

ALMA MATER STUDIORUM · UNIVERSITÀ DI BOLOGNA

FACOLTÀ DI SCIENZE MATEMATICHE, FISICHE E NATURALI
Corso di Laurea Triennale in Informatica

Strumenti per la generazione automatica di documentazione di profili d'uso di linguaggi standard basati su Core Components

Tesi di Laurea in Tecnologie WEB

Relatore:
Chiar.mo Prof.
Fabio Vitali

Presentata da:
Daniele D'Agnese

Correlatori:
Ing.
Piero De Sabbata

Dott.
Nicola Gessa

Dott.
Arianna Brutti

Sessione II
Anno Accademico 2010/2011

*Alla mia Famiglia,
con affetto*

Indice

Introduzione	1
Capitolo 1. Stato dell'arte	7
1.1 XS3P.....	8
1.2 <oXygen/> XML Editor.....	12
1.3 DocFlex/XML.....	17
1.4 Tabella comparativa degli applicativi.....	22
1.5 Documentazione UBL.....	23
1.6 Documentazione MODA-ML.....	25
1.7 Documentazione eBiz-TCF.....	27
1.8 Considerazioni finali.....	29
Capitolo 2. Progettazione	31
2.1 Requisiti dell'applicativo.....	31
2.1.1 Documenti XSD.....	32
<i>Core Components</i>	32
2.1.2 Albero dell'istanza XML.....	33
2.1.3 Semplificazione dei rami dell'albero.....	33
2.1.4 Liste di codici.....	34
<i>Generic Code</i>	34
2.1.5 Business rules.....	35
<i>Schematron</i>	35
2.1.6 Parametrizzazione della documentazione.....	36
2.1.7 Personalizzazione della documentazione.....	36
2.1.8 Output HTML e PDF.....	36
2.1.9 Output statistiche.....	36
2.1.10 Modalità d'esecuzione.....	37
2.2 Funzionalità dell'applicativo.....	37
2.2.1 Supporto documenti XSD	38
2.2.2 Rappresentazione dell'albero dell'istanza XML.....	39
2.2.3 Semplificazione rami dell'albero dell'istanza.....	40
2.2.4 Gestione Code List.....	41

2.2.5	Supporto Business rules.....	42
2.2.6	Parametrizzazione della documentazione.....	43
2.2.7	Personalizzazione della documentazione.....	43
2.2.8	Output guida per profilo d'uso.....	44
2.2.9	Output statistiche.....	45
2.2.10	Linea di comando e modalità batch.....	45
2.3	Architettura XDG.....	46
2.3.1	Ciclo d'esecuzione.....	46
2.3.2	Analisi dei componenti.....	48
	<i>Componente XML Schema analyzer</i>	48
	<i>Componente xml2html</i>	49
	<i>Componente PDF&stats</i>	51
	<i>Componente Schematron handler</i>	53
2.3.3	Input.....	55
	<i>File configurazione TASK</i>	56
	<i>File configurazione output HTML</i>	58
	<i>File configurazione output PDF</i>	60
	<i>Template DocBook</i>	62
	<i>Configurazione TRIM</i>	65
2.3.4	Output.....	66
	<i>Output XML</i>	66
	<i>Output HTML</i>	67
	<i>Output PDF</i>	67
	<i>Output Statistiche</i>	68
2.4	Considerazioni finali.....	70
Capitolo 3. Implementazione.....		71
3.1	Implementazione applicativo XDG.....	71
3.1.1	Classe principale	71
	<i>Task Handler</i>	72
	<i>XML-Schema Analyzer</i>	73
	<i>Schematron Handler</i>	76
	<i>XML2HTML</i>	79
	<i>PDF&stats</i>	80
	<i>Utility</i>	87
3.1.2	Classi ausiliarie.....	87
3.1.3	Fogli di stile XSLT.....	88
	<i>xml2html.xsl</i>	88
	<i>skeleton.xsl</i>	91
	<i>ExtractSchFromXSD-2.xsl</i>	92
3.1.4	Configurazioni di default.....	93
	<i>Configurazione HTML</i>	93
	<i>Configurazione PDF</i>	93

	<i>Template docbook</i>	93
3.1.5	Documenti di validazione.....	94
3.2	Analisi output prodotti.....	95
3.2.1	XML.....	95
	<i>Struttura generale documento XML</i>	95
	<i>Building block dell'albero dell'istanza XML</i>	96
	<i>Attraversamento dell'albero dell'istanza XML</i>	97
3.2.2	HTML & PDF.....	99
	<i>Copertina</i>	99
	<i>Indice</i>	100
	<i>Capitoli</i>	100
	<i>Tabella del profilo d'uso</i>	101
	<i>Tabella business rules</i>	105
	<i>Bibliografia</i>	106
3.2.3	STATISTICHE.....	107
3.3	Considerazioni finali.....	108
	Conclusioni	109
	Appendice A – Invoice (PEPPOL)	113
	Bibliografia	141

Indice delle figure

Figura 1: Es. di documentazione generata da XS3P.....	9
Figura 2: Es. istanza XML generata da XS3P.....	9
Figura 3: Es. rappresentazione XML-Schema generata da XS3P.....	10
Figura 4: Es. codelist generato da XS3P.....	10
Figura 5: Es. di documentazione generata da <oXygen/>.....	13
Figura 6: Es. di diagramma generato da <oXygen/>.....	13
Figura 7: Es. d' istanza generata da <oXygen/>.....	14
Figura 8: Es. rappresentazione XML-Schema generato da <oXygen/>.....	14
Figura 9: Es. di codelist generato da oXygen.....	15
Figura 10: Es. di parametrizzazione della documentazione in oXygen.....	15
Figura 11: Es. documentazione basata su FramedDoc template in DocFlex/XML.....	18
Figura 12: Es. di codelist generato da DocFlex/XML.....	19
Figura 13: Es. di parametrizzazione documentazione in DocFlex/XML.....	19
Figura 14: Es. di personalizzazione della documentazione in DocFlex/XML.	20
Figura 15: DocFlex/XML Template Designer.....	21
Figura 16: Es. documentazione UBL (Spreadsheet Model).....	24
Figura 17: Es. documentazione MODA-ML (Struttura del documento XML).	26
Figura 18: Es. documentazione MODA-ML (Descrizione elementi).....	27
Figura 19: Es. documentazione eBiz-TCF.....	28
Figura 20: Diagramma UML dei componenti XML-Schema gestiti.....	38
Figura 21: Ciclo d'esecuzione dell'applicativo XDG.....	46
Figura 22: Diagramma componente XML schema analyzer.....	48
Figura 23: Pipeline XML Schema analyzer.....	49
Figura 24: Diagramma componente xml2html.....	50
Figura 25: Diagramma componente PDF&stats.....	51
Figura 26: Diagramma pipeline Schematron.....	53
Figura 27: Struttura del treegrid.....	88
Figura 28: Sostituzione parametri nello XSLT skeletron.....	92
Figura 29: Template docbook di default.....	94
Figura 30: Struttura generale documento XML.....	95
Figura 31: Es. copertina.....	99
Figura 32: Es. indice.....	100
Figura 33: Es. capitolo.....	100
Figura 34: Es. tabella del profilo d'uso.....	101
Figura 35: Es. compositor.....	102
Figura 36: Es. documentation info.....	102

Figura 37: Es. codelist.....	103
Figura 38: Es. facets,constraint, advanced e wilcard info.....	104
Figura 39: Es. derivation info.....	104
Figura 40: Es. additional constraint info.....	104
Figura 41: Tabella business rules con fasi e regole extra.....	105

Indice delle tabelle

Tabella 1: Tabella comparativa degli applicativi esaminati.....	22
Tabella 2: Tabella test documenti XSD UBL-2.0.....	111

Introduzione

Lo scopo di questa tesi è quello di fornire un tool che consenta, in modo automatico, di generare la documentazione *human-readable* di formati XML standard basati sui Core Components di UN/CEFACT e OASIS, strutturandola in guide per profili d'uso che siano corrette, pratiche e corredate da una veste grafica professionale affinché anche utenti, non particolarmente esperti, possano facilmente apprendere il contenuto.

Le transazioni di e-Business ed e-Commerce si basano spesso sullo scambio di documenti digitali, come ad esempio ordini, cataloghi, fatture, i cui formati vengono definiti da organizzazioni di standardizzazione. Tali formati standard sono per natura molto generici in quanto cercano di coprire un ampio spettro di casi possibili. Questa forte caratteristica di generalità fa sorgere la necessità, per i diversi utilizzatori dei documenti, di personalizzarli in funzione delle proprie specifiche esigenze, che spesso non sono risolte in maniera soddisfacente dai documenti standard così come sono.

A tale fine, nel processo di personalizzazione, vengono definiti sottoinsiemi del formato scegliendo solo le parti veramente necessarie oppure ne viene raffinata la semantica ridefinendo ad esempio alcuni elementi del vocabolario XML. Questi formati personalizzati consentono sia di definire la struttura dei documenti digitali da scambiare sia di validarli.

Di frequente tale processo di personalizzazione comporta anche l'utilizzo di ulteriori linguaggi di validazione, come Schematron, per esprimere ad

esempio le *Business rules*, ovvero vincoli aggiuntivi da applicare in determinati contesti specifici di business, e le *liste di codici* ossia gli elenchi di valori consentiti per valute, pesi, misure ecc. fondamentali per una vera interoperabilità.

L'architettura Core Components definisce un insieme di *building block*, ovvero componenti base i più generici possibili svincolati da ogni contesto, in modo da poter essere utilizzato in diversi settori di business per realizzare formati di scambio dati.

I componenti generici, per essere utilizzati, devono essere contestualizzati in un ambito specifico di business. Questo avviene ad esempio nel linguaggio standard UBL (Universal Business Language) il quale definisce una serie di documenti commerciali, in formato elettronico, basati appunto sull'aggregazione di elementi Core Components.

Tali formati vengono definiti attraverso il linguaggio XML-SCHEMA (d'ora in avanti semplicemente XSD), ma risultano ancora molto generici per essere utilizzati in contesti reali; pertanto è necessario un ulteriore passaggio che consiste nel creare un profilo d'uso.

Il profilo d'uso quindi rappresenta la versione finale del formato, libera da tutte le parti non necessarie e corredata da una serie di informazioni di business aggiuntive (business rules) specifiche del contesto.

Per favorire la comprensione e l'adozione dei profili d'uso si rende poi necessaria la stesura di linee guida per profili d'uso (d'ora in avanti semplicemente guide).

Quindi il problema principale che questo elaborato di tesi tenta di risolvere è quello di produrre, in modo automatico, guide capaci di integrare armoniosamente informazioni di diverso tipo, quali appunto la struttura e le annotazioni dei documenti XSD, le business rules, le liste di codici ed

eventuale contenuto esterno, come ad esempio pagine di presentazione, testo libero, immagini e bibliografia, in modo da rendere la guida il più possibile completa, fruibile e rispondente al linguaggio da descrivere.

Dal punto di vista di chi produce le guide, l'assenza di un tool di questo genere comporta la necessità di creare ed integrare manualmente la documentazione con un notevole sforzo. Tale operazione manuale può risultare sia costosa in termini di tempo (una guida di medie dimensioni può richiedere parecchie ore) sia rischiosa in quanto si possono introdurre errori: ad esempio le informazioni della guida potrebbero non essere coerenti con quelle definite nei documenti XSD, anche per semplici errori di battitura.

L'utilizzo di un tale applicativo semplifica notevolmente situazioni dove sono presenti uno o più documenti XSD molto complessi, ma in particolar modo semplifica quelle dove è importante creare restrizioni dello standard, generando nuovi documenti personalizzati e semplificati rispetto a quello principale.

Dal punto di vista dell'utilizzatore delle guide, lo studio di nuovi documenti da adottare, avendo a disposizione informazioni troppo o troppo poco dettagliate, mal strutturate o dispersive, ne rallenta l'apprendimento, ne ostacola l'eventuale adozione e può favorire una errata comprensione e quindi l'insorgere di errori.

Ad esempio il formato eBiz-TCF-ArticleCatalogueValidator, appartenente alla specifica eBiz-TCF e basato sul formato Catalogue di UBL, è stato personalizzato sia riducendo notevolmente la struttura (da circa 40.000 elementi totali nel documento UBL di partenza a 178 nel profilo d'uso eBiz) sia inserendo le business rules (78 regole Schematron); tutto ciò ha semplificato di molto la comprensibilità del formato stesso.

Per cercare di capire come queste guide debbano essere composte, in

mancanza di un vero e proprio riferimento formale sulla strutturazione e generazione di documentazione, ci si è concentrati inizialmente sull'analisi di tool commerciali ed opensource *XML-Schema documentation generator* attualmente disponibili quali xs3p, <Oxygen/> XML Editor e DocFlex/XML.

Dall'analisi effettuata è emerso che le guide generate da questi applicativi, seppur di ottima fattura, sono troppo particolareggiate, frammentate, poco personalizzabili e pertanto poco fruibili: ne consegue che tali guide risultano essere più adeguate per gli sviluppatori che per gli utilizzatori.

In un secondo momento sono state prese in esame le documentazioni reperibili su alcuni formati basati sui Core Components, come ad esempio MODA-ML ed eBiz-TCF. Analizzando tali documentazioni si è notato che un approccio più conveniente poteva essere quello di presentare esempi e frammenti di documenti XML piuttosto che un dizionario sistematico di vocaboli e delle caratteristiche del loro uso: questa modalità consente di farsi un'idea generale di come è strutturato il documento XML velocemente ed in modo agevole, anche senza particolari conoscenze informatiche.

Di seguito si è proceduto con l'individuare altri aspetti per rendere la guida ancora più fruibile, in particolare è stata migliorata ed unificata la gestione delle liste di codici sia interne al documento XSD sia espresse mediante un altro linguaggio.

Le guide, generate con queste caratteristiche, potevano essere parzialmente soddisfacenti; tuttavia era necessario, per chi doveva generarle, avere a disposizione dei metodi più flessibili per poter adattare il contenuto delle guide a contesti d'uso differenti e con scopi diversi.

Per risolvere questo problema, prendendo spunto dalle soluzioni offerte dall'applicativo DocFlex/XML ed estendendole, si è scelto di dare la possibilità di:

- parametrizzare la documentazione da generare, scegliendo ad esempio con una granularità più fine, quali informazioni riportare
- personalizzare la documentazione attraverso l'aggiunta di testo esterno al documento XSD, per spiegare ad esempio parti più complesse della guida o per aggiungere informazioni non contenute negli XSD
- selezionare alcuni rami dell'albero dell'istanza XML da non includere nella guida, allo scopo di renderla più concisa e quindi meno dispersiva, caratteristica questa utile in particolar modo in guide in formato cartaceo

Inoltre, a differenza degli applicativi esaminati, si è scelto di arricchire la guida consentendo di riportare informazioni aggiuntive, quali i vincoli aggiuntivi espressi tramite il linguaggio Schematron e le liste di codici. Questi vincoli, qualora presenti, devono essere documentati per consentire una maggior comprensione su come l'istanza XML debba essere composta per poter risultare corretta.

Dopo aver individuato le caratteristiche principali delle guide per profili d'uso, si è quindi progettato e sviluppato un applicativo, denominato XDG (XML-Schema Documentation Generator), che fosse in grado di gestire gli aspetti sopra menzionati. Tale applicativo è stato implementato nell'ambiente di sviluppo Eclipse, utilizzando sia codice JAVA, con l'ausilio di librerie quali EMF per la gestione dei documenti XSD e iText per la generazione dell'output PDF, sia fogli di stile XSLT per la generazione dell'output HTML.

L'architettura dell'applicativo XDG è stata opportunamente divisa in quattro componenti ben distinti, ognuno con uno scopo preciso:

- XML-Schema Analyzer – *gestisce documenti XSD*
- Schematron Handler – *gestisce documenti Schematron*

- XML2HTML – *genera output HTML*
- PDF&stats – *genera output PDF*

L'applicativo XDG genera le guide per profili d'uso in due formati: HTML e PDF; questi, anche se simili, presentano differenze: la guida nel formato HTML risulta l'unica possibile per documenti molto estesi, invece quella in formato PDF si rivela idonea per documenti di piccole dimensioni e per la stampa.

Dopo la fase d'implementazione, l'applicativo XDG è stato sottoposto ad una fase di test per valutare sia la compatibilità delle guide generate sui principali web browser e pdf reader sia la correttezza e la concreta fruibilità delle guide generate per documenti appartenenti alla specifica UBL-2.0 ed infine per valutare la generazione di guide su customizzazioni di schemi UBL utilizzati in ambiti reali, nello specifico quello usato nel progetto PEPPOL.

Dai test effettuati è emerso che l'applicativo XDG è in grado di generare guide per profili d'uso corrette, complete e graficamente gradevoli, tanto che l'Intercent-ER (Agenzia regionale di sviluppo dei mercati telematici) ha scelto di usarlo per generare la documentazione di supporto all'implementazione delle specifiche PEPPOL.

Si è scelto di focalizzare l'attenzione sul “motore” dell'applicativo XDG invece che sull'interfaccia grafica, cercando però allo stesso tempo di adottare soluzioni adeguate in modo da favorirne un'implementazione futura; quindi possibili sviluppi futuri possono interessare l'implementazione di comode interfacce grafiche per consentire sia di configurare tutti gli aspetti possibili supportati dall'applicativo, come ad esempio l'impostazione dei documenti XSD da processare, le opzioni di configurazione per gli output HTML e PDF, la configurazione del contenuto esterno allo XSD, sia di generare report statistici con dati prodotti dall'applicativo stesso.

Capitolo 1. Stato dell'arte

In questo capitolo saranno descritti i principali applicativi attualmente disponibili che rappresentano lo stato dell'arte della generazione automatica della documentazione per documenti XML-Schema Definition [WST11]: xs3p, <oxygen/> XML Editor e DocFlex/XML. Per ogni applicativo verranno analizzate le seguenti caratteristiche:

- la licenza offerta
- le tecniche usate per generare la documentazione
- la modalità d'uso
- il tipo di visualizzazione offerta
- l'output generato
- la possibilità di parametrizzare la documentazione
- la possibilità di personalizzare la documentazione
- la capacità di gestire documenti XSD molto complessi e di grandi dimensioni
- la possibilità di gestire *codelist* (elenco di valori ammissibili per alcuni elementi come ad esempio valute, unità di misura e codici identificativi delle agenzie ecc.) definiti sia nel documento XSD stesso mediante i tag “xsd:enumeration” e sia tramite i *Generic Code* (formato standard XML per definire vocabolari controllati)
- l'usabilità effettiva della documentazione generata

Dopo l'analisi degli applicativi verrà proposta una tabella riepilogativa che riassume i vantaggi e gli svantaggi riscontrati nei tool presi in esame. Infine sarà esaminata la documentazione disponibile di tre specifiche standard basate sui Core Component che sono UBL, Moda-ML e eBiz-TCF.

1.1 XS3P

L'applicativo `xs3p` [XS11], distribuito sotto licenza *Mozilla Public License* (MPL), è uno *schema documentation generator* che prende in input un documento XSD Schema e genera un documento XHTML. `xs3p` è costituito essenzialmente da un unico foglio di stile XSLT e può essere eseguito da linea di comando utilizzando un *XSLT processor*, come XALAN, Saxon, MSXML e `xsltproc` oppure può essere integrato in altri programmi come oXygen XML Editor e Stylus Studio.

Il documento XHTML generato in output è organizzato nel seguente modo:

- indice (Table of Contents)
- descrizione dei tipi e degli elementi XML-Schema presenti nel documento d'input
- legenda
- glossario della terminologia XML-Schema

Per tutti i tipi e gli elementi presenti nel documento XSD d'input, vengono riportate le informazioni generali (il nome e lo schema d'appartenenza), le informazioni sulla derivazione (per estensione o restrizione), la rappresentazione dell'istanza XML e infine la definizione XML-Schema relativa al componente (Fig.1).

Complex Type: **person_Type**

Super-types:	xsd:string < sub_person_Type (by restriction) < person_Type (by extension)
Sub-types:	None

Name	person_Type
Abstract	no

XML Instance Representation

```

<...
  email="email_Type [0..1]">
    sub_person_Type
  </...>

```

Schema Component Representation

```

<xsd:complexType name="person_Type">
  <xsd:simpleContent>
    <xsd:extension base="sub_person_Type">
      <xsd:attribute name="email" type="email_Type" use="optional"/>
    </xsd:extension>
  </xsd:simpleContent>
</xsd:complexType>

```

Figura 1: Es. di documentazione generata da XS3P.

Nella rappresentazione dell'istanza XML (Fig.2) sono presenti informazioni sulla cardinalità, ovvero il numero massimo e minimo d'occorrenze che un elemento o attributo può presentare (es. [0..1]), sul *Model Group* come ad esempio “xsd:choice” (es. Start Choice ... End Choice), sul *Constraint* dell'elemento o attributo (es. in verde country=”Australia”), sulle *Facets* come “xsd:pattern” o “xsd:enumeration” (es. <<pattern = [1-9][0-9]{3}>>) ed infine sulla documentazione contenuta nel tag “xsd:documentation” visualizzabile in una finestra popup cliccando sul simbolo “?”.

XML Instance Representation

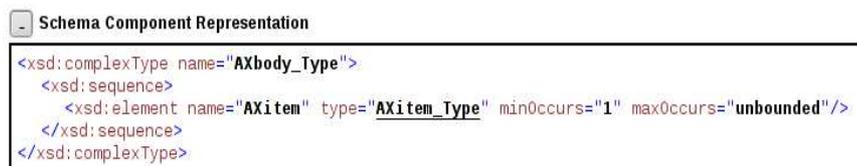
```

<... country="Australia">
  <unitNo> string </unitNo> [0..1]
  <houseNo> string </houseNo> [1]
  <street> string </street> [1]
  Start Choice[1]
    <city> string </city> [1]
    <town> string </town> [1]
  End Choice
  <state> AusStates </state> [1]
  <postcode> string <<pattern = [1-9][0-9]{3}>> </postcode> [1] ?
</...>

```

Figura 2: Es. istanza XML generata da XS3P.

Nella rappresentazione XML-Schema viene visualizzato il codice XML utilizzato per definire il componente descritto, escluso il tag “xsd:annotation” (Fig.3).



```

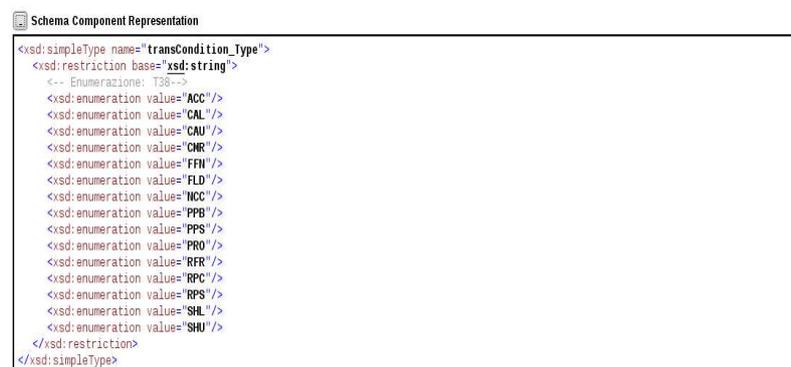
Schema Component Representation
<xsd:complexType name="AXbody_Type">
  <xsd:sequence>
    <xsd:element name="AXitem" type="AXitem_Type" minOccurs="1" maxOccurs="unbounded"/>
  </xsd:sequence>
</xsd:complexType>

```

Figura 3: Es. rappresentazione XML-Schema generata da XS3P.

xs3p supporta i *codelist* definiti nel documento XSD tramite il tag “xsd:enumeration” (Fig.4), ma non quelli espressi mediante i *Generic Code*.

Name	transCondition_Type
Content	<ul style="list-style-type: none"> • Base XSD Type: string • value comes from list: {ACC CAL CAU CMR FFN FLD NCC PPB PPS PRO RFR RPC RPS SHL SHU}
Documentation	transport condition - special condition of transport or special service requested to the Forwarder



```

Schema Component Representation
<xsd:simpleType name="transCondition_Type">
  <xsd:restriction base="xsd:string">
    <!-- Enumerazione: T38-->
    <xsd:enumeration value="ACC"/>
    <xsd:enumeration value="CAL"/>
    <xsd:enumeration value="CAU"/>
    <xsd:enumeration value="CMR"/>
    <xsd:enumeration value="FFN"/>
    <xsd:enumeration value="FLD"/>
    <xsd:enumeration value="NCC"/>
    <xsd:enumeration value="PPB"/>
    <xsd:enumeration value="PPS"/>
    <xsd:enumeration value="PRO"/>
    <xsd:enumeration value="RFR"/>
    <xsd:enumeration value="RPC"/>
    <xsd:enumeration value="RPS"/>
    <xsd:enumeration value="SHL"/>
    <xsd:enumeration value="SHU"/>
  </xsd:restriction>
</xsd:simpleType>

```

Figura 4: Es. codelist generato da XS3P.

La rappresentazione dei *codelist* risulta essere sia poco fruibile specialmente in presenza di un elevato numero di valori possibili e di altre *facet* diverse da “xsd:enumeration”, sia ridondante nel caso in cui lo stesso *codelist* venga utilizzato in parti diverse dello schema.

Per quanto riguarda la parametrizzazione e la personalizzazione della documentazione da generare, xs3p permette unicamente di impostare alcuni parametri da passare al foglio di stile XSLT.

I parametri che possono essere definiti sono i seguenti:

- *title* – definisce il titolo del documento
- *sortByComponent* – esprime l'ordine con cui visualizzare i componenti
- *useJavaScript* – abilita l'uso di javascript
- *printAllSuperTypes* – mostra la definizione di tutta la gerarchia oppure solo del padre
- *printAllSubTypes* – mostra la definizione di tutti i sottotipi
- *printLegend* – include la legenda
- *printGlossary* – stampa il glossario
- *printNSPrefixes* – mostra i prefissi dei namespace
- *searchIncludedSchemas* – segue le dichiarazioni “xsd:include”
- *searchImportedSchemas* – segue le dichiarazioni “xsd:import”.

Purtroppo non vi è modo di personalizzare la documentazione inserendo nella guida contenuti esterni allo schema, se non modificando manualmente il foglio di stile XSLT: operazione che può risultare molto complessa.

xs3p riesce a gestire documenti XSD molto grandi e definiti su più librerie tramite le direttive “xsd:include” e “xsd:import”, ma purtroppo non è in grado di supportare la direttiva “xsd:redefine”.

In generale xs3p risulta essere un buon tool d'immediato utilizzo ed abbastanza completo; ma alcuni fattori, come l'assenza dell'albero dell'istanza XML di tutto il documento, la non gestione del tag “xsd:redefine”, l'impossibilità di personalizzare la documentazione e l'XHTML come unica modalità di output, sono fortemente limitativi nella generazione di guide per profili d'uso.

1.2 <oXygen/> XML Editor

<oXygen/> XML Editor [OX11] è un IDE (*Integrated Desktop Environment*) proprietario, scritto in Java, molto completo e con tante funzionalità per la creazione, manipolazione, validazione e documentazione di documenti XML. Tra la nuove *feature* presenti nelle ultime versioni di <oXygen/> analizzeremo l'*XML Schema Documentation* che, con un click del mouse su una voce del menu presente nell'interfaccia grafica oppure da linea di comando, consente appunto di generare documentazione a partire da documenti XSD in vari formati come HTML, PDF, DocBook e in un formato personalizzato specificando un foglio di stile XSLT da usare per la trasformazione.

Il documento generato in output nei formati HTML e PDF è suddiviso nel modo seguente:

- indice (Table of Contents)
- descrizione di tutti i componenti XML-Schema (elementi, tipi, attributi, gruppi, gruppi di attributi) presenti nel documento d'input raggruppati secondo il *namespace* d'appartenenza

Per ogni componente XML-Schema presente nello XSD vengono riportati le informazioni generali, come *namespace*, *SchemaLocation*, *Type* e *Content Model*, le annotazioni, il diagramma dello schema, l'istanza XML e la definizione XML-Schema relativa al componente (Fig.5).

Element ACSDesRequest_Type / shipInfo

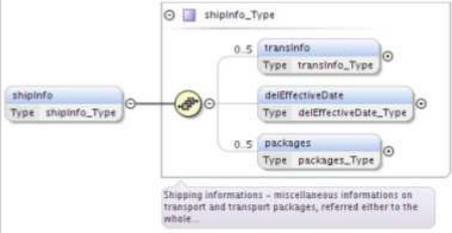
Namespace	No namespace						
Diagram							
Type	shipInfo_Type						
Properties	<table border="1"> <tr> <td>content:</td> <td>complex</td> </tr> <tr> <td>minOccurs:</td> <td>0</td> </tr> <tr> <td>maxOccurs:</td> <td>1</td> </tr> </table>	content:	complex	minOccurs:	0	maxOccurs:	1
content:	complex						
minOccurs:	0						
maxOccurs:	1						
Model	transInfo{0,5} , delEffectiveDate{0,1} , packages{0,5}						
Children	delEffectiveDate, packages, transInfo						
Instance	<pre><shipInfo> <transInfo transReason="" {0,5}</transInfo> <delEffectiveDate dateForm="" {0,1}</delEffectiveDate> <packages packageLevel="" {0,5}</packages> </shipInfo></pre>						
Source	<xsd:element name="shipInfo" type="shipInfo_Type" minOccurs="0" maxOccurs="1"/>						
Schema location	file [redacted]moda-ml/ACSDesRequest.xsd						

Figura 5: Es. di documentazione generata da <oXygen/>.

Il diagramma rappresenta in modo fedele ogni caratteristica XML-Schema relativa al componente ed ai figli di primo livello, come il tipo, la cardinalità, il Content Model, la derivazione ed anche le annotazioni presenti (Fig.6).

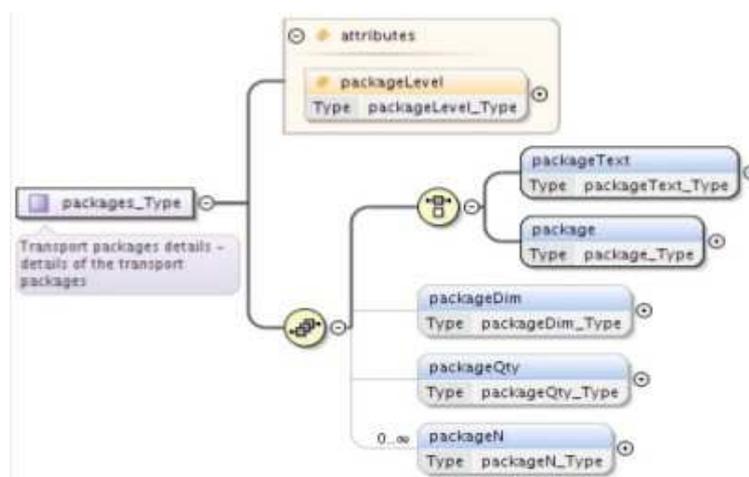


Figura 6: Es. di diagramma generato da <oXygen/>.

La rappresentazione dell'istanza del documento XML (Fig.7) è limitata all'elemento corrente e fornisce informazioni sugli elementi, con le relative cardinalità indicate tra parentesi graffe (es. {1,1}), e sugli attributi.

Instance	<pre> <packages packageLevel="" <packageText>{1,1}</packageText> <package>{1,1}</package> <packageDim>{0,1}</packageDim> <packageQty>{0,1}</packageQty> <packageN numberingOrg="">{0,unbounded}</packageN> </packages> </pre>
----------	--

Figura 7: Es. d' istanza generata da <oXygen/>.

E' da notare però che, a differenza dell'applicativo xs3p, non sono indicate le informazioni sui *Model Group* come “xsd:choice”: infatti nell'esempio (Fig.7) gli elementi “packageText” e “package” nello XSD sono definiti in una “xsd:choice”, ma ciò non si evince dall'istanza XML.

Nella rappresentazione XML-Schema viene visualizzato il codice XML utilizzato per definire il componente descritto così com'è espresso nel documento XSD (Fig.8).

Source	<pre> <xsd:complexType name="shipInfo_Type"> <xsd:annotation> <xsd:documentation>Shipping informations · miscellaneous informations on transport and transport packages, referred either to the whole document or to the single item</ xsd:documentation> </xsd:annotation> <xsd:sequence> <xsd:element name="transInfo" type="transInfo_Type" minOccurs="0" maxOccurs="5"/> <xsd:element name="delEffectiveDate" type="delEffectiveDate_Type" minOccurs="0" maxOccurs="1"/> <xsd:element name="packages" type="packages_Type" minOccurs="0" maxOccurs="5"/> </xsd:sequence> </xsd:complexType> </pre>
--------	--

Figura 8: Es. rappresentazione XML-Schema generato da <oXygen/>.

<oXygen/>, così come xs3p, supporta i *codelist* definiti nel documento XSD tramite il tag “xsd:enumeration” (Fig.9), ma non quelli espressi mediante i *Generic Code*.

Facets	enumeration	ACC
	enumeration	CAL
	enumeration	CAU
	enumeration	CMR
	enumeration	FFN
	enumeration	FLD
	enumeration	NCC
	enumeration	PPB
	enumeration	PPS
	enumeration	PRO
	enumeration	RFR
	enumeration	RPC
	enumeration	RPS
	enumeration	SHL
	enumeration	SHU

Figura 9: Es. di codelist generato da oXygen.

Anche qui la rappresentazione dei *codelist* risulta essere sia poco fruibile specialmente in presenza di un elevato numero di valori possibili e di altre *facet* diverse da “xsd:enumeration”, sia ridondante nel caso dello stesso *codelist* ripetuto in più parti dello schema.

Per quanto riguarda la parametrizzazione della documentazione da generare, <oXygen/> permette di selezionare tutte le caratteristiche dello XML-Schema (Fig.10); in particolare mediante la checkbox “Espace XML content” (evidenziato in verde) è possibile includere o escludere in blocco (tutti o nessuno) gli elementi appartenenti ad altri *namespace*, ovvero diversi dal vocabolario XML-Schema, che si possono trovare annidati all'interno dei tag “xsd:annotation”.

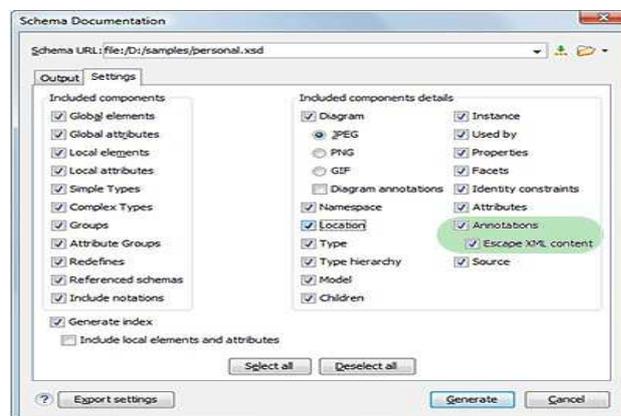


Figura 10: Es. di parametrizzazione della documentazione in oXygen.

Però è da notare che non è possibile specificare, con una granularità più fine, quali tag, appartenenti ad altri namespace, includere o meno. Tale limite comporta un problema qualora si voglia realizzare una guida relativa a schemi basati su Core Components. Infatti all'interno del tag “xsd:documentation” possiamo trovare molti altri tag appartenenti al vocabolario *Core Component Technical Specification*: in tal caso non vi è possibilità di filtrare per esempio elementi che specificano informazioni più avanzate oppure altri come “ccts:Example” e “ccts:Definition” che forniscono informazioni meno tecniche ma più esplicative.

Anche in <oXygen/>, così come in xs3p, di base non è consentito personalizzare la documentazione con del contenuto esterno allo XSD; l'unico modo possibile è quello di scrivere un foglio di stile XSLT personalizzato in grado d'integrare la documentazione derivante dallo schema con altri contenuti esterni organizzati in documenti XML.

Per quanto riguarda la gestione di schemi complessi e organizzati su più librerie, oXygen non è in grado di produrre l'output PDF in quanto termina con il seguente messaggio di errore “Exception in thread main java.lang.OutOfMemoryError: Java heap space Specify more memory for the FOP as explained in the section 'Performance Problems', subsection 'External Processes' in the User Manual” [AF11]. Questo tipo di errore è dovuto al fatto che il processore XSLT-FO Apache FOP, utilizzato per generare il pdf, esaurisce la memoria disponibile poiché mantiene tutto il contenuto da renderizzare in memoria. Come soluzione è possibile o provare ad aumentare la memoria disponibile da utilizzare, come suggerito nel messaggio d'errore, oppure generare la documentazione suddividendola in più file per ciascun componente da documentare ottenendo in questo modo però una guida frammentata su più pagine.

In definitiva <oXygen/> è un ottimo tool per generare documentazione tecnica per schemi medio/piccoli; l'unica considerazione negativa è che la documentazione generata risulta essere troppo frammentata facendo perdere di vista la struttura d'insieme.

1.3 DocFlex/XML

DocFlex/XML [DF11] è un applicativo Java rilasciato sotto varie licenze sia freeware che commerciali; esso consente, a partire da qualsiasi documento XML, di modellare tramite l'uso di *template* (in stile XSLT) la documentazione in output da generare nei formati HTML, RTF, TXT.

L'applicativo DocFlex/XML può essere eseguito sia da linea di comando, sia tramite una finestra di dialogo la quale permette facilmente di selezionare il documento in input, il tipo di *template* da utilizzare e il percorso della documentazione da generare. In aggiunta può essere integrato in altri applicativi come XML Spy.

La struttura della documentazione da generare è fortemente condizionata dal template scelto: di base DocFlex/XML fornisce due template che sono *FramedDoc.tpl* e il *PlainDoc.tpl*.

Nel primo caso, il FramedDoc template, la documentazione generata ricalca molto lo stile *Javadoc* (Fig.11) con riquadri contenenti informazioni generali sugli schemi, l'elenco dei componenti processati e la descrizione del componente stesso. Nel secondo caso, il PlainDoc template, la documentazione generata presenta le informazioni dei componenti in modo sequenziale.

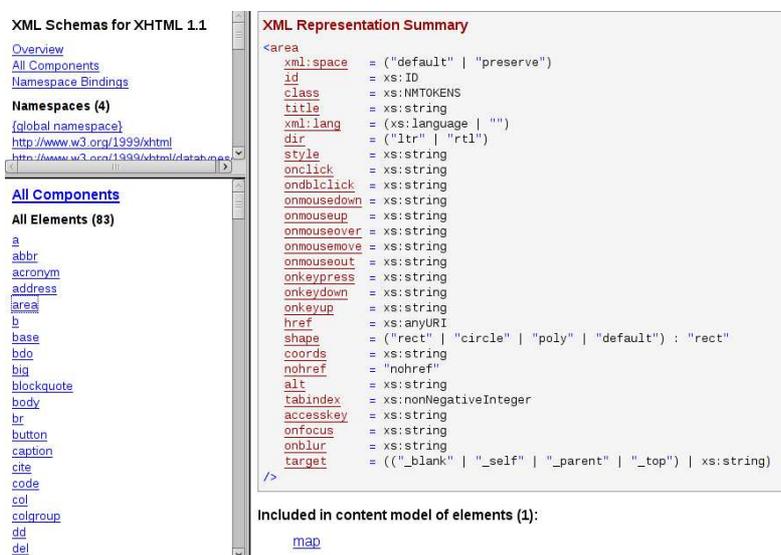


Figura 11: Es. documentazione basata su FramedDoc template in DocFlex/XML.

