

Università degli Studi di Bologna

FACOLTÀ DI SCIENZE MATEMATICHE, FISICHE E NATURALI

Corso di Laurea in Informatica
(Indirizzo: Sistemi Distribuiti)
(Materia: Laboratorio di Tecnologie Web)

ESTENSIONE E CREAZIONE DI PROFILI EBXML DI COLLABORAZIONE AZIENDALE PER L'E-BUSINESS

Relatore:
Prof. FABIO VITALI

Tesi di Laurea di:
CRISTIANO NOVELLI

Correlatori:
Ing. PIERO DE SABBATA
Dott. NICOLA GESSA

PAROLE CHIAVE:
ebXML, BPSS, CPP, CPA, e-business

Sessione II

Anno Accademico 2003 - 2004

*Alla mia Famiglia
con affetto*

Indice

1 INTRODUZIONE	1
2 MODA-ML.....	11
2.1 Tessile/Abbigliamento italiano.....	11
2.2 Adozione di nuove tecnologie e standard.....	13
2.3 I Business Process MODA-ML.....	14
2.4 Il Dizionario.....	17
2.4.1 Il contesto e i requisiti.....	17
2.4.2 Processi e Attività di filiera	19
2.4.3 Dai processi di filiera ai documenti scambiati.....	19
2.4.4 Dalla radice del documento ai singoli componenti.....	20
3 EBXML	23
3.1 Il framework ebXML.....	24
3.2 ebXML BPSS	28
3.2.1 Concetti chiave dello standard ebXML BPSS.....	29
3.2.1.1 <i>Business Collaboration</i>	31
3.2.1.2 <i>Business Activity</i>	32
3.2.1.3 <i>Business Transaction</i>	32
3.2.1.4 <i>Business Document Flows</i>	33
3.2.1.5 <i>Business Transaction Choreography</i>	33
3.2.2 Struttura XML dello standard ebXML BPSS.....	34
3.3 ebXML Collaboration Protocol	36
3.3.1 ebXML CPP.....	38
3.3.2 ebXML CPA.....	38
3.3.3 Struttura XML dello standard ebXML CPP e CPA.....	40
4 PROGETTAZIONE	47
4.1 Il Problema.....	47
4.2 Architettura CP-NET	48
4.3 Specifiche MODA-ML.....	52

4.3.1 MODA-ML BPSS	52
4.3.2 MODA-ML CPP e CPA	57
4.3.2.1 Adattamento dello standard ebXML CPP a MODA-ML.....	57
4.3.2.2 Il nuovo elemento aggiunto al CPP	58
4.3.2.3 Gestione del nuovo elemento nel CPA.....	64
4.4 Progettazione del BPSS-Generator	67
4.5 Progettazione del CPP-Editor	68
4.5.1 Obiettivo del CPP-Editor	68
4.5.2 Creazione o Modifica	68
4.5.3 Funzionamento del CPP-Editor	69
4.5.4 Preferenze sulle entità opzionali dei documenti	71
5 IMPLEMENTAZIONE	75
5.1 Organizzazione di Files e Directory	76
5.2 Implementazione di MODA-ML BPSS-Generator	78
5.3 Implementazione di MODA-ML CPP-Editor	81
5.3.1 Pagina iniziale del CPP-Editor	82
5.3.2 CPP-Editor Step 1	83
5.3.3 CPP-Editor Step 2.....	85
5.3.4 CPP-Editor Step 3.....	88
5.3.5 CPP-Editor Step 4.....	92
5.3.6 Libertà di navigazione nelle cinque fasi del CPP-Editor.....	93
5.4 Gestione degli errori	95
5.5 Installazione delle applicazioni sul server MODA-ML.....	97
6 CONCLUSIONI	99
APPENDICE A: BPSS “Produzione Tessuto”	105
APPENDICE B: BPSS “Fornitura Tessuti”	107
APPENDICE C: CPP di default	111
APPENDICE D: CPP relativo al BPSS “Fornitura Tessuti”	115
BIBLIOGRAFIA	119
Ringraziamenti.....	123

1 INTRODUZIONE

Scopo di questa tesi è la creazione di un'architettura di supporto alle aziende del settore Tessile e Abbigliamento italiano per la creazione di Profili di Collaborazione atti a favorire il raggiungimento di Business Collaborations.

