-3 : 2009

Agency (xs:string)

Or (OrType)

And (AndType)

**ConceptWhereType:** Phần tử ConceptWhere biểu diễn truy vấn cho một khái niệm hoặc nhiều khái niệm. Phần tử này chứa tất cả các mệnh đề trong truy vấn đó và được biểu diễn bởi các phần tử con.

*Element Content (Type):*

(Choice)

Concept (xs:string)

Agency (xs:string)

Or (OrType)

And (AndType)

**AgencyWhereType:** Phần tử AgencyWhere biểu diễn truy vấn về các thông tin chi tiết của Agency. Phần tử này bao gồm tất cả các mệnh đề trong truy vấn đó và được biểu diễn bởi các thành phần con.

*Element Content (Type):*

(Choice)

DataSet (xs:string) - giá trị nhỏ nhất: 0 - giá trị lớn nhất: vô cùng

KeyFamily (xs:string) - giá trị nhỏ nhất: 0 - giá trị lớn nhất: vô cùng

Codelist (CodelistType) - giá trị nhỏ nhất: 0 - giá trị lớn nhất: vô cùng

Category (CategoryType) - giá trị nhỏ nhất: 0 - giá trị lớn nhất: vô cùng

Concept (xs:string) - giá trị nhỏ nhất: 0 - giá trị lớn nhất: vô cùng

Agency (xs:string) - giá trị nhỏ nhất: 0 - giá trị lớn nhất: vô cùng

Or (OrType) - giá trị nhỏ nhất: 0 - giá trị lớn nhất: vô cùng

And (AndType) - giá trị nhỏ nhất: 0 - giá trị lớn nhất: vô cùng

**TimeType**(Kiểu thời gian): chứa điểm thời gian hoặc chu kỳ thời gian mà các kết quả được cung cấp. Khi StartTime (thời gian bắt đầu) và EndTime (thời gian kết thúc) được sử dụng thì cần hiểu đây là toàn bộ các dải thời gian.

*Element Content (Type):*

(Choice)

StartTime (common:TimePeriodType)

EndTime (common:TimePeriodType) - giá trị nhỏ nhất: 0

Or:

Time (common:TimePeriodType)

### 5.4.3 Kiểu đơn giản

**AttachmentLevelType** (Kiểu đính kèm): Kiểu này cung cấp bảng liệt kê các mức đính kèm tương ứng với các mức đính kèm trong Mô hình thông tin SDMX, cộng với giá trị "Any" mà trong đó, việc tìm kiếm được thay thế bằng ký tự.

*Restricts xs:NMTOKEN*

Code: DataSet - Được gắn ở mức tập dữ liệu

Code: Group - Được gắn ở mức nhóm

Code: Series - Được gắn ở mức chuỗi

Code: Observation - Được gắn ở mức quan sát

Code: Any - Được gắn ở mọi mức đính kèm

## 5.5 Mô đun tên miền chung về SDMX

**Tên miền:** [http://www.SDMX.org/resources/SDMXML/schemas/v1\\_0/common](http://www.SDMX.org/resources/SDMXML/schemas/v1_0/common)

**Nhập:** <http://www.w3.org/XML/1998/namespace> (xml.xsd)

### 5.5.1 Kiểu phức

**TextType** (Kiểu văn bản): đưa ra một tập các phương án khác nhau về ngôn ngữ để cung cấp các kết cấu mà con người có thể đọc.

[data] (xs:string)

**AnnotationType** (Kiểu ghi chú): cung cấp các chú thích phi tài liệu và ghi chú được nhúng trong dữ liệu và các thông điệp cấu trúc. Kiểu này đưa ra các trường tùy chọn để cung cấp nhan đề, mô tả kiểu, URL và văn bản ghi chú.

*Element Content (Type):*

AnnotationTitle (xs:string) - giá trị nhỏ nhất: 0

AnnotationType (xs:string) - giá trị nhỏ nhất: 0

AnnotationURL (xs:anyURI) - giá trị nhỏ nhất: 0

AnnotationText (TextType) - giá trị nhỏ nhất: 0 - giá trị lớn nhất: vô cùng

**AnnotationsType** (Kiểu các ghi chú): cung cấp danh sách các ghi chú được đính kèm tới dữ liệu và các thông điệp cấu trúc.

*Element Content (Type):*

Annotation (AnnotationType) - giá trị lớn nhất: vô cùng

### 5.5.2 Kiểu đơn giản

**TimePeriodType** (Kiểu chu kỳ thời gian): TIME\_PERIOD không thể trình bày một cách đầy đủ trong kiểu ngày tháng của lược đồ XML: chúng ta sử dụng liên kết của dateTime, date, gYearMonth và gYear để thay thế. Tên mặc định cho khái niệm là TIME\_PERIOD. Các chu kỳ hàng quý và nửa năm được mô tả dưới dạng tháng bắt đầu, các chu kỳ hàng tuần dưới dạng thứ hai đầu tuần: ví dụ: quý thứ hai của năm 2002 là 2002-4, bắt đầu với tháng tư.

**ActionType** (Kiểu hoạt động): cung cấp danh sách các hoạt động, mô tả mục đích truyền dữ liệu từ phía người gửi. Mỗi hoạt động áp dụng cho toàn bộ các tập dữ liệu mà nó đưa ra.

*Restricts xs:NMTOKEN*

Code: Update - Dữ liệu là một cập nhật gia tăng dần về tập dữ liệu hiện có hoặc cung cấp dữ liệu hoặc tài liệu mới (các giá trị thuộc tính) vắng mặt trước đó.

Code: Delete - Dữ liệu được xóa.

**AlphaType** (Kiểu alpha): Kiểu này được sử dụng cho việc định kiểu các nội dung của thuộc tính không được mã hóa. Không có giới hạn nào trên các ký tự được sử dụng, nhưng mang ngữ nghĩa của trình thiết kế tập khóa theo dạng tương tự với trình thiết kế của thông điệp SDMX EDI.

*Restricts xs:string*

**AlphaNumericType** (Kiểu chữ số): Kiểu này được sử dụng cho việc định kiểu các nội dung của thuộc tính không được mã hóa. Không có giới hạn nào trên các ký tự được sử dụng, nhưng mang ngữ nghĩa của trình thiết kế tập khóa theo dạng tương tự với trình thiết kế của thông điệp SDMX EDI.

*Restricts xs:string*

## 5.6 Định dạng dữ liệu và mã hóa ký tự

Trong tất cả các tài liệu SDMX-ML - cho dù các tài liệu đó có chuyên về tập khóa hay không - ký tự mã hóa phải là UTF-8. Để đơn giản hóa việc trao đổi siêu dữ liệu và dữ liệu thông kê ở phạm vi toàn cục, các giới hạn cũng áp dụng cho việc trình bày các định dạng số: dấu phân cách thập phân luôn là ("."). Trong dữ liệu, không dùng ký tự để ngăn cách hàng nghìn.

## 6 Lược đồ tập khóa chuyên dụng: Cấu trúc chính và ánh xạ chuẩn

Các lược đồ đặc trưng cho tập khóa, do đó không có lược đồ đơn nào dành cho tất cả đối tượng sử dụng. Trong các trường hợp này, các ánh xạ chuẩn được cung cấp nhằm ngăn việc đưa ra các lược đồ, các lược đồ có thể được báo trước từ việc kiểm tra các thông điệp cấu trúc SDMX mô tả các tập khóa chúng được lấy làm cơ sở. Việc tạo tự động các lược đồ tập

khóa chuyên dụng theo các ánh xạ này là một hệ quả tự nhiên và không có công cụ miễn phí nào dùng cho việc tạo lược đồ tập khóa chuyên dụng này.