Le informazioni dei componenti, nella documentazione generata da entrambi i template, sono divise nel seguente modo:

- Overview Info
- Content Model Diagram
- XML Rappresentation
- Attribute Detail

Nelle Overview Info sono presenti informazioni sul *namespace*, *type* e *content* sul componente in esame; nel Content Model Diagram troviamo l'immagine del diagramma relativo al componente; nello XML Rappresentation è riportato il codice XML così com'è definito nello XSD; infine nello Attribute Detail vengono descritti i dettagli sugli attributi presenti.

Le informazioni presenti sono chiare e complete; purtroppo però bisogna evidenziare che non vi è riportata l'istanza del documento XML.

element <Quantity>

Namespace: [urn:opalis:names:specification:ubli:schema:xsd:CommonBasicComponent-2](#)

Type: QuantityType

Content: simple, 1 attribute

Defined: globally in [URN_CommonBasicComponent-2.0.xsd](#); see [XML source](#)

Used: at 4 locations

XML Representation Summary

```
<quantity
  unitCode = {"04" | "05" | "08" | "10" | "11" | "13" | "14" | "15" | "16" | "17" | "18" | "19" | "20" | "21" | "22" |
"23" | "24" | "25" | "26" | "27" | "28" | "29" | "30" | "31" | "32" | "33" | "34" | "35" | "36" | "37" |
"38" | "40" | "41" | "43" | "44" | "45" | "46" | "47" | "48" | "50" | "51" | "52" | "53" | "54" | "55" | "56" |
"60" | "61" | "62" | "63" | "64" | "66" | "68" | "70" | "72" | "73" | "74" | "76" | "77" | "78" | "80" |
"81" | "84" | "85" | "86" | "88" | "90" | "92" | "93" | "94" | "95" | "96" | "97" | "98" | "100" | "110" |
"130" | "140" | "150" | "16" | "17" | "18" | "19" | "20" | "21" | "22" | "23" | "24" | "25" | "26" | "27" |
"28" | "29" | "30" | "31" | "32" | "33" | "34" | "35" | "36" | "37" | "38" | "39" | "40" | "41" | "42" | "43" |
"44" | "45" | "46" | "47" | "48" | "49" | "50" | "51" | "52" | "53" | "54" | "55" | "56" | "57" | "58" | "59" |
"60" | "61" | "62" | "63" | "64" | "65" | "66" | "67" | "68" | "69" | "70" | "71" | "72" | "73" | "74" | "75" |
"76" | "77" | "78" | "79" | "80" | "81" | "82" | "83" | "84" | "85" | "86" | "87" | "88" | "89" | "90" | "91" |
"92" | "93" | "94" | "95" | "96" | "97" | "98" | "99" | "100" | "110" | "120" | "130" | "140" | "150" |
"160" | "170" | "180" | "190" | "200" | "210" | "220" | "230" | "240" | "250" | "260" | "270" | "280" |
"290" | "300" | "310" | "320" | "330" | "340" | "350" | "360" | "370" | "380" | "390" | "400" | "410" |
"420" | "430" | "440" | "450" | "460" | "470" | "480" | "490" | "500" | "510" | "520" | "530" | "540" |
"550" | "560" | "570" | "580" | "590" | "600" | "610" | "620" | "630" | "640" | "650" | "660" | "670" | "680" |
"690" | "700" | "710" | "720" | "730" | "740" | "750" | "760" | "770" | "780" | "790" | "800" | "810" |
"820" | "830" | "840" | "850" | "860" | "870" | "880" | "890" | "900" | "910" | "920" | "930" | "940" |
"950" | "960" | "970" | "980" | "990" | "1000" | "1100" | "1200" | "1300" | "1400" | "1500" | "1600" |
"1700" | "1800" | "1900" | "2000" | "2100" | "2200" | "2300" | "2400" | "2500" | "2600" | "2700" | "2800" |
"2900" | "3000" | "3100" | "3200" | "3300" | "3400" | "3500" | "3600" | "3700" | "3800" | "3900" |
"4000" | "4100" | "4200" | "4300" | "4400" | "4500" | "4600" | "4700" | "4800" | "4900" | "5000" | "5100" |
"5200" | "5300" | "5400" | "5500" | "5600" | "5700" | "5800" | "5900" | "6000" | "6100" | "6200" | "6300" |
"6400" | "6500" | "6600" | "6700" | "6800" | "6900" | "7000" | "7100" | "7200" | "7300" | "7400" | "7500" |
"7600" | "7700" | "7800" | "7900" | "8000" | "8100" | "8200" | "8300" | "8400" | "8500" | "8600" |
"8700" | "8800" | "8900" | "9000" | "9100" | "9200" | "9300" | "9400" | "9500" | "9600" | "9700" |
"9800" | "9900" | "10000"}
 />
```

Figura 12: Es. di codelist generato da DocFlex/XML.

DocFlex/XML, così come xs3p e <oxygen>, supporta i *codelist* definiti nel documento XSD tramite il tag “xsd:enumeration” (Fig.12), ma non quelli espressi mediante i *Generic Code*.

Anche qui la rappresentazione dei *codelist* risulta essere poco fruibile in presenza di un elevato numero di valori possibili e ridondante nel caso in cui lo stesso *codelist* venga utilizzato in parti diverse dello schema.

Per quanto riguarda la parametrizzazione della documentazione, DocFlex/XML consente, sia da linea di comando sia tramite interfaccia grafica (Parameter Inspector), di selezionare molti parametri tra i quali i componenti XML-Schema da includere, il livello dei dettagli e la modalità di ordinamento dei componenti stessi (Fig.13).

Parameter	Value	Default
<input checked="" type="checkbox"/> <documentation>		
Language		en
Alternative Language		
Source URI	document as deep-link	
<input checked="" type="checkbox"/> Tags		...
XHTML		
For schemas		...
Include Images	<input checked="" type="checkbox"/>	
Copy Images	<input checked="" type="checkbox"/>	
Other Tags		
Encode markup characters	<input checked="" type="checkbox"/>	
Show part headings	<input checked="" type="checkbox"/>	
For schemas		...
References/Links		
Parameter Description		
Parameter Name: proc. annotation, tags, other		
Data Type: enumeration		
Possible Values: "show", "html", "no"		
Controls what to do with any other XML tags that have not been specifically processed as XHTML markup.		
Possible Choices:		
"show"		
All unprocessed XML tags will appear in the generated documentation.		
"process as XHTML"		
The unprocessed tags will be treated as XHTML tags. (The "XHTML" parameter should be also selected for this.)		
OK Cancel		

Figura 13: Es. di parametrizzazione documentazione in DocFlex/XML.

Così come `<oxygen>`, DocFlex/XML consente di selezionare gli eventuali tag, appartenenti ad altri *namespace*, presenti in “xsd:annotation”; come si evince dalla fig.13 nel riquadro “Parameter Description”, sono possibili tre opzioni per la voce *Other Tags*:

- show – mostra i tag non XHTML in modalità escaping (`<ccts:Example>` → `<ccts:Example>`);
- process as XHTML – processa i tag non XHTML come se fossero tag XHTML.
- no processing – ignora i tag non XHTML.

E' da notare comunque che non è possibile specificare con un dettaglio maggiore quali tag, presenti in “xsd:annotation”, includere o escludere nella documentazione.

Per quanto riguarda la personalizzazione della documentazione, DocFlex/XML di base è in grado di gestire tag appartenenti al vocabolario XHTML presenti all'interno del tag “xsd:annotation”. La fig.14 mostra un esempio di personalizzazione mediante l'aggiunta di tag XHTML come “p”, “i”, “a” e “img” all'interno del tag “xsd:annotation/xsd:documentation”.

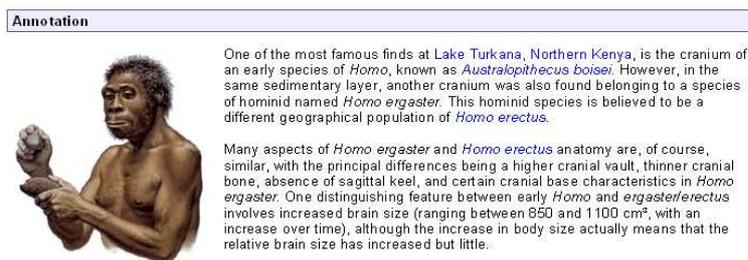


Figura 14: Es. di personalizzazione della documentazione in DocFlex/XML.

Fino ad ora abbiamo descritto le caratteristiche di base (presenti nella versione Freeware), ma le vere potenzialità di DocFlex/XML sono nel Template Designer (Fig.15) (disponibile solo nella versione proprietaria), che

consente di definire un nuovo template da usare al posto dei template di base *FramedDoc* e *PlainDoc*. Il Template Designer, attraverso un meccanismo simile alla definizione dei template in XSLT, permette un'elevata personalizzazione della documentazione da generare: infatti è possibile selezionare i documenti XML d'input da processare, specificare su quali componenti iterare e come visualizzarli.

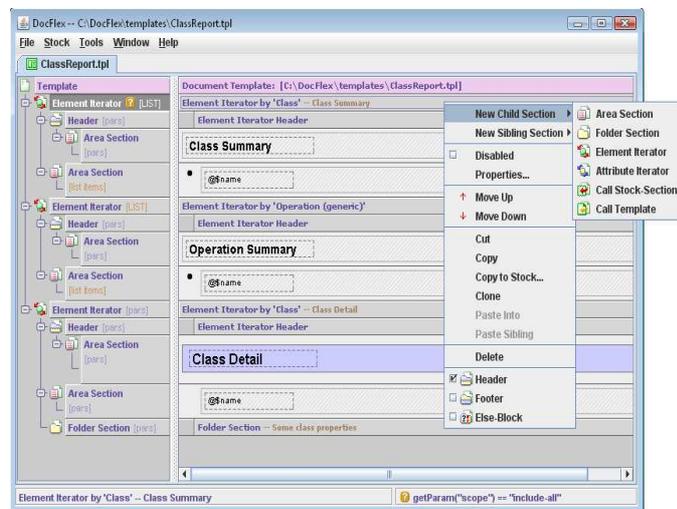


Figura 15: DocFlex/XML Template Designer.

Questa elevata flessibilità di modellare a proprio piacimento la documentazione da generare permette di creare delle guide d'utilizzo che, oltre alla pura descrizione dello schema, includono informazioni aggiuntive esterne, facendo sì che la guida generata sia più adatta ad un fruitore meno tecnico. Si evidenzia che DocFlex/XML si rivela efficiente nel gestire schemi complessi e di grandi dimensioni in breve tempo.

In conclusione DocFlex/XML, tramite il Parameter Inspector e il Template Designer, risulta essere un ottimo applicativo sia nella parametrizzazione e sia nella personalizzazione della documentazione, superando i limiti dei template di base; l'unica mancanza riscontrata è quella relativa all'istanza XML.

1.4 Tabella comparativa degli applicativi

Di seguito è riportata la tabella comparativa degli applicativi presi in esame, per evidenziare le caratteristiche emerse durante la loro analisi.

Parametri considerati	X3SP	<oXygen/>	DocFlex/XML
<i>Licenza</i>	MPL	Proprietaria	Proprietaria
<i>Tecnologia utilizzata</i>	XSLT	JAVA	JAVA
<i>Modalità d'esecuzione</i>	Linea di Comando, Integrato in altri applicativi	Interfaccia Grafica	Linea di Comando, Interfaccia Grafica, Integrato in altri applicativi
<i>Istanza XML</i>	SI, presente	SI, presente	NO, non presente
<i>Codelist XSD</i>	SI	SI	SI
<i>Codelist Generic Code</i>	NO	NO	NO
<i>Supporto XSD di grandi dimensioni</i>	SI	NO	SI
<i>Parametrizzazione</i>	SI	SI	SI
<i>Parametrizzazione xsd:annotation</i>	NO	NO (solo escaping dei tag di altri vocabolari)	SI (tramite il Template Designer)
<i>Personalizzazione</i>	NO	NO	SI
<i>Formati d' output</i>	XHTML	HTML,PDF, DOCBOOK, (personalizzato)	HTML,RTF,TXT

Tabella 1: Tabella comparativa degli applicativi esaminati.

Fra i tre applicativi esaminati, il più completo ed adeguato per realizzare guide per profili d'uso risulta essere DocFlex/XML anche se alcuni aspetti, come la licenza proprietaria e la mancanza della rappresentazione dell'istanza, ne costituiscono un limite d'utilizzo in contesti professionali.

1.5 Documentazione UBL

“Universal Business Language (UBL) is a library of standard electronic XML business documents such as purchase orders and invoices. UBL was developed by an OASIS Technical Committee with participation from a variety of industry data standards organizations. UBL is designed to plug directly into existing business, legal, auditing, and records management practices.. It is designed to eliminate the re-keying of data in existing fax and paper-based business correspondence and provide an entry point into electronic commerce for small and medium-sized businesses.” [UBL11]

UBL (versione 2.0) si prefigge di creare un formato generico per consentire alle diverse imprese commerciali di “dialogare” elettronicamente permettendo loro di aumentare la rapidità, l'efficienza e riducendo errori. Per adempiere questo scopo UBL definisce una libreria di componenti riusabili (es: “Address”, “Item”) che sono i tipici elementi di un documento di business, implementa trentuno modelli di tipi comuni di documenti di business (es: Catalogue, Order, Invoice) ed infine fornisce una metodologia che consente di *caratterizzare* UBL per uno specifico ambito commerciale. Nel *package* della specifica UBL, oltre ai documenti XSD dei componenti riusabili e dei documenti di business, come materiale di supporto troviamo:

- diagrammi UML (*semantica* del documento)
- spreadsheet (*struttura* del documento)
- istanze XML

I diagrammi UML rappresentano i *diagrammi delle classi* sia dei componenti dei documenti sui quali si basano gli XSD e sia dei componenti che descrivono tutti i documenti assemblati. Gli *spreadsheet* sono tabelle (Fig.16), sia in formato *.xls* che *.ods*, le quali definiscono i documenti

assemblati; in esse le righe rappresentano gli elementi (sfondo rosa sono ABIE, quelli con sfondo bianco sono BBIE, mentre quelli con sfondo verde sono ASBIE) e le colonne rappresentano i metadati associati al rispettivo componente, ovvero i vari tag appartenenti al vocabolario CCTS che saranno presenti all'interno del nodo “xsd:documentation” dello schema. Questi *spreadsheet* possono essere creati manualmente oppure in maniera guidata utilizzando la web application eDoCreator [EDO11] che consente, una volta completata la definizione dello spreadsheet, di generare automaticamente gli XSD i quali rappresentano la *sintassi* del documento. Va ricordato che la forma normativa dei documenti UBL sono gli schema XSD e non gli *spreadsheet*. Infine troviamo le istanze XML che sono gli esempi pratici dei documenti assemblati di business come Order, Invoice e CreditNote.

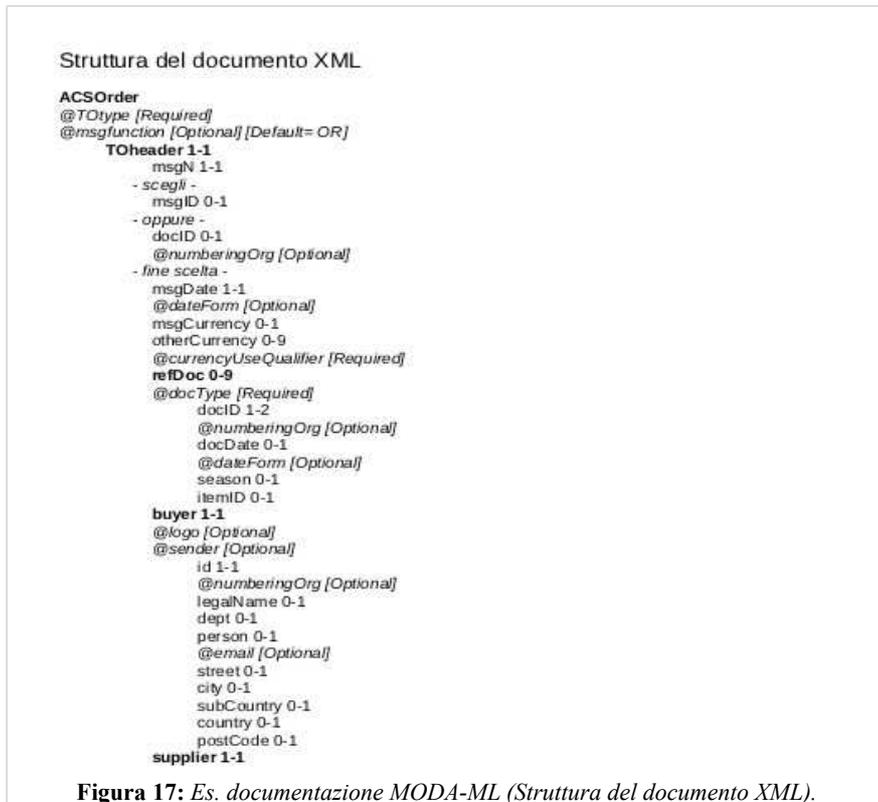
	A	B	C	D	E	F
1	UBL Name	Dictionary Entry Name	Object Class Qualifier	Object Class	Property Term Qualifier	Property Term Possessive Noun
2	ApplicationResponse	Application Response, Details		Application Response		
3	UBLVersionID	Application Response, UBL Version Identifier, Identifier		Application Response		UBL Version
4	CustomizationID	Application Response, Customization Identifier, Identifier		Application Response		Customization
5	ProfileID	Application Response, Profile Identifier, Identifier		Application Response		Profile
6	ID	Application Response, Identifier		Application Response		
7	UUID	Application Response, UUID, Identifier		Application Response		
8	IssueDate	Application Response, Issue Date, Date		Application Response		Issue
9	IssueTime	Application Response, Issue Time, Time		Application Response		Issue
10	ResponseDate	Application Response, Response Date, Date		Application Response		Response
11	ResponseTime	Application Response, Response Time, Time		Application Response		Response
12	Note	Application Response, Note, Text		Application Response		
13	VersionID	Application Response, Version Identifier, Identifier		Application Response		Version
14	Signature	Application Response, Signature		Application Response		
15	SenderParty	Application Response, Sender Party, Party		Application Response	Sender	
16	ReceiverParty	Application Response, Receiver Party, Party		Application Response	Receiver	
17	DocumentResponse	Application Response, Document Response		Application Response		
18						

Figura 16: Es. documentazione UBL (Spreadsheet Model).

1.6 Documentazione MODA-ML

“Il progetto MODA-ML è stato finanziato dalla CE (IST Take-Up Action Line IV.2.5 "Computing, communications and networks take-up measures") del V Programma Quadro (appartiene al cluster di progetti Eutist-AMI). Il progetto, iniziato nel luglio 2001, si è concluso nel marzo 2003; le sue attività proseguono con il sostegno dei partner sotto la definizione di iniziativa Moda-ML. Moda-ML si è posto l'obiettivo di rendere agevole la circolazione di informazioni tecniche e gestionali tra aziende della filiera del Tessile Abbigliamento tramite lo scambio di documenti XML via Internet. Nel concreto il progetto voleva definire un formato dati comune per lo scambio di informazioni via Internet tra le aziende della filiera: un linguaggio con caratteristiche tali da poter essere incorporato, a costi contenuti, tanto negli attuali sistemi informativi aziendali quanto nelle nuove soluzioni informatiche offerte dai fornitori di tecnologia. L'ambizione era operare con livelli qualitativi tali da poterne poi proporre la formalizzazione come standard internazionale per il settore.” [MML06]

Dalla homepage di MODA-ML [MML11] è possibile reperire, per ogni schema appartenente al package, la relativa guida in formato .pdf. All'interno di queste guide è possibile trovare sia la descrizione del documento XML (Fig.17), sia la descrizione dettagliata degli elementi del documento (Fig.18).



Questo tipo di visualizzazione dell'istanza del documento XML risulta essere molto chiara e di facile comprensione:

- I tag in grassetto rappresentano *elementi non foglia*, ovvero che contengono altri elementi al loro interno (es. “buyer”).
- I tag non in grassetto rappresentano *elementi foglia*, ovvero che NON contengono altri elementi all'interno, ma solo eventuali attributi (es. “docID”).
- I tag preceduti dal simbolo “@” rappresentano gli *attributi* (es. @sender).
- Elementi ed attributi sono allineati verticalmente secondo il loro livello di annidamento.

Oltre alla struttura sono riportate le informazioni più rilevanti che sono la cardinalità (es. 0–1), il possibile uso dell'attributo (es. required, optional) e la presenza di *compositor* (es. choice, scegli – oppure- fine scelta).

L'unico limite di questo metodo di visualizzazione è che non si adatta bene nel caso di documenti XSD molto lunghi in quanto risulta essere difficoltoso seguire lo schema perché distribuito su troppe pagine.

La struttura del documento è seguita dalla descrizione degli elementi semplici e complessi presenti nell'istanza XML, con i corrispettivi *XPath*, ovvero tutti i percorsi dove è possibile trovare l'elemento.

acsCode	codifica che identifica un prodotto accessorio di abbigliamento (XPath: ACSOrder/ACbody/AItem/acsCode) 1-2
acsComInfo	informazioni varie di natura commerciale, specificabili in un Ordine di accessori, riferibili all'intero documento come al singolo item (XPath: ACSOrder/ACbody/AItem/acsComInfo) 0-1 (XPath: ACSOrder/acsComInfo) 0-1
acsPack	modalità di confezione dell'accessorio (XPath: ACSOrder/acsComInfo/acsPack) 0-1 (XPath: ACSOrder/ACbody/AItem/acsComInfo/acsPack) 0-1

Figura 18: Es. documentazione MODA-ML (Descrizione elementi).

E' da notare che questo tipo di documentazione viene generato in automatico utilizzando un database MS-Access contenente tutte le definizioni degli elementi, tipi e attributi definiti.

1.7 Documentazione eBiz-TCF

Il progetto eBIZ-TCF è stato lanciato dalla Commissione Europea ed è condotto da EURATEX (Confederazione Europea dell'Industria Tessile e Abbigliamento), CEC (Confederazione Europea dell'Industria della Calzatura) e dall'italiana ENEA, con l'obiettivo di armonizzare le procedure di scambio dati e-Business per le imprese europee della moda: Tessile, Abbigliamento e Calzaturiero (TCF acronimo in inglese). [PD10]

Nella documentazione per gli schemi eBiz-TCF [EB11] troviamo sia la struttura dell'istanza del documento e sia la descrizione degli elementi in un'unica tabella (Fig.19).

XML Tag	Content description
Catalogue	
UBLVersionID 1-1	Constant: 2.0
CustomizationID 1-1	Constant: "eBiz-TCF-v2008-1"
ProfileID 1-1	Constant: "eBiz-TCF-ArticleCatalogue"
ID 1-1	Document number: an identifier for the Catalogue assigned by the sender.
Name 0-1	Supplier catalogue ID: the identifier of a special assortment defined by the supplier (for example the NOS articles of the Supplier).
IssueDate 1-1	Document date: the date on which the Catalogue was issued.
Note 0-1	Note: free-form text (note or similar information) applying to the document.
VersionID 0-1	Supplier catalogue version: catalogue version according to the Supplier's system.
PreviousVersionID 0-1	Catalogue reference: an associative reference to a Catalogue that can be "changed" or "replaced" by this.
ValidityPeriod 0-1	Validity period: the period during which (or the date from which) the information in the Catalogue is effective. The Start date is mandatory, the End date should be specified only if really need, for example in case of special offer.
StartDate 1-1	The start date of the period.
EndDate 0-1	The end date of the period.
Provider Party 1-1	
PhysicalLocation 1-1	
ID 1-1	GLN catalogue sender: GLN code of the place where the catalogue sender's physical location is.
@schemeName [optional]	Constant: "GLN"
Receiver Party 1-1	
PhysicalLocation 1-1	
ID 1-1	GLN catalogue recipient: GLN code of the place where the catalogue recipient's physical location is.

Figura 19: Es. documentazione eBiz-TCF.

Questo modello di visualizzazione risulta essere chiaro e d'immediata comprensione: infatti nella colonna di sinistra è possibile, come nella documentazione di MODA-ML vista in precedenza, seguire la struttura dell'istanza XML, mentre nella colonna di destra, per ogni riga, vi sono le relative informazioni. Anche questo modello di visualizzazione però non si adatta bene per documenti XSD molto lunghi. È da notare che questa documentazione non è generata né in modo automatico né a partire da file XML-Schema, ma è interamente realizzata a mano.

1.8 Considerazioni finali

Dall'esame delle documentazioni generate dagli applicativi e di quelle disponibili dei linguaggi esaminati, risulta che il modello di eBiz-TCF si addice maggiormente per creare guide per profili d'uso. Tuttavia essendo questo modello prodotto in modo manuale, si rende necessario realizzare un *applicativo* che automatizzi il processo di creazione e che abbia le seguenti *feature*:

- generazione a partire da documenti XSD
- modello di visualizzazione in stile eBiz-TCF
- gestione separata dei codelist XSD dalle altre Facets
- supporto per codelist espressi tramite *Generic Code*
- supporto business rule (*schematron*)
- possibilità di parametrizzare la documentazione da generare
- possibilità di integrare l'output finale con documentazione esterna
- output nei formati XML, HTML e PDF
- generazione output statistiche

Capitolo 2. Progettazione

In questo capitolo verrà discussa la progettazione dell'applicativo XDG (*XML-Schema Documentation Generator*) tenendo conto dei seguenti aspetti:

- requisiti richiesti
- funzionalità fornite
- analisi della struttura dell'applicativo

2.1 Requisiti dell'applicativo

L'applicativo XDG deve essere in grado di generare guide per profili d'uso derivate da specifiche standard rispettando vari requisiti, alcuni dei quali esaminati nella sezione 1.4. In particolar modo deve:

1. utilizzare documenti XSD
2. rappresentare l'albero dell'istanza XML
3. semplificare su richiesta i rami dell'albero dell'istanza XML
4. documentare le liste di codici
5. documentare le Business rules
6. consentire la parametrizzazione della documentazione
7. consentire la personalizzazione della documentazione
8. produrre output HTML e PDF
9. produrre output statistiche
10. eseguire da linea di comando e in modalità batch

2.1.1 Documenti XSD

Il primo requisito che l'applicativo deve supportare è quello di essere in grado di generare le guide utilizzando come punto di partenza esclusivamente documenti XSD definiti attraverso il linguaggio XML-Schema [PD04] e non altri formati.

Questa scelta è motivata dal fatto che XSD è uno standard ormai largamente affermato per la creazione di template di documenti. Inoltre l'applicativo deve essere in grado di gestire necessariamente documenti XSD strutturati e annotati secondo la specifica *Core Components*.

Core Components

La Core Components Technical Specification [UC09] fornisce un modo per identificare e massimizzare il riuso delle business information in modo tale da supportare e favorire l'interoperabilità delle informazioni sia in un formato fruibile per esseri umani sia in un formato comprensibile alle macchine.

L'approccio seguito dalla specifica CC è stato quello di fornire una semantica syntax-neutral: questo tipo di approccio consente, mediante l'uso del linguaggio naturale, di creare modelli di dati scambiabili, evitando le limitazioni tipiche dei linguaggi caratterizzati da una sintassi rigorosa.

Questa soluzione si adatta perfettamente alle variazioni dei modelli di dati, della semantica e della struttura dei messaggi: variazioni che in passato hanno condotto all'incompatibilità e conseguente mancanza d'interoperabilità.

I concetti chiave della specifica Core Components si basano su due livelli: core components e business information entities. I Core Components sono elementi riusabili che identificano un concetto astratto e, prima di poter essere utilizzati, devono essere contestualizzati entro un determinato ambito di business ovvero trasformati in Business Information Entities.

2.1.2 Albero dell'istanza XML

Il secondo requisito dell'applicativo è che esso generi guide contenenti l'intero albero dell'istanza XML, composto da elemento radice, nodi foglia e non foglia ed attributi.

Infatti, presentando l'albero dell'istanza XML dell'intero documento, si ottiene una visione globale della sua struttura che ne favorisce la comprensione in maniera più semplice ed immediata rispetto invece ad una visione frammentaria, come avviene ad esempio nella documentazione generata dall'applicativo <oxygen> XML Editor nella quale le istanze degli elementi vengono riportate singolarmente.

L'albero dell'istanza deve anche presentare eventuali informazioni sui nodi per facilitare la loro comprensione all'utilizzatore della guida.

2.1.3 Semplificazione dei rami dell'albero

Come terzo requisito l'applicativo deve fornire, all'utente che deve generare la guida, un metodo per semplificare l'albero dell'istanza XML ovvero un modo per specificare quali nodi, con eventuali nodi annidati al suo interno, rimuovere dall'albero dell'istanza XML.

Questo tipo di *customizzazione* dell'albero dell'istanza XML si rivela vantaggioso ad esempio nel caso in cui si debba generare, senza alterare la cardinalità degli elementi e/o attributi presenti nel documento XSD originale, una guida, di prova, relativa ad un documento da adattare ad un contesto di business specifico dove, però, molti elementi opzionali non sono necessari e pertanto eliminabili.

Poter effettuare questo tipo di operazione rende la guida più semplice e immediata da leggere e da capire in quanto vengono appunto rimosse le parti superflue del documento.

2.1.4 Liste di codici

Il quarto requisito è che l'applicativo deve essere in grado di gestire in modo efficiente e completo le liste di codici:

- in modo efficiente perché gli elementi e gli attributi, che utilizzano le liste di codici, potrebbero presentarsi un numero elevato di volte nell'albero dell'istanza XML, appesantendo notevolmente la guida
- in modo completo perché le liste di codici, fondamentali nei documenti di *ebusiness*, possono essere definite sia mediante il documento XSD stesso e sia per mezzo di altri linguaggi come ad esempio *Generic Code*

Generic Code

Il consorzio OASIS-OPEN ha definito la specifica “Code List Representation (Genericode) Version 1.0” [OA07] per rappresentare le liste di codici, indicate anche come *vocabolari controllati* o *enumeration*. L'obiettivo principale è quello di fornire una modalità standard per rappresentare e scambiare liste di codici cercando di superare le classiche problematiche legate ad esse, come ad esempio la traduzione dei codici nelle diverse lingue (es: domenica, dimanche, domingo, sunday, sonntag).

I documenti generic code sono espressi concettualmente mediante UML e concretamente in un formato XML generalmente con estensione “.gc”; essi possono contenere un vasto insieme di informazioni sulle liste di codici. Il formato XML è stato adottato proprio per favorire la distribuzione e l'interoperabilità delle liste di codici tra sistemi differenti. Le liste di codici vengono strutturate secondo un modello tabellare dove:

- le righe rappresentano “conceptual code”
- le colonne rappresentano “associated value”

L'idea di fondo è la stessa dei database relazionali: le righe rappresentano concettualmente le voci della lista (*distinct entry*), le colonne rappresentano l'insieme di valori definiti per ogni voce della lista (*codelist metadata*). Per ogni riga deve essere presente almeno una colonna che assume la funzione di chiave (*key*).

2.1.5 Business rules

Come quinto requisito l'applicativo deve essere in grado di riportare nella guida le informazioni sulle *Business rules*, espresse mediante il linguaggio Schematron, per indicare e documentare la presenza di eventuali *vincoli aggiuntivi* applicabili in determinati contesti del documento XSD.

Schematron

Il linguaggio Schematron fa parte dello standard ISO/IEC 1975 [II06], il quale suddivide il processo di validazione di un documento XML (o SGML) in più parti proponendo tecnologie specifiche per funzioni diverse:

- *Regular-grammar-based validation* — RELAX NG [JM01]
- *Rule-based validation* — Schematron
- *Namespace-based Validation Dispatching Language* — NVDL

Fra queste tre tecnologie Schematron consente, utilizzando un approccio basato sulla definizione di regole, di esprimere vincoli sul documento XML altrimenti non esprimibili mediante l'uso di linguaggi *grammar-based* come ad esempio RELAX-NG e XML-SCHEMA (1.0). Il linguaggio Schematron è caratterizzato dai seguenti aspetti:

- Sintassi XML
- Regole organizzate in documenti con estensione “.sch”
- Uso del linguaggio XPath per esprimere il *context* e il *test* della regola

2.1.6 Parametrizzazione della documentazione

L'applicativo deve permettere all'utente di parametrizzare la documentazione cioè di poter scegliere sia le informazioni, estratte dal file XSD, da includere nella guida sia i parametri per impostare la struttura dell'output, in modo tale da generare selettivamente guide mirate in base alle esigenze specifiche del fruitore e all'ambito di utilizzo.

2.1.7 Personalizzazione della documentazione

L'applicativo deve permettere all'utente di personalizzare la guida con contenuto extra, come ad esempio capitoli, sezioni ed immagini, che possono essere utili nel dettagliare e spiegare meglio lo schema del profilo d'uso, in modo tale da renderla meno schematica, più discorsiva e quindi maggiormente fruibile.

2.1.8 Output HTML e PDF

L'ottavo requisito è che l'applicativo deve produrre la guida nei formati HTML e PDF. Il formato HTML è adatto per visionare la guida mediante un *web browser* mentre il formato PDF è utilizzabile sia per poter visionare la guida nella versione elettronica tramite un *PDF reader* sia per produrla nella versione stampabile.

2.1.9 Output statistiche

Come nono requisito l'applicativo deve generare dei documenti contenenti le statistiche sia per ottenere dei parametri con i quali valutare l'applicativo stesso in termini di *performance* sia per produrre *report* sulla struttura dell'albero in modo tale da analizzarla nel dettaglio. Si è ritenuto opportuno aggiungere a posteriori tale requisito per avere un metro di giudizio sulla bontà sia degli algoritmi sia degli output prodotti.

2.1.10 Modalità d'esecuzione

Infine l'applicativo deve poter essere eseguito da linea di comando e in modalità *batch*.

L'esecuzione da linea di comando è stata scelta per i seguenti motivi:

- per concentrarsi maggiormente sul *kernel* dell'applicativo anziché sull'interfaccia
- per consentire l'utilizzo dell'applicativo all'interno di altri programmi o script.

La modalità batch si adatta ad un uso non interattivo dell'applicativo, utile ad esempio nel caso in cui si debba processare un gran numero di documenti XSD i quali, eventualmente, potrebbero richiedere tempi molto lunghi.

2.2 Funzionalità dell'applicativo

Partendo dall'analisi dei requisiti sono stati identificati, di conseguenza, le funzionalità che l'applicativo XDG deve fornire:

1. supporto documenti XSD
2. rappresentazione dell'albero dell'istanza XML
3. semplificazione su richiesta dei rami dell'albero dell'istanza XML
4. gestione Code List
5. supporto Business rules
6. supporto parametrizzazione documentazione
7. supporto personalizzazione documentazione
8. output guida per profilo d'uso
9. output statistiche
10. gestione parametri d'input linea di comando e modalità batch

2.2.1 Supporto documenti XSD

La prima funzionalità su cui ci si è concentrati è stata quella di generare guide utilizzando come input documenti XSD. Pertanto si è scelto di fornire inizialmente un supporto completo ai componenti XML-Schema usati nella specifica OASIS UBL 2.0 fino a renderlo il più generale possibile.

Di seguito è rappresentato il diagramma dei componenti del linguaggio XML-Schema (Fig.21) nel quale sono riportate in verde le parti gestite ed in rosso quelle non gestite dall'applicativo.

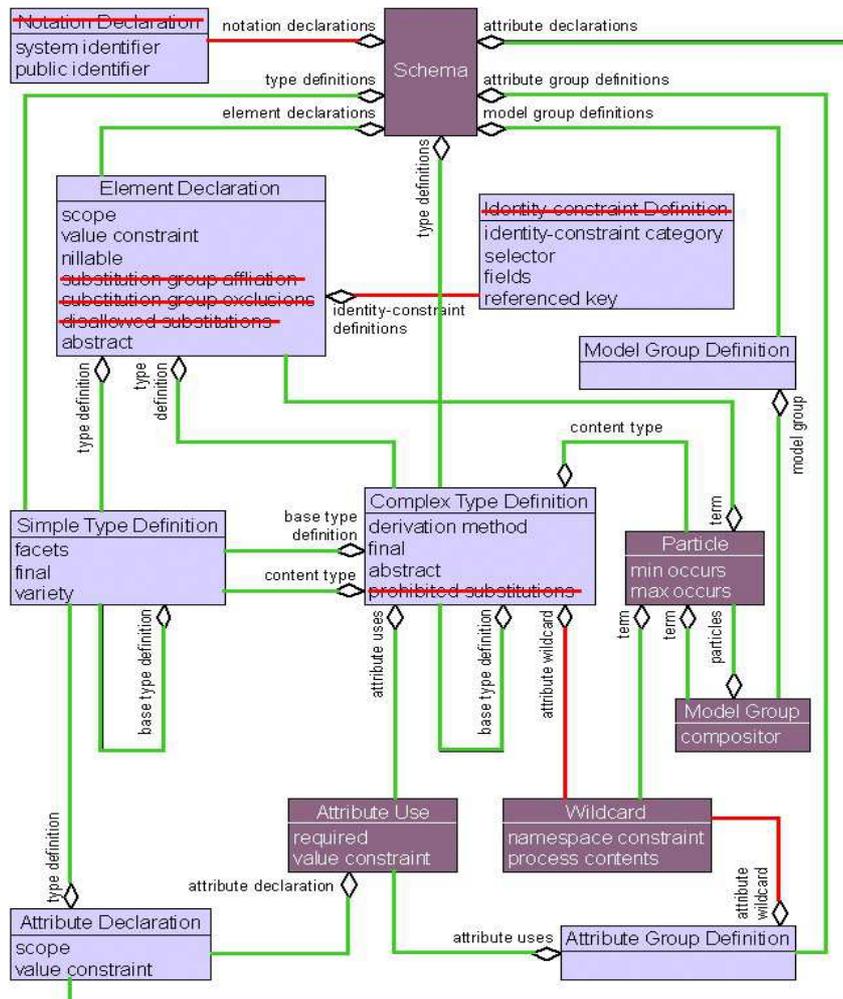


Figura 20: Diagramma UML dei componenti XML-Schema gestiti.

L'applicativo è in grado di gestire la maggior parte dei componenti XML-Schema tranne i tipi anonimi; di conseguenza i modelli di progettazione supportati sono i seguenti:

- giardino dell'eden – È caratterizzato sia dal definire come globali tutti gli elementi e i tipi sia dall'usare sistematicamente “@ref” per tipare gli elementi.
- tende alla veneziana – È caratterizzato dal possedere un unico elemento globale, l'elemento radice, e tanti tipi nominali ciascuno dei quali contiene definizioni di elementi locali associati a tipi nominali ed esterni.