I Profili di Collaborazione consentono alle aziende di poter descrivere in maniera formale le proprie capacità di intraprendere Business Process. Sono creati con lo scopo di facilitare il coordinamento e l'instaurazione di Business Collaborations, con conseguente diminuzione del tempo totale del processo produttivo.

L'architettura sviluppata si basa sugli standard ebXML ed è stata applicata al contesto del progetto italiano MODA-ML. Il framework ebXML è un insieme di specifiche e standard atti a facilitare l'evoluzione dell'e-commerce, mentre MODA-ML è un progetto europeo con lo scopo di fornire alle aziende del settore Tessile e Abbigliamento italiano uno standard per lo scambio di documenti business.

Il risultato concreto di questo lavoro è stato la definizione di specifiche MODA-ML basate su uno degli standard ebXML (chiamato Profili di Collaborazione) e lo sviluppo di due applicazioni web che ne supportano la gestione.

L'architettura qui descritta, chiamata CP-NET (Collaboration Profile Networking Enterprises Technology), è nata come supporto al progetto MODA-ML (Middleware tOols and Documents to enhAncE the textile/clothing supply chain through xML) condotto dall'ENEA (Ente Nazionale per le Nuove Tecnologie, l'Energia e l'Ambiente), nel contesto del settore Tessile e Abbigliamento italiano.

Il settore Tessile e Abbigliamento è caratterizzato da un consistente numero di aziende che operano nei diversi stadi della lavorazione dei tessuti e che compongono la catena produttiva. Questa situazione vede una grossa interdipendenza tra le aziende che molto spesso, però, sono di natura diversa sia in termini di dimensioni sia nella tipologia di

gestione e dei sistemi gestionali utilizzati. La comunicazione elettronica, in mancanza di un protocollo comune, è di conseguenza macchinosa e rallenta l'intero processo produttivo.

MODA-ML ha come scopo quello di agevolare la comunicazione elettronica delle aziende del settore Tessile e Abbigliamento tramite un insieme di documenti XML definiti secondo una specifica creata. Il progetto MODA-ML ha previsto lo sviluppo di un Dizionario in cui sono contenuti tutti i dati inerenti alla struttura dei documenti di scambio e l'organizzazione dei *Business Process*. Ha inoltre implementato alcuni prototipi software per rendere concrete le modalità di scambio dei documenti creati.

Prima di instaurare la comunicazione, le aziende devono però accordarsi sui Business Process che vogliono eseguire, sulle regole in cui vogliono svolgerli, sui documenti che vogliono utilizzare e su come definire gli stessi documenti affinché siano immediatamente comprensibili e non ambigui da entrambe le parti.

Lo scambio d'informazioni fra due o più aziende richiede che ciascuna parte sia a conoscenza delle *Business Collaborations*, e quindi dei Business Process, supportate dalle altre parti, i ruoli da queste ricoperti e i dettagli tecnologici circa il modo in cui tali parti inviano e ricevono i messaggi.

Per instaurare una Business Collaboration è necessario che le aziende raggiungano un accordo su tali dettagli e quindi è necessario un dialogo preliminare.

L'architettura CP-NET s'inserisce in questo contesto per fornire un supporto per l'instaurazione delle comunicazioni e-business utilizzando *Profili di Collaborazione* basati sullo standard ebXML.

ebXML (Electronic Business using eXtensible Markup Language) nasce dall'iniziativa di UN/CEFACT (United Nations Center for Trade Facilitation and Electronic Business) e OASIS (Organization for the Advancement of Structured Information Standards). Definisce un insieme di standard e un'infrastruttura modulare basata su XML che consente l'utilizzo globale d'informazioni relative all'e-business, favorendo l'interoperabilità tra le aziende in modo sicuro e coerente. Ogni modulo ebXML può essere implementato in maniera indipendente dal resto dell'architettura.