Điều quan trọng cần chú ý là tất cả lược đồ tập khóa chuyên dụng đều dựa trên điểm trung tâm của các cấu trúc định danh, cho phép có được số lượng nhỏ nhất các thẻ phân biệt tập khóa với



tập khóa. Điều này mô tả các cấu trúc "lõi", mỗi cấu trúc có trong môđun vùng tên SDMX của chính nó, sau đó thảo luận các ánh xạ từ tập khóa tới giản đồ tập khóa

Các lược đồ này là tương tự nhau. Chúng biến đổi theo vị trí trong các thuộc tính và giá trị khóa về cấu trúc chung có thể được quy định, ngoài ra trong trường hợp dữ liệu phần giao - chấp nhận việc thời gian được quy định một lần, ở mức tập dữ liệu, cùng với sự tác động của nhiều quan sát. Sự khác biệt ít hơn được thấy trong lược đồ Utility (tiện ích), được thiết kế để mang nhiều siêu dữ liệu cấu trúc, cho phép các công cụ XML "điển hình" (ví dụ như các trình soạn thảo và phân tích được hướng dẫn bằng lược đồ) mang lại lợi ích từ tính sẵn có của dữ liệu này - nhìn chung, các công cụ này không có khả năng tra cứu tập khóa về siêu dữ liệu có cấu trúc.

### 6.1 Cấu trúc chính của thông điệp Compact Data (dữ liệu cô đọng)

Tên miền: [http://www.SDMX.org/resources/SDMXML/schemas/v1\\_0/compact](http://www.SDMX.org/resources/SDMXML/schemas/v1_0/compact)

Nhập: [http://www.SDMX.org/resources/SDMXML/schemas/v1\\_0/common](http://www.SDMX.org/resources/SDMXML/schemas/v1_0/common)

(SDMXCommon.xsd)

#### 6.1.1 Phần tử toàn cục

**DataSet(DataSetType)** (Tập dữ liệu (Kiểu tập dữ liệu)): Phần tử DataSet chứa tập dữ liệu.

**Group(GroupType)** (Nhóm (Kiểu nhóm)): Phần tử Group chứa nhóm.

**Series(SeriesType)** (Chuỗi (Kiểu chuỗi)): Phần tử Series chứa các chuỗi.

**Obs(ObsType)** (Quan sát (Kiểu quan sát)): Phần tử Obs chứa quan sát.

#### 6.1.2 Kiểu phức

**DataSetType** (Kiểu tập dữ liệu): thực hiện vai trò của cơ sở có cấu trúc, được mở rộng qua việc bổ sung các giá trị thuộc tính để phản ánh các nhu cầu riêng của tập khóa sử dụng xs:extends element.

**GroupType** (Kiểu nhóm): thực hiện vai trò của cơ sở có cấu trúc, được đặt lại tên và mở rộng qua việc bổ sung các thuộc tính để phản ánh các nhu cầu riêng của tập khóa cụ thể sử dụng xs:extends element.

**SeriesType** (Kiểu chuỗi): thực hiện vai trò của cơ sở có cấu trúc, được mở rộng qua việc bổ sung các thuộc tính để phản ánh các nhu cầu riêng của tập khóa cụ thể sử dụng xs:extends element.

**ObsType** (Kiểu quan sát): thực hiện vai trò của cơ sở có cấu trúc, được mở rộng qua việc bổ sung các thuộc tính để phản ánh các nhu cầu riêng của tập khóa cụ thể sử dụng xs:extends element.

### 6.2 Cấu trúc chính của thông điệp Utility Data (dữ liệu tiện ích)

Tên miền: [http://www.SDMX.org/resources/SDMXML/schemas/v1\\_0/utility](http://www.SDMX.org/resources/SDMXML/schemas/v1_0/utility)

Nhập: [http://www.SDMX.org/resources/SDMXML/schemas/v1\\_0/common](http://www.SDMX.org/resources/SDMXML/schemas/v1_0/common)

(SDMXCommon.xsd)

## TCVN 7981-3 : 2009

### 6.2.1 Phần tử toàn cục

**DataSet(DataSetType)** (Tập dữ liệu (Kiểu tập dữ liệu)): tồn tại để thực hiện vai trò như tiêu đề của nhóm thay thế mà các phần tử và thuộc tính về tập khóa được giới hạn.

**Group(GroupType)** (Nhóm (Kiểu nhóm)): tồn tại để thực hiện vai trò như tiêu đề của nhóm thay thế mà các phần tử và thuộc tính về tập khóa được giới hạn.

**Series(SeriesType)** (Chuỗi (Kiểu chuỗi)): tồn tại để thực hiện vai trò như tiêu đề của nhóm thay thế mà các phần tử và thuộc tính về tập khóa được giới hạn.

**Key(KeyType)** (Khóa (Kiểu khóa)): là phần tử mà các dùng làm tiêu đề của nhóm thay thế chứa các giá trị khóa về tập khóa.

**Obs(ObsType)** (Quan sát (Kiểu quan sát)): tồn tại để thực hiện vai trò như tiêu đề của nhóm thay thế mà các phần tử và thuộc tính về tập khóa bị giới hạn.

### 6.2.2 Kiểu phức

**DataSetType** (Kiểu tập dữ liệu): thực hiện vai trò của cơ sở có cấu trúc, được mở rộng qua việc bổ sung các giá trị thuộc tính để phản ánh các nhu cầu riêng của tập khóa sử dụng xs:extends element.

**GroupType** (Kiểu nhóm): thực hiện vai trò của cơ sở có cấu trúc, được đặt lại tên và mở rộng qua việc bổ sung các thuộc tính để phản ánh các nhu cầu riêng của tập khóa cụ thể sử dụng xs:extends element.

**SeriesType** (Kiểu chuỗi): thực hiện vai trò của cơ sở có cấu trúc, được mở rộng qua việc thêm các thuộc tính để phản ánh các nhu cầu riêng của tập khóa cụ thể sử dụng xs:extends element.

**KeyType** (Kiểu khóa): mô tả kiểu trừu tượng, kiểu này xác định phần tử khóa.

**ObsType** (Kiểu quan sát): thực hiện vai trò của cơ sở có cấu trúc, được mở rộng qua việc bổ sung các thuộc tính để phản ánh các nhu cầu riêng của tập khóa cụ thể sử dụng xs:extends element.

## 6.3 Cấu trúc chính của thông điệp Cross-Sectional Data (dữ liệu phần giao)

Tên miền: [http://www.SDMX.org/resources/SDMXML/schemas/v1\\_0/cross](http://www.SDMX.org/resources/SDMXML/schemas/v1_0/cross)

Nhập: [http://www.SDMX.org/resources/SDMXML/schemas/v1\\_0/common](http://www.SDMX.org/resources/SDMXML/schemas/v1_0/common)

(SDMXCommon.xsd)

### 6.3.1 Phần tử toàn cục

**DataSet(DataSetType)** (Tập dữ liệu (Kiểu tập dữ liệu)): chứa tập dữ liệu.

**Group(GroupType)** (Nhóm (Kiểu nhóm)): chứa nhóm.

**Section(SectionType)** (Phần (Kiểu phần)): chứa phần.

**Obs(ObsType)** (Quan sát (Kiểu quan sát)): chứa quan sát với một hay nhiều đơn vị đo.

### 6.3.2 Kiểu phức

**DataSetType** (Kiểu tập dữ liệu): thực hiện vai trò của cơ sở có cấu trúc, được mở rộng qua việc bổ sung các giá trị thuộc tính để phản ánh các nhu cầu riêng của tập khóa sử dụng xs:extends element.

**GroupType** (Kiểu nhóm): thực hiện vai trò của cơ sở có cấu trúc, được mở rộng qua việc bổ sung các giá trị thuộc tính để phản ánh các nhu cầu riêng của tập khóa sử dụng xs:extends element. Thuộc tính thời gian giữ giá trị về khái niệm miền dữ liệu thời gian được quy định trong tập khóa. Nếu thời gian không được sử dụng như khái niệm trong tập khóa, thì không có giá trị nào cần được cung cấp.

*Attribute: time (common:TimePeriodType) - tùy chọn*

**SectionType** (Kiểu phần): thực hiện vai trò của cơ sở có cấu trúc, được mở rộng qua việc bổ sung các giá trị thuộc tính để phản ánh các nhu cầu riêng của tập khóa sử dụng xs:extends element.