I modelli di progettazione *bambole russe* e *fette di salame* non sono stati supportati poiché definendo tutti i tipi localmente non ne consentono il riuso, contrariamente a quanto accade negli XSD basati sui Core Components.

2.2.2 Rappresentazione dell'albero dell'istanza XML

La guida comprende la rappresentazione dell'albero dell'istanza XML. Nella costruzione dell'albero ci si è concentrati nell'adottare due diversi approcci: *statico* e *dinamico*. L'approccio *statico* consiste nel rappresentare l'albero dell'istanza XML per intero, in tutta la sua lunghezza, come ad esempio accade nell'output PDF. L'approccio *dinamico* consiste nel caricare e quindi visualizzare rami dell'albero dell'istanza XML in modo selettivo.

L'approccio statico si adatta bene a documenti di piccole dimensioni, consentendo di visionare per intero l'albero dell'istanza in modo diretto e completo. L'approccio dinamico si rivela l'unico possibile quando i documenti XSD possono consentire istanze XML molto grandi, come ad esempio il documento “Order” presente in UBL 2.0: infatti, poiché questo può dar vita, nella versione minima, ad un'istanza composta addirittura da

circa 3.000.000 di nodi tra elementi e attributi, l'approccio dinamico consente di ridurre la dimensione dell'output notevolmente rispetto a quella con l'approccio statico.

Si è scelto di rappresentare l'albero dell'istanza XML seguendo le modalità spiegate nelle sezioni 1.6 e 1.7 sulla documentazione relativa alle specifiche MODA-ML ed eBiz-TCF. Tale rappresentazione comporta la gestione dei seguenti aspetti:

- elementi foglia e non foglia, attributi, compositor e wildcard
- livelli di annidamento indentati e numerati
- cardinalità, mixed content e nil come informazioni aggiuntive

Inoltre, parallelamente all'albero dell'istanza XML, come per la documentazione eBiz-TCF, sono riportate le informazioni relative agli elementi, attributi e wildcard.

2.2.3 Semplificazione rami dell'albero dell'istanza

Alla rappresentazione dell'albero dell'istanza XML, sopra citata, è stata aggiunta la funzionalità di riduzione, su richiesta, dei rami dell'albero dell'istanza XML. Questa consente di specificare, mediante una sintassi XPath [JD99] semplificata, un elenco di path da non includere nella rappresentazione dell'albero dell'istanza XML.

Tale operazione avviene nella generazione *statica* della rappresentazione dell'albero dell'istanza XML poiché, solo in questo caso, esso viene percorso interamente: per ogni ramo dell'albero si confronta il corrispondente *canonical path* con i path definiti nell'elenco di quelli da togliere e, se vi è almeno una corrispondenza, il ramo viene “potato” (in inglese *trim*) e quindi non processato. L'applicativo XDG è in grado di gestire solo un sottoinsieme

semplice di espressioni XPath poiché non si ha l'istanza XML vera e propria sulla quale applicare le query XPath, ma solo una sua *rappresentazione*, per giunta parziale, nel senso che per ogni ramo si conoscono solo i nodi già processati e non quelli successivi (questo punto verrà spiegato meglio nel capitolo sull'implementazione).

Le espressioni XPath semplificate devono avere la seguente forma:

- canonical path (/a/b/c , /a/b/@a , /cbc:a/udt:b , /@xml:lang)
- descendant axis nella forma contratta (//a , /a/b//y:c , //a//x:b//c//@d)

2.2.4 Gestione Code List

L'applicativo riesce a gestire, in modo compatto ed efficiente, le liste di codici espresse sia mediante i nodi “xsd:enumeration” presenti nello XSD e sia mediante i *generic code*.

- In modo compatto perché le facet enumeration, a differenza delle altre facet, durante l'analisi del documento XSD d'input, sono state decontestualizzate per essere raggruppate insieme ai codelist espressi mediante i *generic code* e quindi documentate nello stesso modo.
- In modo efficiente in quanto le liste di codici, poiché possono sia essere costituite da molti valori e sia presentarsi nell'albero dell'istanza XML in più nodi, sono state gestite in modo diverso, *statico* e *dinamico*, a seconda del tipo di output della guida (PDF o HTML).

La modalità *statica* consiste nell'indicare nei nodi dell'albero solo il nome della lista di codici e nel riportare tutti i valori del codelist una sola volta in una sezione della guida. La modalità *dinamica* consiste nel caricare, nei nodi dell'albero dell'istanza XML interessati, l'intera lista di codici solo quando viene esplicitamente richiesta.

Nella guida le liste di codici vengono etichettate nei seguenti modi per evidenziarne la modalità d'uso:

- *mandatory* – *d'uso obbligatorio, definito nello XSD*
- *recommended* – *d'uso consigliato, definito nel file Generic Code.*

2.2.5 Supporto Business rules

L'applicativo è in grado di riportare le informazioni sulle *Business rules* espresse mediante il linguaggio Schematron, il quale permette appunto di fare asserzioni su ciò che il documento XML può o non può contenere. Si è scelto di supportare un set di componenti *Core Elements*, appartenenti alla specifica Schematron, sufficiente a gestire le informazioni sui vincoli da documentare.

Oltre a riportare le informazioni sui vincoli, ci si è concentrati maggiormente nel cercare di associare i vincoli ai corrispettivi nodi contesto nell'albero dell'istanza. Per fare ciò, come visto nella sezione 2.2.3 sulla semplificazione dei rami dell'albero dell'istanza XML, è necessario che l'espressione XPath, presente nell'attributo *context* della “sch:rule”, sia in una delle seguenti forme:

- canonical path (/a/b/c , /a/b/@a , /cbc:a/udt:b , /@xml:lang)
- descendant axis nella forma contratta (//a , /a/b//y:c , //@a ,
//a//x:b//c//@d)

Nel caso in cui l'espressione XPath presenti *predicati*, ovvero espressioni racchiuse tra parentesi quadre utilizzate per filtrare in una sequenza solo alcuni nodi, questi vengono eliminati e, qualora l'espressione risultante rientri in una delle due forme sopraelencate, essa può essere associata. Di conseguenza vi saranno due tabelle: la prima contenente le regole linkate ai nodi dell'albero, la seconda contenente invece le regole che non si sono potute associare in nessun punto dell'albero.

2.2.6 Parametrizzazione della documentazione

L'applicativo consente di parametrizzare la documentazione dell'output delle guide. Come visto per gli applicativi descritti nei capitoli 1.1, 1.2 e 1.3, è possibile selezionare quali tipologie di informazioni, relative alla documentazione, includere nelle guide per differenziarle a seconda del contesto d'utilizzo.

In particolar modo ci si è concentrati nel fornire un metodo per scegliere, con una granularità fine, quali tag, presenti all'interno degli elementi “*xsd:documentation*” appartenenti per es. al vocabolario CCTS, includere nella documentazione.

2.2.7 Personalizzazione della documentazione

Oltre alla parametrizzazione della documentazione ci si è focalizzati nel fornire un metodo per personalizzare il contenuto della guida.

Si è scelto pertanto di utilizzare il formato standard *docbook* [NR10] come mezzo per arricchire le guide con contenuto extra allo XSD come ad esempio la copertina, i capitoli, le sezioni, le immagini e la bibliografia.

Il formato *docbook* è particolarmente adatto a questo scopo in quanto esso, essendo un linguaggio di markup semantico, fornisce un metodo per definire la struttura e il contenuto di un documento a prescindere da quella che sarà la sua presentazione.

Quindi, per consentire all'utente di personalizzare il contenuto delle guide, si è definito un “*template*” composto da un sottoinsieme di tag ammissibili, tra i molti messi a disposizione dalla specifica *docbook*. Il *template docbook* verrà esaminato nel dettaglio nella sezione 2.3.3.

2.2.8 Output guida per profilo d'uso

Come da requisito, l'applicativo è in grado di generare la guida nei seguenti formati:

- HTML – *per visualizzare la guida nei web browser*
- PDF – *per visualizzare la guida su dispositivi elettronici o produrla nella versione cartacea.*

Oltre agli output HTML e PDF, l'applicativo produce un documento di output intermedio nel formato XML il quale contiene sia tutti i componenti XML-Schema processati nello XSD sia tutti i codelist. Esso viene utilizzato dall'applicativo stesso per generare gli output HTML e PDF.

Per l'uso dell'output HTML sono stati scelti come riferimento i seguenti *web browser*:

- internet explorer
- firefox
- chrome/chromium
- safari
- opera

Per l'uso dell'output PDF sono stati scelti come riferimento i seguenti *pdfreader*:

- Acrobat reader
- foxit
- evince
- xpdf
- ePDFView

2.2.9 Output statistiche

Oltre alla guida nei formati HTML, PDF e l'output intermedio XML, si è scelto di generare documenti con dati statistici contenenti sia le informazioni generali riepilogative sull'esecuzione del *task* completato e sia le informazioni più tecniche relative all'albero dell'istanza XML. Il contenuto di ciascun documento statistico generato sarà presentato nella sezione 2.3.4.

2.2.10 Linea di comando e modalità batch

L'applicativo viene eseguito da linea di comando ed in modalità batch. L'utilizzo della linea di comando è utile ad esempio nei casi cui si ha l'esigenza di integrare l'applicativo in *script* o in altri programmi, mentre la modalità batch si adatta all'utilizzo dell'applicativo nel generare molteplici guide, che potrebbero richiedere molto tempo per la loro realizzazione.

Per adempiere a questo scopo si è scelto di utilizzare un file di configurazione dove possono essere specificati i *task*, ovvero i processi di generazione delle guide per i file XSD d'input, con le relative impostazioni di configurazione. Questa scelta si è rivelata adeguata al fine di permettere, in modo semplice ed efficace, sia una configurazione particolareggiata nei dettagli per un singolo file XSD e sia una configurazione generica comune a più file XSD d'input.

Inoltre l'applicativo prevede un file di *log* che, come l'output della linea di comando, presenta le seguenti informazioni relative ad ogni *task* da eseguire:

- riepilogo parametri d'input specificati
- passaggi intermedi completati
- eventuali messaggi con severità: info, warning ed error

2.3 Architettura XDG

Dopo aver analizzato nelle sezioni precedenti i requisiti e le funzionalità relativi all'applicativo XDG, sarà presentata ed esaminata la sua architettura nei seguenti aspetti:

- ciclo d'esecuzione
- analisi dei componenti
- input
- output

2.3.1 Ciclo d'esecuzione

Il ciclo di esecuzione dell'applicativo XDG, come si evince dalla figura sottostante (Fig.21), risulta essere organizzato in modo semplice e lineare.

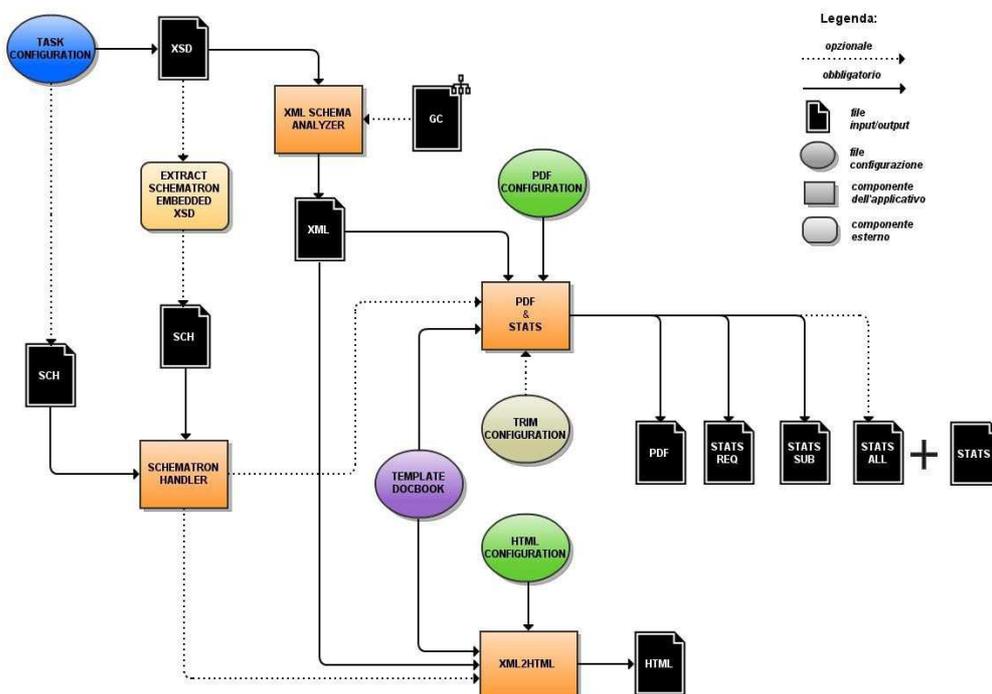


Figura 21: Ciclo d'esecuzione dell'applicativo XDG.

Esso inizia con l'esaminare il file "*configurazione TASK*" in modo da ottenere l'elenco dei documenti XSD con le relative opzioni di configurazione da processare. Ogni documento d'input XSD viene analizzato e documentato mediante il componente "*XML schema analyzer*" il quale, tenendo conto anche degli eventuali documenti contenenti le liste di codici espresse tramite i Generic Code, genera in output il documento XML.

In seguito si provvede, qualora necessario, a generare una versione minimale del documento Schematron schema, mediante il componente "*Schematron handler*", in modo tale da poter essere utilizzata nei passaggi successivi. Una volta ottenuti l'output XML ed opzionalmente il documento Schematron schema nella versione minimale, si procede con il generare dapprima l'output HTML e successivamente l'output PDF insieme ad alcuni documenti contenenti le statistiche.

Il componente "*xml2html*", utilizzando sia il file "*configurazione HTML*" sia il "*template docbook*" e, se specificato, il documento Schematron schema nella versione minimale, trasforma l'output XML nel documento HTML.

Similmente, il componente "*PDF&stats*" partendo dall'output XML, assieme al file di "*configurazione PDF*", al "*template docbook*" e, se definiti, ai documenti Schematron schema e "*configurazione trim*", genera l'output PDF e tre documenti statistici sull'albero dell'istanza XML, rispettivamente "*stats required*", "*stats subordinated*" e "*stats all*".

Il ciclo d'esecuzione dell'applicativo XDG termina con la generazione di un documento, denominato "*stats*", contenente i dati statistici riepilogativi sia sui tre output principali generati, XML, HTML e PDF, sia sull'albero dell'istanza.

2.3.2 Analisi dei componenti

L'applicativo XDG, come visto nel suo ciclo d'esecuzione, utilizza i seguenti componenti:

- XML schema analyzer
- xml2html
- PDF&Stats
- SCHEMATRON handler

Componente XML Schema analyzer

Il componente *XML schema analyzer* rappresenta la parte nevralgica dell'applicativo sia perché esso gestisce i componenti del linguaggio XML-Schema presenti nel documento XSD sia perché produce l'output XML, il quale successivamente sarà l'input di base per generare i documenti HTML e PDF.

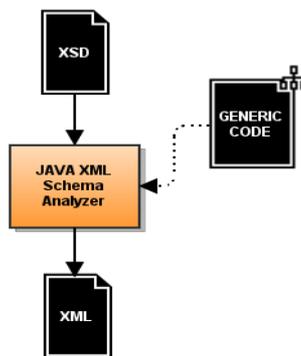


Figura 22: Diagramma componente XML schema analyzer.

Come si evince chiaramente dalla figura (Fig.22), lo *XML schema analyzer* ha come input principale il documento XSD, che può essere strutturato anche su più librerie, eventualmente situate in rete, ed ha come input opzionale un insieme di documenti *Generic Code*, nei quali sono definite le liste di codici.

Lo XML *schema analyzer*, durante l'analisi del documento XSD, costruisce le varie parti dell'output XML ed usa quest'ultimo come struttura dati per memorizzare tutte le informazioni processate e per “raffinare”, in seguito, l'output XML attraverso passaggi successivi, come rappresentato nella figura sottostante (Fig.23).

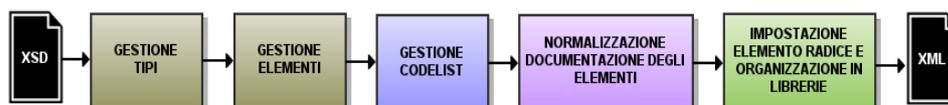


Figura 23: Pipeline XML Schema analyzer.

Nei primi due passaggi vengono gestiti sia tutti i tipi (“xs:simpleType” e “xs:complexType”) sia tutti gli elementi definiti globalmente presenti nel documento XSD principale ed eventualmente anche nelle librerie importate, incluse e ridefinite. Durante il primo e il secondo passaggio vengono gestiti tutti i componenti XML-Schema supportati dall'applicativo come attributi, compositor, wildcard e facet, che è possibile trovare definiti all'interno dei tipi e degli elementi.

Il terzo passaggio gestisce le informazioni dei codelist espressi mediante i *generic code*, qualora presenti, inserendo le informazioni sulle liste di codici.

Nel quarto passaggio si provvede a completare le informazioni degli elementi, sia quelli definiti globalmente che quelli definiti localmente, con le informazioni dei relativi tipi.

L'ultimo passaggio definisce l'elemento radice dell'albero dell'istanza XML ed organizza, sempre all'interno del file XML, tutti i tipi e gli elementi in sezioni secondo il rispettivo namespace di appartenenza.

Componente xml2html

Il componente *xml2html* ha il compito di generare la guida nel formato d'output HTML. Come rappresentato nel diagramma sottostante (Fig.24), il

componente prende in input i seguenti parametri: il file di configurazione HTML per le impostazioni di configurazione, il template docbook per la personalizzazione della guida con contenuto extra ed infine, opzionalmente, lo Schematron schema nella versione minimale per documentare le *business rules*.

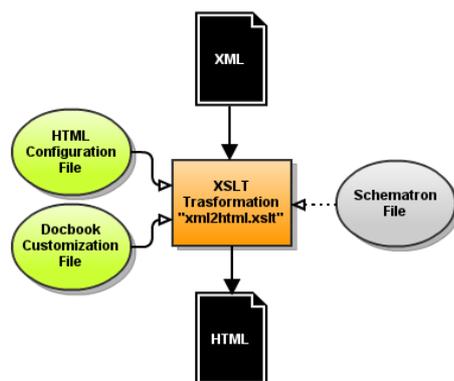


Figura 24: Diagramma componente xml2html..

Il componente xml2html per generare l'output HTML compie una serie di passaggi. Il primo passaggio consiste nell'adottare i parametri d'input definiti nel file di configurazione dell'output HTML, come ad esempio i colori da utilizzare per differenziare elementi, attributi, compositor e wildcard presenti nell'albero dell'istanza XML.

Il passaggio successivo imposta la copertina, l'indice, i capitoli descrittivi ed il capitolo tabella del profilo d'uso, tutti definiti nel template docbook.

Dopo questi primi due passaggi, il terzo inserisce la tabella del profilo d'uso, la quale contiene momentaneamente solo l'elemento radice dell'albero dell'istanza in quanto, l'output HTML, come visto nella sezione 2.2.2, utilizza la modalità *dinamica*.

Il quarto passaggio aggiunge il capitolo business rules definito nel template docbook ed il quinto ne riporta le regole presenti nel documento Schematron schema nella sua versione minimale.

Il sesto passo completa la guida con la bibliografia specificata nel template docbook.

L'ultimo passaggio crea una sezione contenente le righe della tabella del profilo d'uso, le quali rappresentano i nodi (elementi, attributi, compositor e wildcard) utilizzabili per assemblare dinamicamente l'albero dell'istanza XML.

Componente PDF&stats

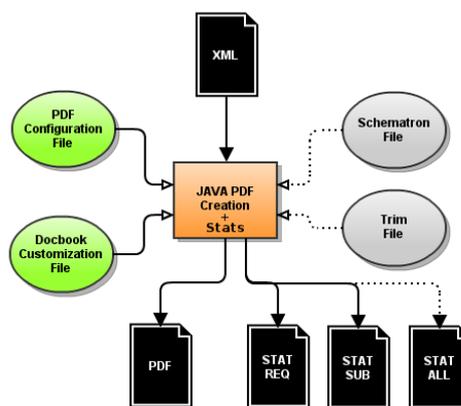


Figura 25: Diagramma componente PDF&stats.

Il componente PDF&stats ha il compito di generare sia la guida nel formato d'output PDF sia tre documenti statistici. Il componente, come si evince nel diagramma (Fig.25), ha in input i seguenti parametri: il file di configurazione PDF per le impostazioni di configurazione, il template docbook per personalizzare la guida con contenuto extra, opzionalmente lo schematron schema nella versione minimale contenente le *business rules* ed infine il file trim, anch'esso facoltativo, per specificare i rami dell'albero dell'istanza da non includere.

Il componente PDF&stats, per generare l'output PDF e i documenti statistici, compie una serie di passaggi. Il primo passaggio consiste nell'adottare i parametri d'input specificati nel file di configurazione dell'output PDF.

Il passaggio successivo imposta la copertina, i capitoli descrittivi ed il capitolo profilo d'uso, tutti definiti nel template docbook.

Il terzo passaggio, di notevole importanza, genera la tabella del profilo d'uso percorrendo l'intero albero dell'istanza XML, dalla radice ai nodi foglia, e, per ogni nodo, compie le seguenti operazioni:

- controllo trim: *in caso positivo si passa al nodo successivo.*
- processo nodo: *si inserisce la riga nella tabella del profilo d'uso.*
- controllo Schematron: *in caso positivo si inserisce info sul vincolo.*
- controllo per le statistiche: *in caso positivo si aggiunge il path del ramo nel corrispettivo output delle statistiche.*
- controllo ricorsione: *in caso positivo si scartano i nodi annidati.*

Quest'ultima operazione, *controllo ricorsione*, assicura che, qualora all'interno del ramo corrente sia presente più di una volta lo stesso elemento, questo venga documentato solo alla sua prima occorrenza: si evita in questo modo di incorrere ciclicamente (*loop*) nel medesimo percorso.

Il quarto passaggio aggiunge il capitolo business rules definito nel template docbook ed il quinto ne riporta le regole, gestite e non gestite, presenti nel documento Schematron schema nella sua versione minimale.

Il sesto passo organizza nell'appendice le liste di codici definite nel documento d'input XML; il settimo genera il glossario utilizzando le definizioni dei termini impostate nel documento di configurazione PDF; l'ottavo inserisce la bibliografia specificata nel template docbook.

Il nono passaggio genera l'indice posizionandolo inizialmente in coda al documento PDF, per poi riordinarlo tra la copertina ed il primo capitolo.

L'ultimo passaggio scrive sul disco i documenti statistici: stats required, stats subordinated e stats all.

Componente Schematron handler

Per gestire i file SCH Schematron schema, come si evince dalla figura sottostante (Fig.26), il componente *Schematron handler* esegue una “pipeline” di trasformazioni, sia mediante il linguaggio JAVA sia attraverso fogli di stile XSLT, fino ad arrivare ad una versione minimale del documento SCH stesso. Di seguito saranno esaminati i passi della pipeline.

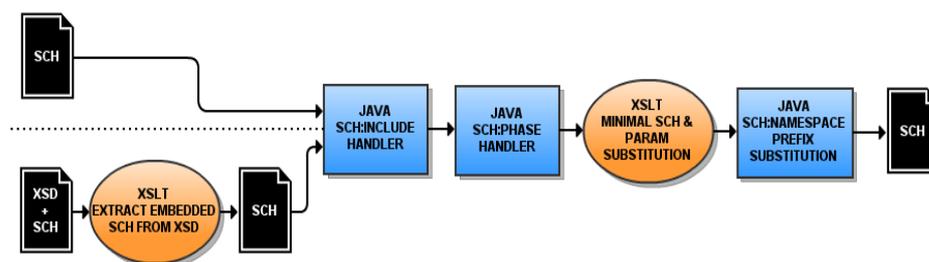


Figura 26: Diagramma pipeline Schematron.

Le regole Schematron possono trovarsi organizzate in un documento SCH separato dal documento XSD oppure inglobate (*embedded*) nel documento XSD: in quest'ultimo caso, prima di iniziare il processo di “minimizzazione” del documento SCH, è necessario estrarre le regole Schematron dal documento XSD, organizzarle in un documento separato e proseguire con i passaggi successivi. Questa trasformazione viene attuata utilizzando il foglio di stile “*ExtractSchFromXSD-2.xsl*” che è disponibile nel package “schematron iso xslt2” reperibile online.

Il passaggio successivo, effettuato utilizzando il linguaggio JAVA, consiste nel risolvere tutte le direttive “sch:include”, eventualmente a loro volta presenti anche nei file inclusi, per ottenere come risultato un unico documento.

Di seguito, sempre utilizzando JAVA, vengono impostate le fasi (sch:phase) nel seguente modo: se nel file di configurazione TASK sono state

specificate delle fasi, allora le fasi diverse da quelle specificate vengono rimosse dal documento SCH; altrimenti, se non è stata specificata neanche una fase, viene impostato il parametro “*phaseDirective=#DEFAULT*” da passare al foglio di stile XSLT per il passaggio successivo.

Dopo l'impostazione delle fasi viene effettuata una trasformazione utilizzando il foglio di stile XSLT denominato “*skeleton.xslt*” il quale si occupa dei seguenti tre aspetti: gestione delle fasi, gestione dei pattern astratti (*abstract pattern*) ed infine sostituzione dei parametri.

La gestione delle fasi nel foglio di stile è realizzata nel modo seguente: se il parametro *phaseDirective* è impostato su “*#DEFAULT*” si controlla il valore dell'attributo *defaultPhase* presente nel tag “*sch:schema*” e, qualora esso sia stato definito, si mantiene la fase indicata eliminando le altre eventuali fasi e/o regole, altrimenti vengono mantenute tutte le fasi e/o regole presenti nel documento; invece, se il parametro *phaseDirective* non è stato impostato, significa che sono state specificate una o più fasi nel file di configurazione TASK e quindi le regole non organizzate in fasi, se presenti, vengono tutte eliminate.

Oltre alla gestione delle fasi, il foglio di stile XSLT provvede a risolvere i pattern astratti ovvero quei pattern che definiscono i valori dei parametri i quali vengono utilizzati nelle espressioni XPath degli *instance pattern*, e provvede ad effettuare la sostituzione dei parametri. L'ultimo passo della *pipeline* consiste, utilizzando il codice JAVA, nel sostituire, in tutte le espressioni XPath definite nel documento Schematron, il valore dell'attributo *prefix* presente nei tag “*sch:ns*” con il valore del prefisso corrispondente definito nel documento XSD.

Dopo aver effettuato tutti i passaggi della *pipeline* descritti, il risultato è un documento Schematron schema minimale che può essere utilizzato nei

successivi moduli dell'applicativo per generare documentazione sulle *business rules* negli output HTML e PDF.

2.3.3 Input

In questa sezione verranno presentati i meccanismi di input utilizzati dall'applicativo per i seguenti scopi:

- configurazione parametri d'input
 - file configurazione TASK
- parametrizzazione documentazione
 - file configurazione HTML
 - file configurazione PDF
- personalizzazione documentazione
 - file template DocBook
- semplificazione su richiesta dei rami dell'albero dell'istanza XML
 - file configurazione TRIM

Inoltre per i file di configurazione task, html, pdf e trim è stato abbinato il relativo XSD per consentire la validazione a runtime.

Nelle cartelle “doc/task/”, “doc/htmlconfig”, “doc/pdfconfig” e “doc/trimconfig”, presenti nel package, si trovano le relative guide, specificate dettagliatamente e generate in modo automatico dall'applicativo proprio a partire dai file XSD stessi.

Invece nella cartella “doc/docbookconfig/”, anch'essa presente nel package, vi è un esempio completo di template docbook nel quale sono riportati e documentati tutti i tag, appartenenti alla specifica docbook, utilizzabili per specificare template docbook personalizzati.

File configurazione TASK

Il file di configurazione TASK è un documento XML mediante il quale è possibile specificare, in modo semplice e completo, una serie di file XSD da processare con le relative opzioni. Il file task può contenere una serie di nodi “singleTask” e “multipleTask”: l'idea di base è quella di permettere, mediante l'uso del *singleTask*, di specificare una configurazione completa nei minimi dettagli per un singolo file XSD; al contrario, mediante l'utilizzo del *multipleTask*, di specificare una sola volta le opzioni di configurazione per un insieme di file XSD, impostando esclusivamente alcuni parametri di configurazione comuni a tutti gli XSD d'input ed utilizzando le opzioni di configurazione di default per i rimanenti parametri.

Il tag “singleTask” si divide in due macro sezioni: “input” e “output”: la sezione input contiene i parametri di input e deve essere definita obbligatoriamente, invece la sezione output contiene le opzioni di configurazione per i file di output da generare e può essere omessa.

Nel tag “input” è possibile specificare le seguenti opzioni:

- xsd [1-1] – *filename dello XSD d'input ed eventuale elemento radice*
- codelist [0-1] – *cartella contenente i file Generic Code, il nome del campo da visualizzare e la grandezza massima del file .gc consentita*
- docbook [0-1] – *filename del file docbook per la personalizzazione dell'output con contenuto esterno*
- trim [0-1] – *filename del file trim contenente i path dell'albero d'istanza da escludere nell'output PDF*
- additionalConstraint [0-1] – *file schematron schema e le fasi da attivare*
- additionalEmbeddedConstraint [0-1] – *flag indicante se documentare i vincoli Schematron embedded*

Nel tag “output” è possibile specificare le seguenti opzioni:

- xml [0-1] – *filename dell'output XML*
- html [0-1] – *filename dell'output HTML ed il filename del file di configurazione HTML*
- pdf [0-1] – *filename dell'output PDF ed il filename del file di configurazione PDF*
- stats [0-1] – *filename degli output per le statistiche*

Il tag “multipleTask” contiene la sola sezione di “input” dove è possibile specificare le seguenti opzioni:

- xsd [1-1] – *cartella contenente i file XSD da processare*
- codelist [0-1] – *cartella contenente i file Generic Code, il nome del campo da visualizzare e la grandezza massima del file .gc consentita*
- additionalEmbeddedConstraint [0-1] – *flag indicante se documentare i vincoli Schematron embedded*

Per tutte le altre opzioni si utilizzano i valori di default e i path costruiti a partire dal filename dello XSD d'input ma con i prefissi diversi a seconda dell'opzione.

Ad esempio, considerando un documento XSD denominato “order.xsd”, vengono cercati nella stessa directory i seguenti file di input:

- order.db – *template docbook*
- order.sch – *documento Schematron schema*
- order_trim.xml – *documento semplificazione rami dell'albero*
- order_html_conf.xml – *configurazione output HTML*
- order_pdf_conf.xml – *configurazione output PDF*

Come output invece vengono utilizzati i seguenti filename:

- *order.xml – output XML*
- *order.html – output HTML*
- *order.pdf – output PDF*
- *order_stats.txt – output stats*
- *order_stats_required.xml – output stats required*
- *order_stats_subordinated.xml – output stats subordinated*

File configurazione output HTML

Il file di configurazione HTML è un documento XML mediante il quale è possibile impostare le opzioni da utilizzare per la generazione dell'output HTML. La struttura del file di configurazione è suddivisa principalmente in tre parti: *costanti della tabella*, *mappatura colori* e *selezione documentazione*.

Nella prima sezione, identificata dal tag “constantsInfo”, è possibile impostare i seguenti parametri della tabella del profilo d'uso:

- *table_second_th_width – larghezza di default della colonna documentata espressa in pixel*
- *info_structured_indent – larghezza dell'annidamento degli elementi dell'albero dell'istanza espressa in pixel*
- *indent_width – larghezza in pixel della spaziatura delle voci nella colonna documentazione*

Nella seconda sezione, identificata dal tag “colorMapping”, è possibile specificare i colori (rgb) da utilizzare nella tabella per identificare più facilmente i seguenti elementi:

- *complex – elementi che hanno all'interno altri elementi e/o attributi*

- base – *elementi che hanno all'interno attributi*
- simple – *elementi che non hanno all'interno né altri elementi né attributi*
- attribute – *nodi attributo*
- wildcard – *nodi wildcard (<any/>)*
- compositor – *nodi compositor come sequence, choice e all*

La terza ed ultima sezione consente di selezionare, mediante l'attributo “make”, fra i seguenti tipi di documentazione quelli da inserire nella colonna *documentation*:

- documentationInfo – *informazioni contenute nel tag “xsd:documentation”*
- facetsInfo – *informazioni sulle facets come “pattern”*
- constraintInfo – *informazioni sugli attributi “fixed” o “default”*
- codelistInfo – *informazioni sulle liste di codici (sia XSD sia Generic Code)*
- derivationInfo – *informazioni di derivazione (“extension” e “restriction”)*
- advancedInfo – *informazioni sugli attributi “block”, “final”, “abstract”*
- wildcardInfo – *informazioni sugli attributi “namespace” e “processContents”*

Il tag “documentationInfo” rappresenta la parte principale del file di configurazione: infatti esso consente di attuare, con una granularità fine, la selezione dei singoli tag, appartenenti ad esempio al vocabolario CCTS, presenti all'interno del nodo “xsd:documentation”.

File configurazione output PDF

Il file di configurazione PDF è un documento XML mediante il quale è possibile impostare le opzioni da utilizzare per la generazione dell'output PDF. La struttura del file di configurazione è suddivisa principalmente in quattro parti: *costanti della tabella*, *larghezza delle colonne*, *mappatura colori e selezione documentazione*.

Nella prima sezione, identificata dal tag “constantsInfo”, è possibile impostare i seguenti parametri della tabella del profilo d'uso:

- *flush* – il numero modulo che regola la scrittura parziale sul disco
- *indentFactor* – il numero di spazi bianchi da usare per l'annidamento degli elementi dell'albero dell'istanza
- *indentChar* – il carattere da usare per l'incolonnamento

Il tag “flush” è un parametro che regola le *performance* dell'applicativo durante la costruzione dell'albero dell'istanza XML: esso stabilisce il numero massimo di righe che si possono inserire nella tabella del profilo d'uso prima di riversare parte di essa dalla memoria centrale al disco, in modo tale da poter gestire tabelle di grandi dimensioni senza occupare tutta la memoria disponibile.

Nella seconda sezione, identificata dal tag “tableWidth”, è possibile adattare le dimensioni in larghezza delle tre colonne della tabella:

- *indentWidth* – la larghezza della colonna “livello annidamento”
- *tagWidth* – la larghezza della colonna “tag”
- *documentationWidth* – la larghezza della colonna “documentation”

Le dimensioni della tabella sono espresse in *pt* e la larghezza totale, essendo la pagina nel formato A4 (595pt) e considerando i margini di sinistra

e destra (2*25pt), non deve superare i 545 pt.

Nella terza sezione, identificata dal tag “colorMapping”, è possibile specificare i colori (rgb) da utilizzare nella tabella per identificare facilmente i seguenti elementi:

- *complex – elementi che hanno all'interno altri elementi e/o attributi*
- *simple – elementi che non hanno all'interno altri elementi*
- *attribute – nodi attributo*
- *wildcard – nodi wildcard (<any/>)*
- *compositor – nodi compositor come sequence, choice e all*

La quarta ed ultima sezione consente di selezionare, mediante l'attributo “make”, fra i seguenti tipi di documentazione, quelli da inserire nella colonna *documentation*:

- *documentationInfo – informazioni contenute nel tag “xsd:documentation”*
- *facetsInfo – informazioni sulle facets come “pattern”*
- *constraintInfo – informazioni sugli attributi “fixed” o “default”*
- *codelistInfo – informazioni sulle liste di codici (sia XSD sia Generic Code)*
- *derivationInfo – informazioni di derivazione (“extension” e “restriction”)*
- *advancedInfo – informazioni sugli attributi “block”, “final”, “abstract”*
- *wildcardInfo – informazioni sugli attributi “namespace” e “processContents”*
- *additionalConstraintInfo – informazioni sulle business rules*
- *pathInfo – informazione sul canonical path*

Come visto per il file di configurazione HTML nella precedente sezione, il tag “documentationInfo” rappresenta la parte principale del file di configurazione: infatti esso consente di attuare, con una granularità fine, la selezione dei singoli tag, appartenenti ad esempio al vocabolario CCTS, presenti all'interno del nodo “xsd:documentation”.

Template DocBook

Il template DocBook, come specificato nella sezione 2.2.7, è un documento XML mediante il quale è possibile personalizzare la guida.

Il template consente di definire le seguenti parti:

- copertina [1-1]
- label per appendice, indice e glossario [0-1]
- capitoli descrittivi [0-N]
- capitolo profilo d'uso [1-1]
- capitolo business rules [0-1]
- bibliografia [0-1]

La parte del template docbook relativa alla copertina consente di definire i seguenti elementi:

1. logo [0-N]
2. informazioni progetto [0-1]
 - acronimo [0-1]
 - nome/descrizione [0-N]
 - link [0-N]
 - altro [0-N]
3. titolo guida [1-1]
4. lista informazioni sulla guida [0-1]

- voce informazione sulla guida [1-N]
 - nome [1-1]
 - valore [1-1]

La copertina quindi è costituita da una o più immagini, da informazioni sul progetto come per esempio l'acronimo, le frasi descrittive, i link a pagine web ed altro contenuto addizionale, dal titolo della guida ed infine da una lista contenente coppie nome/valore utili per specificare dettagli come il numero di versione, la data di pubblicazione, il nome del file del documento ed altro.

Dopo la copertina è possibile specificare le etichette per l'appendice, l'indice ed il glossario da utilizzare negli output HTML e PDF. Se non specificate, assumono rispettivamente i valori “Appendix”, “Table Of Content” e “Glossary”.

Definita la copertina, è possibile specificare uno o più capitoli descrittivi che possono essere utilizzati per inserire del contenuto a proprio piacimento. Il capitolo descrittivo presenta i seguenti elementi comuni a tutti i capitoli:

1. titolo [1-1]
2. paragrafo [0-N]
3. immagine [0-N]
4. sottocapitolo [0-N]
 - titolo [1-1]
 - paragrafo [0-N]
 - immagine [0-N]

Il capitolo profilo d'uso ricalca la struttura del capitolo descrittivo; l'unica differenza è che alla fine presenterà la tabella del profilo d'uso generata a

partire dallo XSD e quindi l'elemento:

5. tabella del profilo d'uso [1-1]

L'elemento *tabella del profilo d'uso*, seppur obbligatorio, è puramente indicativo: esso rappresenta una sorta di segnaposto per la tabella che verrà generata automaticamente utilizzando lo XSD.

Il capitolo business rules può essere utilizzato per specificare contenuto extra prima delle tabelle delle regole Schematron. La struttura del capitolo business rules è la stessa dei capitoli descrittivi con in più gli elementi:

5. tabella regole gestite [1-1]
6. tabella regole non gestite [1-1]

Anche in questo caso gli elementi *tabella regole gestite* e *tabella regole non gestite* sono dei semplici segnaposti per le tabelle che verranno generate a partire da documenti Schematron schema. All'interno dell'elemento *tabella regole non gestite* è possibile specificare l'etichetta da utilizzare per la tabella delle regole non gestite.