Per poter sviluppare il mio progetto ho preso in considerazione un modulo che prevede i seguenti standard:

1. Business Process Specification Schema (BPSS) per la descrizione formale dei Business Process;
2. Collaboration Protocol Profile (CPP) per la definizione dei Profili di Collaborazione;
3. Collaboration Protocol Agreement (CPA) per la definizione dei Contratti di Collaborazione.

I BPSS rappresentano l'astrazione dei Business Process e sono indispensabili per essere usati come riferimenti univoci ai Business Process che essi descrivono. La loro essenza si concentra sulla definizione delle *Business Transaction* e sulla loro combinazione all'interno dei Business Process. Una Business Transaction è l'unità logica di business condotta tra due o più parti che genera uno stato computabile di successo o fallimento [ebGLOSS]. I BPSS sono utilizzati come complemento per la definizione dei Profili e Contratti di Collaborazione.

I CPP sono Profili di Collaborazione che ogni azienda può creare per descrivere le proprie capacità riguardanti i Business Process che decide di eseguire.

I CPA, infine, sono la rappresentazione formale dell'accordo instaurato tra le aziende dopo aver messo a confronto i propri CPP in maniera manuale o semiautomatica.

Il lavoro qui svolto consiste nella definizione di un'architettura, CP-NET, che prevede:

- l'adattamento degli standard BPSS, CPP e CPA al contesto MODA-ML;
- lo sviluppo di due applicazioni web che permettono rispettivamente la gestione degli standard BPSS e CPP.

Nel seguito della tesi descriverò l'architettura CP-NET, le differenze sostanziali con il framework ebXML e il lavoro svolto per applicarla al contesto di MODA-ML.

Il framework ebXML prevede che l'applicazione dei Profili di Collaborazione sia effettuata nel seguente modo: l'azienda A accede ad un Repository per ottenere le informazioni e i dettagli relativi ai componenti standard disponibili e concernenti i Business Process della propria catena produttiva; successivamente l'azienda A crea il proprio CPP in cui specifica le proprie caratteristiche e capacità di business; registra

tale CPP nel Repository per rendersi disponibile ad intraprendere Business Collaborations con altre aziende. I successivi passi prevedono che un'altra azienda B, tramite la ricerca di un partner sul Repository, verifichi la compatibilità del proprio CPP con quello dell'azienda A e provi a trovare un accordo commerciale che vede la sua concretizzazione nella stipulazione del CPA.

In parziale contrasto con i requisiti di ebXML, MODA-ML non contempla la creazione di un supporto alle aziende per reperire i componenti standard appropriati per creare CPP, ma ha un Dizionario contenente tutti i dati relativi alla struttura dei Business Process e dei documenti di scambio.

Il progetto MODA-ML si è sviluppato concentrandosi essenzialmente sulla definizione dei Business Process e dei documenti di scambio; l'obiettivo primario è raggiungere uno standard di scambio tra le aziende del T/A e metterle in grado di agevolare le comunicazioni elettroniche. Sviluppare un repository ebXML è una scelta non prioritaria e in contrasto con la natura non centralizzata del progetto.

L'architettura CP-NET implementa i Profili di Collaborazione secondo l'approccio, consigliato da ENEA, che descrivo di seguito:

1. L'azienda A accede al Repository (il server MODA-ML) per creare immediatamente un CPP. Un'applicazione web permette di interfacciarsi con il Dizionario e guida l'azienda nella creazione o modifica del proprio CPP.
2. Il CPP creato è inviato all'azienda. (Non è prevista, come richiederebbe ebXML, la fase d'immagazzinamento del CPP sul server per la ricerca di nuovi partner commerciali).
3. L'azienda B accede al Repository per crearsi un proprio CPP.
4. L'azienda B vuole instaurare una Business Collaboration con A, che già ben conosce. I due CPP vengono messi a confronto e, in maniera manuale, viene stipulato l'accordo con il CPA.

Prima di poter cominciare la progettazione e lo sviluppo dell'applicazione per supportare le aziende nella creazione di Profili di Collaborazione è stato necessario adattare lo standard ebXML CPP al contesto MODA-ML.