**ObsType** (Kiểu quan sát): thực hiện vai trò của cơ sở có cấu trúc, được mở rộng qua việc bổ sung các giá trị thuộc tính để phản ánh các nhu cầu riêng của tập khóa sử dụng xs:extends element. Nó có khả năng trình bày giá trị và các thuộc tính của đơn vị đo phần giao sẵn có (khi mở rộng).

## 6.4 Ánh xạ tới các lược đồ tập khóa chuyên dụng

### 6.4.1 Các quy tắc chung

Đối với tất cả lược đồ về tập khóa (Compact (cô đọng), Utility(tiện ích) và Cross-Sectional (phần giao)) SDMX cung cấp tên miền được sử dụng làm cơ sở để mở rộng các lược đồ về tập khóa. Lược đồ về tập khóa chuyên dụng được tạo ra trong tên miền đích, được sở hữu và duy trì bởi cơ quan tạo ra nó. Sử dụng thuộc tính targetNamespace của phần tử lược đồ để định danh tên miền chứa lược đồ về tập khóa. Mô đun tên miền được cung cấp bởi SDMX mặc dù lớp lược đồ tập khóa chuyên dụng được kết hợp chặt chẽ bằng cách sử dụng phần tử nhập trong lược đồ tập khóa chuyên dụng. Mô đun tên miền chung về SDMX cũng phải được nhập vào lược đồ. Các thuộc tính xml:namespace có thể được bổ sung cho phần tử lược đồ khi được yêu cầu.

Thuộc tính elementFormDefault trên phần tử lược đồ đưa ra giá trị "giới hạn", còn thuộc tính attributeFormDefault sẽ đưa ra giá trị "vô cùng".

Tất cả các bổ sung cho mô đun SDMX sẽ được tạo ra bằng cách sử dụng phần tử mở rộng từ Lược đồ XML của W3C. Thuật ngữ "các mức cấu trúc" khi tham chiếu tới các mô đun SDMX nhập, bao gồm:

- Mức DataSet (tập dữ liệu)
- Mức Group (nhóm)
- Mức Series (chuỗi)

## TCVN 7981-3 : 2009

- Mức Observation (quan sát)

Các mức này thường tham chiếu đến phần tử được cung cấp bởi môđun SDMX trong đó các thuộc tính và phần tử có thể được gán. Trong một vài trường hợp, các kết cấu được đặt tên cụ thể trong tập khóa sẽ trở thành các thành viên của một tập các phần tử tương ứng với một trong các mức được đặt tên ở trên.

Với tất cả các ánh xạ về tập khóa ở trên, các môđun tên miền SDMX-ML được định danh với các từ viết tắt được sử dụng trong các lược đồ chuẩn ("compact:" tham chiếu tới môđun CompactData; "common:" tới môđun tên miền Common, "utility:" tới môđun tên miền UtilityData; và "cross:" tới môđun CrossSectionalData).

Chú ý rằng tất cả các ánh xạ sau đây, thuật ngữ "tên khái niệm" là giá trị của thuộc tính id của khái niệm khi được tìm thấy trong thông điệp SDMX-ML mô tả tập khóa.

### 6.4.2 Lược đồ Compact (cô đọng)

Lược đồ Compact (cô đọng) biểu diễn tất cả các giá trị thuộc tính và giá trị miền dữ liệu như các thuộc tính XML. Các thuộc tính này được đặt ở các mức khác nhau trong cấu trúc "cô đọng" được nhập. Lược đồ về tập khóa sử dụng các nhóm thay thế XSD để gán các phần tử và thuộc tính tập khóa tới các cấu trúc được cung cấp trong tên miền "compact:".

Phần tử toàn cục có tên "DataSet" được khai báo, với thuộc tính XSD substitutionGroup có giá trị tham chiếu đến phần tử DataSet trong "compact:" namespace. Thuộc tính kiểu tham chiếu DataSetType trong tên miền về tập khóa.

XSD complexType được khai báo có tên "DataSetType". Có XSD complexContent chứa phần tử mở rộng XSD, với thuộc tính cơ sở của DataSetType trong "compact:" namespace. Việc mở rộng bao gồm một phần tử lựa chọn XSD, với thuộc tính minOccurs có giá trị "0" và maxOccurs có giá trị "vô cùng". Lựa chọn chứa tham chiếu phần tử XSD đối với mỗi nhóm được đặt tên khai báo trong tập khóa. Chúng có thuộc tính tham chiếu XSD với giá trị của tên nhóm được cung cấp trong tập khóa. (Các phần tử này lấy các tên của nhóm được khai báo trong tập khóa). Ngoài ra, phần tử XSD sẽ được khai báo theo lựa chọn với thuộc tính tham chiếu có giá trị của Series (chuỗi). Thêm nữa, phần tử có tên Annotations (ghi chú) sẽ được khai báo theo lựa chọn với kiểu AnnotationsType từ tên miền "common:"

Đối với mỗi thuộc tính được khai báo trong tập khóa với attachmentLevel của "DataSet", thuộc tính XML cũng được khai báo trong phần mở rộng. Thuộc tính này có cùng tên với khái niệm của thuộc tính trong tập khóa, có giá trị thuộc tính sử dụng là "tùy chọn". Đối với các thuộc tính được mã hóa, thuộc tính XML được gán với một kiểu giá trị là tên của danh sách mã biểu diễn nó. Trong tên miền về tập khóa, danh sách mã này được biểu diễn bởi khai báo simpleType, khai báo này chứa danh sách các bảng liệt kê, tương đương với các giá trị của danh sách mã như đã mô tả trong tập khóa. Chúng là phần mở rộng của kiểu dữ liệu "string" XSD. Các giá trị liệt kê là giá trị của các mã. Mô tả của các mã được đặt bên trong các phần tử "tài liệu" của XSD và được chứa trong các phần tử "ghi chú", bản thân chúng được chứa trong các phần tử "liệt kê" giống như

trường hợp đầu tiên của phần tử tài liệu XSD. Không có văn bản nào khác xuất hiện trong trường hợp riêng của phần tử tài liệu XSD, mặc dù các phần tử tài liệu XSD khác có thể xuất hiện trong mọi phần tử liệt kê XSD.

Các thuộc tính không mã hóa cũng được biểu diễn với các phần tử simpleType của XSD khai báo trong tên miền của tập khóa, với các tên được tạo bằng cách lấy tên của thuộc tính trong tập khóa và viết thêm "Type". Nếu không bị giới hạn thì các thuộc tính sẽ nằm trong lược đồ XML của W3C với kiểu gốc "string (chuỗi ký tự)"; nếu các giới hạn được quy định trong tập khóa, thì các thuộc tính sẽ là các giới hạn của kiểu dữ liệu "string" XSD, trừ khi chúng có độ dài tối đa được quy định trong tập khóa. Nếu độ dài tối đa được cung cấp trong mô tả tập khóa thì nó sẽ áp dụng các điều sau đây:

- Nếu là dạng số thì cơ sở giới hạn sẽ là kiểu dữ liệu "decimal".
- Nếu là dạng chữ cái thì cơ sở giới hạn sẽ là kiểu dữ liệu common:AlphaType.
- Nếu là dạng vừa số vừa chữ thì cơ sở giới hạn sẽ là kiểu dữ liệu common:AlphaNumericType

Nếu độ dài tối đa được quy định, nhưng giá trị của thuộc tính không phải là độ dài cố định thì maxLength trong XSD simpleType nên được lấy bằng với độ dài tối đa của thuộc tính như đã quy định trong tập khóa. Nếu giá trị của thuộc tính là độ dài cố định thì thuộc tính XSD minLength nên được lấy cùng một giá trị. Nếu một thuộc tính không được mã hóa là kiểu số và số lượng các phân số thập phân được quy định trong tập khóa, thì fracDig của kiểu đơn giản nên lấy giá trị được quy định trong tập khóa.