Dopo la copertina, i capitoli descrittivi, il capitolo profilo d'uso ed il capitolo business rules è possibile definire la parte inerente la bibliografia. La specifica del docbook mette a disposizione molti elementi per organizzare la bibliografia nel modo più completo possibile; tuttavia nel *template* si è scelto di utilizzare solo un set minimo di tag in quanto risulta sufficientemente adeguato per bibliografie da inserire nelle guide.

Gli elementi che possono essere definiti nella bibliografia sono i seguenti:

1. titolo etichetta bibliografia [1-1]
2. voce bibliografica [1-N]
 - abbreviazione [1-1]

- gruppo autori [1-1]
 - autore [1-N]
 - ◆ nome [1-1]
 - ◆ cognome [1-1]
- titolo [1-1]
- testo libero [0-1]
- risorse esterne [0-1]
 - link [1-N]

Nella bibliografia è possibile specificare una o più voci bibliografiche, ciascuna di esse costituita da: l'abbreviazione utilizzata, i nomi di uno o più autori, il titolo della fonte bibliografica, testo semplice contenente informazioni non classificate secondo la specifica del docbook ed infine risorse esterne quali link a pagine web.

All'interno dei paragrafi situati sia nei capitoli che nelle sezioni presenti nei capitoli descrittivi, tabella del profilo d'uso e business rules, è possibile inserire del testo formattato in grassetto o in corsivo. La specifica docbook consente di mettere in rilievo una parte di testo attraverso il tag “*emphasis*”; si è scelto di utilizzare il tag *emphasis* con l'attributo `role="bold"` per indicare il testo in grassetto e con l'attributo `role="italic"` per indicare il corsivo.

Configurazione TRIM

Il file XML “*TRIM*” consente di specificare, mediante una sintassi XPath semplificata e motivata nella sezione 2.2.3, un elenco di path, da non includere nell'output PDF. La struttura del file TRIM è molto semplice: in essa troviamo unicamente il nodo radice “*trim*” con all'interno una sequenza di tag “*p*” contenenti l'espressione XPath semplificata che indica l'elemento o l'attributo da rimuovere dall'albero dell'istanza XML.

2.3.4 Output

In questa sezione verranno esaminati i seguenti output generati dall'applicativo:

- XML
- HTML
- PDF
- STATISTICHE

Output XML

L'output XML, a differenza di quelli HTML e PDF, rappresenta la versione *machine readable* della guida priva però del contenuto extra e delle business rules: infatti esso viene utilizzato come input per generare la tabella del profilo d'uso da inserire nella guida negli altri due formati.

La struttura dell'output XML è suddivisa prettamente in due blocchi: *header* e *body*. L'header è stato pensato per raggruppare tutte le informazioni aggiuntive e riepilogative relative al documento XSD, come ad esempio il filename, il target namespace e l'identificativo della versione. Il body è stato pensato per contenere tutte le strutture dati necessarie per generare l'albero dell'istanza XML e per documentare correttamente le liste di codici.

A sua volta la sezione body presenta tre sottosezioni interne denominate rispettivamente: *tree*, *libraries* e *codelists*. La sezione tree indica l'elemento radice e rappresenta il punto di partenza per costruire l'albero dell'istanza XML; la sezione libraries organizza, raggruppando a seconda del namespace d'appartenenza, i tipi e gli elementi che potranno essere utilizzati per assemblare dinamicamente l'albero dell'istanza XML; infine la sezione codelists contiene tutte le liste di codici processate.

Output HTML

L'output HTML rappresenta la versione *ibrida* della guida, a metà tra quella *machine readable*, XML, e quella *human readable*, PDF.

La guida nel formato HTML racchiude sia l'efficienza e la praticità d'uso, derivanti dal fatto che la struttura dell'albero dell'istanza XML viene costruita dinamicamente, sia la completezza d'informazione, poiché comprende il contenuto extra espresso nel template docbook e le Business rules definite nel documento Schematron schema.

Inoltre, grazie alla modalità dinamica, l'output HTML consente di percorrere tutto l'albero dell'istanza XML proseguendo anche nei punti di loop, ovvero quei nodi che sono già stati attraversati almeno una volta.

La struttura della guida nel formato HTML si compone dei seguenti elementi:

- copertina
- indice
- capitoli descrittivi
- capitolo profilo d'uso
 - tabella del profilo d'uso
- capitolo business rules
 - tabella business rules
- bibliografia

Output PDF

L'output PDF rappresenta la versione *human readable* per eccellenza della guida. Infatti la guida viene presentata con una veste grafica sobria ed elegante e quindi adatta all'utilizzo in contesti professionali.

A differenza della tabella del profilo d'uso presente nell'output HTML,

quella nell'output PDF viene rappresentata per intero e questo, per istanze molto grandi, comporta sia tempi lunghi per generarla sia un output di notevoli dimensioni. Per ovviare a ciò, è possibile, mediante l'uso del file di configurazione trim, semplificare alcuni rami dell'albero in modo da rendere l'output PDF più leggero.

La struttura della guida nel formato PDF si compone dei seguenti elementi:

- copertina
- indice
- capitoli descrittivi
- capitolo profilo d'uso
 - tabella del profilo d'uso
- capitolo business rules
 - tabella business rules regole linkate all'albero
 - tabella business rules regole non linkate all'albero
- appendice codelist
- glossario
- bibliografia

Output Statistiche

L'output delle statistiche consiste in un set di quattro file che sono rispettivamente *stats*, *stats required*, *stats subordinated* e *stats all*, contenenti sia informazioni generali riepilogative sul task completato (*stats*) e sia informazioni più tecniche relative all'albero dell'istanza XML creato durante la generazione dell'output PDF (*required*, *subordinated* e *all*).

L'output *stats* è un file di testo TXT nel quale vengono riportate informazioni organizzate in quattro blocchi.

Nel primo blocco vi sono i tempi impiegati per la generazione degli output XML, HTML e PDF espressi nel formato h:m:s e le dimensioni in KB dei relativi file di output.

Nel secondo blocco è possibile trovare il conteggio dei componenti processati durante la creazione dell'albero dell'istanza XML come attributi, elementi foglia (cioè elementi che non contengono altri elementi), elementi non foglia (ovvero elementi che contengono altri elementi) ed infine il loro totale.

Nel terzo blocco sono riportati i nodi, sia elementi che attributi, *required* e *subordinated*. Un nodo è *required* se ha sia cardinalità minima obbligatoria (o *use=required* se è un attributo) e sia tutti i nodi *ancestor* con cardinalità minima obbligatoria. Un nodo è *subordinated* se ha sia cardinalità minima obbligatoria (o *use=required* se è un attributo) e sia almeno un nodo *ancestor* con cardinalità minima opzionale.

Nel quarto ed ultimo blocco sono riportati i valori *loop point* e *max nested level* che rappresentano rispettivamente il conteggio totale dei punti in cui è stata gestita la ricorsione (es. /Invoice/Signature/Agency/Agency) ed il massimo livello di profondità raggiunto durante la costruzione dell'albero dell'istanza XML.

Gli output *stats required*, *stats subordinated* e *stats all* sono file XML contenenti tutti i path rispettivamente dei nodi *required*, dei nodi *subordinated* e di tutti i nodi, *required* e *subordinated*, processati durante la costruzione dell'albero dell'istanza XML.

Di base l'output *stats all* non viene abilitato perché, se l'istanza presenta molti nodi, produce un file di grandi dimensioni; quindi, affinché questo output venga generato, è necessario che tale funzionalità venga espressamente specificata nel file di configurazione task.

La struttura di questi tre file XML risulta essere simile e molto semplice: infatti essa è costituita solo dal nodo radice, rispettivamente “required”, “subordinated” e “all”, contenente una serie di nodi “p” (path) i quali presentano all'interno il *canonical path* del nodo in questione. Inoltre i nodi “p” possono avere l'attributo “alt” il quale indica che si è in presenza di una scelta tra rami differenti dell'albero. I nodi “p” che presentano l'attributo “alt” sono sia i nodi presenti direttamente nei compositor “xsd:choice” sia i nodi annidati che derivano dalla loro scelta.

2.4 Considerazioni finali

La fase di progettazione è stata portata avanti di pari passo con la fase d'implementazione e quindi di testing, in quanto a volte è accaduto che una scelta implementativa, la quale sembrava lineare ed ottimale, all'atto dell'implementazione, a causa delle tecnologie adottate, non si rivelava funzionale e pertanto bisognava cercare soluzioni alternative.

Nel prossimo capitolo verranno esaminati i componenti dell'applicativo, descritti finora a livello progettuale, nel loro dettaglio implementativo.

Capitolo 3. Implementazione

In questo capitolo sarà discussa l'implementazione dei componenti visti nella sezione 2.3 considerando in particolare gli aspetti più rilevanti ed infine verranno analizzati gli output prodotti.

3.1 Implementazione applicativo XDG

L'applicativo XDG è stato implementato utilizzando il linguaggio JAVA con l'ausilio delle librerie EMF XSD, iText e SAXON, tre fogli di stile XSLT, quattro documenti XSD di validazione ed infine tre file XML per le configurazioni di default. Si procede con l'analisi delle seguenti parti:

- classe principale
- classi ausiliarie
- fogli di stile XSLT
- file configurazioni di default
- documenti di validazione

3.1.1 Classe principale

La classe principale dell'applicativo XDG è *XDocumentationGenerator*, la quale definisce l'unico metodo pubblico *configurationParse()*, una serie di metodi privati utilizzati per gestire lo Schematron, gli output XML, HTML e PDF, ed alcuni metodi di *utility*.

Task Handler

Per gestire e lanciare i task, con le relative opzioni, definiti nel file di configurazione TASK, l'applicativo XDG espone l'unico metodo pubblico *configurationParse()*, il quale procede nel modo seguente:

1. imposta file e console logger
2. per ogni task presente nel file di configurazione
 - caso *singleTask*
 - analisi configurazione INPUT e OUTPUT e validazione dei file di configurazione
 - esecuzione task
 - caso *multipleTask*
 - per ogni documento XSD presente nella directory
 - analisi INPUT e validazione dei file di configurazione
 - esecuzione task

Nel primo punto viene creato un oggetto *Formatter*, ossia un gestore della formattazione del testo, sia per personalizzare i messaggi a seconda del livello di severità *info*, *warning* e *severe*, sia per visualizzare le medesime informazioni di log in entrambi gli output: sul file e su console.

Nel secondo punto, dopo l'analisi dei parametri di input/output e la validazione dei documenti di configurazione, si esegue il task che consiste nel lanciare le seguenti operazioni, descritte nelle sezioni successive:

- genera output XML – *XML-Schema Analyzer*
- gestisce eventuale documento Schematron – *Schematron Handler*
- genera output HTML – *XML2HTML*
- genera output PDF – *PDF&stats*
- genera output STATISTICHE stats – *utilityLogStats()*

XML-Schema Analyzer

Il componente XML Schema analyzer, descritto nella sezione 2.3.2, è stato realizzato con l'uso della libreria EMF XSD [DFME08] che consente di implementare, in modo semplice ed efficace, i metodi qui elencati e raggruppati in base alle funzionalità svolte:

1. metodo principale:
xmlCreate()
2. gestione risorse XSD:
xmlGetSchemaResourceSet(), xmlGetSchemaResource()
3. gestione documentation, facet e codelist:
xmlDoInfo(), xmlHandlerFacets(), xmlHandlerCodeList()
4. creazione albero dell'istanza XML:
*xmlHandlerAttributeUse(), xmlHandlerElementDeclaration(),
xmlHandlerTypeDefinition(), xmlHandlerCompositor(),
xmlNormalizeRedefineNSid()*
5. gestione documenti Generic Code:
xmlGetGenericCode()

Il metodo principale *xmlCreate()* genera l'output intermedio XML implementando i passaggi descritti nella sezione 2.3.2 figura 23, utilizzando i metodi descritti nei punti successivi. Di seguito vengono presentati i passi principali di questo metodo:

- organizza le risorse XSD – *xmlGetSchemaResourceSet()*
- ottieni XSD principale – *xmlGetSchemaResource()*
- analizza tutti i tipi – *xmlHandlerTypeDefinition()*
- analizza tutti gli elementi – *xmlHandlerElementDeclaration()*
- associa i Generic Code

In particolare nel metodo *xmlCreate()* si provvede ad associare i GC ai nodi dell'albero nel modo seguente:

- ottieni mappa GC – *xmlGetGenericCode()*
- vengono prelevati tutti i tipi che derivano “udt:CodeType”
- per ogni tipo ...
 - prova ad associare il corrispettivo GC nel modo seguente:
 - se è definito l'attributo “listURI” usalo per identificare il GC
 - altrimenti prova con il nome del type (senza il suffisso Type)
 - se il generic code è presente nella mappa GC
 - se rientra nelle dimensioni consentite aggiungilo nello XML
 - altrimenti inserisci solo un collegamento

I metodi appartenenti al secondo punto (*gestione risorse XSD*), utilizzando la struttura dati *ResourceSet* fornita dalla libreria EMF, gestiscono le risorse XSD in modo da semplificare la lettura e la gestione dello schema principale ed eventualmente di quelli inclusi, importati e ridefiniti. In particolare il metodo *xmlGetSchemaResource()* restituisce il documento XSD principale.

I metodi appartenenti al terzo punto (*gestione documentation, facet e codelist*) si occupano rispettivamente della gestione dei nodi e del testo che possono presentarsi all'interno del tag “xsd:annotation”, della gestione di tutte le facet escluse le *xsd:enumeration*, ed infine della gestione dei codelist espressi con la facet *xsd:enumeration*, tutte le volte che si presentano durante l'analisi del documento XSD.

Nel quarto punto (*creazione albero dell'istanza XML*) sono raggruppati tutti i metodi utilizzati per gestire la struttura dell'albero dell'istanza XML: gli attributi, gli elementi globali, i tipi semplici e complessi ed infine i

compositor. Durante l'analisi dei componenti XML-Schema si procede ad esaminarli per poi creare i relativi nodi all'interno dell'output intermedio XML. In particolar modo il metodo *xmlHandlerTypeDefinition()* è quello più complesso: esso, procedendo ricorsivamente, analizza i tipi semplici e complessi e li documenta in base ad una flag denominata *WalkDocumentation*; questa flag specifica la modalità di ricorsione: se impostata a FALSE, nell'output XML, viene creato un nuovo nodo nel quale inserire le informazioni. Di seguito vengono presentati i passi principali effettuati dall'algoritmo (in pseudo codice):

```
function typexmlHandlerTypeDefinition() {
  if( simpleType ) {
    if( WalkDocumentation == FALSE ) {
      CreaNodo();
    }
    ottieni_informazioni(); // final,annotation, facet, enumeration
    if( restriction @base ) {
      if( @base != builtin type )
        xmlHandlerTypeDefinition( ); // WalkDocumentation=TRUE
    }
  }
  else { // complexType
    if( WalkDocumentation == FALSE ) CreaNodo();

    ottieni_informazioni(); // mixed,block,final,abstract
    switch( type ) {
      case EMPTY:
        ottieni_informazioni(); // annotation
      case SIMPLE:
        ottieni_informazioni(); // annotation
        documenta_derivation( @base );
        typexmlHandlerTypeDefinition(); // WalkDocumentation=TRUE
      case ELEMENT_ONLY e MIXED:
        ottieni_informazioni(); // annotation
        for( particle : particles ) xmlHandlerCompositor( particle );
    }
    xmlHandlerAttributeUse(); // documenta attributi
  }
}
```

Ogni tipo, elemento ed attributo, per essere catalogato, necessita di un identificativo: il metodo *xmlNormalizeRedefineNSid()* si occupa di associare un id corretto per i componenti ridefiniti all'interno della direttiva *xsd:redefine*. La libreria EMF XSD gestisce le direttive “*xsd:redefine*” riportando, in duplice copia, tutti i componenti del relativo XSD: una copia

contenente i componenti originali così come sono ed un'altra copia contenente tutti i componenti fra i quali quelli ridefiniti al posto degli originali. Per distinguere a quale copia un componente appartiene, il metodo *xmlNormalizeRedefineNSid()* procede esaminando le *referencing directive* ovvero i riferimenti inversi a librerie che importano, includono o ridefiniscono quella in esame: il componente appartenente alla copia “ridefinita” avrà una sola direttiva inversa *redefine*.

Infine nel quinto punto vi è il metodo *xmlGetGenericCode()* che organizza i documenti Generic Code, creando una mappa nome-valore, dove il nome è il campo “*gc:ShortName*” definito nel file (.gc) ed il valore è il pathname del file (.gc).

Schematron Handler

La gestione delle Business rules, espresse mediante il linguaggio Schematron, come esposto nella sezione 2.3.2 *Componente Schematron*, è stata implementata mediante i metodi *schematronExtractEmbeddedXSD()* e *schematronParse()*; quest'ultimo, a sua volta, si avvale del metodo ausiliario *schematronHandlerInclude()*.

Il metodo *schematronExtractEmbeddedXSD()* ha il compito di estrarre, mediante lo stylesheet *ExtractSchFromXSD-2.xsl*, le regole Schematron, eventualmente definite all'interno dei documenti XSD, e di organizzarle in un file esterno da processare in un secondo momento. L'algoritmo è costituito dai seguenti passi:

1. crea file temporaneo nel quale inserire le regole estratte
2. setta opzioni del trasformatore XSLT 2.0
 - imposta lo stylesheet
 - effettua la trasformazione
3. controlla se il file Schematron temporaneo è vuoto.

Lo stylesheet `ExtractSchFromXSD-2.xsl` definisce nel file temporaneo il nodo radice “sch:schema” a prescindere dalla presenza o meno di regole trovate all'interno del documento XSD; quindi nel punto tre si controlla il contenuto del file Schematron temporaneo verificando la presenza di almeno un tag “sch:ns” e di un nodo “sch:phase” o “sch:pattern”.

Il metodo `schematronParse()`, partendo da un documento Schematron specificato esternamente nel file di task oppure estratto dallo XSD mediante il metodo `schematronExtractEmbeddedXSD()`, crea un documento XML contenente la versione minimale del file Schematron d'input, dove per minimale s'intende il documento Schematron nel quale sono state gestiti, se presenti, le inclusioni, le fasi e i pattern astratti; tale documento Schematron minimale verrà utilizzato in seguito, dagli altri componenti, per documentare le business rules all'interno degli output HTML e PDF. L'algoritmo è il seguente:

1. crea file temporaneo per Schematron minimale
2. gestisci inclusioni “sch:include” – `schematronHandlerInclude()`
3. gestisci fasi “sch:phase”
4. effettua la trasformazione mediante lo stylesheet `skeleton.xsl`
5. controlla coerenza tra prefissi “sch:ns” e “xsd:schema/@ns”
 - eventualmente sostituisci prefissi

Il file temporaneo Schematron minimale, creato nel primo punto, viene memorizzato nella *default temporary-file directory* in base al sistema operativo utilizzato (ad esempio nella cartella `/tmp` nei sistemi GNU/Linux) per essere rimosso all'uscita dall'esecuzione dell'applicativo (proprietà `deleteOnExit`).

Nel secondo punto si procede a gestire le direttive Schematron `include` mediante il metodo `schematronHandlerInclude()`, il quale ha il compito di

sostituire, ricorsivamente, le direttive “sch:include” con il contenuto dei documenti indicati nell'attributo @href. Per fare ciò si eseguono i seguenti passi:

1. si ottengono tutte le inclusioni “sch:include”
2. per ogni inclusione presente
 - si ottiene il path del documento dall'attributo “@href”
 - ricorsivamente chiama il metodo *schematronHandlerInclude()*
 - dopo aver percorso tutte le direttive, a ritroso, vengono sostituiti i nodi “sch:include” con il contenuto presente nei documenti

È da notare che l'algorithm descritto non gestisce la ricorsione circolare poiché la direttiva “sch:include” non genera tale problema in quanto il *content-model* degli elementi Schematron, gestiti dall'applicativo XDG, non presentano la ricorsione nel loro interno.

Nel terzo punto vengono gestite le fasi (sch:phase), eventualmente attivate, da documentare ed in particolare viene impostato il valore del parametro *phaseDirective* da passare al trasformatore che, come si evince nel quarto punto, esegue la trasformazione XSLT 2.0 utilizzando lo stylesheet *skeleton.xsl*, il quale si occupa in particolare della sostituzione dei parametri in presenza di pattern astratti.

Il quinto punto è quello più delicato in quanto si provvede ad accordare, qualora differenti, i prefissi definiti nel documento Schematron, all'interno dei tag “sch:ns”, con quelli dichiarati nel documento XSD, in base al valore del namespace URI.

Questa operazione è necessaria per consentire una corretta associazione dei link delle business rules nei nodi dell'albero dell'istanza XML.

Si procede in questo modo:

1. prepara hashmap <@prefix,@uri> dei nodi “sch:ns” (hm_sch) e dei nodi “xsd:schema/@ns” (hm_xsd)
2. per ogni elemento dello hm_sch, controlla:
 - se l'@uri dell'elemento corrente è presente anche nello hm_xsd ...
 - usando il @prefix dell'elemento corrente prova ad ottenere l'@uri di un elemento presente nello hm_xsd con lo stesso @prefix di quello corrente
 - se i due elementi hanno stesso @prefix ma @uri diversi ...
 - se il @prefix dell'elemento corrente non è vuoto ...
 - ciclando nello hm_xsd trova un elemento con lo stesso @uri dell'elemento corrente
 - se trovato, marca il prefisso per la sostituzione
 - altrimenti il @prefix è vuoto, quindi non sarà possibile effettuare la sostituzione
 - altrimenti @prefix e @uri coincidono:nessuna sostituzione
 - altrimenti avvisa con un *warning* e prosegui
3. sostituisci tutti i prefissi nelle espressioni degli attributi @context e @test

XML2HTML

Il metodo *htmlCreate()* implementa il componente xml2html, presentato nella sezione 2.3.2. Tale metodo esegue una trasformazione XSLT 2.0 [MK08] tramite il foglio di stile xml2html.xsl utilizzando la libreria SAXON ed effettua i seguenti passaggi:

1. imposta parametri di trasformazione da passare allo stylesheet
 - (filename) file di configurazione HTML
 - (filename) template docbook
 - nome del campo da visualizzare per i generic code
 - (filename) documento Schematron
2. effettua la trasformazione

Nel primo punto, per assicurare un corretto funzionamento anche su piattaforme Windows, si è reso necessario adattare i path dei file, come ad esempio quelli dei file di configurazione html e il template docbook, convertendoli in URI.

Nel secondo punto è stato usato un oggetto *streamResult* come file di output della trasformazione in modo da supportare efficientemente eventuali documenti HTML di grandi dimensioni, anche dell'ordine di 20MB.

PDF&stats

Il componente PDF&stats, descritto nella sezione 2.3.2, è stato realizzato con l'uso della libreria iText [BL10] la quale consente, in modo semplice e programmatico, di creare documenti PDF anche di grandi dimensioni, senza richiedere un eccessivo consumo di risorse.

Inizialmente si era scelto di generare l'output PDF utilizzando le tecnologie XSLT-FO e APACHE-FOP; ma queste, anche se in fase di progettazione si erano presentate idonee, in fase di implementazione e consequenziale fase di test, applicate a documenti molto grandi, si sono poi rivelate non adatte.

Di seguito vengono presentati, raggruppati in base alle funzionalità svolte, i metodi che sono stati definiti per realizzare sia l'output PDF sia gli output delle statistiche required, subordinated e all, ponendo l'attenzione esclusivamente sulle parti più significative.

1. metodo principale:
pdfCreate()
2. utility:
pdfGetConf(), *pdfBinaryToRoman()*
3. gestione documentazione:
pdfDocumentationInfo(), *pdfCCTSInfo()*, *pdfFacetstInfo()*,

pdfFacetInfo(), *pdfConstraintInfo()*, *pdfCodelistInfo()*,
pdfDerivationInfo(), *pdfAdvancedInfo()*, *pdfWildcardInfo()*,
pdfDoInfo()

4. gestione trim e business rules:

pdfTrim(), *pdfCoconsDoc()*, *pdfReplacePredicate()*,
pdfSchematronXPathSanity(), *pdfCheckPath()*,
pdfSchematronMapContext(), *pdfSchematronTable()*

5. gestione albero:

pdfWalk(), *pdfDoIndent()*, *pdfTableFlush()*

6. gestione docbook:

pdfDoCoverAndChapterDescriptionDocBook(),
pdfDoAdditionalConstraintDocBook(), *pdfDoBibliographyDocBook()*,
pdfDoChapterTextDocBook(), *pdfGetNumberRow()*,
pdfGetNumberColumn()

7. creazione statistiche:

pdfLogStats(), *pdfPathAlternativeCheck()*

8. gestione label:

pdfGetIndexLabel(), *pdfGetAppendixLabel()*, *pdfGetGlossyLabel()*,
pdfGetSchematronUnhandledLabel()

Il metodo principale *pdfCreate()* ha il compito di generare l'output PDF attraverso i seguenti passi:

- crea report pdf, imposta opzioni tabella del profilo d'uso
- ottieni impostazioni di configurazione – *pdfGetConf()*
- crea mappa path trim
- crea mappa business rules – *pdfSchematronMapContext()*
- crea cover e capitoli – *pdfDoCoverAndChapterDescriptionDocBook()*
- popola la tabella del profilo d'uso – *pdfWalk()*
- crea capitolo business rules – *pdfDoAdditionalConstraintDocBook()*
- ottieni label regole non linkate – *pdfGetSchematronUnhandledLabel()*

- crea tabelle business rules – *pdfSchematronTable()*
- ottieni label appendice – *pdfGetAppendixLabel()*
- crea appendice/liste di codice
- ottieni label glossario – *pdfGetGlossyLabel()*
- crea glossario
- crea bibliografia – *pdfDoBibliographyDocBook()*
- ottieni label indice – *pdfGetIndexLabel()*
- crea indice
- genera statistiche – *pdfLogStats()*

Nel metodo *pdfCreate()* vengono specificate due opzioni importanti: *tabella in modalità parziale e ordinamento non gerarchico delle pagine*.

L'esigenza della tabella in modalità parziale nasce dal fatto che un oggetto, solo dopo che sia stato completato, può essere inserito nel documento PDF, causando un elevato consumo di memoria. Per risolvere questo problema la libreria iText mette a disposizione l'interfaccia *LargeElement*. Come si evince dall'esempio di codice sottostante, per utilizzare questa modalità, sono necessari tre passaggi: dichiarare la tabella non ancora completa (1), aggiungere parte delle tabella incompleta (2), impostare tabella completa ed aggiungerla nel documento PDF (3).

```
Document document = new Document(); // crea documento PDF
PdfWriter.getInstance(document, new FileOutputStream(filename));
document.open();
PdfPTable table = new PdfPTable( numColumn ); // crea tabella
table.setComplete(false); // 1) IMPOSTA TABELLA INCOMPLETA
int count = 0;
for( PdfPCell c : cells ) {
    // aggiungi righe alla tabella
    table.addCell( c );
    if( (count++ %10) == 0 )
        document.add( table ); // 2) RIVERSA PARTE TABELLA DALLA RAM → HDD
}
table.setComplete( true ); // 3) IMPOSTA TABELLA COMPLETA
// aggiungi la tabella al documento PDF
document.add( table );
document.close();
```

Per spostare l'indice dalla fine all'inizio del documento, occorre specificare alla libreria iText di gestire l'elenco delle pagine in modalità lineare invece che come un albero (1), ottenere l'array contenente l'ordinamento delle pagine (2), riposizionare le pagine (3) ed infine effettuare il riordino delle pagine (4).

```
// 1) IMPOSTA MODALITA' LINEARE INVECE CHE GERARCHICA
writer.setLinearPageMode();
// 2) ottieni il totale delle pagine e l'array contenente le posizioni delle pagine
pdf.newPage();
int total = writer.reorderPages(null);
int[] order = new int[total];
// 3) riposiziona le pagine del documento PDF ...
// 4) EFFETTUA IL RIORDINO DELLE PAGINE
writer.reorderPages(order);
```

Tale operazione è necessaria poiché l'indice può essere generato solo dopo aver scritto tutto il documento, in quanto il numero delle pagine relativo ai capitoli, sezioni, appendice e glossario non è noto a priori.

Ritornando alla lista dei metodi definiti, il secondo punto (*utility*) comprende due metodi ausiliari: *pdfGetConf()* recupera le impostazioni di configurazione del PDF e *pdfBinaryToRoman()* converte i numeri decimali nella numerazione in romano.

Nel terzo punto (*gestione documentazione*) vi sono tutti i metodi utilizzati per inserire le informazioni nella colonna “documentation” della tabella del profilo d'uso. In particolar modo il metodo *pdfDoInfo()* si comporta come un dispatcher ovvero provvede a chiamare, a seconda della tipologia di informazione, il corrispettivo metodo. Gli altri metodi procedono nel creare ed impostare un elemento *chunk*, da inserire nella tabella, nel quale porre l'informazione.

Al quarto punto (*gestione trim e business rules*) appartengono i metodi utilizzati per documentare le business rules e per eliminare selettivamente rami dall'albero dell'istanza XML:

per ogni *step*, si verifica se i nodi di entrambe le espressioni individuano il medesimo elemento. In caso positivo è possibile associare la regola business rules nell'albero dell'istanza XML.

Nel quinto punto (*gestione albero*) sono presenti i metodi per la gestione dell'albero dell'istanza. Il metodo *pdfWalk()* ricorsivamente costruisce la tabella del profilo d'uso, seguendo i riferimenti dei componenti organizzati in librerie nell'output intermedio XML, come sarà presentato nella sezione 3.2.1 *Attraversamento dell'albero dell'istanza*. Il metodo *pdfDoIndent()* consente di rappresentare il livello di annidamento dei nodi inserendo, a partire dal margine sinistro della tabella, tanti spazi bianchi quanti sono i livelli di annidamento, intervallati da un carattere (ad esempio “.”). L'ultimo metodo, *pdfTableFlush()*, riversa la tabella parziale dalla RAM al disco, utilizzando la funzione della libreria iText *add (report.add(table))*: per ottenere questo comportamento è necessario che, all'atto della creazione della tabella, sia stata specificata l'opzione *table.setComplete(false)* la quale appunto indica che la tabella non è completa e quindi è possibile chiamare più volte il metodo *add* per inserire le righe finora aggiunte.

I metodi del sesto punto (*docbook*) gestiscono la parte relativa al template docbook. I primi tre metodi, *pdfDoCoverAndChapterDescriptionDocBook()*, *pdfDoAdditionalConstraintDocBook()*, *pdfDoBibliographyDocBook()*, hanno il compito rispettivamente di creare la copertina e i capitoli descrittivi, la tabella delle business rules ed infine la parte relativa alla bibliografia. Gli ultimi tre metodi invece vengono usati per gestire il testo e gli elenchi puntati (*pdfDoChapterTextDocBook*) e per stabilire la disposizione degli eventuali loghi nella copertina (*pdfGetNumberRow*, *pdfGetNumberColumn*). Per gestire in modo preciso e ottimale la copertina, è stata usata una tabella divisa in tre parti: nella prima vengono ridimensionati e disposti i loghi, nella seconda le informazioni principali come ad esempio il titolo e l'acronimo,

nell'ultima parte le informazioni sulla guida.

I metodi del settimo punto (*creazione statistiche*) *pdfLogStats()* e *pdfPathAlternativeCheck()* servono a creare gli output statistici required, subordinated e all.

Durante la costruzione della tabella del profilo d'uso, nel metodo *pdfWalk()* per ogni elemento, attributo e compositor, viene stabilito in quale tipologia di output statistico inserire il canonical path corrente e, una volta scelta, viene aggiunto alla corrispettiva lista (*requiredList*, *subordinatedList* e *allList*).

Nel caso in cui un elemento faccia parte di una choice, per evidenziare l'eventuale mutua esclusione con altri elementi della choice stessa, il path che verrà aggiunto nella lista presenterà la seguente forma: “**id?canonical_path**”.

Alla fine della costruzione della tabella, il metodo *pdfLogStats()* genera i file di output utilizzando le liste e, per ciascun elemento, viene analizzato il path tramite il metodo *pdfPathAlternativeCheck()* che restituisce TRUE se trova il simbolo “?” all'interno del path: in tal caso verrà creato un nodo “p” contenente la parte destra del path (dopo il “?”) e l'attributo “alt” contenente la parte sinistra del path (prima del “?”), come spiegato nella sezione 2.3.4 *output statistiche*.

Infine i metodi appartenenti all'ottavo punto (*gestione label*) hanno il compito di impostare il valore delle label rispettivamente per l'indice, l'appendice, il glossario e la tabella delle regole non linkate, cercandole prima nel template docbook, tramite una query XPath, e, qualora non definite, utilizzando quelle di default.

Utility

L'applicativo XDG, oltre ai metodi visti in precedenza, utilizza i seguenti metodi ausiliari:

- *utilityParseDocument()*
- *utilityXPathQuery()*
- *utilityTimeFormat()*
- *utilityLogStats()*

Il metodo *utilityParseDocument()* carica il DOM di un documento XML in modo da essere esaminato e/o modificato successivamente attraverso i metodi della libreria Java JAXP.

Il metodo *utilityXPathQuery()* esegue query XPath 2.0 mediante la libreria Java SAXON, permettendo di specificare opzionalmente espressioni che contengono *wildcard namespace* (es. *:party invece in cac:party).

Il metodo *utilityTimeFormat()* converte il tempo espresso in ms nel formato h:m:s (es: 00:03:45).

Il metodo *utilityLogStats()* definisce il template dell'output delle statistiche stats; template che verrà presentato nella sezione 3.2.3.

3.1.2 Classi ausiliarie

L'applicativo XDG si avvale delle seguenti classi private ausiliarie:

- HeaderFooter – *gestisce voci dell'indice e numero di pagina*
- HeaderFooterTOC – *gestisce il numero di pagina dell'indice*
- SchemaValidator – *valida documenti XML tramite file XSD*
- Validator – *gestisce gli errori di validazione warn, error, fatal*
- RuleNumber – *tiene traccia del numero di regole business rules*

3.1.3 Fogli di stile XSLT

L'applicativo XDG fa uso di tre fogli di stile XSLT che sono: *xml2html.xsl*, *skeleton.xsl* e *ExtractSchFromXSD-2.xsl*. Di seguito verranno esaminati i suddetti stylesheet nelle parti maggiormente significative. Per un maggior dettaglio è possibile consultare la relativa documentazione presente nel package dell'applicativo XDG, situata nella cartella “*stylesheetDoc/*”.

xml2html.xsl

Il foglio di stile *xml2html.xsl* è utilizzato dal componente *xml2html*, presentato nella sezione 2.3.2 *xml2html*, per generare, a partire dall'output intermedio XML, la guida nel formato HTML.

Le parti principali del foglio di stile sono due:

- struttura del *treegrid*
- codice javascript

La tabella del profilo d'uso, nell'output HTML, è stata implementata mediante l'utilizzo del *treegrid* ossia una tabella nella quale, in particolare, la parte sinistra è indentata a seconda del livello di annidamento. La struttura di ogni elemento del *treegrid* è rappresentata nell'immagine sottostante (Fig.27).

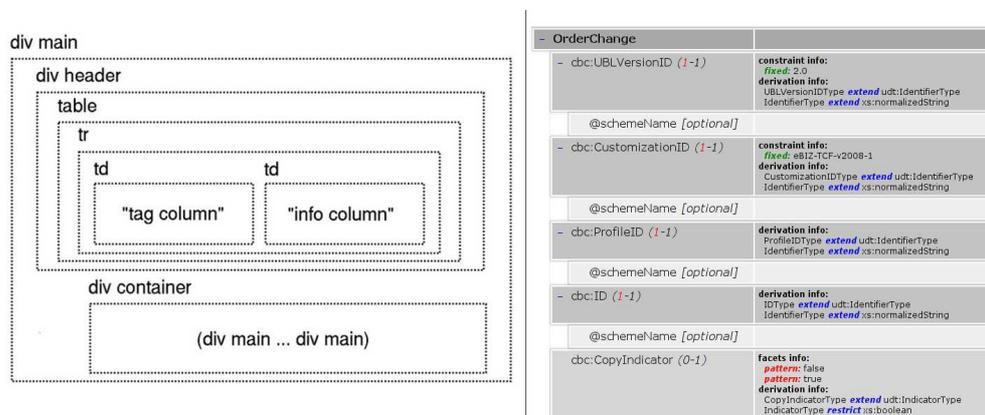


Figura 27: Struttura del treegrid.

La struttura dell'elemento è costituita da un “div” *main* esterno che, al suo interno, presenta altri due “div”, *header* e *container*: il primo contiene le informazioni dell'elemento stesso (la singola riga della tabella del profilo d'uso), mentre il secondo è utilizzato come contenitore nel quale inserire, a *runtime*, gli elementi annidati.

A sua volta il “div” *header* contiene l'elemento “table” con una sola riga “tr” divisa in due colonne “td”: la prima contiene il nodo dell'albero dell'istanza XML, mentre la seconda le relative informazioni.

La proprietà CSS [BTIH11] *margin-left*, applicata all'elemento “div” *container*, viene utilizzata per rappresentare il livello di annidamento.

La struttura dati *treegrid* utilizza codice javascript [SP10] per i seguenti scopi:

- evidenziare riga della tabella
 - *spot()* – *evidenzia l'entry corrente.*
 - *spotout()* – *fine evidenziamento.*
- manipolare DOM
 - *hideshow()* – *mostra/nasconde elementi annidati.*
 - *codelist()* – *carica le liste di codici.*
 - *showpathcall()* – *visualizza il canonical path.*
- ridimensionare colonne della tabella
 - *startDrag()* – *inizio ridimensionamento della colonna.*
 - *dragging()* – *ridimensionamento della colonna.*
 - *endDrag()* – *fine ridimensionamento della colonna.*

Le funzioni `spot()` e `spotout()` evidenziano l'elemento d'interesse cambiando il colore dell'elemento "tr" al passaggio del mouse.

La funzione `hideshow()` è la più importante in quanto consente di espandere/collassare i nodi dell'albero dell'istanza XML, permettendo un'agevole navigazione.

Essa effettua i seguenti passi (in pseudo codice):

```
function hideshow()
{
  // caso (-): elementi già aggiunti
  if( stato == espanso )
  {
    // nascondi il div container
    nascondi( "div container" );
    // imposta lo stato come collassato (+)
    stato = collassato;
  }
  // caso (+): div container nascosto
  else
  {
    // se nel div container non è ancora stato aggiunto niente
    if( empty( "div container" ) )
    {
      // recupera gli elementi annidati tramite query XPath
      doXPathQuery();
      // crea copia degli elementi recuperati
      doCloneNodes();
      // inserisci elementi copiati nel div container
      insertNodes( "div_container" );
    }

    // mostra il div container
    mostra( "div_container" );
    // impostato lo stato come espanso (-)
    stato = espanso;
  }
}
```

I punti salienti della funzione `hideshow()` sono la query XPath e la copia degli elementi, entrambe in modalità *cross-browser*: per eseguire la query XPath la funzione `hideshow()` utilizza la `selectNodes()` o l'`evaluate()` a seconda del tipo di supporto al DOM 3 XPath fornito dal browser; per effettuare la copia degli elementi utilizza la funzione `cloneNode()` nei browser che la supportano, mentre per gli altri si è reso necessario implementare una funzione clone *ad hoc* che ricostruisca l'elemento da zero.