Đối với mỗi nhóm được đặt tên trong tập khóa, phần tử XSD toàn cục được khai báo, lấy tên của nhóm. Thuộc tính kiểu XSD có giá trị được tạo bằng cách lấy tên của phần tử và thêm "Type" vào cuối. Nó có thuộc tính substitutionGroup mà các tham chiếu phần tử Group (nhóm) được khai báo trong tên miền "compact:".

XSD complexType được khai báo cho mỗi nhóm được đặt tên trong tập khóa, với tên được tạo bằng cách lấy tên của nhóm trong tập khóa và viết thêm "Type". Nó có một phần tử XSD complexContent chứa XSD extends với giá trị thuộc tính cơ sở compact:GroupType. Phần tử mở rộng chứa phần tử chuỗi XSD. Một phần tử có tên Annotations (ghi chú) được bổ sung cho đoạn cuối của chuỗi, với kiểu AnnotationsType từ "common:"namespace. Nó cũng có giá trị minOccurs là "0".

Đối với mỗi thuộc tính trong tập khóa với attachmentLevel của "Group (nhóm)", phần tử thuộc tính XSD được bổ sung phần tử mở rộng, với thuộc tính sử dụng là "tùy chọn" và thuộc tính kiểu được xác định cho mức DataSet ở trên. Thuộc tính Name (tên) đề cập ở đây là tên khái niệm của thuộc tính trong tập khóa.

Đối với mỗi miền dữ liệu được tham chiếu bởi phần tử DimensionRef (tham chiếu miền dữ liệu) trong khai báo Group (nhóm) được đặt tên ở tập khóa XML, một phần tử thuộc tính XSD cũng được bổ sung cho phần tử mở rộng, với thuộc tính sử dụng đặt là "theo quy định" và thuộc tính

## TCVN 7981-3 : 2009

kiểu được xác định cho các thuộc tính được mã hóa ở mức tập dữ liệu ở trên. Thuộc tính Name(tên) đề cập ở đây là tên khái niệm của miền dữ liệu trong tập khóa.

Phần tử toàn cục XSD có tên Series (chuỗi) được khai báo trong tên miền của tập khóa, với kiểu thuộc tính SeriesType (kiểu chuỗi) và substitutionGroup tham chiếu compact:Series.

XSD complexType được khai báo với tên SeriesType (kiểu chuỗi). Nó chứa XSD complexContent với phần tử mở rộng XSD mà có giá trị thuộc tính cơ sở của compact:SeriesType. Phần tử mở rộng chứa phần tử chuỗi XSD, bao gồm phần tử XSD với thuộc tính ref có giá trị là "Obs". Thuộc tính minOccurs có giá trị "0" và maxOccurs có giá trị "vô cùng". Phần tử có tên Annotations (ghi chú) được bổ sung cho đoạn cuối của chuỗi, với kiểu AnnotationsType từ "common:" namespace. Nó cũng có giá trị minOccurs là "0".

Đối với mỗi thuộc tính trong tập khóa với attachmentLevel là "Series (chuỗi)", phần tử thuộc tính XSD được bổ sung cho phần tử mở rộng, với thuộc tính sử dụng đặt là "tùy chọn" và thuộc tính kiểu được xác định cho mức DataSet (tập dữ liệu) ở trên. Thuộc tính Name là tên của khái niệm thuộc tính trong tập khóa. Ngoại lệ là khi một thuộc tính có giá trị thuộc tính isTimeFormat là "true" -trong trường hợp này, nó được xử lý như các thuộc tính mức chuỗi khác trừ thuộc tính sử dụng có giá trị "theo quy định".

Phần tử toàn cục XSD được khai báo có tên "Obs". Phần tử này có thuộc tính substitutionGroup với giá trị "compact:Obs" và kiểu "ObsType".

Phần tử XSD complexType được khai báo với tên "ObsType" và XSD complexContent. Nó chứa phần tử XSD extends với thuộc tính cơ sở "compact:ObsType" và phần tử chuỗi XSD. Phần tử chuỗi chứa phần tử có tên là Annotation (ghi chú), với kiểu AnnotationsType từ "common:" namespace. Phần tử này có giá trị minOccurs là "0".

Phần tử mở rộng cũng có phần tử thuộc tính XSD trong đó, sẽ có một thuộc tính tên mà giá trị của nó là tên của khái niệm TimeDimension (miền dữ liệu thời gian) từ tập khóa. Phần tử này có thuộc tính sử dụng là "tùy chọn" và thuộc tính kiểu là "common:TimePeriodType".

Phần tử mở rộng có phần tử thuộc tính XSD trong đó, có tên thuộc tính mà giá trị của nó là tên khái niệm của đơn vị đo đầu tiên từ tập khóa. Phần tử này có thuộc tính sử dụng là "tùy chọn" và thuộc tính kiểu là XSD "double".

Đối với mỗi thuộc tính được khai báo trong tập khóa với attachmentLevel của "Observation(quan sát)", thuộc tính XSD được bổ sung cho phần tử mở rộng. Mỗi thuộc tính XSD lấy tên của khái niệm thuộc tính được khai báo trong tập khóa và có thuộc tính sử dụng là "tùy chọn". Thuộc tính kiểu được xác định cho các thuộc tính mức DataSet (tập dữ liệu) mô tả ở trên.

Không có kết cấu hay khai báo nào khác được bổ sung cho các lược đồ được tạo bằng cách sử dụng ánh xạ này.

**Các dải thời gian trong CompactData** (dữ liệu cô đọng): Không giống với định dạng dữ liệu SDMX-ML khác, định dạng CompactData (dữ liệu cô đọng) của tập khóa có thể biểu diễn một tập

các giá trị quan sát mà không cho phép cung cấp thời gian cho mỗi giá trị. Nếu một Series (chuỗi) có thời gian được cung cấp cho quan sát đầu tiên, thì các quan sát tiếp theo trong chuỗi có thể bỏ qua thời gian, chỉ cung cấp giá trị quan sát (giá trị về thuộc tính được đặt tên sau đơn vị đo đầu tiên) và bất kỳ thuộc tính nào được yêu cầu (xem bên dưới). Các lần quan sát tiếp theo có thể được tính toán theo tần suất được quy định bởi giá trị thuộc tính về định dạng thời gian liên quan (nếu không có giá trị miền dữ liệu tần suất), có thể được tính toán bởi ứng dụng. Chú ý rằng hỗ trợ về chức năng này không bắt buộc cho các ứng dụng không đòi hỏi hỗ trợ này trong các tuyên bố về sự phù hợp của chúng. Nó cũng cho phép cung cấp giá trị thời gian cho quan sát cuối cùng trong chuỗi, cho phép kiểm tra kép việc tính toán mặc dù điều này không bắt buộc.

**Xóa và cập nhật các thông điệp trong CompactData** (dữ liệu cơ động): Trong phần tử Header (tiêu đề), trường hành động quy định thông điệp là loại cập nhật hay loại xóa. Nếu nó là thông điệp cập nhật thì sẽ được sử dụng để gửi thông tin mới hoặc thông tin cập nhật mà có thể chỉ chứa dữ liệu, tài liệu (đó là, các giá trị thuộc tính được mô tả trong tập khóa) hoặc cả hai. (Các thỏa thuận liên quan đến việc sử dụng các thông điệp cập nhật nếu được quy định giữa các bên đối tác). Đối với thông điệp xóa, yêu cầu là khóa của chuỗi thời gian luôn được gửi đi để xóa dữ liệu, được định danh như toàn bộ các chuỗi do thiếu các chu kỳ thời gian được quy định hoặc một tập các chu kỳ thời gian cụ thể, bao gồm các đoạn thời gian đó. Các giá trị thuộc tính có thể được xóa bằng cách gửi toàn bộ hoặc từng phần thuộc tính, với mọi giá trị hợp lệ về thuộc tính (theo lược đồ XSD) được lấy ra để biểu thị rằng giá trị thuộc tính hiện hành nên được xóa.

#### 6.4.3 Lược đồ Cross-Sectional (phần giao)

Các lược đồ phần giao của tập khóa biểu diễn cách trình bày dữ liệu dựa trên các chuỗi phi thời gian được tạo ra trong tập khóa. Chúng cũng có khả năng biểu diễn dữ liệu thống kê trong đó thời gian không phải là một khái niệm - đó là, chúng có thể cung cấp định dạng SDMX-ML cho dữ liệu mà vốn đã là dữ liệu phần giao. Với định dạng CompactData (dữ liệu cơ động), các giá trị khóa và giá trị thuộc tính được gắn với cấu trúc gồm bốn mức như các thuộc tính XML. Đối với dữ liệu phần giao, thuật ngữ "Series (chuỗi)" là từ viết tắt của "chuỗi thời gian" được thay thế bởi kết cấu "Section (phần)" tương đương.

Chú ý rằng các nhóm được khai báo trong tập khóa không quan tâm đến các mục đích của định dạng dữ liệu phần giao. Chúng được thay thế bởi phần tử Group (nhóm) chung, bỏ qua ứng dụng xử lý hoặc ghi để tuân theo tính hợp lệ của các giá trị thuộc tính về các nhóm Section (phần). Điều này là đúng bởi lược đồ phần giao đơn của SDMX-ML có thể được mô tả trong tập khóa như vậy nó cho phép nhiều miền dữ liệu được biểu diễn ở mức quan sát, thay thế vai trò của thời gian trong các định dạng hướng các chuỗi thời gian, do đó cho phép các giá trị khóa và giá trị thuộc tính được đính kèm ở nhiều mức.

Phần tử toàn cục có tên "DataSet (tập dữ liệu)" được khai báo với thuộc tính XSD substitutionGroup có giá trị tham chiếu đến phần tử DataSet trong tên miền "cross:". Thuộc tính kiểu tham chiếu DataSetType(kiểu tập dữ liệu) trong tên miền của tập khóa.

XSD complexType được khai báo có tên "DataSetType". Nó có XSD complexContent chứa phần

## TCVN 7981-3 : 2009

tử mở rộng XSD, với thuộc tính cơ sở của DataSetType trong tên miền "cross:". Phần mở rộng bao gồm phần tử lựa chọn XSD, với minOccurs là "0" và maxOccurs là "vô cùng". Phần tử lựa chọn bao gồm tham chiếu phần tử XSD với giá trị "Group (nhóm)". Ngoài ra, phần tử XSD được khai báo trong lựa chọn với thuộc tính ref, giá trị của nó là Section (phần). Thêm nữa, phần tử có tên Annotations (ghi chú) được bổ sung cho lựa chọn, với kiểu AnnotationsType (kiểu ghi chú) từ tên miền "common:". Nó có thuộc tính minOccurs là "0".

Đối với mỗi thuộc tính hoặc miền dữ liệu được khai báo trong tập khóa với crossSectionalAttachDataSet là "true", thuộc tính XML cũng được khai báo trong phần mở rộng. Nó có cùng tên với khái niệm thuộc tính hoặc khái niệm miền dữ liệu trong tập khóa. Và có giá trị thuộc tính sử dụng là "tùy chọn". Đối với các thuộc tính được mã hóa, thuộc tính XML đưa ra giá trị kiểu là tên của danh sách mã biểu diễn nó. Trong tên miền của tập khóa, danh sách mã này được biểu diễn bởi khai báo simpleType bao gồm danh sách các bảng liệt kê, tương đương với các giá trị của danh sách mã, như đã được mô tả trong tập khóa. Chúng là phần mở rộng của kiểu dữ liệu "string (chuỗi ký tự)". Các giá trị liệt kê là giá trị của các mã. Các mô tả của các mã được đặt bên trong các phần tử "tài liệu" XSD, được chứa trong các phần tử "ghi chú" XSD mà bản thân chúng được chứa trong các phần tử "liệt kê" XSD như trường hợp đầu tiên của phần tử tài liệu XSD. Không có một văn bản nào khác xuất hiện với trường hợp riêng của phần tử tài liệu XSD này, mặc dù các phần tử tài liệu XSD khác có thể xuất hiện trong mọi phần tử liệt kê XSD.

Các thuộc tính không được mã hóa cũng được biểu diễn với các phần tử XSD simpleType khai báo trong tên miền của tập khóa, với các tên được tạo bằng cách lấy tên của khái niệm thuộc tính trong tập khóa và viết thêm "Type". Nếu không được giới hạn, chúng sẽ là lược đồ W3C XML với kiểu ban đầu là "string"; nếu các giới hạn được quy định trong tập khóa, chúng sẽ là các giới hạn của kiểu dữ liệu "string", trừ khi chúng có độ dài lớn nhất quy định trong tập khóa. Nếu độ dài lớn nhất được cung cấp trong mô tả tập khóa thì nó vận dụng các điều sau:

- Nếu là dạng số thì cơ sở giới hạn là kiểu dữ liệu "decimal".
- Nếu là dạng chữ cái thì cơ sở giới hạn là kiểu dữ liệu common:AlphaType.
- Nếu là dạng vừa số vừa chữ thì cơ sở giới hạn là kiểu dữ liệu common:AlphaNumericType (trong đó, "common:" biểu thị mô đun tên miền SDMX)

Nếu độ dài lớn nhất được quy định, nhưng giá trị thuộc tính không phải là độ dài cố định thì maxLength trong XSD simpleType nên được đặt bằng với độ dài tối đa của thuộc tính như đã quy định trong tập khóa. Nếu giá trị của thuộc tính là độ dài cố định thì thuộc tính XSD minLength nên được đặt cùng một giá trị. Nếu một thuộc tính không được mã hóa là kiểu số và số lượng các phân số thập phân được quy định trong tập khóa, thì fracDig của kiểu đơn giản nên lấy giá trị được quy định trong tập khóa.

Phần tử XSD toàn cục được khai báo có tên Group (nhóm). Thuộc tính kiểu XSD có giá trị GroupType (kiểu nhóm). Phần tử này có thuộc tính substitutionGroup mà tham chiếu phần tử Group được khai báo trong tên miền "cross:".



XSD complexType có tên GroupType (kiểu nhóm) được khai báo. Nó có phần tử complexContent chứa XSD extends với giá trị thuộc tính cơ sở compact:GroupType. Phần tử mở rộng chứa phần tử chuỗi XSD, bao gồm phần tử XSD với một tham chiếu đến phần tử Section. Thuộc tính minOccurs của nó có giá trị là "0" và maxOccurs có giá trị là "vô cùng". Phần tử có tên Annotations (ghi chú) được bổ sung đến phần cuối của chuỗi, với kiểu AnnotationsType từ tên miền "common:". Phần tử này có giá trị minOccurs là "0"

Đối với mỗi thuộc tính hoặc miền dữ liệu trong tập khóa với crossSectionalAttachGroup có giá trị "true" isFrequencyDimension có giá trị "true", phần tử thuộc tính XSD được bổ sung đến phần tử mở rộng, với thuộc tính sử dụng đặt là "tùy chọn" và thuộc tính kiểu được xác định cho mức DataSet (tập dữ liệu) ở trên. Thuộc tính Name(tên) là tên của khái niệm thuộc tính hoặc khái niệm miền dữ liệu trong tập khóa.

Phần tử toàn cục XSD có tên Section (phần) được khai báo trong tên miền của tập khóa, với kiểu SectionType và thuộc tính substitutionGroup tham chiếu compact:Section.

XSD complexType được khai báo với tên SectionType. Nó chứa XSD complexContent với phần tử mở rộng XSD mang giá trị thuộc tính cross:SectionType. Phần tử mở rộng chứa phần tử lựa chọn XSD với minOccurs là "0" và maxOccurs là "vô cùng", chứa một phần tử XSD cho mỗi CrossSectionalMeasure (đơn vị đo phần giao) được khai báo trong tập khóa, với một thuộc tính ref mà giá trị của nó là tên của khái niệm đơn vị đo. Phần tử có tên Annotations được bổ sung đến đoạn cuối của phần tử lựa chọn, với kiểu AnnotationsType từ "common:" namespace.