La funzione *codelist()* carica le liste di codici: esse sono visualizzate usando gli elementi HTML “select” che mostrano liste di valori in tendine. Il funzionamento è lo stesso di quello visto per la funzione *hideshow()*, ossia il recupero dei valori tramite query XPath, la copia dei valori ed infine l'inserimento nell'apposito elemento.

La funzione *showpathcall()* mostra, al passaggio del mouse su un elemento, un tooltip contenente il canonical path del relativo elemento. La stringa contenente il path viene costruita a ritroso, partendo dal nodo foglia per poi risalire alla radice e man mano aggiornando la stringa con il nome del nodo in questione, memorizzato in un attributo posto sul div *main*.

Le funzioni *startDrag()*, *dragging()* e *endDrag()* permettono di cambiare il valore della proprietà *width* degli elementi “td”, in modo da adattare le dimensioni della tabella qualora necessario.

skeleton.xsl

Il foglio di stile *skeleton.xsl* viene utilizzato dal componente Schematron handler in un passo della pipeline la quale, partendo da un documento Schematron originale, genera un documento Schematron in una forma minimale, come discusso nella sezione 2.3.2 *Schematron handler*.

In particolare nel suddetto foglio di stile si provvede alla sostituzione dei parametri, qualora si trovino definiti nei pattern astratti. Tale sostituzione prevede due passaggi:

- organizzazione dei parametri
- sostituzione dei parametri

L'organizzazione dei parametri consiste nel creare un parametro XSLT (plist) contenente la lista dei parametri, ordinata in base alla lunghezza del nome del parametro in modo discendente, da passare ai template successivi.

La sostituzione dei parametri viene effettuata in due passi:

- costruzione della regexp
- utilizzo funzione XSLT 2.0 *xsl:analyze-string*

Per prima cosa viene costruita l'espressione regolare a partire dal parametro *plist*, contenente la lista dei parametri definiti nello *instance pattern*, nel seguente modo: “\$paramName1|.....|\$paramNameN”. Dopo aver impostato l'espressione regolare, si passa all'analisi della stringa per sostituire i parametri, come mostrato nella figura sottostante (Fig.28).

```
<xsl:analyze-string select="$context" regex="{ $regexp }">
  <!-- MATCHING PARAM FOUND -->
  <xsl:matching-substring>
    <xsl:variable name="paramMatch"
      select="substring-after(., '$ ' )"/>
    <xsl:variable name="replace"
      select="$paramList/param[ @name = $paramMatch ]/text()"/>
    <xsl:value-of select="$replace"/>
  </xsl:matching-substring>
  <!-- MATCHING PARAM NOT FOUND -->
  <xsl:non-matching-substring>
    <xsl:value-of select="."/>
  </xsl:non-matching-substring>
</xsl:analyze-string>
```

Figura 28: Sostituzione parametri nello XSLT skeleton.

ExtractSchFromXSD-2.xsl

Il foglio di stile *ExtractSchFromXSD-2.xsl*, usato anch'esso dal componente Schematron schema, permette di estrarre le regole Schematron embedded da un documento XSD per organizzarle in un file separato; tale stylesheet, situato nel package *iso-schematron-xslt2.zip* reperibile online, è l'unico non sviluppato in questa tesi (a differenza degli altri).

Questo foglio di stile procede nel mettere in un'unica variabile tutti gli schemi XSD, compresi quelli inclusi, importati e ridefiniti; di seguito genera il documento Schematron schema copiandovi gli elementi namespace “sch:ns”, le fasi “sch:phase” ed infine i pattern “sch:pattern”.

3.1.4 Configurazioni di default

L'applicativo XDG è stato fornito di tre file contenenti le configurazioni di default per gli output html, pdf e per il template docbook in modo da evitare che venga richiesta necessariamente la definizione di tali file da parte dell'utente che deve generare la guida. Inoltre tali file di configurazione consentono di mostrare a primo acchito con quali caratteristiche di base vengono generate le guide.

Configurazione HTML

Tra i valori più significativi impostati nel file di configurazione HTML (*html_configure.xml*) troviamo:

- dimensione seconda colonna “*documentation*”: 600px
- nodi all'interno del tag *annotation* appartenenti al vocabolario CCTS
- tutte le tipologie di documentazione abilitate

Configurazione PDF

Nel file di configurazione PDF (*pdf_configure.xml*) sono definiti i seguenti parametri:

- flush value: 100 (scrittura su disco ogni n righe inserite nella tabella)
- dimensioni colonne della tabella: 14pt , 266pt , 265pt
- nodi all'interno del tag *annotation* appartenenti al vocabolario CCTS
- tutte le tipologie di documentazione abilitate

Template docbook

Il template docbook di default (*docbook_default_template.db*) imposta la copertina con solamente il titolo ed il numero di versione, definisce le etichette per l'appendice, l'indice ed il glossario ed infine crea il capitolo profilo d'uso senza però alcun contenuto (Fig. 29).

```

<book xmlns="http://docbook.org/ns/docbook" xmlns:xlink="http://www.w3.org/1999/xlink" version="5.0">
  <info>
    <title>GUIDA PER PROFILO D'USO</title>
    <cover>
      <variablelist>
        <varlistentry>
          <term>
            <literal>Versione</literal>
          </term>
          <listitem>
            <simpara>0.1</simpara>
          </listitem>
        </varlistentry>
      </variablelist>
    </cover>
  </info>
  <appendix label="Appendice">
    <info>
      <title>Nodo info necessario per la validazione</title>
    </info>
    <para>Nodo para necessario per la validazione</para>
  </appendix>
  <index label="Indice"/>
  <glossary label="Glossario"/>
  <part role="profilo_uso">
    <info>
      <title>profilo d'uso</title>
    </info>
    <chapter>
      <title>Tabella del profilo d'uso</title>
      <informaltable>
        <tr><td>Tabella</td></tr>
      </informaltable>
    </chapter>
  </part>
</book>

```

Figura 29: Template docbook di default.

3.1.5 Documenti di validazione

Sono stati definiti i seguenti documenti XSD sia per consentire all'applicativo XDG di validare a *runtime* i file di configurazione passati in input sia per utilizzare l'applicativo stesso al fine di auto-generare la documentazione utile a comprendere i file di configurazione XDG.

1. task.xsd – *configurazione parametri d'input.*
2. trim.xsd – *semplificazione su richiesta dei rami dell'albero.*
3. html_configure.xsd – *parametrizzazione documentazione HTML.*
4. pdf_configure.xsd – *parametrizzazione documentazione PDF.*

3.2 Analisi output prodotti

Dopo aver esaminato l'implementazione dell'applicativo XDG in questa sezione verranno riportate la struttura dell'output intermedio XML, la veste grafica degli output HTML e PDF ed infine quella del documento stats.

3.2.1 XML

In questa sezione verrà analizzata la struttura dell'output intermedio XML, ponendo l'attenzione in particolare sugli elementi utilizzati per rappresentare i nodi dell'albero dell'istanza XML ed infine sarà proposto un algoritmo d'esempio su come utilizzare l'output intermedio XML per attraversare l'albero.

Struttura generale documento XML

Come descritto nella sezione 2.3.4 *Output XML*, il documento XML si suddivide nelle seguenti parti raffigurate nell'immagine sottostante (Fig.30).

```
<XDocumentationGenerator>
  <header>                                <!-- META INFORMAZIONI -->
  ...
</header>
  <body>                                    <!-- INFO SULLA STRUTTURA -->
  <tree/>                                   <!-- RADICE DELL'ALBERO -->
  <libraries>                               <!-- SEZIONE LIBRERIE -->
    <library>                               <!-- ENTRY LIBRERIA -->
      <type> ... </type>                   <!-- TIPI APPARTENENTI ALLA LIBRERIA -->
      ...
      <element> ... </element>           <!-- ELEMENTI APPARTENENTI ALLA LIBRERIA -->
      ...
    </library>
    ...
  </libraries>
  <codelists>                              <!-- SEZIONE CODELISTS -->
    <codelist> ... </codelist>           <!-- ENTRY CODELIST -->
    ...
    <codelist> ... </codelist>
  </codelists/>
</body>
</XDocumentationGenerator>
```

Figura 30: Struttura generale documento XML.

Building block dell'albero dell'istanza XML

Per organizzare la struttura e le informazioni dei nodi dell'albero dell'istanza XML sono stati definiti i seguenti componenti base:

- Componenti singoli
 - *type* – *identifica un tipo semplice o complesso.*
 - *element* – *identifica un elemento.*
 - *attribute* – *identifica un attributo.*
 - *any* – *identifica un elemento wildcard.*
- Componenti aggregati
 - *compositor* – *identifica i compositor: sequence, choice e all.*

La struttura dei componenti singoli è divisa in due parti: *containerInfo* e *containerContent*. La prima, obbligatoria, è utilizzata per memorizzare tutte le informazioni che sono:

- *documentationInfo* *informazioni sulle annotation*
- *facetsInfo* *informazioni sulle facets*
- *constraintInfo* *informazioni sui vincoli fixed o default*
- *codelistInfo* *informazioni sulle liste di codici*
- *derivationInfo* *informazioni di derivazione*
- *advancedInfo* *informazioni attributi block, final e abstract*

La seconda, opzionale, contiene i nodi annidati qualora presenti:

- *attribute*
- *element*
- *any*
- *compositor*

La struttura dei componenti aggregati è molto semplice: essa presenta esclusivamente il tag “compositor” dove all'interno si possono trovare element, any oppure altri compositor.

Attraversamento dell'albero dell'istanza XML

Di seguito viene presentato un algoritmo che mostra come percorrere ricorsivamente l'albero dell'istanza XML, utilizzando l'output intermedio XML e tenendo conto anche dell'eventuale possibilità di incorrere in un ciclo continuo in punti di loop.

I passaggi principali sono:

- ottenere la radice dell'albero
- impostare variabile path (per controllo loop ricorsione)
- percorrere l'albero ricorsivamente

```
// ottieni nodo radice
getRootNode( name , nsID )
{
    element = queryXPath("/XDocumentationGenerator/body/libraries/library/
                        element[ @nsID = '$nsID' and @name = '$name' ]" );
    return element;
}

// ottieni il nodo containerContent del tipo
getElementContent( typeName , nsID )
{
    element = queryXPath( "/XdocumentationGenerator/body/libraries/library/
                        type[ @nsID='"+nsID+"' and @name='"+typeName+"/
                        containerContent" )
    return element;
}

// controlla eventuale punto di loop
checkRecursion( path , name )
{
    if( path.contains( name ) == FALSE )
    {
        return NOT_LOOPING;
    }
    else
    {
        return LOOPING;
    }
}
```

```

walk( node )
{
  switch( node )
  {
    case ELEMENT:
    {
      // aggiorna il path corrente
      path += node->name+' ';

      doSomethingWithElement( node );

      // ottieni il rispettivo tipo
      follow = getElementTypeContent( node->refTypeName , node->refNsID );
      // ottieni tutti i nodi all'interno del containerContent del relativo tipo
      childList = getAllChildren( follow );
      // percorri ricorsivamente ogni nodo
      for( child : childList )
      {
        // controlla possibile punto di loop
        if( checkRecursion( path , child->name ) == NOT_LOOPING ) walk( child );
      }
    }
    case ATTRIBUTE:
    {
      doSomethingWithAttribute( node );
    }
    case WILDCARD:
    {
      doSomethingWithWildcard( node );
    }
    case COMPOSITOR:
    {
      switch( node->compositorType )
      {
        case sequence:
        {
          doSomethingWithSequence( node );
        }
        case choice:
        {
          doSomethingWithChoice( node );
        }
        case all:
        {
          doSomethingWithAll( node );
        }
      }

      // ricorsivamente percorri i nodi annidati
      childList = getAllChildren( node );
      for( child : childList ) walk( child );
    }
  }
}

main()
{
  root = getRootNode( tree->rootElementName , tree->rootElementID );
  path = "/";
  walk( root );
}

```

3.2.2 HTML & PDF

Di seguito saranno presentate, con l'ausilio di screenshot, le sezioni della guida per gli output HTML e PDF evidenziando le parti comuni e differenti.

Copertina

La struttura della copertina per entrambi gli output HTML e PDF è organizzata, dall'alto verso il basso, in fasce (1-7) ciascuna contenente una tipologia di elementi (Fig.31).



Figura 31: Es. copertina.

1. logo – immagini dei loghi di associazioni, aziende e applicazioni.
2. titolo guida per profilo d'uso – titolo principale della guida.
3. acronimo – acronimo del progetto.
4. nome/descrizione – spiegazione dell'acronimo.
5. link – link a pagine web.
6. altro – testo libero descrittivo.
7. informazioni guida – informazioni sulla generazione della guida.

Indice

L'indice nell'output HTML è proposto con un elenco puntato contenente i link ai relativi capitoli e sezioni, mentre nell'output PDF l'indice è presentato come normalmente avviene nei libri (Fig.32).

(HTML)	(PDF)																																														
<p>Indice</p> <ul style="list-style-type: none"> • 1. DESCRIZIONE DI BUSINESS <ul style="list-style-type: none"> • 1.1 Sezione 1 • 1.2 Sezione 2 • 2. DESCRIZIONE DI BUSINESS 2 <ul style="list-style-type: none"> • 2.1 Sezione 1 • 2.2 Sezione 2 • 3. Tabella profilo d'uso <ul style="list-style-type: none"> • 3.1 Sezione 1 • 3.2 Sezione 2 • 4. Business rule <ul style="list-style-type: none"> • 4.1 Sezione 1 • 4.2 Sezione 2 • Bibliografia 	<p>Indice</p> <hr/> <table border="0"> <tr><td>1. DESCRIZIONE DI BUSINESS</td><td>1</td></tr> <tr><td> 1.1 Scopo</td><td>1</td></tr> <tr><td> 1.2 Generalità</td><td>1</td></tr> <tr><td>2. DESCRIZIONE DI BUSINESS 2</td><td>3</td></tr> <tr><td> 2.1 Scopo 2</td><td>3</td></tr> <tr><td> 2.2 Generalità 2</td><td>3</td></tr> <tr><td>3. Tabella profilo d'uso</td><td>5</td></tr> <tr><td>4. Business rule</td><td>6002</td></tr> <tr><td> Appendix i - CodeList: AllowanceChargeReasonCode</td><td>6004</td></tr> <tr><td> Appendix ii - CodeList: BinaryObjectMimeCodeContent</td><td>6005</td></tr> <tr><td> Appendix iii - CodeList: ChargeCode</td><td>6008</td></tr> <tr><td> Appendix iv - CodeList: CityCode</td><td>6009</td></tr> <tr><td> Appendix v - CodeList: CountryIdentificationCode</td><td>6010</td></tr> <tr><td> Appendix vi - CodeList: CurrencyCode</td><td>6012</td></tr> <tr><td> Appendix vii - CodeList: CurrencyCodeContent</td><td>6013</td></tr> <tr><td> Appendix viii - CodeList: LatitudeDirectionCode</td><td>6014</td></tr> <tr><td> Appendix ix - CodeList: LineStatusCode</td><td>6015</td></tr> <tr><td> Appendix x - CodeList: LongitudeDirectionCode</td><td>6016</td></tr> <tr><td> Appendix xi - CodeList: OperatorCode</td><td>6017</td></tr> <tr><td> Appendix xii - CodeList: PaymentMeansCode</td><td>6018</td></tr> <tr><td> Appendix xiii - CodeList: UnitCodeContent</td><td>6019</td></tr> <tr><td>Glossario</td><td>6025</td></tr> <tr><td>Bibliografia</td><td>6026</td></tr> </table>	1. DESCRIZIONE DI BUSINESS	1	1.1 Scopo	1	1.2 Generalità	1	2. DESCRIZIONE DI BUSINESS 2	3	2.1 Scopo 2	3	2.2 Generalità 2	3	3. Tabella profilo d'uso	5	4. Business rule	6002	Appendix i - CodeList: AllowanceChargeReasonCode	6004	Appendix ii - CodeList: BinaryObjectMimeCodeContent	6005	Appendix iii - CodeList: ChargeCode	6008	Appendix iv - CodeList: CityCode	6009	Appendix v - CodeList: CountryIdentificationCode	6010	Appendix vi - CodeList: CurrencyCode	6012	Appendix vii - CodeList: CurrencyCodeContent	6013	Appendix viii - CodeList: LatitudeDirectionCode	6014	Appendix ix - CodeList: LineStatusCode	6015	Appendix x - CodeList: LongitudeDirectionCode	6016	Appendix xi - CodeList: OperatorCode	6017	Appendix xii - CodeList: PaymentMeansCode	6018	Appendix xiii - CodeList: UnitCodeContent	6019	Glossario	6025	Bibliografia	6026
1. DESCRIZIONE DI BUSINESS	1																																														
1.1 Scopo	1																																														
1.2 Generalità	1																																														
2. DESCRIZIONE DI BUSINESS 2	3																																														
2.1 Scopo 2	3																																														
2.2 Generalità 2	3																																														
3. Tabella profilo d'uso	5																																														
4. Business rule	6002																																														
Appendix i - CodeList: AllowanceChargeReasonCode	6004																																														
Appendix ii - CodeList: BinaryObjectMimeCodeContent	6005																																														
Appendix iii - CodeList: ChargeCode	6008																																														
Appendix iv - CodeList: CityCode	6009																																														
Appendix v - CodeList: CountryIdentificationCode	6010																																														
Appendix vi - CodeList: CurrencyCode	6012																																														
Appendix vii - CodeList: CurrencyCodeContent	6013																																														
Appendix viii - CodeList: LatitudeDirectionCode	6014																																														
Appendix ix - CodeList: LineStatusCode	6015																																														
Appendix x - CodeList: LongitudeDirectionCode	6016																																														
Appendix xi - CodeList: OperatorCode	6017																																														
Appendix xii - CodeList: PaymentMeansCode	6018																																														
Appendix xiii - CodeList: UnitCodeContent	6019																																														
Glossario	6025																																														
Bibliografia	6026																																														

Figura 32: Es. indice.

Capitoli

La struttura dei capitoli è la stessa per entrambi gli output HTML e PDF (Fig.33).

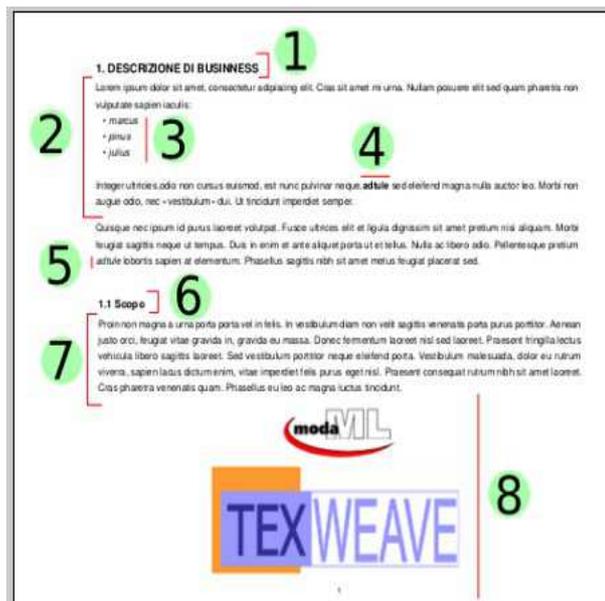


Figura 33: Es. capitolo.

Vengono gestiti i seguenti tipi di contenuti:

1. titolo capitolo – *il titolo principale del capitolo.*
2. paragrafo capitolo – *esempio di testo nel paragrafo del capitolo.*
3. elenco puntato – *esempio di elenco puntato.*
4. testo grassetto – *testo enfatizzato in grassetto.*
5. testo corsivo – *testo enfatizzato in corsivo.*
6. titolo sezione – *il titolo della sezione contenuta nel capitolo.*
7. paragrafo sezione – *esempio di testo nel paragrafo della sezione.*
8. immagine – *esempio immagine inserita.*

Tabella del profilo d'uso

La parte principale della guida è rappresentata dalla tabella del profilo d'uso: *dinamica* per l'output HTML, *statica* per l'output PDF (Fig.34).

TAG	DOCUMENTATION
- taskConfig	Tasks Configuration File.
- singleTask (0-unbounded)	Task configuration for single xsd file.
- input (1-1)	Input configuration options.
- xsd (1-1)	XSD Schema input file configuration options.
@rootElementName [optional]	Choice a specified tree if more then one root element are available in the XSD input file. facets info: minLength: 1 derivation info: rootElementNameType restrict xs:string
fileName (1-1)	The xsd.schema input file name.

(HTML)

TAG	DOCUMENTATION
0 taskConfig	Tasks Configuration File.
1 singleTask (0-unbounded)	Task configuration for single xsd file.
2 . input (1-1)	Input configuration options.
3 . . xsd (1-1)	XSD Schema input file configuration options.
. . . @rootElementName [optional]	Choice a specified tree if more then one root element are available in the XSD input file. facets info: minLength: 1 derivation info: rootElementNameType restrict xs:string
4 . . . fileName (1-1)	The xsd.schema input file name.
3 . . . codelist (0-1)	Codelist configuration options.
4 . . . directoryName (1-1)	The codelist directory.
4 . . . displayName (0-1)	The codelist (generic code) field name to display with value field. constraint info: default: name

(PDF)

Figura 34: Es. tabella del profilo d'uso.

Oltre agli elementi, attributi e wildcard, sono rappresentati in modo chiaro i compositor *sequence*, *choice* ed *all*, visualizzando l'inizio/fine e riportando la relativa cardinalità (Fig.35).

(HTML)	(PDF)
CHOICE - Start (1-1)	CHOICE - Start (1-1)
- DSAKeyValue (1-1)	9 DSAKeyValue (1-1)
SEQUENCE - Start (0-1)	SEQUENCE - Start (0-1)
P (1-1)	10 P (1-1)
Q (1-1)	10 Q (1-1)
SEQUENCE - End	SEQUENCE - End
G (0-1)	10 G (0-1)
Y (1-1)	10 Y (1-1)
J (0-1)	10 J (0-1)
SEQUENCE - Start (0-1)	SEQUENCE - Start (0-1)
Seed (1-1)	10 Seed (1-1)
PgenCounter (1-1)	10 PgenCounter (1-1)
SEQUENCE - End	SEQUENCE - End
OR	OR
+ RSAKeyValue (1-1)	9 RSAKeyValue (1-1)
OR	10 Modulus (1-1)
<any/> (1-1)	10 Exponent (1-1)
OR	OR
<any/> (1-1)	<any/> (1-1)
CHOICE - End	CHOICE - End

Figura 35: Es. compositor.

Per quanto riguarda i nodi situati all'interno del tag “xsd:documentation”, per entrambi gli output, sono riportati nella forma nome-valore.

(HTML)

info Core Components: ComponentType: BBIE DictionaryEntryName: Instruction For Returns, UBL Version Identifier.Identifier Definition: The earliest version of the UBL 2 schema for this document type that defines all of the elements that might be encountered in the current instance. ObjectClass: Measure Unit PropertyTermName: Code RepresentationTermName: Code PrimitivType: string UsageRule: Reference UNECE Rec 20 and X12 355.

(a)

(PDF)

(b)

info Core Components:
 UniqueID: UDT0000013-SC2
 CategoryCode: SC
 DictionaryEntryName: Measure Unit, Code
 Definition: The type of unit of measure.
 ObjectClass: Measure Unit
 PropertyTermName: Code
 RepresentationTermName: Code
 PrimitivType: string
 UsageRule: Reference UNECE Rec 20 and X12 355.

Glossario

Data Type
 Data Type – Defines the set of valid values that can be used for a particular Business Information Entity Property. It is defined by specifying restrictive basis of the Data Type.

Definition
 Definition – This is the unique semantic meaning of a Core Component, Data Type.

Figura 36: Es. documentation info.

Le spiegazioni dei nodi situati nelle *annotation*, impostabili nei relativi file di configurazione, vengono visualizzate nell'output HTML al passaggio del mouse sul nodo interessato, mentre nell'output PDF sono organizzate e consultabili nella sezione *glossario* (Fig.36).

Le liste di codici (mandatory e recommended) nell'output HTML sono riportate nei nodi che ne fanno uso mediante un menu a tendina, apribile dinamicamente al bisogno; nell'output PDF, nei nodi vengono presentati solo i nomi identificativi delle liste di codici, rimandando all'*appendice* per l'elenco dei rispettivi valori (Fig.37).

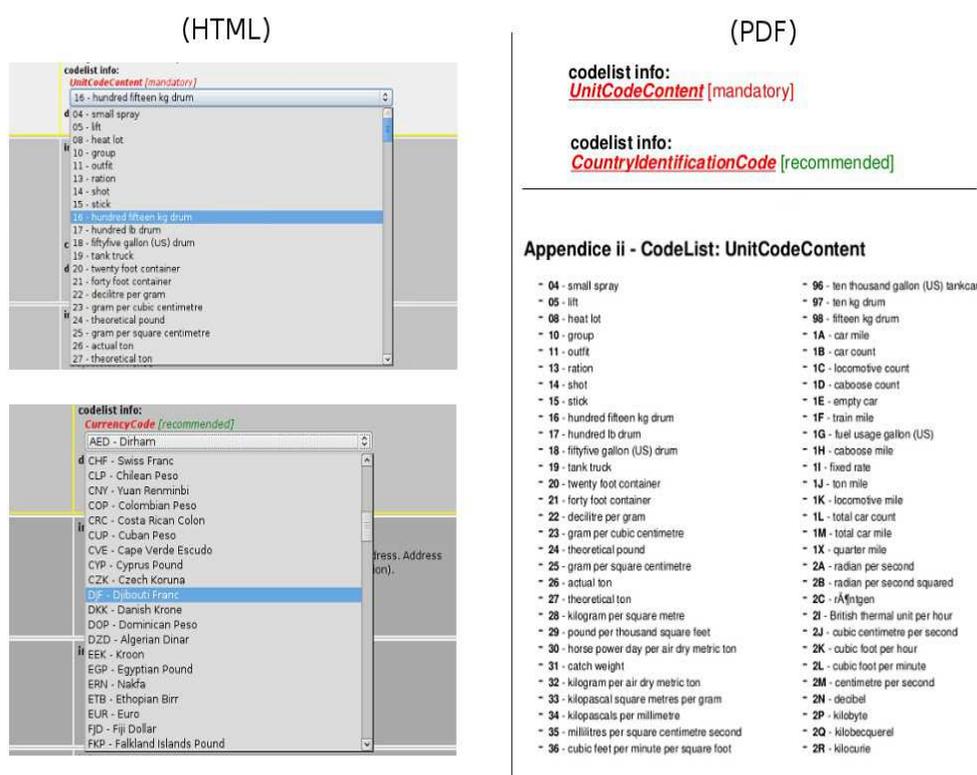


Figura 37: Es. codelist.

Per ciascun elemento dell'albero dell'istanza vengono visualizzate, allo stesso modo in entrambi gli output HTML e PDF, le tipologie d'informazione di seguito elencate, sotto forma di coppia nome-valore (Fig.38):

- facets info – informazioni sulle facets come pattern e lenght.
- constraint info – informazioni sui vincoli fixed o default.
- advanced info – informazioni sugli attributi block, final e abstract.
- wildcard info – informazioni sul processContents e sul namespace.

```

facets info:
pattern: [0-9ABCDEFabcdef]{6}

constraint info:
default: en

advanced info:
block: #all
final: extension
abstract: true

wildcard info:
processContents: skip
namespace: ##any

```

Figura 38: Es. facets, constraint, advanced e wildcard info.

In aggiunta le *derivation info*, in entrambi gli output HTML e PDF, sono presentate nel seguente modo: a sinistra il nome del tipo che effettua la derivazione, al centro l'etichetta *extend* o *restrict* che specifica l'operazione di estensione o di restrizione effettuata ed infine a destra il nome del tipo che ha subito la derivazione (Fig.39).

```

derivation info:
IdentificationCodeType extend CountryIdentificationCodeType
CountryIdentificationCodeType restrict udt:CodeType
CodeType extend xs:normalizedString

```

Figura 39: Es. derivation info.

Esclusivamente nell'output PDF, i nodi dell'albero dell'istanza XML contengono i link alle informazioni sulle business rules: i numeri presenti sono gli *ID* identificativi delle regole documentate nella tabella che si trova nel capitolo business rules (Fig.40).

```

additional constraint info:
19, 35

```

Figura 40: Es. additional constraint info.

Tabella business rules

Per entrambi gli output HTML e PDF, la tabella business rules è strutturata in sezioni, contenenti le regole, delimitate in testa da righe nere la cui presenza varia in base a come è costituito il documento Schematron; sono possibili tre diversi casi:

- regole organizzate in fasi
- regole organizzate in fasi più regole non organizzate in fasi
- regole non organizzate in fasi

La tabella nel primo caso avrà tante righe nere quante sono le fasi attivate, ognuna contenente l'ID della fase e del pattern attivato, separate da un trattino.

Nel secondo caso ci saranno tante righe nere quante sono le fasi attivate, ognuna contenente l'ID della fase e del pattern attivato ed inoltre ci sarà una riga nera con l'etichetta “Additional rules” contenente le regole che non sono state organizzate in fasi (Fig.41).

Phase ID: a - testa	
1) "/Catalogue/cbc:ProfileID"	
• ASSERT	
{text() = "eBIZ-TCF-ArticleCatalogue"}	
<i>The value of the element ProfileID must be eBIZ-TCF-ArticleCatalogue</i>	
• REPORT	
@schemeName	
<i>The attribute schemeName must not be present in this path.</i>	
Phase ID: a - testb	
2) "/Catalogue/cbc:ID"	
• REPORT	
@schemeName	
<i>The 'schemeName' attribute must not be present in this path.</i>	
Phase ID: b - testc	
3) "//cbc:ID"	
• REPORT	
@schemeName	
<i>The 'schemeName' attribute must not be present in this path.</i>	
Additional rules	
4) "/Catalogue/cbc:VersionID"	
• REPORT	
@schemeName	
<i>The 'schemeName' attribute must not be present in this path.</i>	

Figura 41: Tabella business rules con fasi e regole extra.

Nel terzo caso ci sarà esclusivamente una sola riga nera con l'etichetta “Additional rules”.

Dopo la riga nera si può trovare almeno una riga grigia nella quale sono riportati il numero identificativo associato alla regola ed il nodo contesto della regola definito nell'espressione XPath presente nell'attributo *context*. Ogni riga grigia è seguita una o più volte da tre righe bianche.

Nella prima riga, identificata dal simbolo “•”, sono presenti l'etichetta ASSERT o REPORT a seconda del tipo di asserzione ed eventualmente il valore della *flag* warning o fatal. Nella seconda riga c'è il valore dello *assertion test* identificato dall'espressione XPath contenuta nell'attributo *test*. Nella terza ed ultima riga si trova la descrizione verbale dello *assertion test*.

Infine, esclusivamente nell'output PDF, si può trovare la tabella business rules delle regole non linkate, ovvero quelle regole non associabili in nessun punto dell'albero dell'istanza XML. Tale tabella è strutturata come quella nella figura precedente (Fig.41) ed è identificata tramite una label configurabile nel template docbook (Es. “*Tabella regole non linkate*”).

Bibliografia

Infine la bibliografia è presentata nel seguente modo (Fig.42):

The diagram shows a bibliography entry with six numbered callouts:

- 1: Points to the label 'Bibliografia'.
- 2: Points to the identifier '[EJ11]'.
- 3: Points to the author names 'Erwin Folmer, Jack Verhoosel'.
- 4: Points to the title 'State of the Art on Semantic IS Standardization, Interoperability & Quality'.
- 5: Points to the publisher 'Wiley'.
- 6: Points to the URL 'http://ebxml.xml.org/node/47'.

Figura 42: Es. bibliografia.

Vengono gestiti i seguenti tipi di contenuti:

1. label – *etichetta della bibliografia*.
2. abbreviazione – *identificativo della fonte bibliografica*.

3. gruppo autore/autori – *nome e cognome di uno o più autori.*
4. titolo – *titolo della fonte bibliografica.*
5. testo libero – *testo libero per aggiungere informazioni sulla fonte.*
6. link – *link a pagine web riguardanti la risorsa bibliografica.*

3.2.3 STATISTICHE

Le informazioni dell'output *stats*, presentate nella sezione 2.3.4 “*Output statistiche*”, sono visualizzate come nella figura sottostante (Fig.43).

```
#=====
Stats generated by: eBiz-TCF-DespatchAdviceDeliveryBasedValidator.xsd
#=====
XML TIME:           | 00:00:09 h/m/s
HTML TIME:          | 00:00:02 h/m/s
PDF TIME:           | 00:00:04 h/m/s
XML FILE SIZE:      | 440 KB
HTML FILE SIZE:     | 314 KB
PDF FILE SIZE:      | 53 KB
#-----
ATTRIBUTE:          | 152
ELEMENT LEAF:       | 182
ELEMENT:            | 95
TOTAL:              | 429
#-----
NODES REQUIRED:      | 59
NODES SUBORDINATED: | 83
#-----
LOOP POINT:         | 0
MAX NESTED LEVEL:  | 8
#=====
```

Figura 43: Es. documento stats.

3.3 Considerazioni finali

In questo capitolo sono stati esaminati solo i principali dettagli implementativi dell'applicativo XDG, mentre per un maggior approfondimento si rimanda sia alla documentazione Javadoc disponibile nel package dell'applicativo XDG all'interno della cartella “doc/javadoc/” sia direttamente al codice sorgente: questo infatti è stato commentato in modo chiaro e minuzioso proprio per favorirne la comprensione e/o eventuale modifica.

Durante la fase di sviluppo dell'applicativo ci si è concentrati, oltre che sulla correttezza della documentazione generata, anche sull'aspetto delle *performance* cercando appunto di minimizzare l'uso delle risorse.

Nel prossimo capitolo, *Conclusioni*, verrà valutato l'applicativo XDG in un contesto reale come quello dei documenti XSD appartenenti alla specifica UBL-2.0 ed inoltre saranno delineati i suoi possibili sviluppi futuri.

Conclusioni

Lo scopo della tesi è stato quello di fornire un valido strumento per la generazione della documentazione relativa a schemi XSD, organizzata in chiare e pratiche guide per profili d'uso, in modo da favorirne la comprensione e l'utilizzo senza richiedere un *background* informatico, sfruttando sia in maniera flessibile le annotazioni all'interno del documento stesso sia informazioni esterne come le business rules e le liste di codici.

Si è partiti dall'analisi dei principali applicativi disponibili appartenenti alla categoria XML-Schema documentation generator, individuandone sia i pregi che i difetti; inoltre sono state esaminate le documentazioni esistenti per alcuni linguaggi basati sulla specifica Core Components, quali UBL 2.0, MODA-ML ed eBiz-TCF (capitolo 1).

Effettuata questa prima fase di ricerca, si è passati alla progettazione dell'applicativo, delineando, in relazione a quanto sopra esposto, i requisiti richiesti e le funzionalità che esso deve fornire e motivando le scelte progettuali intraprese. Ci si è poi focalizzati sull'architettura dell'applicativo: partendo dalla descrizione del ciclo d'esecuzione, sono stati definiti tutti i componenti necessari assieme ai meccanismi di input/output (capitolo 2).

Infine nell'ultimo capitolo, sono stati descritti i punti salienti dell'implementazione dell'applicativo, quali i metodi della classe principale ed i fogli di stile XSLT; inoltre sono stati proposti alcuni esempi di output generati, evidenziandone le parti simili e quelle differenti.

Dopo la fase d'implementazione, l'applicativo XDG è stato sottoposto ad una fase di test in modo da valutare in primo luogo la compatibilità delle guide generate sui web browser/pdf reader previsti nella sezione 2.2.8, in secondo luogo la correttezza e l'effettiva usabilità delle guide relative alla specifica UBL-2.0 ed infine soprattutto è stato sottoposto ad una vera prova sul campo in un contesto reale come quello del progetto PEPPOL [PPL11].

Dalle prove effettuate risulta che la compatibilità web browser della guida è stata rispettata completamente in tutte le funzionalità principali, quali il treegrid, i codelist e il tooltip XPath. Anche per quanto riguarda la compatibilità con i pdf reader, la guida viene visualizzata correttamente su tutti i programmi presi in esame; l'unica piccola anomalia è stata riscontrata nella visualizzazione dei link a pagine web nel programma Acrobat Reader: è stato comunque trovato un workaround.

Per testare la correttezza e l'usabilità delle guide generate sono stati scelti i documenti UBL-2.0 più significativi con diverso numero totale di elementi.

Si è verificata l'effettiva correttezza delle guide sia confrontando le statistiche sulla composizione dell'albero generate dall'applicativo XDG (quale il numero degli elementi non foglia, foglia, attributi e quello totale) con le statistiche generate da un altro applicativo sia effettuando dei test a campione su varie parti nella guida.

Per quanto riguarda l'usabilità delle guide generate dall'applicato XDG, come si evince dalla tabella sottostante, per documenti XSD che generano istanze ridotte, le guide nel formato PDF forniscono un reale supporto alla comprensione della struttura dei documenti ed, essendo costituite da poche pagine, non fanno perdere la visione d'insieme e quindi non risultano essere dispersive. Invece per documenti XSD che generano istanze molto estese, la guida nel formato PDF risulta non avere una reale praticità d'uso a causa del

numero elevato di pagine: in questo caso è appunto indispensabile disporre della versione HTML.

Document	Html time (h:m:s)	Pdf time (h:m:s)	Html size (MB)	Pdf size (MB)	Total element
AttachedDocument	00:00:04	00:00:19	9,338	1,514	6.576
Reminder	00:00:05	00:00:38	9,418	3,614	15.928
Catalogue	00:00:05	00:01:35	9,365	9,028	40.087
Waybill	00:00:05	00:18:04	9,363	105,524	468.891
DespatchAdvice	00:00:04	00:37:13	9,357	214,017	950.021
Quotation	00:00:04	00:59:14	9,371	339,437	1.505.292
Order	00:00:04	01:56:57	9,418	677,136	3.001.765

Tabella 2: *Tabella test documenti XSD UBL-2.0.*

Questo test ha confermato, come previsto nella fase di progettazione, che nel formato HTML le guide, anche per documenti XSD che possono dar vita ad istanze molto grandi, vengono generate in tempi (circa 4/6 secondi) e con dimensioni (circa 10MB) costanti rispetto a quelle nel formato PDF le quali invece, all'aumentare del numero totale di nodi, risultano via via sempre più pesanti e generate in tempi molto lunghi, come ad esempio accade per il documento *Order*.

Infine l'applicativo è stato testato in una vera prova sul campo inerente il progetto europeo PEPPOL (Pan-European Public Procurement On-Line). Questo ha l'obiettivo di sviluppare una soluzione pilota di eProcurement pubblico transnazionale per definire un set condiviso di elementi che consenta ai soggetti nei diversi Paesi di interagire, nonostante le differenze legislative ed in particolare tecnologiche.

In appendice A è stato posto un esempio di guida, relativa al documento *Invoice*, basato sulla specifica UBL e utilizzato per emettere fatture in formato elettronico.

Tale guida soddisfa pienamente le aspettative ed integra perfettamente le informazioni di vario tipo, quali le Business rules, liste di codici e testo aggiuntivo, tanto che l'applicativo XDG sarà utilizzato per generare la documentazione che Intercent-ER (*Agenzia regionale di sviluppo dei mercati telematici* che si occupa di promuovere e sostenere il processo di ottimizzazione degli acquisti e di gestire la piattaforma tecnologica predisposta dalla Regione Emilia-Romagna) fornirà per supportare l'implementazione delle specifiche PEPPOL.

Come esposto nel capitolo 2 sulla progettazione, si è scelto di focalizzare l'attenzione sul “motore” dell'applicativo XDG invece che sull'interfaccia grafica, cercando però allo stesso tempo di adottare soluzioni adeguate in modo da favorirne un'implementazione futura.

L'applicativo XDG può essere facilmente integrato in altre applicazioni desktop oppure in web application semplicemente istanziando la classe `XDocumentationGenerator` e chiamando il metodo `configurationParse()`, passando come parametro il pathname del file di configurazione TASK.

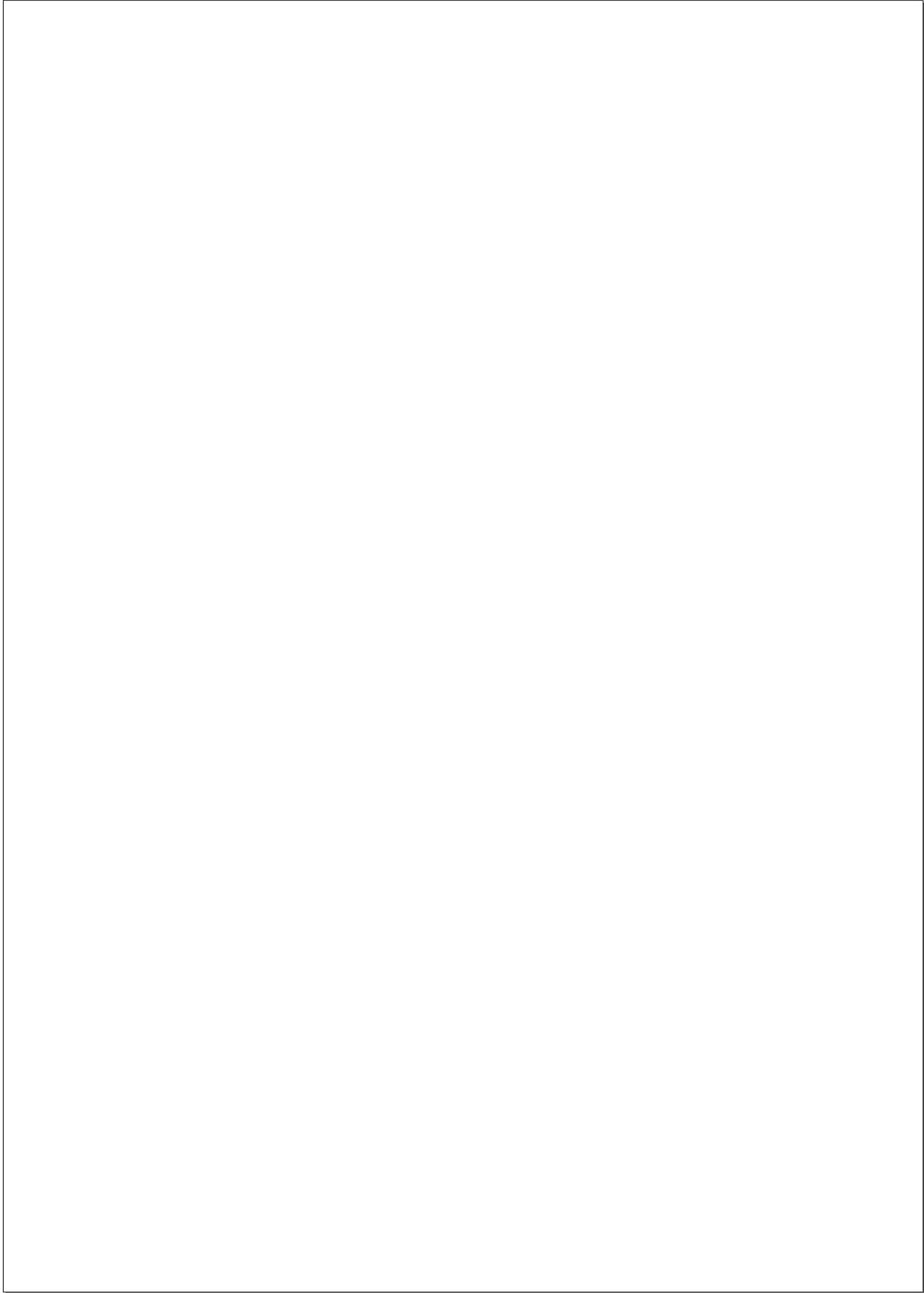
La maggior parte degli sviluppi futuri si può incentrare sulla creazione di comode GUI (*graphical user interface*) per la creazione e per l'impostazione dei principali file di configurazione fin qui visti. Essendo questi documenti di configurazione basati sul linguaggio XML e provvisti dei rispettivi XSD, è possibile costruire interfacce in qualunque linguaggio che consenta la manipolazione, la creazione e la validazione di documenti XML: infatti tali interfacce, una volta ottenuti i parametri, dovranno semplicemente creare il relativo file XML e validarlo con il corrispettivo documento XSD. Un altro aspetto interessante potrebbe essere quello d'implementare tool di analisi e di reportistica, basati sui documenti statistici generati dall'applicativo: ad esempio potrebbe essere utile generare automaticamente grafici oppure verificare la composizione degli elementi dell'albero.

Appendice A – Invoice (PEPPOL)

Di seguito viene presentato un esempio di guida per profilo d'uso, generata dall'applicativo XDG, relativa al documento *Invoice* appartenente alla specifica PEPPOL.

**GUIDA DI IMPLEMENTAZIONE DEL
PROFILO D'USO UBL PER LA
FATTURA PEPPOL**

Versione: 1.0
Data: 02/11/2011
Note: questo è un documento di esempio



Indice

1. Contesto	1
2. Descrizione	2
3. Profilo d'uso	3
Appendice i - CodeList: BinaryObjectMimeCodeContent	12
Appendice ii - CodeList: CurrencyCodeContent	15
Appendice iii - CodeList: UnitCodeContent	16
Glossario	22



1. Contesto

Questo documento descrive il profilo d'uso della fattura UBL 2.0 definito nell'ambito del progetto PEPPOL (www.peppol.eu).

Il Progetto Pan-European Public Procurement On-Line (PEPPOL) nasce nell'ambito del Programma Europeo per l'Innovazione e la Competitività (CIP) che promuove azioni volte a favorire l'imprenditorialità, la competitività industriale e l'innovazione facilitando l'accesso delle PMI al finanziamento e agli investimenti nella loro fase di avviamento e di crescita. L'obiettivo è sviluppare una soluzione pilota di eProcurement pubblico transnazionale, cioè interoperabile a livello europeo, che capitalizzi sui punti di forza dei sistemi nazionali esistenti. Tale soluzione deve consentire a un qualsiasi operatore economico dell'Unione Europea di:

- partecipare ad una procedura d'acquisto elettronica indetta da una stazione appaltante pubblica di un altro Stato Membro;
- inviare le attestazioni necessarie alla partecipazione e la propria offerta in forma elettronica;
- qualora risulti aggiudicatario e se previsto, inviare un catalogo in formato elettronico;
- ricevere ordini in formato elettronico
- emettere fatture elettroniche.

La sintassi adottata per i documenti di business è UBL 2.0 per il quale Peppol ha definito dei profili d'uso, al fine di garantire l'interoperabilità tra le varie implementazioni.

Il profilo d'uso di un documento di business standard è una sua customizzazione (personalizzazione) per un contesto specifico; in termini pratici, il profilo d'uso può definire, ad esempio, sottoinsiemi dello standard (individuando la parte che serve per supportare le informazioni da scambiare) e/o raffinarne la semantica.

2. Descrizione

La fattura e' il documento utilizzato per richiedere il pagamento di beni o servizi forniti alle condizioni concordate tra il fornitore e il cliente.

Le fatture UBL conformi al profilo d'uso qui descritto rispettano la direttiva 2006/112/EC.

Le fatture UBL conformi al profilo d'uso qui descritto devono essere valide rispetto allo schema:

<http://docs.oasis-open.org/ubl/os-UBL-2.0/xsd/maindoc/UBL-Invoice-2.0.xsd>.

Le fatture UBL conformi al profilo d'uso qui descritto devono rispettare un set di regole di business che descrivono requisiti normativi nazionali e europei. Le regole di business, descritte in linguaggio Schematron, sono disponibili al seguente indirizzo:

<http://www.invinet.org/recursos/conformance/download.html> .

3. Profilo d'uso

Tutti i dati che possono essere contenuti nella Fattura sono illustrati nella seguente tabella (che offre una visione semplificata del profilo d'uso Peppol dello schema UBL Invoice 2.0) dove:

1- la colonna TAG indica gli elementi UBL, dove:

- i due numeri che seguono ciascun elemento indicano la sua "cardinalità" (numero minimo e massimo di ripetizioni ammesse); il numero minimo uguale a 0 indica un elemento opzionale, il numero minimo maggiore o uguale a 1 indica un elemento obbligatorio);
- lo sfondo grigio denota gli elementi che sono "complessi semantici" ovvero aggregati di dati elementari;
- il simbolo @ denota un attributo dell'elemento che lo precede; qui la condizione d'uso (required, optional, ecc.) è indicata fra parentesi quadre.

2- la colonna DOCUMENTATION fornisce:

- la descrizione dell'elemento;
- eventuali regole di business applicabili sull'elemento (additional constraint info);
- eventuali codifiche da utilizzare per valorizzare l'elemento (codelist info).

TAG	DOCUMENTATION
0 Invoice	Documento usato per richiedere un pagamento. additional constraint info: S-P
1 cbc:UBLVersionID (1-1)	<i>Definition:</i> Identificatore della versione UBL di cui il documento è istanza.
1 cbc:CustomizationID (1-1)	<i>Definition:</i> Identificatore della transazione di cui il documento è istanza.
1 cbc:ProfileID (1-1)	<i>Definition:</i> Identificatore del processo di business nell'ambito del quale il documento viene scambiato.
1 cbc:ID (1-1)	<i>Definition:</i> Numero fattura.
1 cbc:IssueDate (1-1)	<i>Definition:</i> Data di emissione fattura.
1 cbc:InvoiceTypeCode (0-1)	<i>Definition:</i> Tipo fattura (espresso come codice).
. @listAgencyName [optional]	<i>Definition:</i> Agenzia che mantiene la lista di codici. <i>UsageRule:</i> Default: UNEDIFACT 3055 code list.
. @listID [optional]	<i>Definition:</i> Identificatore della lista di codici.
1 cbc:Note (0-1)	<i>Definition:</i> Testo libero relativo alla Fattura. Può contenere note o altre informazioni non esplicitamente previste nella struttura del documento.
1 cbc:TaxPointDate (0-1)	<i>Definition:</i> La data della Fattura, usata per indicare il momento in cui l'imposta sarà applicabile.
1 cbc:DocumentCurrencyCode (1-1)	<i>Definition:</i> Valuta usata nel documento.
. @listAgencyName [optional]	<i>Definition:</i> Agenzia che mantiene la lista di codici. Default: UNECE
. @listID [optional]	<i>Definition:</i> Identificatore della lista di codici utilizzata. Default:ISO 4217 Alpha
1 cbc:AccountingCost (0-1)	<i>Definition:</i> Il codice contabile del Compratore applicato all'intera Fattura, espresso come testo.
1 cac:InvoicePeriod (0-1)	<i>Definition:</i> Periodo al quale la fattura si riferisce additional constraint info: S
2 cbc:StartDate (0-1)	<i>Definition:</i> Data inizio del periodo di fatturazione.
2 cbc:EndDate (0-1)	<i>Definition:</i> Data fine del periodo di fatturazione.
1 cac:OrderReference (0-1)	<i>Definition:</i> Riferimento all'ordine per cui si è emessa la fattura.
2 cbc:ID (1-1)	<i>Definition:</i> Identificatore dell'ordine (assegnato dal cliente).
1 cac:ContractDocumentReference (0-1)	<i>Definition:</i> Riferimento ad un contratto correlato alla fattura.
2 cbc:ID (1-1)	<i>Definition:</i> Identificatore del documento.
2 cbc:DocumentType (0-1)	<i>Definition:</i> Tipo di documento.
1 cac:AdditionalDocumentReference (0-unbounded)	<i>Definition:</i> Riferimento ad altri documenti correlati alla fattura.
2 cbc:ID (1-1)	<i>Definition:</i> Identificatore del documento.
2 cbc:DocumentType (0-1)	<i>Definition:</i> Tipo di documento.
. cac:Attachment (0-1)	<i>Definition:</i> Documento allegato.
3 . . cbc:EmbeddedDocumentBinaryObject (0-1)	<i>Definition:</i> Contiene un documento incorporato in formato BLOB (binary large

		object)
	@mimeCode [required]	Definition: Tipo mime del binary object. codelist info: BinaryObjectMimeTypeContent [mandatory]
3	cac:ExternalReference (0-1)	Definition: Un documento allegato, riferito esternamente, internamente ad un indirizzo MIME, o incorporato.
4	cbc:URI (0-1)	Definition: L'URI che identifica dove è situato il documento esterno.
1	cac:AccountingSupplierParty (1-1)	Definition: Dati del fornitore.
2	cac:Party (0-1)	additional constraint info: 25, 4
3	cbc:EndpointID (0-1)	Definition: Identificatore di un end point associato al soggetto (ad es.EANLocation Number, GLN).
3	cac:PartyIdentification (0-1)	
4	cbc:ID (1-1)	Definition: Identificatore del soggetto (assegnato secondo le PEPPOL Policy sull'uso degli identificatori).
3	cac:PartyName (1-1)	
4	cbc:Name (1-1)	Definition: Denominazione o ragione sociale.
3	cac:PostalAddress (1-1)	Definition: Indirizzo sede. additional constraint info: 20
4	cbc:ID (0-1)	Definition: Identificatore indirizzo (utilizzare se l'indirizzo e' registrato in uno schema di registrazione indirizzi).
4	cbc:Postbox (0-1)	Definition: Numero di casella postale.
4	cbc:StreetName (0-1)	Definition: Nome via, piazza, ...
4	cbc:BuildingNumber (0-1)	Definition: Numero civico.
4	cbc:Department (0-1)	Definition: Nome dipartimento.
4	cbc:PostalZone (0-1)	Definition: CAP.
4	cbc:CountrySubentity (0-1)	Definition: Provincia.
4	cac:Country (0-1)	Definition: Paese.
5	cbc:IdentificationCode (0-1)	Definition: Identificatore Paese.
	@listAgencyName [optional]	Definition: Agenzia che mantiene la lista di codici utilizzata. Default: UN/ECE
	@listID [optional]	Definition: Identificatore lista di codici utilizzata. Default: ISO3166-1
3	cac:PartyTaxScheme (1-1)	
4	cbc:CompanyID (0-1)	Definition: Numero Partita IVA.
4	cac:TaxScheme (1-1)	
5	cbc:ID (1-1)	Definition: Identificatore del regime fiscale.
3	cac:PartyLegalEntity (0-1)	
4	cbc:RegistrationName (0-1)	Definition: Il nome del soggetto come registrato presso l'autorità competente.
4	cbc:CompanyID (1-1)	Definition: Identificativo del soggetto come registrata mediante lo schema di registrazione aziendale.
4	cac:RegistrationAddress (0-1)	Definition: Indirizzo legale della parte nell'ambito di uno schema di registrazione aziendale.
5	cbc:ID (0-1)	Definition: Identificatore indirizzo (utilizzare se l'indirizzo e' registrato in uno schema di registrazione indirizzi).
5	cbc:Postbox (0-1)	Definition: Numero di casella postale.
5	cbc:StreetName (0-1)	Definition: Nome via, piazza, ...
5	cbc:BuildingNumber (0-1)	Definition: Numero civico.
5	cbc:Department (0-1)	Definition: Nome dipartimento.
5	cbc:PostalZone (0-1)	Definition: CAP.
5	cbc:CountrySubentity (0-1)	Definition: Provincia.
5	cac:Country (0-1)	Definition: Paese.
6	cbc:IdentificationCode (0-1)	Definition: Identificatore Paese.
	@listAgencyName [optional]	Definition: Agenzia che mantiene la lista di codici utilizzata. Default: UN/ECE
	@listID [optional]	Definition: Identificatore lista di codici utilizzata. Default: ISO3166-1
3	cac:Contact (0-1)	Definition: Contatti.
4	cbc:Telephone (0-1)	Definition: Numero di telefono.
4	cbc:Telefax (0-1)	Definition: Numero fax.
4	cbc:ElectronicMail (0-1)	Definition: Indirizzo di posta elettronica.
3	cac:Person (0-1)	Definition: Persona di riferimento.
4	cbc:FirstName (0-1)	Definition: Nome.
4	cbc:FamilyName (0-1)	Definition: Cognome.
4	cbc:MiddleName (0-1)	Definition: Secondo nome o iniziali.
4	cbc:JobTitle (0-1)	Definition: Titolo professionale.
1	cac:AccountingCustomerParty (1-1)	Definition: Dati del cliente.
2	cac:Party (0-1)	additional constraint info: 11
3	cbc:EndpointID (0-1)	Definition: Identificatore di un end point associato al soggetto (ad es.EANLocation Number, GLN).
3	cac:PartyIdentification (0-1)	
4	cbc:ID (1-1)	Definition: Identificatore del soggetto (assegnato secondo le PEPPOL Policy sull'uso degli identificatori).
3	cac:PartyName (1-1)	
4	cbc:Name (1-1)	Definition: Denominazione o ragione sociale.
3	cac:PostalAddress (1-1)	Definition: Indirizzo sede. additional constraint info: 24

4	cbc:ID (0-1)	Definition: Identificatore indirizzo (utilizzare se l'indirizzo e' registrato in uno schema di registrazione indirizzi).
4	cbc:Postbox (0-1)	Definition: Numero di casella postale.
4	cbc:StreetName (0-1)	Definition: Nome via, piazza, ...
4	cbc:BuildingNumber (0-1)	Definition: Numero civico.
4	cbc:Department (0-1)	Definition: Nome dipartimento.
4	cbc:PostalZone (0-1)	Definition: CAP.
4	cbc:CountrySubentity (0-1)	Definition: Provincia.
4	cac:Country (0-1)	Definition: Paese.
5	cbc:IdentificationCode (0-1)	Definition: Identificatore Paese.
	@listAgencyName [optional]	Definition: Agenzia che mantiene la lista di codici utilizzata. Default: UN/ECE
	@listID [optional]	Definition: Identificatore lista di codici utilizzata. Default: ISO3166-1
3	cac:PartyTaxScheme (1-1)	
4	cbc:CompanyID (0-1)	Definition: Numero Partita IVA.
4	cac:TaxScheme (1-1)	
5	cbc:ID (1-1)	Definition: Identificatore del regime fiscale.
3	cac:PartyLegalEntity (0-1)	
4	cbc:RegistrationName (0-1)	Definition: Il nome del soggetto come registrato presso l'autorità competente.
4	cbc:CompanyID (1-1)	Definition: Identificativo del soggetto come registrata mediante lo schema di registrazione aziendale.
4	cac:RegistrationAddress (0-1)	Definition: Indirizzo legale della parte nell'ambito di uno schema di registrazione aziendale.
5	cbc:ID (0-1)	Definition: Identificatore indirizzo (utilizzare se l'indirizzo e' registrato in uno schema di registrazione indirizzi).
5	cbc:Postbox (0-1)	Definition: Numero di casella postale.
5	cbc:StreetName (0-1)	Definition: Nome via, piazza, ...
5	cbc:BuildingNumber (0-1)	Definition: Numero civico.
5	cbc:Department (0-1)	Definition: Nome dipartimento.
5	cbc:PostalZone (0-1)	Definition: CAP.
5	cbc:CountrySubentity (0-1)	Definition: Provincia.
5	cac:Country (0-1)	Definition: Paese.
6	cbc:IdentificationCode (0-1)	Definition: Identificatore Paese.
	@listAgencyName [optional]	Definition: Agenzia che mantiene la lista di codici utilizzata. Default: UN/ECE
	@listID [optional]	Definition: Identificatore lista di codici utilizzata. Default: ISO3166-1
3	cac:Contact (0-1)	Definition: Contatti.
4	cbc:Telephone (0-1)	Definition: Numero di telefono.
4	cbc:Telefax (0-1)	Definition: Numero fax.
4	cbc:ElectronicMail (0-1)	Definition: Indirizzo di posta elettronica.
3	cac:Person (0-1)	Definition: Persona di riferimento.
4	cbc:FirstName (0-1)	Definition: Nome.
4	cbc:FamilyName (0-1)	Definition: Cognome.
4	cbc:MiddleName (0-1)	Definition: Secondo nome o iniziali.
4	cbc:JobTitle (0-1)	Definition: Titolo professionale.
1	cac:PayeeParty (0-1)	Definition: An association to the Payee.
2	cac:PartyIdentification (0-1)	
3	cbc:ID (1-1)	Definition: Identificatore del soggetto (assegnato secondo le PEPPOL Policy sull'uso degli identificatori).
2	cac:PartyName (1-1)	
3	cbc:Name (1-1)	Definition: Denominazione o ragione sociale.
2	cac:PartyLegalEntity (0-1)	
3	cbc:RegistrationName (0-1)	Definition: Il nome del soggetto come registrato presso l'autorità competente.
3	cbc:CompanyID (1-1)	Definition: Identificativo del soggetto come registrata mediante lo schema di registrazione aziendale.
3	cac:RegistrationAddress (0-1)	Definition: Indirizzo legale della parte nell'ambito di uno schema di registrazione aziendale.
4	cbc:ID (0-1)	Definition: Identificatore indirizzo (utilizzare se l'indirizzo e' registrato in uno schema di registrazione indirizzi).
4	cbc:Postbox (0-1)	Definition: Numero di casella postale.
4	cbc:StreetName (0-1)	Definition: Nome via, piazza, ...
4	cbc:BuildingNumber (0-1)	Definition: Numero civico.
4	cbc:Department (0-1)	Definition: Nome dipartimento.
4	cbc:PostalZone (0-1)	Definition: CAP.
4	cbc:CountrySubentity (0-1)	Definition: Provincia.
4	cac:Country (0-1)	Definition: Paese.
5	cbc:IdentificationCode (0-1)	Definition: Identificatore Paese.
	@listAgencyName [optional]	Definition: Agenzia che mantiene la lista di codici utilizzata. Default: UN/ECE
	@listID [optional]	Definition: Identificatore lista di codici utilizzata. Default: ISO3166-1
2	cac:Contact (0-1)	Definition: Contatti.
3	cbc:Telephone (0-1)	Definition: Numero di telefono.
3	cbc:Telefax (0-1)	Definition: Numero fax.
3	cbc:ElectronicMail (0-1)	Definition: Indirizzo di posta elettronica.
2	cac:Person (0-1)	Definition: Persona di riferimento.
3	cbc:FirstName (0-1)	Definition: Nome.

3	cbc.FamilyName (0-1)	Definition: Cognome.
3	cbc.MiddleName (0-1)	Definition: Secondo nome o iniziali.
3	cbc.JobTitle (0-1)	Definition: Titolo professionale.
1	cac:Delivery (0-1)	Definition: Informazioni sulla consegna.
2	cbc.ActualDeliveryDate (0-1)	Definition: Data di consegna richiesta.
2	cac:DeliveryLocation (0-1)	
3	cbc.ID (0-1)	Definition: Identificatore luogo di consegna (es. EAN LocationNumber, GLN).
3	cac:Address (0-1)	Definition: Association to the address of the location. additional constraint info: 14
4	cbc.ID (0-1)	Definition: Identificatore indirizzo (utilizzare se l'indirizzo e' registrato in uno schema di registrazione indirizzi).
4	cbc.Postbox (0-1)	Definition: Numero di casella postale.
4	cbc.StreetName (0-1)	Definition: Nome via, piazza, ...
4	cbc.BuildingNumber (0-1)	Definition: Numero civico
4	cbc.Department (0-1)	Definition: Nome dipartimento
4	cbc.PostalZone (0-1)	Definition: CAP.
4	cbc.CountrySubentity (0-1)	Definition: Provincia.
4	cac:Country (0-1)	Definition: Paese.
5	cbc:IdentificationCode (0-1)	Definition: Identificatore Paese.
	@listAgencyName [optional]	Definition: Agenzia che mantiene la lista di codici utilizzata. Default: UN/ECE
	@listID [optional]	Definition: Identificatore lista di codici utilizzata. Default: ISO3166-1
1	cac:PaymentMeans (0-unbounded)	Definition: Informazioni sulle modalita' di pagamento. additional constraint info: 5
2	cbc:PaymentMeansCode (1-1)	Definition: Metodo di pagamento espresso come codice.
	@listAgencyName [optional]	Definition: Agenzia che mantiene lista di codici utilizzata. Default: UN/ECE.
	@listID [optional]	Definition: Identificatore della lista di codici utilizzata. Default: UN/ECE 4461.
2	cbc:PaymentDueDate (0-1)	Definition: Data di pagamento.
2	cbc:PaymentChannelCode (0-1)	Definition: Canale utilizzato per il pagamento, espresso come codice.
	@listAgencyName [optional]	Definition: Agenzia che mantiene la lista di codici. UsageRule: Default: UN/EDIFACT 3055 code list
	@listID [optional]	Definition: Identificatore della lista di codici.
2	cbc:PaymentID (0-1)	Definition: Identificatore pagamento.
2	cac:PayeeFinancialAccount (0-1)	Definition: Dati del conto del pagante.
3	cbc.ID (1-1)	Definition: Identificatore conto (es. IBAN).
3	cac:FinancialInstitutionBranch (0-1)	Definition: Dati filiale dell'istituzione finanziaria.
4	cbc.ID (0-1)	Definition: Identificatore filiale.
4	cac:FinancialInstitution (0-1)	Definition: Dati istituzione finanziaria.
5	cbc.ID (1-1)	Definition: Identificatore istituzione finanziaria.
3	cac:Country (0-1)	
4	cbc:IdentificationCode (0-1)	Definition: Identificatore Paese.
	@listAgencyName [optional]	Definition: Agenzia che mantiene la lista di codici utilizzata. Default: UN/ECE
	@listID [optional]	Definition: Identificatore lista di codici utilizzata. Default: ISO3166-1
1	cac:PaymentTerms (0-1)	Definition: Informazioni sui termini di pagamento.
2	cbc:Note (0-1)	Definition: Termini di pagamento.
1	cac:AllowanceCharge (0-unbounded)	Definition: Sconti o addebiti che si applicano all'intera Fattura. additional constraint info: 13, 14, 9
2	cbc:ChargeIndicator (1-1)	Definition: Indicatore: valorizzare con "true" se si vuole inserire un costo, valorizzare con "false" se si vuole inserire uno sconto.
2	cbc:AllowanceChargeReason (0-1)	Definition: Motivazione per cui si sta applicando lo sconto, o costo, all fattura.
2	cbc:Amount (1-1)	Definition: Ammontare dello sconto, o costo.
	@currencyID [required]	Definition: Valuta. codelist info: CurrencyCodeContent [mandatory]
1	cac:TaxTotal (0-unbounded)	Definition: Dettaglio IVA.
2	cbc:TaxAmount (1-1)	Definition: Informazione sul subtotale di una particolare categoria d'imposta in un regime fiscale, come un'aliquota standard in ambito IVA (o VAT).
	@currencyID [required]	Definition: Valuta. codelist info: CurrencyCodeContent [mandatory]
2	cac:TaxSubtotal (1-1)	Definition: An association to Tax Subtotal. additional constraint info: 2
3	cbc:TaxableAmount (1-1)	Definition: Importo Imponibile.
	@currencyID [required]	Definition: Valuta. codelist info: CurrencyCodeContent [mandatory]
3	cbc:TaxAmount (1-1)	Definition: Importo Imposta
	@currencyID [required]	Definition: Valuta. codelist info: CurrencyCodeContent [mandatory]
3	cac:TaxCategory (1-1)	additional constraint info: 3
4	cbc.ID (1-1)	Definition: Identificatore categoria di imposta.
4	cbc:Percent (0-1)	Definition: L'aliquota di imposta per la categoria, espressa come percentuale.

4	cbc:TaxExemptionReasonCode (0-1)	Definition: La ragione dell'esenzione d'imposta espressa come codice (usare la code list CWA 15577 tax exemption. Versione 2006).
	@listAgencyName [optional]	Definition: Agenzia che mantiene la lista di codici. UsageRule: Default: UN/EDIFACT 3055 code list.
	@listID [optional]	Definition: Identificatore della lista di codici.
4	cbc:TaxExemptionReason (0-1)	Definition: La ragione dell'esenzione d'imposta espressa.
4	cac:TaxScheme (1-1)	Definition: Regime fiscale.
5	cbc:ID (1-1)	Definition: Identificatore del regime fiscale.
1	cac:LegalMonetaryTotal (1-1)	Definition: Totali fattura. additional constraint info: 1
2	cbc:LineExtensionAmount (1-1)	Definition: Il Totale degli importi di Riga al netto delle imposte e degli sconti concordati, ma inclusivo di qualsiasi importo di arrotondamento applicabile. Definition: Valuta. codelist info: CurrencyCodeContent [mandatory]
	@currencyID [required]	
2	cbc:TaxExclusiveAmount (1-1)	Definition: L'importo totale escluse le tasse. Definition: Valuta. codelist info: CurrencyCodeContent [mandatory]
	@currencyID [required]	
2	cbc:TaxInclusiveAmount (1-1)	Definition: L'importo totale incluse le tasse. Definition: Valuta. codelist info: CurrencyCodeContent [mandatory]
	@currencyID [required]	
2	cbc:AllowanceTotalAmount (0-1)	Definition: L'importo totale di tutti gli abbuoni. Definition: Valuta. codelist info: CurrencyCodeContent [mandatory]
	@currencyID [required]	
2	cbc:ChargeTotalAmount (0-1)	Definition: L'importo totale di tutti gli oneri. Definition: Valuta. codelist info: CurrencyCodeContent [mandatory]
	@currencyID [required]	
2	cbc:PrepaidAmount (0-1)	Definition: L'importo totale prepagato. Definition: Valuta. codelist info: CurrencyCodeContent [mandatory]
	@currencyID [required]	
2	cbc:PayableRoundingAmount (0-1)	Definition: L'importo di arrotondamento (positivo o negativo) aggiunto al Totale Importi Riga calcolato per arrotondario. Definition: Valuta. codelist info: CurrencyCodeContent [mandatory]
	@currencyID [required]	
2	cbc:PayableAmount (1-1)	Definition: L'importo totale da pagare. Definition: Valuta. codelist info: CurrencyCodeContent [mandatory]
	@currencyID [required]	
1	cac:InvoiceLine (1-unbounded)	Definition: Linea di fattura. additional constraint info: 21, 2
2	cbc:ID (1-1)	Definition: Identificatore linea di fattura.
2	cbc:Note (0-1)	Definition: Testo libero relativo alla Riga Fattura. Può contenere note o altre informazioni non esplicitamente previste nella struttura del documento.
2	cbc:InvoicedQuantity (0-1)	Definition: Quantità fatturata. Definition: Unità di misura. codelist info: UnitCodeContent [mandatory]
	@unitCode [optional]	
2	cbc:LineExtensionAmount (1-1)	Definition: Importo dovuto per questa linea al netto dell'IVA. Definition: Valuta. codelist info: CurrencyCodeContent [mandatory]
	@currencyID [required]	
2	cbc:AccountingCost (0-1)	Definition: Il codice contabile del Cliente riferito alla linea di fattura.
2	cac:OrderLineReference (0-unbounded)	
	cbc:LineID (1-1)	Definition: Riferimento ad una riga d'ordine.
2	cac:AllowanceCharge (0-unbounded)	additional constraint info: 13, 14
3	cbc:ChargeIndicator (1-1)	Definition: Indicatore: valorizzare con "true" se si vuole inserire un costo, valorizzare con "false" se si vuole inserire uno sconto.
3	cbc:AllowanceChargeReason (0-1)	Definition: Motivazione per cui si sta applicando lo sconto, o costo, alla fattura.
3	cbc:MultiplierFactorNumeric (0-1)	Definition: Fattore applicato all'importo base per calcolare gli sconti e i costi.
3	cbc:Amount (1-1)	Definition: Ammontare dello sconto, o costo. Definition: Valuta. codelist info: CurrencyCodeContent [mandatory]
	@currencyID [required]	
3	cbc:BaseAmount (0-1)	Definition: Importo al quale e' applicato il Multiplier Factor per il calcolo di sconti e costi. Definition: Valuta. codelist info: CurrencyCodeContent [mandatory]
	@currencyID [required]	
2	cac:TaxTotal (0-unbounded)	Definition: An association to Tax Total.
3	cbc:TaxAmount (1-1)	Definition: Informazione sul subtotale di una particolare categoria d'imposta in un regime fiscale, come un'aliquota standard in ambito IVA (o VAT). Definition: Valuta. codelist info: CurrencyCodeContent [mandatory]
	@currencyID [required]	

3 cac:TaxSubtotal (1-1)	<i>Definition:</i> An association to Tax Subtotal additional constraint info: 2
4 cbc:TaxableAmount (1-1) @currencyID [required]	<i>Definition:</i> Importo Imponibile. <i>Definition:</i> Valuta. codelist info: CurrencyCodeContent [mandatory]
4 cbc:TaxAmount (1-1) @currencyID [required]	<i>Definition:</i> Importo Imposta. <i>Definition:</i> Valuta. codelist info: CurrencyCodeContent [mandatory]
4 cac:TaxCategory (1-1)	additional constraint info: 3
5 cbc:ID (1-1)	<i>Definition:</i> Identificatore categoria di imposta.
5 cbc:Percent (0-1)	<i>Definition:</i> L'aliquota di imposta per la categoria, espressa come percentuale.
5 cbc:TaxExemptionReasonCode (0-1) @listAgencyName [optional] @listID [optional]	<i>Definition:</i> La ragione dell'esenzione d'imposta espressa come codice (usare la code list CWA 15577 tax exemption. Versione 2006). <i>Definition:</i> Agenzia che mantiene la lista di codici. <i>UsageRule:</i> Default: UNEDIFACT 3055 code list. <i>Definition:</i> Identificatore della lista di codici.
5 cbc:TaxExemptionReason (0-1)	<i>Definition:</i> La ragione dell'esenzione d'imposta espressa.
5 cac:TaxScheme (1-1)	<i>Definition:</i> Regime fiscale.
6 cbc:ID (1-1)	<i>Definition:</i> Identificatore del regime fiscale.
2 cac:Item (1-1)	<i>Definition:</i> An association to Item.
3 cbc:Description (0-1)	<i>Definition:</i> Descrizione dettagliata dell'articolo o servizio.
3 cbc:Name (1-1)	<i>Definition:</i> Nome sintetico.
3 cac:SellersItemIdentification (0-1)	<i>Definition:</i> Identificatore assegnato dal fornitore.
4 cbc:ID (1-1)	
3 cac:StandardItemIdentification (0-1)	<i>Definition:</i> Identificatore standard.
4 cbc:ID (1-1)	
3 cac:CommodityClassification (0-1)	<i>Definition:</i> Identificatore per la classificazione dell'articolo o servizio secondo un sistema di classificazione delle merci.
4 cbc:ItemClassificationCode (0-1) @listAgencyName [optional] @listID [optional]	<i>Definition:</i> The trade commodity classification, expressed as acode. <i>Definition:</i> Agenzia che mantiene la lista di codici. <i>UsageRule:</i> Default: UNEDIFACT 3055 code list. <i>Definition:</i> Identificatore della lista di codici. additional constraint info: 19
3 cac:ClassifiedTaxCategory (0-1)	<i>Definition:</i> Identifica la categoria di imposta che si applica all'articolo o servizio. additional constraint info: 14
4 cbc:ID (1-1)	<i>Definition:</i> Identificatore categoria di imposta.
4 cbc:Percent (0-1)	<i>Definition:</i> L'aliquota di imposta per la categoria, espressa come percentuale.
4 cac:TaxScheme (1-1)	<i>Definition:</i> Regime fiscale.
5 cbc:ID (1-1)	<i>Definition:</i> Identificatore del regime fiscale.
3 cac:AdditionalItemProperty (0-unbounded)	<i>Definition:</i> Caratteristiche.
4 cbc:Name (1-1)	<i>Definition:</i> Nome (es. tagoova, colore, anno, ...).
4 cbc:Value (1-1)	<i>Definition:</i> Valore.
2 cac:Price (0-1)	<i>Definition:</i> Prezzo.
3 cbc:PriceAmount (1-1) @currencyID [required]	<i>Definition:</i> Importo. <i>Definition:</i> Valuta. codelist info: CurrencyCodeContent [mandatory]
3 cbc:BaseQuantity (0-1) @unitCode [optional]	<i>Definition:</i> Quantità a cui si applica il prezzo. <i>Definition:</i> Unita' di misura. codelist info: UnitCodeContent [mandatory]
3 cac:AllowanceCharge (0-unbounded)	<i>Definition:</i> Informazioni su sconti o costi applicabili alla linea di fattura. > additional constraint info: 13 , 14
4 cbc:ChargeIndicator (1-1)	<i>Definition:</i> Indicatore: valorizzare con "true" se si vuole inserire un costo, valorizzare con "false" se si vuole inserire uno sconto.
4 cbc:AllowanceChargeReason (0-1)	<i>Definition:</i> Motivazione per cui si sta applicando lo sconto, o costo, alla fattura.
4 cbc:MultiplierFactorNumeric (0-1)	<i>Definition:</i> Fattore applicato all'importo base per calcolare gli sconti e i costi.
4 cbc:Amount (1-1) @currencyID [required]	<i>Definition:</i> Ammontare dello sconto, o costo. <i>Definition:</i> Valuta. codelist info: CurrencyCodeContent [mandatory]
4 cbc:BaseAmount (0-1) @currencyID [required]	<i>Definition:</i> Importo al quale e' applicato il Multiplier Factor per il calcolo di sconti e costi. <i>Definition:</i> Valuta. codelist info: CurrencyCodeContent [mandatory]

Phase ID: EUGENT10_phase - T10	
1) "/cac:LegalMonetaryTotal"	
• ASSERT (fatal)	
	number(cbc:PayableAmount) >= 0
	[EUGEN-T10-R019]-Total payable amount in an invoice MUST NOT be negative
2) "/cac:TaxSubtotal"	
• ASSERT (warning)	
	((cac:TaxCategory/cbc:ID = 'E') and (cac:TaxCategory/cbc:TaxExemptionReason or cac:TaxCategory/cbc:TaxExemptionReasonCode) or (cac:TaxCategory/cbc:ID != 'E'))
	[EUGEN-T10-R009]-If the category for VAT is exempt (E) then an exemption reason SHOULD be provided.
3) "/cac:TaxCategory"	
• ASSERT (fatal)	
	(parent::cac:AllowanceCharge) or (cbc:ID and cbc:Percent) or (cbc:ID = 'AE')
	[EUGEN-T10-R006]-For each tax subcategory except in Reverse Charge VAT the category ID and the applicable tax percentage MUST be provided.
4) "/cac:AccountingSupplierParty/cac:Party"	
• ASSERT (warning)	
	(cac:PostalAddress/cbc:StreetName and cac:PostalAddress/cbc:CityName and cac:PostalAddress/cbc:PostalZone and cac:PostalAddress/cac:Country/cbc:IdentificationCode)
	[EUGEN-T10-R001]-A supplier postal address in an invoice SHOULD contain at least, Street name and number, city name, zip code and country code.
5) "/cac:PaymentMeans"	
• ASSERT (warning)	
	((cbc:PaymentMeansCode = '31') and (cac:PayeeFinancialAccount/cbc:ID/@schemeID and cac:PayeeFinancialAccount/cbc:ID/@schemeID = 'IBAN') and (cac:PayeeFinancialAccount/cac:FinancialInstitutionBranch/cac:FinancialInstitution/cbc:ID/@schemeID and cac:PayeeFinancialAccount/cac:FinancialInstitutionBranch/cac:FinancialInstitution/cbc:ID/@schemeID = 'BIC')) or (cbc:PaymentMeansCode != '31') or ((cbc:PaymentMeansCode = '31') and (not(cac:PayeeFinancialAccount/cbc:ID/@schemeID) or (cac:PayeeFinancialAccount/cbc:ID/@schemeID != 'IBAN')))
	[EUGEN-T10-R004]-If the payment means are international account transfer and the account id is IBAN then the financial institution should be identified by using the BIC id.
6) "/cac:InvoicePeriod"	
• ASSERT (fatal)	
	(cbc:StartDate)
	[EUGEN-T10-R020]-If the invoice refers to a period, the period MUST have a start date.
• ASSERT (fatal)	
	(cbc:EndDate)
	[EUGEN-T10-R021]-If the invoice refers to a period, the period MUST have an end date.
7) "/cac:InvoiceLine"	
• ASSERT (warning)	
	(cbc:InvoicedQuantity and cbc:InvoicedQuantity/@unitCode)
	[EUGEN-T10-R003]-Each invoice line SHOULD contain the quantity and unit of measure
8) "/Invoice"	
• ASSERT (fatal)	
	((cac:TaxTotal[cac:TaxSubtotal/cac:TaxCategory/cac:TaxScheme/cbc:ID = 'VAT']/cbc:TaxAmount) and (cac:AccountingSupplierParty/cac:Party/cac:PartyTaxScheme/cbc:CompanyID) or not((cac:TaxTotal[cac:TaxSubtotal/cac:TaxCategory/cac:TaxScheme/cbc:ID = 'VAT'])))
	[EUGEN-T10-R007]-If the VAT total amount in an invoice exists it MUST contain the suppliers VAT number.
• ASSERT (fatal)	
	not(cac:PayeeParty) or (cac:PayeeParty/cac:PartyName/cbc:Name)
	[EUGEN-T10-R010]-If payee information is provided then the payee name MUST be specified.
• ASSERT (warning)	
	((/cac:AccountingSupplierParty/cac:Party/cac:PostalAddress/cac:Country/cbc:IdentificationCode != /cac:AccountingCustomerParty/cac:Party/cac:PostalAddress/cac:Country/cbc:IdentificationCode) or (/cac:AccountingSupplierParty/cac:Party/cac:PostalAddress/cac:Country/cbc:IdentificationCode != /cac:Delivery/cac:DeliveryLocation/cac:Address/cac:Country/cbc:IdentificationCode) and (/cac:TaxCategory/cbc:ID = 'AE' or not(/cac:TaxCategory/cbc:ID = 'AE'))
	[EUGEN-T10-R014]-An invoice with reverse charge VAT SHOULD have either a Customers country code or a Delivery address country code that is NOT EQUAL to the Suppliers address country code.
• ASSERT (fatal)	
	starts-with(/cac:AccountingCustomerParty/cac:Party/cac:PartyTaxScheme/cbc:CompanyID,/cac:AccountingCustomerParty/cac:Party/cac:PostalAddress/cac:Country/cbc:IdentificationCode) and (/cac:TaxCategory/cbc:ID = 'AE' or not(/cac:TaxCategory/cbc:ID = 'AE'))
	[EUGEN-T10-R015]-An invoice with reverse charge VAT MUST contain customers VAT id prefixed the customers country code.
• ASSERT (warning)	
	((/cac:TaxCategory/cbc:ID = 'AE') and not(/cac:TaxCategory/cbc:ID = 'AE')) or not(/cac:TaxCategory/cbc:ID = 'AE') or not(/cac:TaxCategory)
	[EUGEN-T10-R016]-An invoice with reverse charge VAT MAY NOT contain other VAT categories.
• ASSERT (fatal)	
	((cbc:TaxExclusiveAmount = /cac:TaxTotal[cac:TaxSubtotal[cac:TaxCategory/cbc:ID='AE']/cbc:TaxableAmount) and (/cac:TaxCategory/cbc:ID = 'AE' or not(/cac:TaxCategory/cbc:ID = 'AE'))
	[EUGEN-T10-R017]-The taxable amount for reverse charge tax category MUST equal the invoice taxable amount.