Đối với mỗi thuộc tính hoặc miền dữ liệu trong tập khóa với crossSectionalAttachSection có giá trị "true", phần tử thuộc tính XSD được bổ sung cho phần tử mở rộng, với thuộc tính sử dụng đặt là "tùy chọn" và thuộc tính kiểu được xác định cho mức DataSet (tập dữ liệu) ở trên. Phần tử Name (tên) là tên của khái niệm thuộc tính hoặc khái niệm miền dữ liệu trong tập khóa.

Phần tử toàn cục XSD được khai báo cho mỗi CrossSectionalMeasure (đơn vị đo phần giao) trong tập khóa, với tên khái niệm của đơn vị đo. Phần tử này có một thuộc tính substitutionGroup với giá trị "cross:Obs" và có kiểu "ObsType". Nếu không có CrossSectionalMeasures (các đơn vị đo phần giao) được khai báo thì sử dụng PrimaryMeasure (đơn vị đo chính) để thay thế.

Phần tử XSD complexType được khai báo cho mỗi CrossSectionalMeasure (đơn vị đo phần giao) trong tập khóa với tên được tạo bằng cách viết thêm "Type" vào khái niệm của đơn vị đo. Các khai báo này chứa XSD complexContent. XSD complexContent chứa phần tử mở rộng XSD với thuộc tính cơ sở "cross:ObsType". Nó chứa phần tử chuỗi XSD. Phần tử chuỗi chứa phần tử có tên Annotations (ghi chú), với kiểu AnnotationsType (kiểu ghi chú) từ "common:" namespace. Nó sẽ có giá trị minOccurs là "0".

Phần tử mở rộng cũng có giá trị thuộc tính XSD bên trong về mỗi thuộc tính hoặc miền dữ liệu có crossSectionalAttachObservation giá trị "true" và các danh sách tên khái niệm của phép đo trong phần tử AttachmentMeasure (đơn vị đo kèm theo) ở khai báo của nó. Thuộc tính XSD mang giá trị tên từ tên khái niệm của thuộc tính. Nó có thuộc tính sử dụng là "tùy chọn" và thuộc tính kiểu được mô tả cho mức DataSet(tập dữ liệu). Ngoài ra, thuộc tính được khai báo với tên "value" và

## TCVN 7981-3 : 2009

kiểu XSD "double". Thuộc tính sử dụng là "tùy chọn". (Chú ý rằng với miền dữ liệu mà việc biểu diễn được mã hóa của nó tương ứng với CrossSectionalMeasures không bao giờ có thuộc tính crossSectionalAttachObservation là "true".)

Nếu không có CrossSectionalMeasures (đơn vị đo phần giao) được khai báo trong tập khóa thì sẽ có thuộc tính Name (tên) trong đó giá trị của nó là tên khái niệm của PrimaryMeasure (đơn vị đo chính) từ tập khóa. Nó có thuộc tính sử dụng là "tùy chọn" và thuộc tính kiểu là XSD "double".

Trong trường hợp này, với mỗi thuộc tính khai báo trong tập khóa với attachmentLevel là "Observation (quan sát)", thuộc tính XSD được bổ sung tới phần tử mở rộng. Mỗi thuộc tính XSD lấy tên khái niệm của thuộc tính khai báo trong tập khóa và có thuộc tính sử dụng là "tùy chọn". Thuộc tính kiểu của nó được xác định cho các thuộc tính mức DataSet (tập dữ liệu) mô tả ở trên. Ngoài ra, thuộc tính được khai báo với tên của giá trị và kiểu "xs:double". Thuộc tính sử dụng của nó là "tùy chọn".

Không có khai báo hoặc kết cấu nào khác được bổ sung cho các lược đồ được tạo bằng cách sử dụng ánh xạ này.

**Thông điệp xóa và cập nhật trong Cross-SectionalData (dữ liệu phần giao):** Trong phần tử Header (tiêu đề), trường hành động quy định thông điệp là cập nhật hay xóa. Nếu là thông điệp cập nhật, thì nó được sử dụng để gửi thông tin mới hoặc thông tin cập nhật, có thể chỉ chứa dữ liệu, tài liệu (đó là, các giá trị thuộc tính được mô tả trong tập khóa) hoặc cả hai. (Các thỏa thuận về việc sử dụng các thông điệp cập nhật nên được quy định giữa các bên đối tác.) Đối với thông điệp xóa, thì yêu cầu là khóa toàn bộ có thể được gửi tới để xóa dữ liệu, được định danh như toàn bộ các chuỗi bởi sự vắng mặt của các chu kỳ thời gian hoặc các giá trị có thể bị xóa bằng cách gửi toàn bộ hoặc từng phần bộ thuộc tính, với mọi giá trị hợp lệ về thuộc tính (theo lược đồ XSD) được lấy ra để cho biết giá trị thuộc tính hiện hành nên được xóa.

### 6.4.4 Lược đồ Utility (tiện ích)

Các lược đồ Utility (tiện ích) khác với các lược đồ Compact (cô đọng) và Cross-Sectional (phần giao) bởi vì chúng chỉ ra sự khác biệt giữa việc biểu diễn các thuộc tính và miền dữ liệu được thiết lập trong tập khóa. Thiết kế này duy trì thứ tự các khóa - thiết kế cung cấp nhiều siêu dữ liệu về mặt cấu trúc của tập khóa mà không yêu cầu bộ xử lý truy cập thông điệp cấu trúc mô tả tập khóa. Điều này tạo nên các quy tắc vốn có trong cấu trúc của tập khóa luôn sẵn có cho các công cụ này như trình soạn thảo XML được hướng dẫn bằng lược đồ, chúng là một phần lý do chính về định dạng lược đồ Utility (tiện ích).

Lược đồ Utility (tiện ích) sử dụng kỹ thuật tương tự với các lược đồ tập khóa chuyên dụng và lược đồ Cross-Sectional (phần giao) bằng cách tạo ra các nhóm thay thế được ghi ở đầu bởi các phần tử DataSet (tập dữ liệu), Group (nhóm), Series (chuỗi) và các mức Observation (quan sát). Điều này được làm theo cách mà các thông điệp có thể được thông qua hoàn toàn với trình phân tách XML chung nhưng kích cỡ rộng hơn các định dạng CompactData (dữ liệu cô đọng) hoặc CrossSectionalData (dữ liệu phần giao).

Phần tử toàn cục có tên "DataSet (tập dữ liệu)" được khai báo với thuộc tính XSD substitutionGroup có giá trị tham chiếu đến phần tử DataSet trong tên miền "utility". Thuộc tính kiểu tham chiếu DataSetType (kiểu tập dữ liệu) trong tên miền của tập khóa.

XSD complexType được khai báo có tên "DataSetType". Nó có XSD complexContent chứa phần tử mở rộng XSD với thuộc tính cơ sở của DataSetType trong tên miền "utility:". Phần mở rộng bao gồm phần tử chuỗi XSD chứa phần tử lựa chọn đầu tiên, với giá trị maxOccurs là "vô cùng". Việc lựa chọn chứa tham chiếu phần tử XSD cho mỗi nhóm có tên khai báo trong tập khóa. Chúng có thuộc tính ref XSD với giá trị tên nhóm cung cấp trong tập khóa. (Các phần tử này mang tên của các nhóm khai báo trong tập khóa). Nếu không có nhóm nào được đặt tên thì phần tử XSD sẽ được khai báo trong lựa chọn với thuộc tính ref có giá trị Series(chuỗi). Phần tử có tên Annotations (ghi chú) được bổ sung đến phần cuối của chuỗi, với kiểu AnnotationsType (kiểu ghi chú) từ tên miền "common:" và thuộc tính minOccurs là "0".

Đối với mỗi thuộc tính được khai báo trong tập khóa với attachmentLevel là "DataSet", thuộc tính XML được khai báo trong phần mở rộng. Nó có cùng tên với khái niệm của thuộc tính trong tập khóa. Nó còn có thuộc tính sử dụng với giá trị "theo quy định" nếu thuộc tính khai báo trong tập khóa có assignmentStatus là "bắt buộc" và thuộc tính sử dụng là "tùy chọn", nếu assignmentStatus trong tập khóa là "có điều kiện". Đối với các thuộc tính được mã hóa, thuộc tính XML đưa ra giá trị kiểu chính là id của danh sách mã biểu diễn nó. Trong tên miền của tập khóa, danh sách mã này được biểu diễn bởi khai báo simpleType, khai báo này chứa danh sách các bảng liệt kê, tương đương với các giá trị của danh sách mã khi được mô tả trong tập khóa. Chúng là phần mở rộng của kiểu dữ liệu "string" của XSD. Các giá trị liệt kê là các giá trị của mã. Các mô tả của các mã được đặt bên trong phần tử "tài liệu" của XSD, chứa trong các phần tử "ghi chú" của XSD, mà bản thân chúng được chứa trong các phần tử "liệt kê" XSD như một trường hợp đầu tiên của phần tử tài liệu XSD. Không có văn bản nào khác xuất hiện trong trường hợp cụ thể về phần tử mở rộng này, mặc dù các phần tử tài liệu XSD khác có thể xuất hiện trong phần tử liệt kê XSD cho trước.

Các thuộc tính không được mã hóa cũng được biểu diễn với các phần tử XSD simpleType khai báo trong tên miền của tập khóa, với các tên được tạo bằng cách lấy tên khái niệm của thuộc tính trong tập khóa và viết thêm "Type". Nếu không bị giới hạn, thì chúng sẽ là lược đồ W3C XML có kiểu gốc "string"; nếu các giới hạn được quy định trong tập khóa thì chúng sẽ là các giới hạn của kiểu dữ liệu "string" của XSD, trừ khi chúng có độ dài lớn nhất được quy định trong tập khóa. Nếu độ dài lớn nhất được cung cấp trong mô tả tập khóa, nó sẽ vận dụng các điều sau:

- Nếu là dạng số thì cơ sở giới hạn sẽ là kiểu dữ liệu "decimal".
- Nếu là dạng chữ cái thì cơ sở giới hạn sẽ là kiểu dữ liệu common:AlphaType.
- Nếu là dạng vừa số vừa chữ thì cơ sở giới hạn sẽ là kiểu dữ liệu common:AlphanumericType

Nếu độ dài lớn nhất được quy định, nhưng giá trị của thuộc tính không phải là độ dài cố định thì maxLength trong XSD simpleType nên được đặt bằng với độ dài lớn nhất của thuộc tính như đã quy định trong tập khóa. Nếu giá trị của thuộc tính là độ dài cố định thì thuộc tính XSD minLength

## TCVN 7981-3 : 2009

nên được đặt cùng giá trị. Nếu thuộc tính không được mã hóa là kiểu số và số thập phân được quy định trong tập khóa thì fracDig của kiểu đơn giản nên lấy giá trị được quy định trong tập khóa.

Đối với mỗi nhóm được đặt tên trong tập khóa, phần tử XSD toàn cầu được khai báo, lấy tên của nhóm. Thuộc tính kiểu XSD có giá trị được tạo bằng cách lấy tên của phần tử và thêm "Type" vào cuối thuộc tính đó. Nó có thuộc tính substitutionGroup tham chiếu phần tử Group (nhóm) khai báo trong tên miền "utility:".

XSD complexType được khai báo cho mỗi nhóm có tên được khai báo trong tập khóa, với tên được tạo bằng cách lấy tên của nhóm trong tập khóa và viết thêm "Type". Nó có phần tử XSD complexContent chứa các phần tử mở rộng XSD với giá trị thuộc tính cơ sở là utility:GroupType. Phần tử mở rộng chứa phần tử chuỗi XSD mà bao gồm phần tử XSD với một tham chiếu đến phần tử Series. Thuộc tính maxOccurs có giá trị "vô cùng". Phần tử có tên Annotations được bổ sung đến phần cuối của chuỗi, với kiểu AnnotationsType từ "common:" namespace. Nó cũng có giá trị minOccurs là "0".

Đối với mỗi thuộc tính trong tập khóa với attachmentLevel là "Group", phần tử thuộc tính XSD có thể được bổ sung đến phần tử mở rộng về bất kỳ nhóm cho trước. Để xác định nếu thuộc tính mức Group đã khai báo trong tập khóa được bổ sung đến kiểu XSD nhóm có tên cụ thể, quan sát các phần tử AttachmentGroup trong XML của tập khóa. Nếu phần tử nhóm trong lược đồ về tập khóa đang được khai báo xuất hiện trong phần tử AttachmentGroup (nhóm đính kèm) ở XML của tập khóa thì thuộc tính nên được bao gồm trong lược đồ Utility (tiện ích) đang được tạo. Nếu được bổ sung, thuộc tính này nên được khai báo như đã xác định cho mức DataSet (tập dữ liệu) ở trên. Thuộc tính Name (tên) là tên khái niệm của thuộc tính trong tập khóa.

Phần tử toàn cục XSD có tên Series(chuỗi) được khai báo trong tên miền của tập khóa, với kiểu SeriesType (kiểu chuỗi) và thuộc tính substitutionGroup tham chiếu utility:Series.

XSD complexType được khai báo với tên SeriesType (kiểu chuỗi). Nó có XSD complexContent, với phần tử mở rộng có giá trị thuộc tính cơ sở là utility:SeriesType. Phần tử mở rộng chứa phần tử chuỗi XSD, phần tử này chứa phần tử XSD đầu tiên mà giá trị ref của nó là "Key". Tiếp theo là phần tử XSD với thuộc tính ref có giá trị là "Obs". Thuộc tính maxOccurs của nó có giá trị "vô cùng". Phần tử có tên Annotations (ghi chú) được bổ sung đến phần cuối của chuỗi, với kiểu AnnotationsType(kiểu ghi chú) từ tên miền "common:". Nó cũng có giá trị minOccurs "0".

Đối với mỗi thuộc tính trong tập khóa với attachmentLevel là "Series (chuỗi)", phần tử thuộc tính XSD được bổ sung đến phần tử mở rộng, với các thuộc tính tên, sử dụng, kiểu được xác định cho mức DataSet(tập dữ liệu) ở trên.

Phần tử toàn cục XSD có tên là Key được khai báo. Kiểu phần tử này là KeyType (kiểu khóa) và thuộc tính substitutionGroup với giá trị utility:Key.

XSD complexType được khai báo với tên KeyType (kiểu khóa). Nó có phần tử XSD complexContent với phần tử mở rộng XSD bên trong, thuộc tính cơ sở của nó có giá trị "utility:KeyType". Phần tử mở rộng chứa chuỗi các phần tử XSD, một cho miền dữ liệu phi thời

gian trong tập khóa, theo thứ tự mà chúng xuất hiện trong XML của tập khóa. Các phần tử này có tên giống với các khái niệm của miền dữ liệu trong tập khóa mà chúng biểu diễn. Các thuộc tính kiểu là tên của simpleTypes được tạo cho các thuộc tính được mã hóa ở mức DataSet (tập dữ liệu) ở trên.

Phần tử toàn cục XSD được khai báo có tên "Obs". Phần tử này có thuộc tính substitutionGroup với giá trị "utility:Obs" và kiểu "ObsType".

Phần tử XSD complexType được khai báo với tên "ObsType" và XSD complexContent. Nó chứa phần tử mở rộng XSD với thuộc tính cơ sở "compact:ObsType" và chứa phần tử chuỗi XSD. Phần tử chuỗi chứa phần tử mà tên của nó là tên của khái niệm TimeDimension (miền dữ liệu thời gian) từ tập khóa, với kiểu common:TimePeriodType. Nó được theo sau bởi phần tử mà tên của phần tử đó là tên của PrimaryMeasure khai báo trong tập khóa, với kiểu XSD "double". Cuối cùng là phần tử có tên Annotations(ghi chú), với kiểu AnnotationsType (kiểu ghi chú) từ tên miền "common:". Nó có giá trị minOccurs là "0".