<p>• ASSERT (fatal)</p> <p>//cac:TaxTotal/cbc:TaxAmount = 0 and (//cac:TaxCategory/cbc:ID = 'AE' or not((//cac:TaxCategory/cbc:ID) = 'AE'))</p> <p>[EUGEN-T10-R018]-The tax amount for reverse charge VAT MUST be zero. (since there is only one VAT category allowed it follows that the invoice tax total for reverse charge invoices is zero)</p>
<p>9) //Invoice/cac:AllowanceCharge"</p> <p>• ASSERT (fatal)</p> <p>((//cac:TaxTotal/cac:TaxSubtotal/cac:TaxCategory/cac:TaxScheme/cbc:ID = 'VAT')/cbc:TaxAmount) and (cac:TaxCategory/cac:TaxScheme/cbc:ID = 'VAT') or not((//cac:TaxTotal/cac:TaxSubtotal/cac:TaxCategory/cac:TaxScheme/cbc:ID = 'VAT'))</p> <p>[EUGEN-T10-R006]-If the VAT total amount in an invoice exists then an Allowances Charges amount on document level MUST have Tax category for VAT.</p>
<p>10) //cac:Delivery/cac:DeliveryLocation/cac:Address"</p> <p>• ASSERT (warning)</p> <p>(cbc:CityName and cbc:PostalZone and cac:Country/cbc:IdentificationCode)</p> <p>[EUGEN-T10-R003]-A Delivery address in an invoice SHOULD contain at least, city, zip code and country code.</p>
<p>11) //cac:AccountingCustomerParty/cac:Party"</p> <p>• ASSERT (warning)</p> <p>(cac:PostalAddress/cbc:StreetName and cac:PostalAddress/cbc:CityName and cac:PostalAddress/cbc:PostalZone and cac:PostalAddress/cac:Country/cbc:IdentificationCode)</p> <p>[EUGEN-T10-R002]-A customer postal address in an invoice SHOULD contain at least, Street name and number, city name, zip code and country code.</p>
<p>12) //cac:Item/cac:ClassifiedTaxCategory"</p> <p>• ASSERT (fatal)</p> <p>((//cac:TaxTotal/cac:TaxSubtotal/cac:TaxCategory/cac:TaxScheme/cbc:ID = 'VAT')/cbc:TaxAmount and cbc:ID) or not((//cac:TaxTotal/cac:TaxSubtotal/cac:TaxCategory/cac:TaxScheme/cbc:ID = 'VAT'))</p> <p>[EUGEN-T10-R011]-If the VAT total amount in an invoice exists then each invoice line item must have a VAT category ID.</p>
<p>13) //cac:AllowanceCharge"</p> <p>• ASSERT (fatal)</p> <p>number(Amount)>=0</p> <p>[EUGEN-T10-R022]-An allowance or charge amount MUST NOT be negative.</p>
<p>14) //cac:AllowanceCharge[cbc:ChargeIndicator='false']"</p> <p>• ASSERT (fatal)</p> <p>number(cbc:MultiplierFactorNumeric) >=0</p> <p>[EUGEN-T10-R012]-An allowance percentage MUST NOT be negative.</p> <p>• ASSERT (warning)</p> <p>(cbc:MultiplierFactorNumeric and cbc:BaseAmount) or (not(cbc:MultiplierFactorNumeric) and not(cbc:BaseAmount))</p> <p>[EUGEN-T10-R013]-In allowances, both or none of percentage and base amount SHOULD be provided</p>
<p style="text-align: center;">Phase ID: codelist_phase - CodesT10</p>
<p>15) "cac:FinancialInstitution/cbc:ID//@schemeID"</p> <p>• ASSERT (warning)</p> <p>((not(contains(normalize-space(),' ')) and contains('BIC',concat(' ',normalize-space(),' ')))</p> <p>[PCL-010-002]-Financial Institution SHOULD ONLY be BIC code.</p>
<p>16) "cac:PostalAddress/cbc:ID//@schemeID"</p> <p>• ASSERT (warning)</p> <p>((not(contains(normalize-space(),' ')) and contains('GLN',concat(' ',normalize-space(),' ')))</p> <p>[PCL-010-003]-Postal address identifiers SHOULD ONLY be GLN.</p>
<p>17) "cac:Delivery/cac:DeliveryLocation/cbc:ID//@schemeID"</p> <p>• ASSERT (warning)</p> <p>((not(contains(normalize-space(),' ')) and contains('GLN',concat(' ',normalize-space(),' ')))</p> <p>[PCL-010-004]-Location identifiers SHOULD ONLY be GLN.</p>
<p>18) "cac:Item/cac:StandardItemIdentification/cbc:ID//@schemeID"</p> <p>• ASSERT (warning)</p> <p>((not(contains(normalize-space(),' ')) and contains('GTIN',concat(' ',normalize-space(),' ')))</p> <p>[PCL-010-005]-Standard item identifiers SHOULD ONLY be GTIN.</p>
<p>19) "cac:Item/cac:CommodityClassification/cbc:ItemClassificationCode//@listID"</p> <p>• ASSERT (warning)</p> <p>((not(contains(normalize-space(),' ')) and contains('UNSPSC eCLASS CPV',concat(' ',normalize-space(),' ')))</p> <p>[PCL-010-006]-Commodity classification SHOULD ONLY be one of UNSPSC, eClass or CPV.</p>
<p style="text-align: center;">Phase ID: ITNATT10_phase - T10-IT</p>
<p>20) //cac:AccountingSupplierParty/cac:Party/cac:PostalAddress"</p> <p>• ASSERT (Fatal)</p>

(cbc:StreetName and cbc:BuildingNumber and cbc:CityName and cbc:PostalZone and cbc:CountrySubentity and cac:Country/cbc:IdentificationCode) and (/cac:AccountingSupplierParty/cac:Party/cac:PostalAddress/cac:Country/cbc:IdentificationCode = 1T) or not((/cac:AccountingSupplierParty/cac:Party/cac:PostalAddress/cac:Country/cbc:IdentificationCode = 1T'))
[!T-T10-005]-A suppliers postal address in an invoice MUST contain at least, Street name and number, city name, zip code, country subentity and country code.
21) //cac:InvoiceLine
• ASSERT (Fatal)
(cbc:InvoicedQuantity) and (cbc:InvoicedQuantity/@unitCode) and (/cac:AccountingSupplierParty/cac:Party/cac:PostalAddress/cac:Country/cbc:IdentificationCode = 1T) or not((/cac:AccountingSupplierParty/cac:Party/cac:PostalAddress/cac:Country/cbc:IdentificationCode = 1T'))
[!T-T10-024]-Each invoice line MUST contain the quantity and unit of measure.
• ASSERT (Fatal)
(cac:Price/cbc:PriceAmount) and (/cac:AccountingSupplierParty/cac:Party/cac:PostalAddress/cac:Country/cbc:IdentificationCode = 1T) or not((/cac:AccountingSupplierParty/cac:Party/cac:PostalAddress/cac:Country/cbc:IdentificationCode = 1T'))
[!T-T10-031]-If the supplier country code is "1T", each invoice line MUST contain the product/service unit price.
22) //Invoice
• ASSERT (Fatal)
cbc:InvoiceTypeCode and (/cac:AccountingSupplierParty/cac:Party/cac:PostalAddress/cac:Country/cbc:IdentificationCode = 1T) or not((/cac:AccountingSupplierParty/cac:Party/cac:PostalAddress/cac:Country/cbc:IdentificationCode = 1T'))
[!T-T10-016]-If the supplier country code is "1T", an invoice MUST contain the invoice type.
23) //cac:DespatchDocumentReference
• ASSERT (Fatal)
(cbc:ID and cbc:IssueDate and cbc:DocumentType) and (/cac:AccountingSupplierParty/cac:Party/cac:PostalAddress/cac:Country/cbc:IdentificationCode = 1T) or not((/cac:AccountingSupplierParty/cac:Party/cac:PostalAddress/cac:Country/cbc:IdentificationCode = 1T'))
[!T-T10-011]-If the supplier country code is "1T", the reference to the transport document in an invoice MUST contain document identifier, issue date, reference law.
24) //cac:AccountingCustomerParty/cac:Party/cac:PostalAddress
• ASSERT (Fatal)
(cbc:StreetName and cbc:BuildingNumber and cbc:CityName and cbc:PostalZone and cbc:CountrySubentity and cac:Country/cbc:IdentificationCode) and (/cac:AccountingSupplierParty/cac:Party/cac:PostalAddress/cac:Country/cbc:IdentificationCode = 1T) or not((/cac:AccountingSupplierParty/cac:Party/cac:PostalAddress/cac:Country/cbc:IdentificationCode = 1T'))
[!T-T10-008]-A customer postal address in an invoice MUST contain at least, Street name and number, city name, zip code, country subentity and country code.
25) //cac:AccountingSupplierParty/cac:Party
• ASSERT (Warning)
not((cac:PartyLegalEntity/cbc:CompanyID/@schemeID = 1T:CC) or (cac:PartyLegalEntity/cbc:CompanyID/@schemeID = 1T:CC)/cac:CorporateRegistrationScheme/cac:JurisdictionRegionAddress/cbc:CountrySubentity) or (cac:PartyLegalEntity/cbc:CompanyID/@schemeID = 1T:CC)/cac:CorporateRegistrationScheme/cac:JurisdictionRegionAddress/cbc:CountrySubentityCode) and (/cac:AccountingSupplierParty/cac:Party/cac:PostalAddress/cac:Country/cbc:IdentificationCode = 1T) or not((/cac:AccountingSupplierParty/cac:Party/cac:PostalAddress/cac:Country/cbc:IdentificationCode = 1T'))
[!T-T10-013]-If the supplier country code is "1T" and is registered in the Italian Chamber of Commerce, the information about supplier's Items Registration Company SHOULD include Country Subentity (as code or text) of Chambers of Commerce of company register.
26) //cac:TaxRepresentativeParty
• ASSERT (Warning)
(cac:PartyTaxScheme/cbc:CompanyID/@schemeID = 1T:VAT) and cac:PartyName/cbc:Name) and (/cac:AccountingSupplierParty/cac:Party/cac:PostalAddress/cac:Country/cbc:IdentificationCode = 1T) or not((/cac:AccountingSupplierParty/cac:Party/cac:PostalAddress/cac:Country/cbc:IdentificationCode = 1T'))
[!T-T10-003]-If Tax Representative Party exists and its country code is "1T", an invoice MUST contain VAT Company Identifier and name of the Tax Representative Party.
27) //cac:DocumentReference
• ASSERT (Fatal)
(cbc:ID and cbc:IssueDate and cbc:DocumentType) and (/cac:AccountingSupplierParty/cac:Party/cac:PostalAddress/cac:Country/cbc:IdentificationCode = 1T) or not((/cac:AccountingSupplierParty/cac:Party/cac:PostalAddress/cac:Country/cbc:IdentificationCode = 1T'))
[!T-T10-032]-If the supplier country code is "1T", the reference to the transport document at line level in an invoice MUST contain document identifier, issue date, reference law.

Appendice i - CodeList: BinaryObjectMimeCodeContent

```

- application/CSTAdata+xml
- application/EDI-Consent
- application/EDI-X12
- application/EDIFACT
- application/activemessage
- application/andrew-inset
- application/applefile
- application/atomicmail
- application/batch-SMTP
- application/beep+xml
- application/cals-1840
- application/cnrp+xml
- application/comonground
- application/cpi+xml
- application/csta+xml
- application/cybercash
- application/dca-rtf
- application/de-cdx
- application/dialog-info+xml
- application/dicom
- application/dns
- application/dvcs
- application/epf+xml
- application/eshop
- application/fits
- application/font-tdpfr
- application/http
- application/hyperstudio
- application/iges
- application/im-is-composing+xml
- application/index
- application/index.cmd
- application/index.obj
- application/index.response
- application/index.vnd
- application/iotp
- application/ipp
- application/isup
- application/kpml-request+xml
- application/kpml-response+xml
- application/mca-binhex40
- application/mc-cwrtel
- application/marc
- application/mathematica
- application/mbox
- application/mikey
- application/mpg4-generic
- application/msword
- application/news-message-id
- application/news-transmission
- application/ocsp-request
- application/ocsp-response
- application/octet-stream
- application/oda
- application/ogg
- application/parityfec
- application/pdf
- application/pgp-encrypted
- application/pgp-keys
- application/pgp-signature
- application/pidf+xml
- application/pkcs10
- application/pkcs7-mime
- application/pkcs7-signature
- application/pkix-cert
- application/pkix-crl
- application/pkix-pkipath
- application/pkixcmp
- application/postscript
- application/prs.alvestrand.titrax-sheet
- application/prs.cww
- application/prs.nprend
- application/prs.plucker
- application/qsig
- application/rdfx+xml
- application/reginfo+xml
- application/remote-printing
- application/resource-lists+xml
- application/riscos
- application/ris-services+xml
- application/rtf
- application/samlassertion+xml
- application/samlmetadata+xml
- application/sbml+xml
- application/sdp
- application/set-payment
- application/set-payment-initiation
- application/set-registration
- application/set-registration-initiation
- application/sgml
- application/sgml-open-catalog
- application/shf+xml
- application/sieve
- application/simple-filter+xml
- application/simple-message-summary
- application/slate
- application/soap+xml
- application/spirits-event+xml
- application/timestamp-query
- application/timestamp-reply
- application/ve-trigger
- application/vemml
- application/vnd.3M.Post-it-Notes
- application/vnd.3gpp.pic-bw-large
- application/vnd.3gpp.pic-bw-small
- application/vnd.3gpp.pic-bw-var
- application/vnd.3gpp.sms
- application/vnd.FloGraphit
- application/vnd.Kinar
- application/vnd.Mobius.DAF
- application/vnd.Mobius.DAS
- application/vnd.Mobius.DIS
- application/vnd.Mobius.MBK
- application/vnd.Mobius.MQY
- application/vnd.Mobius.MSL
- application/vnd.Mobius.PLC
- application/vnd.Mobius.TXF
- application/vnd.Quark.QuarkXPress
- application/vnd.RenLearn.rprint
- application/vnd.ccpac.simpily.aso
- application/vnd.ccpac.simpily.imp
- application/vnd.acucobol
- application/vnd.acucorp
- application/vnd.adobe.xfdf
- application/vnd.aether.imp
- application/vnd.amiga.ami
- application/vnd.answer-web-certificate-issue-
  initiation
- application/vnd.answer-web-funds-transfer-
  initiation
- application/vnd.audiograph
- application/vnd.blueice.multipass
- application/vnd.bml
- application/vnd.businessobjects
- application/vnd.canon-cpdl
- application/vnd.canon-lips
- application/vnd.cinderella
- application/vnd.claymore
- application/vnd.commerce-battelle
- application/vnd.commonspace
- application/vnd.contact.cmsg
- application/vnd.cosmocaller
- application/vnd.criticaltools.wbs+xml
- application/vnd.ctc-posml
- application/vnd.cups-postscript
- application/vnd.cups-raster
- application/vnd.cups-raw
- application/vnd.curl
- application/vnd.cybank
- application/vnd.data-vision.rdz
- application/vnd.dna
- application/vnd.dgraph
- application/vnd.dreamfactory
- application/vnd.dxr
- application/vnd.ecdis-update
- application/vnd.ecowin.chart
- application/vnd.ecowin.fileupdate
- application/vnd.ecowin.fileupdate
- application/vnd.ecowin.series
- application/vnd.ecowin.seriesrequest
- application/vnd.ecowin.seriesupdate
- application/vnd.enliven
- application/vnd.epson.esf
- application/vnd.epson.msf
- application/vnd.epson.quickanime
- application/vnd.epson.salt
- application/vnd.epson.ssf
- application/vnd.ericsson.quickcall
- application/vnd.audora.data
- application/vnd.fdf
- application/vnd.ffns
- application/vnd.fints
- application/vnd.frame-maker
- application/vnd.fsc.weblaunch
- application/vnd.fujitsu.oasys
- application/vnd.fujitsu.oasys2
- application/vnd.fujitsu.oasys3
- application/vnd.fujitsu.oasyspp
- application/vnd.fujitsu.oasysprs
- application/vnd.fujixerox.ddd
- application/vnd.fujixerox.docuworks
- application/vnd.fujixerox.docuworks.binder
- application/vnd.fut-misnet
- application/vnd.genomatix.tuxedo
- application/vnd.grafeq
- application/vnd.groove-account
- application/vnd.groove-help
- application/vnd.groove-identity-message
- application/vnd.groove-injector
- application/vnd.groove-tool-message
- application/vnd.groove-tool-template
- application/vnd.groove-vcard
- application/vnd.hbci
- application/vnd.hci-bereports
- application/vnd.hhe.lesson-player
- application/vnd.hp-HPGL
- application/vnd.hp-PCL
- application/vnd.hp-PCLXL
- application/vnd.hp-hpid
- application/vnd.hp-hps
- application/vnd.httpphone
- application/vnd.hzn-3d-crossword
- application/vnd.ibm.Minipay
- application/vnd.ibm.afplinedata
- application/vnd.ibm.electronic-media
- application/vnd.ibm.modcap
- application/vnd.ibm.rights-management
- application/vnd.ibm.secure-container
- application/vnd.informix-visionary
- application/vnd.intercon.formnet
- application/vnd.intertrust.digibox
- application/vnd.intertrust.nncp
- application/vnd.intu.qbo
- application/vnd.intu.qfx

```

```

- application/vnd.ipunplugged.rcprofile
- application/vnd.repository.package+xml
- application/vnd.is-xpr
- application/vnd.japannet-directory-service
- application/vnd.japannet-jpnstore-wakeup
- application/vnd.japannet-payment-wakeup
- application/vnd.japannet-registration
- application/vnd.japannet-registration-wakeup
- application/vnd.japannet-setstore-wakeup
- application/vnd.japannet-verification
- application/vnd.japannet-verification-wakeup
- application/vnd.jsp
- application/vnd.kde.karbon
- application/vnd.kde.kchart
- application/vnd.kde.kformula
- application/vnd.kde.kivio
- application/vnd.kde.kontour
- application/vnd.kde.kpresenter
- application/vnd.kde.kspread
- application/vnd.kde.kword
- application/vnd.kenameapp
- application/vnd.kidspiration
- application/vnd.koan
- application/vnd.liberty-request+xml
- application/vnd.lamasgraphics.life-balance.desktop
- application/vnd.lamasgraphics.life-balance.exchange+xml
- application/vnd.lotus-1-2-3
- application/vnd.lotus-approach
- application/vnd.lotus-freelance
- application/vnd.lotus-notes
- application/vnd.lotus-organizer
- application/vnd.lotus-screencam
- application/vnd.lotus-wordpro
- application/vnd.mcd
- application/vnd.medstation.cdkey
- application/vnd.meridian-slingshot
- application/vnd.mfmp
- application/vnd.micrografx.flo
- application/vnd.micrografx.igx
- application/vnd.mif
- application/vnd.minisoff-hp3000-save
- application/vnd.mitsubishi.misty-guard.trustweb
- application/vnd.mophun.application
- application/vnd.mophun.certificate
- application/vnd.motorola.flexsuite
- application/vnd.motorola.flexsuite.alsi
- application/vnd.motorola.flexsuite.fis
- application/vnd.motorola.flexsuite.golap
- application/vnd.motorola.flexsuite.kmr
- application/vnd.motorola.flexsuite.itc
- application/vnd.motorola.flexsuite.wem
- application/vnd.mozilla.xul+xml
- application/vnd.ms-artgaly
- application/vnd.ms-asf
- application/vnd.ms-excel
- application/vnd.ms-frm
- application/vnd.ms-powerpoint
- application/vnd.ms-project
- application/vnd.ms-thef
- application/vnd.ms-works
- application/vnd.ms-wpl
- application/vnd.mseq
- application/vnd.msimgn
- application/vnd.music-niff
- application/vnd.musician
- application/vnd.nervana
- application/vnd.netfx
- application/vnd.noblenet-directory
- application/vnd.noblenet-sealer
- application/vnd.noblenet-web
- application/vnd.nokia.landmark-wbxml
- application/vnd.nokia.landmark+xml
- application/vnd.nokia.landmarkcollection+xml
- application/vnd.nokia.radio-preset
- application/vnd.nokia.radio-presets
- application/vnd.novadigm.EDM
- application/vnd.novadigm.EDX
- application/vnd.novadigm.EXT
- application/vnd.obn
- application/vnd.omads-email+xml
- application/vnd.omads-file+xml
- application/vnd.omads-folder+xml
- application/vnd.osa.netdeploy
- application/vnd.palm
- application/vnd.paos.xml
- application/vnd.pg.format
- application/vnd.pg.osaill
- application/vnd.picseal
- application/vnd.powerbuilder6
- application/vnd.powerbuilder6-s
- application/vnd.powerbuilder7
- application/vnd.powerbuilder7-s
- application/vnd.powerbuilder75
- application/vnd.powerbuilder75-s
- application/vnd.previewsystems.box
- application/vnd.publshare-delta-tree
- application/vnd.pvi.ptid1
- application/vnd.pwg-multiplexed
- application/vnd.pwg-xml-print+xml
- application/vnd.rapid
- application/vnd.s3ms
- application/vnd.sealed.doc
- application/vnd.sealed.eml
- application/vnd.sealed.mht
- application/vnd.sealed.net
- application/vnd.sealed.ppt
- application/vnd.sealed.xls
- application/vnd.sealedmedia.softseal.html
- application/vnd.sealedmedia.softseal.pdf
- application/vnd.seemail
- application/vnd.shana.informed.formdata
- application/vnd.shana.informed.formtemplate
- application/vnd.shana.informed.interchange
- application/vnd.shana.informed.package
- application/vnd.smaf
- application/vnd.sss-cod
- application/vnd.sss-dtf
- application/vnd.sss-ntf
- application/vnd.street-stream
- application/vnd.sus-calendar
- application/vnd.svd
- application/vnd.swiftview-ics
- application/vnd.syncml+xml
- application/vnd.syncml.ds.notification
- application/vnd.triscape.mxs
- application/vnd.trueapp
- application/vnd.truedoc
- application/vnd.ufdl
- application/vnd.uiq.theme
- application/vnd.uplanet.alert
- application/vnd.uplanet.alert-wbxml
- application/vnd.uplanet.bearer-choice
- application/vnd.uplanet.bearer-choice-wbxml
- application/vnd.uplanet.ca cheop
- application/vnd.uplanet.ca cheop-wbxml
- application/vnd.uplanet.channel
- application/vnd.uplanet.channel-wbxml
- application/vnd.uplanet.list
- application/vnd.uplanet.list-wbxml
- application/vnd.uplanet.listcmd
- application/vnd.uplanet.listcmd-wbxml
- application/vnd.uplanet.signal
- application/vnd.vcx
- application/vnd.vectorworks
- application/vnd.vidsoft.vidconference
- application/vnd.visio
- application/vnd.visionary
- application/vnd.vividence.scriptfile
- application/vnd.vsf
- application/vnd.wap.sic
- application/vnd.wap.sic
- application/vnd.wap.wbxml
- application/vnd.wap.wmlc
- application/vnd.wap.wmlscriptc
- application/vnd.webturbo
- application/vnd.wordperfect
- application/vnd.wqd
- application/vnd.wrq-hp3000-labelled
- application/vnd.wt.stf
- application/vnd.wv.csp+wbxml
- application/vnd.wv.csp+xml
- application/vnd.wv.ssp+xml
- application/vnd.xara
- application/vnd.xfdl
- application/vnd.yamaha.hv-dic
- application/vnd.yamaha.hv-script
- application/vnd.yamaha.hv-voice
- application/vnd.yamaha.smaf-audio
- application/vnd.yamaha.smaf-phrase
- application/vnd.yellowriver-custom-menu
- application/watcherinfo+xml
- application/whoispp-query
- application/whoispp-response
- application/wita
- application/wordperfect5.1
- application/x400-bp
- application/xhtml+xml
- application/xml
- application/xml-dtd
- application/xml-external-parsed-entity
- application/xmpp+xml
- application/xop+xml
- application/zip
- audio/32kadpcm
- audio/3gpp
- audio/AMR
- audio/AMR-WB
- audio/BV16
- audio/BV32
- audio/CN
- audio/DAT12
- audio/DV14
- audio/EVRC
- audio/EVRC-QCP
- audio/EVRC0
- audio/G.722.1
- audio/G722
- audio/G723
- audio/G726-16
- audio/G726-24
- audio/G726-32
- audio/G726-40
- audio/G728
- audio/G729
- audio/G729D
- audio/G729E
- audio/GSM
- audio/GSM-EFR
- audio/L16
- audio/L20
- audio/L24
- audio/L8
- audio/LPC
- audio/MP4A-LATM
- audio/MPA
- audio/PCMA
- audio/PCMU
- audio/QCELP
- audio/RED

```

```

- audio/SMV
- audio/SMV-GCP
- audio/SMV0
- audio/VDVI
- audio/basic
- audio/clearmode
- audio/dsr-es201108
- audio/dsr-es202050
- audio/dsr-es202211
- audio/dsr-es202212
- audio/ILBC
- audio/mpa-robust
- audio/mpeg
- audio/mpeg4-generic
- audio/parityfec
- audio/prs.sid
- audio/telephone-event
- audio/tones
- audio/vnd.3gpp.iufp
- audio/vnd.audiokoz
- audio/vnd.cisco.nse
- audio/vnd.cns.anp1
- audio/vnd.cns.inft1
- audio/vnd.digital-winds
- audio/vnd.everad.plj
- audio/vnd.lucent.voice
- audio/vnd.nokia.mobile-xmf
- audio/vnd.nortel.vbk
- audio/vnd.nuera.ecelp4800
- audio/vnd.nuera.ecelp7470
- audio/vnd.nuera.ecelp9600
- audio/vnd.octel.sbc
- audio/vnd.qcelp
- audio/vnd.rhetorex.32ka-dpcm
- audio/vnd.sealedmedia.softseal.mpeg
- audio/vnd.vmx.csvd
- image/cgm
- image/flts
- image/g3fax
- image/gif
- image/ief
- image/jpeg
- image/jpm
- image/jpx
- image/naplps
- image/png
- image/prs.btif
- image/prs.pti
- image/t38
- image/tiff
- image/tiff-fx
- image/vnd.cns.inf2
- image/vnd.djvu
- image/vnd.dwg
- image/vnd.dxf
- image/vnd.fastbidsheet
- image/vnd.fpx
- image/vnd.fst
- image/vnd.fujixerox.edmics-mmr
- image/vnd.fujixerox.edmics-ric
- image/vnd.globalgraphics.pgb
- image/vnd.microsoft.icon
- image/vnd.mix
- image/vnd.ms-modi
- image/vnd.net-fpx
- image/vnd.sealed.png
- image/vnd.sealedmedia.softseal.gif
- image/vnd.sealedmedia.softseal.jpg
- image/vnd.svf
- image/vnd.wap.wbmp
- image/vnd.xiff
- message/CPIM
- message/delivery-status
- message/disposition-notification
- message/external-body
- message/http
- message/news
- message/partial
- message/rfc822
- message/s-http
- message/sip
- message/sipfrag
- message/tracking-status
- model/iges
- model/mesh
- model/vnd.dwf
- model/vnd.flatland.3dml
- model/vnd.gdl
- model/vnd.gs-gdl
- model/vnd.gtw
- model/vnd.mts
- model/vnd.parasolid.transmit.binary
- model/vnd.parasolid.transmit.text
- model/vnd.vtu
- model/vrml
- multipart/alternative
- multipart/appledouble
- multipart/byteranges
- multipart/digest
- multipart/encrypted
- multipart/form-data
- multipart/header-set
- multipart/mixed
- multipart/parallel
- multipart/related
- multipart/report
- multipart/signed
- multipart/voice-message
- text/RED
- text/calendar
- text/css
- text/csv
- text/directory
- text/dns
- text/enriched
- text/html
- text/parityfec
- text/plain
- text/prs.fallenstein.rst
- text/prs.lines.tag
- text/rfc822-headers
- text/richtext
- text/rtf
- text/sgml
- text/t140
- text/tab-separated-values
- text/troff
- text/uri-list
- text/vnd.dmclientScript
- text/vnd.IPTC.NITF
- text/vnd.IPTC.NewsML
- text/vnd.abc
- text/vnd.curl
- text/vnd.esmertec.theme-descriptor
- text/vnd.fly
- text/vnd.fmi.flexstor
- text/vnd.in3d.3dml
- text/vnd.in3d.spot
- text/vnd.latex-z
- text/vnd.motorola.reflex
- text/vnd.ms-mediapackage
- text/vnd.net2phone.commcenter.command
- text/vnd.sun.j2me.app-descriptor
- text/vnd.wap.sis
- text/vnd.wap.sis
- text/vnd.wap.wml
- text/vnd.wap.wmlscript
- text/xml
- text/xml-external-parsed-entity
- video/3gpp
- video/BMPEG
- video/BT656
- video/CEIB
- video/DV
- video/H261
- video/H263
- video/H263-1998
- video/H263-2000
- video/H264
- video/JPEG
- video/MJ2
- video/MP1S
- video/MP2P
- video/MP2T
- video/MP4V-ES
- video/MPV
- video/SMPT292M
- video/mpeg
- video/mpeg4-generic
- video/ivr
- video/parityfec
- video/pointer
- video/quicktime
- video/raw
- video/vnd.fvt
- video/vnd.motorola.video
- video/vnd.motorola.videoop
- video/vnd.mpegurl
- video/vnd.nokia.interleaved-multimedia
- video/vnd.objectvideo
- video/vnd.sealed.mpeg1
- video/vnd.sealed.mpeg4
- video/vnd.sealed.swf
- video/vnd.sealedmedia.softseal.mov
- video/vnd.vivo

```

Appendice ii - CodeList: CurrencyCodeContent

- AED - Dirham	- IDR - Rupiah	- STD - Dobra
- AFN - Afghani	- ILS - New Israeli Sheqel	- SVC - El Salvador Colon
- ALL - Lek	- INR - Indian Rupee	- SYP - Syrian Pound
- AMD - Dram	- IQD - Iraqi Dinar	- SZL - Lilangeni
- ANG - Netherlands Antillian Guilder	- IRR - Iranian Rial	- THB - Baht
- AOA - Kwanza	- ISK - Iceland Krona	- TJS - Somoni
- ARS - Argentine Peso	- JMD - Jamaican Dollar	- TMM - Manat
- AUD - Australian Dollar	- JOD - Jordanian Dinar	- TND - Tunisian Dinar
- AWG - Aruban Guilder	- JPY - Yen	- TOP - Pa'anga
- AZM - Azerbaijanian Manat	- KES - Kenyan Shilling	- TRL - Turkish Lira
- BAM - Convertible Mark	- KGS - Som	- TTD - Trinidad and Tobago Dollar
- BBD - Barbados Dollar	- KHR - Riel	- TWD - New Taiwan Dollar
- BDT - Taka	- KMF - Comoro Franc	- TZS - Tanzanian Shilling
- BGN - Bulgarian Lev	- KPW - North Korean Won	- UAH - Hryvnia
- BHD - Bahraini Dinar	- KRW - Won	- UGX - Uganda Shilling
- BIF - Burundi Franc	- KWD - Kuwaiti Dinar	- USD - US Dollar
- BMD - Bermudian Dollar (customarily: Bermuda Dollar)	- KYD - Cayman Islands Dollar	- UYU - Peso Uruguayo
- BND - Brunei Dollar	- KZT - Tenge	- UZS - Uzbekistan Sum
- BOB - Boliviano	- LAK - Kip	- VEB - Bolivar
- BRL - Brazilian Real	- LBP - Lebanese Pound	- VND - Dong
- BSD - Bahamian Dollar	- LKR - Sri Lanka Rupee	- VUV - Vatu
- BTN - Ngultrum	- LRD - Liberian Dollar	- WST - Tala
- BWP - Pula	- LSL - Loti	- XAF - CFA Franc
- BYR - Belarussian Ruble	- LTL - Lithuanian Litas	- XAG - Silver
- BZD - Belize Dollar	- LVL - Latvian Lats	- XAU - Gold
- CAD - Canadian Dollar	- LYD - Libyan Dinar	- XCD - East Caribbean Dollar
- CDF - Franc Congolais	- MAD - Moroccan Dirham	- XDR - SDR
- CHF - Swiss Franc	- MDL - Moldovan Leu	- XOF - CFA Franc
- CLP - Chilean Peso	- MGF - Malagasy Franc	- XPD - Palladium
- CNY - Yuan Renminbi	- MKD - Denar	- XPF - CFP Franc
- COP - Colombian Peso	- MMK - Kyat	- XPT - Platinum
- CRC - Costa Rican Colon	- MNT - Tugrik	- YER - Yemeni Rial
- CUP - Cuban Peso	- MOP - Pataca	- YUM - New Dinar
- CVE - Cape Verde Escudo	- MRO - Ouguiya	- ZAR - Rand
- CYP - Cyprus Pound	- MTL - Maltese Lira	- ZMK - Kwacha
- CZK - Czech Koruna	- MUR - Mauritius Rupee	- ZWD - Zimbabwe Dollar
- DJF - Djibouti Franc	- MVR - Rufiyaa	
- DKK - Danish Krone	- MWK - Kwacha	
- DOP - Dominican Peso	- MXN - Mexican Peso	
- DZD - Algerian Dinar	- MYR - Malaysian Ringgit	
- EEK - Kroon	- MZM - Metical	
- EGP - Egyptian Pound	- NAD - Namibia Dollar	
- ERN - Nakfa	- NGN - Naira	
- ETB - Ethiopian Birr	- NIO - Cordoba Oro	
- EUR - Euro	- NOK - Norwegian Krone	
<i>CodeDescription: On 1 January 1999, the euro will become the currency of those Member states of the European Union which adopt the single currency in accordance with the Treaty establishing the European Community. This code has been issued now so that technical preparations can be started. The code element 'EU' has been reserved by the ISO 3166 Maintenance Agency for use within ISO 4217 where 'R' has been appended to make an acceptable mnemonic code.</i>	- NPR - Nepalese Rupee	
- FJD - Fiji Dollar	- NZD - New Zealand Dollar	
- FKP - Falkland Islands Pound	- OMR - Rial Omani	
- GBP - Pound Sterling	- PAB - Balboa	
- GEL - Lari	- PEN - Nuevo Sol	
- GHC - Cedi	- PGK - Kina	
- GIP - Gibraltar Pound	- PHP - Philippine Peso	
- GMD - Dalasi	- PKR - Pakistan Rupee	
- GNF - Guinea Franc	- PLN - Zloty	
- GTQ - Quetzal	- PYG - Guarani	
- GYD - Guyana Dollar	- QAR - Qatari Rial	
- HKD - Honk Kong Dollar	- ROL - Leu	
- HNL - Lempira	- RUB - Russian Ruble	
- HRK - Kuna	- RWF - Rwanda Franc	
- HTG - Gourde	- SAR - Saudi Riyal	
- HUF - Forint	- SBD - Solomon Islands Dollar	
	- SCR - Seychelles Rupee	
	- SDD - Sudanese Dinar	
	- SEK - Swedish Krona	
	- SGD - Singapore Dollar	
	- SHP - St. Helena Pound	
	- SIT - Tolar	
	- SKK - Slovak Koruna	
	- SLL - Leone	
	- SOS - Somali Shilling	
	- SRG - Suriname Guilder	