Đối với mỗi thuộc tính khai báo trong tập khóa với attachmentLevel là "Observation (quan sát)", thuộc tính XSD được bổ sung đến phần tử mở rộng. Mỗi thuộc tính XSD lấy tên khái niệm của thuộc tính được khai báo trong tập khóa và có thuộc tính sử dụng, tên và kiểu được tạo ra khi xác định các thuộc tính mức DataSet mô tả ở trên.

Không có khai báo hay kết cấu nào khác được bổ sung cho các lược đồ được tạo ra bằng cách sử dụng ánh xạ này.

Chú thích: Lược đồ của tập khóa không có bất kỳ cơ chế nào về việc biểu diễn dải thời gian qua tập các giá trị quan sát. Thông điệp có thể dùng được cho kiểu lược đồ này là thông điệp "cập nhật", thông điệp này chứa toàn bộ tập thuộc tính và các giá trị quan sát về chuỗi được truyền. Nếu được quy định thì không có khái niệm của thông điệp "xóa" và trường hành động nào trong thông điệp phần tử Header (tiêu đề) bị bỏ qua.

## 7 Phụ lục: Các thông điệp mẫu về SDMX - ML

Phụ lục này cung cấp các cách trình bày tập dữ liệu mẫu về SDMX-ML, giúp cho việc hiểu dễ dàng hơn. Đối với mỗi tập dữ liệu mẫu, một hoặc nhiều bảng được yêu cầu để chỉ ra cách dữ liệu được định dạng. Chú ý rằng tất cả dữ liệu đều không có thật và chỉ được sử dụng cho các mục đích biểu diễn. (Các con số không nhất quán và được tạo một cách ngẫu nhiên.)

### 7.1 CompactSample.xml (Tập mẫu về dữ liệu cô đọng)

**ID:** Message JD014 (Thông điệp kiểm tra không bị cắt bớt)

**Tên:** Trans46305

**Thời gian:** 2001-03-11T09:30:47-05:00

**Người gửi:** GB Smith from the BIS, +000.000.0000

**Đến:** B.S. Featherstone, Statistics Division, ECB, +000.000.0001

## TCVN 7981-3 : 2009

Thông điệp này chứa dữ liệu mới hoặc cập nhật, được tạo vào lúc T 2001-03-11T09:30:47-05:00.

**Khoản nợ bên ngoài, các kỳ hạn phải thanh toán, các khoản vay ngân hàng của Mexico được biểu diễn bằng cổ phần tính theo số triệu đô la mỹ, hàng tháng lúc bắt đầu chu kỳ. (dữ liệu tự do)**

Thời gian    Dữ liệu

2000-01 - 3.14

2001-02 - 2.29

2000-03 - 3.14

2000-04 - 5.24

2000-05 - 3.14

2000-06 - 3.78

2000-07 - 3.65

2000-08 - 2.37

2000-09 - 3.14

2000-10 - 3.17

2000-11 - 3.34

2000-12 - 1.21

**Khoản nợ bên ngoài, các kỳ hạn phải thanh toán, các khoản vay ngân hàng của Mexico được biểu diễn bằng cổ phần tính theo số triệu đô la mỹ, hàng năm lúc bắt đầu chu kỳ. (dữ liệu tự do)**

Thời gian            Dữ liệu

2000-01                3.14

**Khoản nợ bên ngoài, các kỳ hạn phải thanh toán, sự an toàn của khoản nợ được đưa ra bên ngoài Mexico, biểu diễn bằng cổ phần tính theo số triệu đô la mỹ, hàng tháng lúc bắt đầu chu kỳ. (dữ liệu tự do)**

Thời gian            Dữ liệu

2000-01                5.14

2001-02                3.29

2000-03                6.14

2000-04                2.24

2000-05                3.14

2000-06                7.78

2000-07                3.65

2000-08	5.37
2000-09	3.14
2000-10	1.17
2000-11	4.34
2000-12	1.21

**Khoản nợ bên ngoài, các kỳ hạn phải thanh toán, các khoản vay ngân hàng của Mexico được biểu diễn bằng cổ phần tính theo số triệu đô la mỹ, hàng năm lúc bắt đầu chu kỳ. (dữ liệu tự do)**

<u>Thời gian</u>	<u>Dữ liệu</u>
2000-1	4.14

## 7.2 UtilitySample.xml (Tập mẫu về dữ liệu tiện ích)

**ID:** Message JD01678594 (Thông điệp kiểm tra không bị cắt bớt)

**Tên:** Trans46304

**Thời gian:** 2001-03-11T09:30:47-05:00

**Người gửi:** GB Smith from the BIS, +000.000.0000

**Đến:** B.S. Featherstone, Statistics Division, ECB, +000.000.0001

Thông điệp này chứa dữ liệu mới hoặc cập nhật, được tạo vào lúc 2001-03-11T09:30:47-05:00.

**Khoản nợ bên ngoài, các kỳ hạn phải thanh toán, các khoản vay ngân hàng của Mexico được biểu diễn bằng cổ phần tính theo số triệu đô la mỹ, hàng tháng lúc bắt đầu chu kỳ. (dữ liệu tự do)**

<u>Thời gian</u>	<u>Dữ liệu</u>
2000-01	- 3.14
2000-02	- 3.19
2000-03	- 5.26
2000-04	- 5.12
2000-05	- 4.13
2000-06	- 3.12
2000-07	- 3.14
2000-08	- 3.79
2000-09	- 9.79
2000-10	- 3.14

## **TCVN 7981-3 : 2009**

2000-11 - 3.19

2000-12 - 3.14

### **7.3 GenericSample.xml (Tập mẫu về dữ liệu chung)**

**ID:** Message JD014 (Thông điệp kiểm tra không bị cắt bớt)

**Tên:** Trans46302

**Thời gian:** 2001-03-11T09:30:47-05:00

**Người gửi:** GB Smith from the BIS, +000.000.0000

**Đến:** B.S. Featherstone, Statistics Division, ECB, +000.000.0001

Thông điệp này chứa dữ liệu mới hoặc cập nhật, được tạo vào lúc 2001-03-11T09:30:47-05:00.

**Khoản nợ bên ngoài, các kỳ hạn phải thanh toán, các khoản vay ngân hàng của Mexico được biểu diễn bằng cổ phần tính theo số triệu đô la mỹ, hàng tháng lúc bắt đầu chu kỳ. (dữ liệu tự do)**

#### Thời gian Dữ liệu

2000-01 - 3.14

2001-02 - 3.14

2000-03 - 4.29

2000-04 - 6.04

2000-05 - 5.18

2000-06 - 5.07

2000-07 - 3.13

2000-08 - 1.17

2000-09 - 1.14

2000-10 - 3.04

2000-11 - 1.14

2000-12 - 3.24

### **7.4 CrossSectionalSample.xml (Tập mẫu về dữ liệu phần giao)**

**ID:** Message BIS947586 (Thông điệp kiểm tra không bị cắt bớt)

**Tên:** Trans46305

**Thời gian:** 2001-03-11T09:30:47-05:00

**Người gửi:** GB Smith from the BIS, +000.000.0000

**Đến:** B.S. Featherstone, Statistics Division, ECB, +000.000.0001



Thông điệp này chứa dữ liệu mới hoặc cập nhật, được tạo vào lúc 2001-03-11T09:30:47-05:00.

**Khoản nợ bên ngoài của Mexico tính theo số triệu đô la mỹ vào lúc đầu chu kỳ năm 2000 (dữ liệu tự do)**

<u>Tiêu đề</u>	<u>Cổ phần</u>	<u>Lưu lượng</u>
Tất cả các kỳ hạn phải thanh toán, khoản vay ngân hàng	3.14	1.00
Tất cả kỳ hạn phải thanh toán, sự an toàn của khoản nợ nước ngoài	6.39	2.27
Tất cả kỳ hạn phải thanh toán, các hợp đồng	2.34	-1.00
Tất cả kỳ hạn phải thanh toán, các thẻ tín dụng thương mại phi ngân hàng	3.19	- 1.06

---