Appendice iii - CodeList: UnitCodeContent

- 04 - small spray
- 05 - lift
- 08 - heat lot
- 10 - group
- 11 - outfit
- 13 - ration
- 14 - shot
- 15 - stick
- 16 - hundred fifteen kg drum
- 17 - hundred lb drum
- 18 - fiftyfive gallon (US) drum
- 19 - tank truck
- 20 - twenty foot container
- 21 - forty foot container
- 22 - decilitre per gram
- 23 - gram per cubic centimetre
- 24 - theoretical pound
- 25 - gram per square centimetre
- 26 - actual ton
- 27 - theoretical ton
- 28 - kilogram per square metre
- 29 - pound per thousand square feet
- 30 - horse power day per air dry metric ton
- 31 - catch weight
- 32 - kilogram per air dry metric ton
- 33 - kilopascal square metres per gram
- 34 - kilopascals per millimetre
- 35 - millilitres per square centimetre second
- 36 - cubic feet per minute per square foot
- 37 - ounce per square foot
- 38 - ounces per square foot per 0,01 inch
- 40 - millilitre per second
- 41 - millilitre per minute
- 43 - super bulk bag
- 44 - fivehundred kg bulk bag
- 45 - threehundred kg bulk bag
- 46 - fifty lb bulk bag
- 47 - fifty lb bag
- 48 - bulk car load
- 53 - theoretical kilograms
- 54 - theoretical tonne
- 56 - sitas
- 57 - mesh
- 58 - net kilogram
- 59 - part per million
- 60 - percent weight
- 61 - part per billion (US)
- 62 - percent per 1000 hour
- 63 - failure rate in time
- 64 - pound per square inch, gauge
- 66 - oersted
- 69 - test specificscale
- 71 - volt ampere per pound
- 72 - watt per pound
- 73 - ampere tum per centimetre
- 74 - millipascal
- 76 - gauss
- 77 - milli-inch
- 78 - kilogauss
- 80 - pounds per square inch absolute
- 81 - henry
- 84 - kilopound per square inch
- 85 - foot pound-force
- 87 - pound per cubic foot
- 89 - poise
- 90 - Saybold universal second
- 91 - stokes
- 92 - calorie per cubic centimetre
- 93 - calorie per gram
- 94 - curt unit
- 95 - twenty thousand gallon (US) tankcar
- 96 - ten thousand gallon (US) tankcar
- 97 - ten kg drum
- 98 - fifteen kg drum
- 1A - car mile
- 1B - car count
- 1C - locomotive count
- 1D - caboose count
- 1E - empty car
- 1F - train mile
- 1G - fuel usage gallon (US)
- 1H - caboose mile
- 1I - fixed rate
- 1J - ton mile
- 1K - locomotive mile
- 1L - total car count
- 1M - total car mile
- 1X - quarter mile
- 2A - radian per second
- 2B - radian per second squared
- 2C - rÅnggen
- 2I - British thermal unit per hour
- 2J - cubic centimetre per second
- 2K - cubic foot per hour
- 2L - cubic foot per minute
- 2M - centimetre per second
- 2N - decibel
- 2P - kilobyte
- 2Q - kilobecquerel
- 2R - kilocurie
- 2U - megagram
- 2V - megagram per hour
- 2W - bin
- 2X - metre per minute
- 2Y - millirÅnggen
- 2Z - millivolt
- 3B - megajoule
- 3C - manmonth
- 3E - pound per pound of product
- 3G - pound per piece of product
- 3H - kilogram per kilogram of product
- 3I - kilogram per piece of product
- 4A - bobbin
- 4B - cap
- 4C - centistokes
- 4E - twenty pack
- 4G - microlitre
- 4H - micrometre (micron)
- 4K - milliampere
- 4L - megabyte
- 4M - milligram per hour
- 4N - megabecquerel
- 4O - microfarad
- 4P - newton per metre
- 4Q - ounce inch
- 4R - ounce foot
- 4T - picofarad
- 4U - pound per hour
- 4W - ton(US) per hour
- 4X - kilolitre per hour
- 5A - barrel per minute
- 5B - batch
- 5C - gallon(US) per thousand
- 5E - MMSCF/day
- 5F - pounds per thousand
- 5G - pump
- 5H - stage
- 5I - standard cubic foot
- 5J - hydraulic horse power
- 5K - count per minute
- 5P - seismic level
- 5Q - seismic line
- A1 - 15 C calorie
- A10 - ampere square metre per joule second
- A11 - ÅngstrÅm
- A12 - astronomical unit
- A13 - attojoule
- A14 - barn
- A15 - barn per electron volt
- A16 - barn per steradian electron volt,
- A17 - barn per steradian
- A18 - becquerel per kilogram
- A19 - becquerel per metre cubed
- A2 - ampere per centimetre
- A20 - British thermal unit per second square foot degree Rankin
- A21 - British thermal unit per pound degree Rankin
- A22 - British thermal unit per second foot degree Rankin
- A23 - British thermal unit per hour square foot degree Rankin
- A24 - candela per square metre
- A25 - cheval vapeur
- A26 - coulomb metre
- A27 - coulomb metre squared per volt
- A28 - coulomb per cubic centimetre
- A29 - coulomb per cubic metre
- A3 - ampere per millimetre
- A30 - coulomb per cubic millimetre
- A31 - coulomb per kilogram second
- A32 - coulomb per mole
- A33 - coulomb per square centimetre
- A34 - coulomb per square metre
- A35 - coulomb per square millimetre
- A36 - cubic centimetre per mole
- A37 - cubic decimetre per mole
- A38 - cubic metre per coulomb
- A39 - cubic metre per kilogram
- A4 - ampere per square centimetre
- A40 - cubic metre per mole
- A41 - ampere per square metre
- A42 - curie per kilogram
- A43 - deadweight tonnage
- A44 - decalitre
- A45 - decametre
- A47 - decitex
- A48 - degree Rankin
- A49 - denier
- A5 - ampere square metre
- A50 - dyn second per cubic centimetre
- A51 - dyne second per centimetre
- A52 - dyne second per centimetre to the fifth
- A53 - electronvolt
- A54 - electronvolt per metre
- A55 - electronvolt square metre
- A56 - electronvolt square metre per kilogram
- A57 - erg
- A58 - erg per centimetre
- A6 - ampere per square metre kelvin squared
- A60 - erg per cubic centimetre
- A61 - erg per gram
- A62 - erg per gram second
- A63 - erg per second
- A64 - erg per second square centimetre
- A65 - erg per square centimetre second
- A66 - erg square centimetre
- A67 - erg square centimetre per gram
- A68 - exajoule
- A69 - farad per metre
- A7 - ampere per square millimetre
- A70 - femtojoule
- A71 - femtometre

- A73 - foot per second squared	- B25 - kilobecquerel per kilogram	- B94 - microohm
- A74 - foot pound-force per second	- B26 - kilocoulomb	- B95 - microohm metre
- A75 - freight ton	- B27 - kilocoulomb per cubic metre	- B96 - micropascal
- A76 - gal	- B28 - kilocoulomb per square metre	- B97 - microradian
- A77 - Gaussian CGS unit of displacement	- B29 - kiloelectronvolt	- B98 - microsecond
- A78 - Gaussian CGS unit of electric current	- B3 - batting pound	- B99 - microsiemens
- A79 - Gaussian CGS unit of electric charge	- B31 - kilogram metre per second	- BAR - bar
- A8 - ampere second	- B32 - kilogram metre squared	- BB - base box
- A80 - Gaussian CGS unit of electric field strength	- B33 - kilogram metre squared per second	- BD - board
- A81 - Gaussian CGS unit of electric polarization	- B34 - kilogram per cubic decimetre	- BE - bundle
- A82 - Gaussian CGS unit of electric potential	- B35 - kilogram per litre	- BFT - board foot
- A83 - Gaussian CGS unit of magnetization	CodeDescription: <i>Remarks by GEFEG</i>	- BG - bag
- A84 - gigacoulomb per cubic metre	<i>mbH-Synonym B35 = kilogram per litre of product</i>	- BH - brush
- A85 - gigaelectronvolt	- B36 - thermochemical calorie per gram	- BHP - brake horse power
- A86 - gigahertz	- B37 - kilogram-force	- BIL - trillion (US)
- A87 - gigaohm	- B38 - kilogram-force metre	CodeDescription: <i>Remarks by GEFEG</i>
- A88 - gigaohm metre	- B39 - kilogram-force metre per second	<i>mbH-Synonym BIL = billion (EUR)</i>
- A89 - gigapascal	- B4 - barrel, imperial	- BJ - bucket
- A9 - rate	- B40 - kilogram-force per square metre	- BK - basket
- A90 - gigawatt	- B41 - kilojoule per kelvin	- BL - bale
- A91 - gon	- B42 - kilojoule per kilogram	- BLD - dry barrel (US)
CodeDescription: <i>Remarks by GEFEG</i>	- B43 - kilojoule per kilogram kelvin	- BLL - barrel (US) (petroleum etc.)
<i>mbH-Synonyms A91 = grade</i>	- B44 - kilojoule per mole	- BO - bottle
- A93 - gram per cubic metre	- B45 - kilomole	- BP - hundred board feet
- A94 - gram per mole	- B46 - kilomole per cubic metre	- BQL - becquerel
- A95 - gray	- B47 - kilonewton	- BR - bar
- A96 - gray per second	- B48 - kilonewton metre	- BT - bolt
- A97 - hectopascal	- B49 - kiloohm	- BTU - British thermal unit
- A98 - henry per metre	- B5 - billet	- BUA - bushel (US)
- AA - ball	- B50 - kiloohm metre	- BUI - bushel (UK)
- AB - bulk pack	- B51 - kilopond	- BW - base weight
- ACR - acre	- B52 - kilosecond	- BX - box
- AD - byte	- B53 - kilosiemens	- BZ - million BTUs
- AE - ampere per metre	- B54 - kilosiemens per metre	- C0 - call
- AH - additional minute	- B55 - kilovolt per metre	- C1 - composite product pound (total weight)
- AI - average minute per call	- B56 - kiloweber per metre	- C10 - millifarad
- AJ - cop	- B57 - light year	- C11 - milligal
- AK - fathom	- B58 - litre per mole	- C12 - milligram per metre
- AL - access line	- B59 - lumen hour	- C13 - milligray
- AM - ampoule	- B6 - bun	- C14 - millihenry
- AMH - ampere hour	- B60 - lumen per square metre	- C15 - millijoule
- AMP - ampere	- B61 - lumen per watt	- C16 - millimetre per second
- ANN - year	- B62 - lumen second	- C17 - millimetre squared per second
- AP - aluminium pound only	- B63 - lux hour	- C18 - millimole
- APZ - troy ounce or apothecaries' ounce	- B64 - lux second	- C19 - mole per kilogram
- AQ - anti-hemophilic factor (AHF) unit	- B65 - maxwell	- C2 - carset
- AR - suppository	- B66 - megaampere per square metre	- C20 - millinewton
- ARE - are	- B67 - megabecquerel per kilogram	- C22 - millinewton per metre
- AS - assortment	- B69 - megacoulomb per cubic metre	- C23 - milliohm metre
- ASM - alcoholic strength by mass	- B7 - cycle	- C24 - millipascal second
- ASU - alcoholic strength by volume	- B70 - megacoulomb per square metre	- C25 - milliradian
- ATM - standard atmosphere	- B71 - megaelectronvolt	- C26 - millisecond
- ATT - technical atmosphere	- B72 - megagram per cubic metre	- C27 - millisiemens
- AV - capsule	- B73 - meganewton	- C28 - millisievert
- AW - powder filled vial	- B74 - meganewton metre	- C29 - millitesla
- AY - assembly	- B75 - megaohm	- C3 - microvolt per metre
- AZ - British thermal unit per pound	- B76 - megaohm metre	- C30 - millivolt per metre
- B0 - Btu per cubic foot	- B77 - megasiemens per metre	- C31 - milliwatt
- B1 - barrel (US) per day	- B78 - megavolt	- C32 - milliwatt per square metre
- B11 - joule per kilogram kelvin	- B79 - megavolt per metre	- C33 - millweber
- B12 - joule per metre	- B8 - joule per cubic metre	- C34 - mole
- B13 - joule per square metre	- B81 - reciprocal metre squared reciprocal second	- C35 - mole per cubic decimetre
CodeDescription: <i>Remarks by GEFEG</i>	- B83 - metre to the fourth power	- C36 - mole per cubic metre
<i>mbH-Synonym B13 = joule per metre squared</i>	- B84 - microampere	- C38 - mole per litre
- B14 - joule per metre to the fourth power	- B85 - microbar	- C39 - nanoampere
- B15 - joule per mole	- B86 - microcoulomb	- C4 - carload
- B16 - joule per mole kelvin	- B87 - microcoulomb per cubic metre	- C40 - nanocoulomb
- B18 - joule second	- B88 - microcoulomb per square metre	- C41 - nanofarad
- B2 - bunk	- B89 - microfarad per metre	- C42 - nanofarad per metre
- B20 - joule square metre per kilogram	- B9 - batt	- C43 - nanohenry
- B21 - kelvin per watt	- B90 - microhenry	- C44 - nanohenry per metre
- B22 - kiloampere	- B91 - microhenry per metre	- C45 - nanometre
- B23 - kiloampere per square metre	- B92 - micronewton	- C46 - nanoohm metre
- B24 - kiloampere per metre	- B93 - micronewton metre	- C47 - nanosecond

- C48 - nanotesla
- C49 - nanowatt
- C5 - cost
- C50 - neper
- C51 - neper per second
- C52 - picometre
- C53 - newton metre second
- C54 - newton metre squared kilogram squared
- C55 - newton per square metre
- C56 - newton per square millimetre
- C57 - newton second
- C58 - newton second per metre
- C59 - octave
- C6 - cell
- C60 - ohm centimetre
- C61 - ohm metre
- C62 - one
- C63 - parsec
- C64 - pascal per kelvin
- C65 - pascal second
- C66 - pascal second per cubic metre
- C67 - pascal second per metre
- C68 - petajoule
- C69 - phon
- C7 - centipoise
- C70 - picoampere
- C71 - picocoulomb
- C72 - picofarad per metre
- C73 - picohenry
- C75 - picowatt
- C76 - picowatt per square metre
- C77 - pound gage
- C78 - pound-force
- C8 - millicoulomb per kilogram
- C80 - rad
- C81 - radian
- C82 - radian meter squared per mole
- C83 - radian metre squared per kilogram
- C84 - radian per metre
- C85 - reciprocal Ångström
- C86 - reciprocal cubic metre
- C87 - reciprocal cubic metre per second
- C88 - reciprocal electron volt per cubic metre
- C89 - reciprocal henry
- C9 - coil group
- C90 - reciprocal joule per cubic metre
- C91 - reciprocal kelvin or kelvin to the power minus one
- C92 - reciprocal metre
- C93 - reciprocal square metre
- CodeDescription: *Remarks by GEFEG*
mbH:Synonym C93 = reciprocal metre squared
- C94 - reciprocal minute
- C95 - reciprocal mole
- C96 - reciprocal pascal or pascal to the power minus one
- C97 - reciprocal second
- C98 - reciprocal second per cubic metre
- C99 - reciprocal second per metre squared
- CA - can
- CCT - carrying capacity in metric ton
- CDL - candela
- CEL - degree Celsius
- CEN - hundred
- CG - card
- CGM - centigram
- CH - container
- CJ - cone
- CK - connector
- CKG - coulomb per kilogram
- CL - coil
- CLF - hundred leave
- CLT - centilitre
- CMK - square centimetre
- CMQ - cubic centimetre
- CMT - centimetre
- CNP - hundred pack
- CNT - cental (UK)
- CO - carboy
- COU - coulomb
- CQ - cartridge
- CR - crate
- CS - case
- CT - carton
- CTM - metric carat
- CU - cup
- CUR - curie
- CV - cover
- CWA - hundred pounds (cwt)/hundred weight (US)
- CWI - hundred weight (UK)
- CY - cylinder
- CZ - combo
- D1 - reciprocal second per steradian
- D10 - siemens per metre
- D12 - siemens square metre per mole
- D13 - sievert
- D14 - thousand linear yard
- D15 - sone
- D16 - square centimetre per erg
- D17 - square centimetre per steradian erg
- D18 - metre kelvin
- D19 - square metre kelvin per watt
- D2 - reciprocal second per steradian metre squared
- D20 - square metre per joule
- D21 - square metre per kilogram
- D22 - square metre per mole
- D23 - pen gram (protein)
- D24 - square metre per steradian
- D25 - square metre per steradian joule
- D26 - square metre per volt second
- D27 - steradian
- D28 - syphon
- D29 - terahertz
- D30 - terajoule
- D31 - terawatt
- D32 - terawatt hour
- D33 - tesla
- D34 - tex
- D35 - thermochemical calorie
- D37 - thermochemical calorie per gram kelvin
- D38 - thermochemical calorie per second centimetre kelvin
- D39 - thermochemical calorie per second square centimetre kelvin
- D40 - thousand litre
- D41 - tonne per cubic metre
- D42 - tropical year
- D43 - unified atomic mass unit
- D44 - var
- D45 - volt squared per kelvin squared
- D46 - volt - ampere
- D47 - volt per centimetre
- D48 - volt per kelvin
- D49 - millivolt per kelvin
- D5 - kilogram per square centimeter
- D50 - volt per metre
- D51 - volt per millimetre
- D52 - watt per kelvin
- D53 - watt per metre kelvin
- D54 - watt per square metre
- D55 - watt per square metre kelvin
- D56 - watt per square metre kelvin to the fourth power
- D57 - watt per steradian
- D58 - watt per steradian square metre
- D59 - weber per metre
- D6 - röntgen per second
- D60 - weber per millimetre
- D61 - minute
- D62 - second
- D63 - book
- D64 - block
- D65 - round
- D66 - cassette
- D67 - dollar per hour
- D68 - inch to the fourth power
- D7 - sandwich
- D70 - International Table (IT) calorie
- D71 - International Table (IT) calorie per second centimetre kelvin
- D72 - International Table (IT) calorie per second square centimetre kelvin
- D73 - joule square metre
- D74 - kilogram per mole
- D75 - International Table (IT) calorie per gram
- D76 - International Table (IT) calorie per gram kelvin
- D77 - megacoulomb
- D79 - beam
- D8 - draize score
- D80 - microwatt
- D81 - microtesla
- D82 - microvolt
- D83 - millinewton metre
- D85 - microwatt per square metre
- D86 - millicoulomb
- D87 - millimole per kilogram
- D88 - millicoulomb per cubic metre
- D89 - millicoulomb per square metre
- D9 - dyne per square centimeter
- D90 - cubic metre (net)
- D91 - rem
- D92 - band
- D93 - second per cubic metre
- D94 - second per radian cubic metre
- D95 - joule per gram
- D96 - pound gross
- D97 - pallet/unit load
- D98 - mass pound
- D99 - sleeve
- DAA - decade
- DAD - ten day
- DAY - day
- DB - dry pound
- DC - disk (disc)
- DD - degree
- DE - deal
- DEC - decade
- DG - decigram
- DI - dispenser
- DJ - decagram
- DLT - declitre
- DMK - square decimetre
- DMQ - cubic decimetre
- DMT - decimetre
- DN - decinewton metre
- DPC - dozen piece
- DPR - dozen pair
- DPT - displacement tonnage
- DQ - data record
- DR - drum
- DRA - dram (US)
- DRI - dram (UK)
- DRL - dozen roll
- DRM - drachm (UK)
- DS - display
- DT - dry ton
- DTN - decitonne
- CodeDescription: *Remarks by GEFEG*
mbH:Synonym DTN = centner, metricSynonym
DTN = quintal, metric

- DU - dyne	- HA - hank	- KL - kilogram per metre
- DWT - pennyweight	- HAR - hectare	- KMH - kilometre per hour
- DX - dyne per centimetre	- HBA - hectobar	- KMK - square kilometre
- DY - directory book	- HBX - hundred boxe	- KMQ - kilogram per cubic metre
- DZN - dozen	- HC - hundred count	- KNI - kilogram of nitrogen
- DZP - dozen pack	- HD - half dozen	- KNS - kilogram named substance
- E2 - belt	- HE - hundredth of a carat	- KNT - knot
- E3 - trailer	- HF - hundred feet	- KO - milliequivalence caustic potash per gram of product
- E4 - gross kilogram	- HGM - hectogram	- KPA - kilopascal
- E5 - metric long ton	- HH - hundred cubic feet	- KPH - kilogram of potassium hydroxide (caustic potash)
- EA - each	- HI - hundred sheet	- KPO - kilogram of potassium oxide
- EB - electronic mail box	- HIU - hundred international unit	- KPP - kilogram of phosphorus pentoxide (phosphoric anhydride)
- EC - each per month	- HJ - metric horse power	- KR - kiloröntgen
- EP - eleven pack	- HK - hundred kilogram	- KS - thousand pound per square inch
- EQ - equivalent gallon	- HL - hundred feet (linear)	- KSD - kilogram of substance 90 % dry
- EV - envelope	- HLT - hectolitre	- KSH - kilogram of sodium hydroxide (caustic soda)
- F1 - thousand cubic feet per day	- HM - mile per hour	- KT - kit
- F9 - fibre per cubic centimetre of air	- HMQ - million cubic metre	- KTM - kilometre
- FAH - degree Fahrenheit	- HMT - hectometre	- KTN - kilotonne
- FAR - farad	- HN - conventional millimetre of mercury	- KUR - kilogram of uranium
- FB - field	- HO - hundred troy ounce	- KVA - kilovolt - ampere
- FC - thousand cubic feet	- HP - conventional millimetre of water	- KVR - kilovar
- FD - million particle per cubic foot	- HPA - hectolitre of pure alcohol	- KVT - kilovolt
- FE - track foot	- HS - hundred square feet	- KW - kilograms per millimeter
- FF - hundred cubic metre	- HT - half hour	- KWH - kilowatt hour
- FG - transdermal patch	- HTZ - hertz	- KWT - kilowatt
- FH - micromole	- HUR - hour	- KX - millilitre per kilogram
- FL - flake ton	- HY - hundred yard	- L2 - litre per minute
- FM - million cubic feet	- IA - inch pound (pound inch)	- LA - pound per cubic inch
- FOT - foot	- IC - count per inch	- LBR - pound
- FP - pound per square foot	- IE - person	CodeDescription: <i>Remarks by GEFEG</i>
- FR - foot per minute	- IF - inches of water	<i>mbH.Synonym LBR = pound decimal</i>
- FS - foot per second	- II - column inch	- LBT - troy pound (US)
- FTK - square foot	- IL - inch per minute	- LC - linear centimetre
- FTQ - cubic foot	- IM - impression	- LD - litre per day
- G2 - US gallon per minute	- INH - inch	- LE - lite
- G3 - Imperial gallon per minute	- INK - square inch	- LEF - leaf
- G7 - microfiche sheet	- INQ - inch cubed	- LF - linear foot
- GB - gallon (US) per day	- IP - insurance policy	- LH - labour hour
- GBQ - gigabecquerel	- IT - count per centimetre	- LI - linear inch
- GC - gram per 100 gram	- IU - inch per second (linear speed)	- LJ - large spray
- GD - gross barrel	CodeDescription: <i>Remarks by GEFEG</i>	- LK - link
- GE - pound per gallon (US)	<i>mbH.Synonym IU = inch per second (vibration)</i>	- LM - linear metre
- GF - gram per metre (gram per 100 centimetres)	- IV - inch per second squared (acceleration)	- LN - length
- GF1 - gram of fissile isotope	CodeDescription: <i>Remarks by GEFEG</i>	- LO - lot
- GGR - great gross	<i>mbH.Synonym IV = inch per second squared (vibration acceleration)</i>	- LP - liquid pound
- GH - half gallon (US)	- J2 - joule per kilogram	- LPA - litre of pure alcohol
- GIA - gill (US)	- JB - jumbo	- LR - layer
- GII - gill (UK)	- JE - joule per kelvin	- LS - lump sum
- GJ - gram per millilitre	- JG - jug	- LTN - ton (UK) or longton (US)
- GK - gram per kilogram	- JK - megajoule per kilogram	- LTR - litre
- GL - gram per litre	- JM - megajoule per cubic metre	- LUM - lumen
- GLD - dry gallon (US)	- JO - joint	- LUX - lux
- GLI - gallon (UK)	- JOU - joule	- LX - linear yard per pound
- GLL - gallon (US)	- JR - jar	- LY - linear yard
- GM - gram per square metre	- K1 - kilowatt demand	- M0 - magnetic tape
- GN - gross gallon	- K2 - kilovolt ampere reactive demand	- M1 - milligrams per litre
- GO - milligrams per square metre	- K3 - kilovolt ampere reactive hour	- M4 - monetary value
- GP - milligram per cubic metre	- K5 - kilovolt ampere (reactive)	- M5 - microcurie
- GQ - microgram per cubic meter	- K6 - kilolitre	- M7 - micro-inch
- GRM - gram	- KA - cake	- M9 - million Btu per 1000 cubic feet
- GRN - grain	- KB - kilocharacter	- MA - machine per unit
- GRO - gross	- KBA - kilobar	- MAL - mega litre
- GRT - gross register ton	- KB - kilogram decimal	- MAM - megametre
- GT - gross ton	- KEL - kelvin	- MAW - megawatt
CodeDescription: <i>Remarks by GEFEG</i>	- KF - kilopacket	- MBE - thousand standard brick equivalent
<i>mbH.Synonym GT = metric gross ton</i>	- KG - keg	- MBF - thousand board feet
- GV - gigajoule	- KGM - kilogram	- MBR - millibar
- GW - gallon per thousand cubic feet	- KGS - kilogram per second	- MC - microgram
- GWH - gigawatt hour	- KHZ - kilohertz	- MCU - millicurie
- GY - gross yard	- KI - kilogram per millimetre width	- MD - air dry metric ton
- GZ - gage system	- KJ - kilosegment	
- H1 - half page - electronic	- KJO - kilojoule	
- H2 - half litre		

- MF - milligram per square foot per side
- MGM - milligram
- MHZ - megahertz
- MIK - square mile
- MIL - thousand
- MIN - minute
- MIO - million
- MIU - million international unit
- MK - milligram per square inch
- MLD - milliard
- CodeDescription: *Remarks by GEFEG*
- mbH:Synonym MLD = billion (US)*
- MLT - millilitre
- MMK - square millimetre
- MMQ - cubic millimetre
- MMT - millimetre
- MON - month
- MPA - megapascal
- MQ - thousand metre
- MQH - cubic metre per hour
- MQS - cubic metre per second
- MSK - metre per second squared
- MT - mat
- MTK - square metre
- MTQ - cubic metre
- CodeDescription: *Remarks by GEFEG*
- mbH:Synonym MTQ = metre cubed*
- MTR - metre
- MTS - metre per second
- MV - number of volts
- MVA - megavolt - ampere
- MWH - megawatt hour (1000 kW.h)
- N1 - pen calorie
- N2 - number of lines
- N3 - print point
- NA - milligram per kilogram
- NAR - number of articles
- NB - barge
- NBB - number of bobbins
- NC - car
- NCL - number of cells
- ND - net barrel
- NE - net litre
- NEW - newton
- NF - message
- NG - net gallon (us)
- NH - message hour
- NI - net imperial gallon
- NIU - number of international units
- NJ - number of screens
- NL - load
- NMI - nautical mile
- NMP - number of packs
- NN - train
- NPL - number of parcels
- NPR - number of pairs
- NPT - number of parts
- NQ - mho
- NR - micromho
- NRL - number of rolls
- NT - net ton
- CodeDescription: *Remarks by GEFEG*
- mbH:Synonym NT = metric net ton*
- NTT - net register ton
- NU - newton metre
- NV - vehicle
- NX - part per thousand
- NY - pound per air dry metric ton
- OA - panel
- OHM - ohm
- ON - ounce per square yard
- ONZ - ounce
- OP - two pack
- OT - overtime hour
- OZ - ounce av
- OZA - fluid ounce (US)
- OZI - fluid ounce (UK)
- P0 - page - electronic
- P1 - percent
- P2 - pound per foot
- P3 - three pack
- P4 - four pack
- P5 - five pack
- P6 - six pack
- P7 - seven pack
- P8 - eight pack
- P9 - nine pack
- PA - packet
- PAL - pascal
- PB - pair inch
- PD - pad
- PE - pound equivalent
- PF - pallet (lift)
- PG - plate
- PGL - proof gallon
- PI - pitch
- PK - package
- CodeDescription: *Remarks by GEFEG*
- mbH:Synonym PK = pack*
- PL - pall
- PM - pound percentage
- PN - pound net
- PO - pound per inch of length
- PQ - page per inch
- PR - pair
- PS - pound-forceper square inch
- PT - pint (US)
- PTD - dry pint (US)
- PTI - pint (UK)
- PTL - liquid pint (US)
- PU - tray / tray pack
- PV - half pint (US)
- PW - pound per inch of width
- PY - peck dry (US)
- PZ - peck dry (UK)
- Q3 - meal
- QA - page - facsimile
- QAN - quarter (of a year)
- QB - page - hardcopy
- QD - quarter dozen
- QH - quarter hour
- QK - quarter kilogram
- QR - quire
- QT - quart (US)
- QTD - dry quart (US)
- QTI - quart (UK)
- QTL - liquid quart (US)
- QTR - quarter (UK)
- R1 - pica
- R4 - calorie
- R9 - thousand cubic metre
- RA - rack
- RD - rod
- RG - ring
- RH - running or operating hour
- RK - roll metric measure
- RL - reel
- RM - ream
- RN - ream metric measure
- RO - roll
- RP - pound per ream
- RPM - revolutions per minute
- RPS - revolutions per second
- RS - reset
- RT - revenue ton mile
- RU - run
- S3 - square foot per second
- S4 - square metre per second
- S5 - sixty fourths of an inch
- S6 - session
- S7 - storage unit
- S8 - standard advertising unit
- SA - sack
- SAN - half year (6 months)
- SCO - score
- SCR - scruple
- SD - solid pound
- SE - section
- SEC - second
- SET - set
- SG - segment
- SHT - shipping ton
- SIE - siemens
- SK - split tanktruck
- SL - slipsheet
- SMI - mile (statute mile)
- SN - square rod
- SO - spool
- SP - shelf package
- SQ - square
- SR - strip
- SS - sheet metric measure
- SST - short standard (7200 matches)
- ST - sheet
- STI - stone (UK)
- STN - ton (US) or short ton (UK/US)
- CodeDescription: *Remarks by GEFEG*
- mbH:Synonym STN = net ton (2000 lb)*
- SV - skid
- SW - skein
- SX - shipment
- T0 - telecommunication line in service
- T1 - thousand pound gross
- T3 - thousand piece
- T4 - thousandbag
- T5 - thousand casing
- T6 - thousand gallon (US)
- T7 - thousand impression
- T8 - thousand linear inch
- TA - tenth cubic foot
- TAH - kiloampere hour (thousand ampere hour)
- TC - truckload
- TD - therm
- TE - tote
- TF - ten square yard
- TI - thousand square inch
- TJ - thousand square centimetre
- TK - tank, rectangular
- TL - thousand feet (linear)
- TN - tin
- TNE - tonne (metric ton)
- CodeDescription: *Remarks by GEFEG*
- mbH:Synonym TNE = metric ton*
- TP - ten pack
- TPR - ten pair
- TQ - thousand feet
- TQD - thousand cubic metre per day
- TR - ten square feet
- TRL - trillion (EUR)
- TS - thousand square feet
- TSD - tonne of substance 90 % dry
- TSH - ton of steam per hour
- TT - thousand linear metre
- TU - tube
- TV - thousand kilogram
- TW - thousand sheet
- TY - tank, cylindrical
- U1 - treatment
- U2 - tablet
- UA - torr
- UB - telecommunication line in service average
- UC - telecommunication port

- **UD** - tenth minute
- **UE** - tenth hour
- **UF** - usage per telecommunication line average
- **UH** - ten thousand yard
- **UM** - million unit
- **VA** - volt ampere per kilogram
- **VI** - vial
- **VLT** - volt
- **VQ** - bulk
- **VS** - visit
- **W2** - wet kilo
- **W4** - two week
- **WA** - watt per kilogram
- **WB** - wet pound
- **WCD** - cord
- **WE** - wet ton
- **WEB** - weber
- **WEE** - week
- **WG** - wine gallon
- **WH** - wheel
- **WHR** - watt hour
- **WI** - weight per square inch
- **WM** - working month
- **WR** - wrap
- **WSD** - standard
- **WTT** - watt
- **WW** - millilitre of water
- **X1** - chain
- **YDK** - square yard
- **YDQ** - cubic yard
- **YL** - hundred linear yard
- **YRD** - yard
- **YT** - ten yard
- **Z1** - lift van
- **Z2** - chest
- **Z3** - cask
- **Z4** - hogshead
- **Z5** - lug
- **Z6** - conference point
- **Z8** - newspaper agate line
- **ZP** - page
- **ZZ** - mutually defined

Glossario**Definition**

Significato semantico dell'elemento.

UsageRule

Indicazioni su come e/o quando utilizzare l'elemento.

Bibliografia

- [AF11] Apache FOP home page, ultima visita Maggio 2011
<http://xmlgraphics.apache.org/fop/0.95/running.html#memory>
- [BL10] B. Lowagie *"iText in Action"* 2th Edition, Manning Publ. 2010
- [BTIH11] B.Bos, T.Çelik, I.Hickson, H.Wium Lie *"Cascading Style Sheets Level 2 Revision 1 (CSS 2.1) Specification"*, W3C Recommendation, Giugno 2011, <http://www.w3.org/TR/2011/REC-CSS2-20110607>
- [DF11] DocFlex/XML home page, ultima visita Marzo 2011
http://www.filigris.com/products/docflex_xml/xsddoc/
- [DFME08] D. Steinberg, Frank Budinsky, M. Paternostro, E. Merks *"EMF: Eclipse Modeling Framework"* 2th Edition, Addison-Wesley 2008
- [EB11] eBiz-TCF home page, ultima visita Aprile 2011 <http://ebiz-tcf.eu/>
- [EDO11] eDoc creator home page, ultima visita Ottobre 2011
<http://79.125.15.30:8080/srdc/eDoCreator/>
- [EL02] E. van der Vlist *"XML Schema"* 1th Edition, O'Reilly 2002
- [II06] ISO/IEC, *"Information technology - Document Schema Definition Language (DSDL) - Part 3: Rule-based validation - Schematron"* 1th Edition, ISO/IEC 19757-3:2006(E) 2006. Disponibile all'URL:
[http://standards.iso.org/ittf/PubliclyAvailableStandards/c040833_ISO_IEC_19757-3_2006\(E\).zip](http://standards.iso.org/ittf/PubliclyAvailableStandards/c040833_ISO_IEC_19757-3_2006(E).zip)
- [JD99] J. Clark, S. DeRose *"XML Path Language (XPath) Version 1.0"*, W3C Recommendation, Novembre 1999 <http://www.w3.org/TR/1999/REC-xpath-19991116>
- [JM01] J. Clark, M. Makoto *"RELAX NG Specification"*, OASIS, Dicembre

2001, <http://relaxng.org/spec-20011203.html>

[MK08] M. Kay *"XSLT 2.0 and XPath 2.0 Programmer's Reference"* 4th Edition, Wiley 2008

[MML06] Moda-ML Lisea home page, ultima visita Aprile 2006 <http://cross-lab.it/lisealab/imple/pgcl.asp?p=299&CMS=moda-ml-cms&lingua=it&nomenu=0>

[MML11] Moda-ML home page, ultima visita Giugno 2011 <http://www.moda-ml.org/>

[NR10] N. Walsh, R. L. Hamilton *"DocBook 5: The Definitive Guide"* Rev Upd edition, O'Reilly Media 2010

[OA07] OASIS, *"Code List Representation (Genericcode) Version 1.0"*, 2007 <http://docs.oasis-open.org/codelist/cs-genericcode-1.0/doc/oasis-code-list-representation-genericcode.pdf>

[OX11] <oXygen/> XML Editor home page, ultima visita Marzo 2011 <http://www.oxygenxml.com/>

[PD04] P. Walmsley, D. C. Fallside *"XML Schema Part 0: Primer Second Edition"*, W3C Recommendation, Ottobre 2004, <http://www.w3.org/TR/2004/REC-xmlschema-0-20041028/>

[PD10] P. De Sabbata *"Da Moda-ML a eBIZ-TCF, una sintesi a livello direzionale"*, 2010 http://www.cross-lab.it/moda-ml/download/documentazione/pubbl_modaml.asp?lingua=it

[PPL11] Peppol home page, ultima visita Ottobre 2011 <http://www.peppol.eu/>

[SP10] S. Powers, *"JavaScript Cookbook"* 1th Ed, O'Reilly 2010

[UBL11] UBL-italia home page, ultima visita Settembre 2011 http://www.lisealab.it/ubl-italia/default.asp?lingua=it&node_id=1

[UC09] UN/CEFACT, *"Core Components Technical Specification Version 3.0"*, 2009 <http://live.unece.org/cefact/codesfortrade/CCTS/CCTS-Version3.pdf>

[WST11] W3C XML Schema tools home page, ultima visita Marzo 2011 <http://www.w3.org/XML/Schema#Tools>

[XS11] XS3P home page, ultima visita Marzo 2011 xml.fiforms.org/xs3p/

Ringraziamenti

Ringrazio il Prof. Fabio Vitali per avermi dato la possibilità di svolgere questa tesi e per la sua disponibilità.

Ringrazio Piero De Sabbata, Nicola Gessa e Arianna Brutti per essere stati sempre presenti, disponibili e per avermi sopportato durante lo svolgimento di questa tesi.

Ringrazio i miei genitori per essermi stati sempre vicini.