Per utilizzare i CPP secondo la visione ebXML è indispensabile utilizzare i Business Process Specification Schema (BPSS).

Per poter implementare i BPSS è stata dunque fornito un loro adattamento al contesto MODA-ML e sviluppata un'applicazione che ne ha permesso la creazione direttamente dal Dizionario.

MODA-ML si è sviluppato a partire dall'analisi dei Business Process del settore T/A, per poter poi conseguentemente creare una struttura organizzata e più efficiente. Ogni processo è stato analizzato e suddiviso in attività, all'interno delle quali sono state specificate le transazioni necessarie e i relativi tipi di documento di scambio. Tutte le informazioni riguardanti la struttura dei processi sono immagazzinate nel Dizionario MODA-ML.

I BPSS ebXML permettono di descrivere i Business Process in un formato XML con una struttura ben precisa. L'approccio di base è il riutilizzo sistematico di elementi predefiniti.

Per applicare lo standard BPSS al contesto MODA-ML si è dunque reso necessario mappare la struttura dei Business Process MODA-ML nella struttura dei BPSS di ebXML. L'approccio utilizzato è stato di far corrispondere ad ogni Business Process MODA-ML un BPSS ebXML.

Durante la definizione di questa mappatura si sono eseguite alcune scelte per diminuire il numero degli elementi disponibili ma opzionali degli ebXML BPSS limitandosi ad utilizzare quelli indispensabili per i nostri fini.

Lo scopo ultimo di quest'operazione è stato quello di ottenere un BPSS MODA-ML descritto tramite l'utilizzo dello standard ebXML BPSS.

L'adattamento dei CPP al contesto MODA-ML ha richiesto uno sforzo maggiore rispetto a quello dei BPSS.

Un CPP può essere visto come strutturato in quattro livelli distinti ma interconnessi: Process Specification, Delivery Channels, Document Exchange e Transport. Il livello Process Specification si occupa di descrivere le caratteristiche dell'azienda riguardanti i Business Process che vuole eseguire nella Business Collaboration. Per fare questo è indispensabile fare riferimento ai BPSS. Gli altri tre livelli sono invece dedicati alla specificazione dei canali di Trasporto utilizzati, dei Protocolli web supportati, dei

parametri relativi alla gestione della busta ebXML [ebMS], della sicurezza e crittografia.

Nel progetto di tesi si è sviluppato con particolare attenzione il livello di Specificazione dei Business Process (*Layer Process Specification*) limitandosi a supportare gli altri livelli in maniera superficiale. L'obiettivo al quale si è voluti giungere è utilizzare i Profili di Collaborazione per descrivere:

- le capacità relative ai Business Process;
- le personalizzazioni ai documenti sviluppati da MODA-ML.

Il CPP ebXML non possiede la capacità di descrivere preferenze concernenti i documenti di scambio. I tipi di documento MODA-ML sono caratterizzati da un alto numero di elementi ed attributi, alcune di queste entità risultano essere opzionali. Come conseguenza si è riscontrato che i documenti XML sono molte volte dispersivi, sono utilizzati elementi che potrebbero essere ignorati da entrambe le parti ma che vengono allo stato attuale utilizzati per evitare incomprensioni reciproche. Inoltre i documenti MODA-ML, per come sono definiti, permettono di descrivere la stessa transazione in maniera diversa. Nasce dunque la necessità di accordarsi non solo sul Business Process comune e sui documenti che lo compongono, ma anche sulle entità opzionali che l'una o l'altra parte possono esigere come obbligatori o non utilizzati.

Per risolvere questo problema, si è eseguita un'estensione dello standard ebXML CPP, inserendo un nuovo elemento che contenesse l'espressione di questo tipo di preferenze. Mantenendo invariata la politica ebXML che vede il CPP come descrittore delle capacità dell'azienda e il CPA come concretizzazione formale dell'accordo raggiunto, si è attuata l'estensione modificando lo XML Schema ebXML, comune sia a CPP che a CPA.

Anche nel CPA compare dunque un elemento aggiuntivo, ma se nel CPP esso descrive le preferenze espresse da un'azienda, nel Contratto di Collaborazione assume invece il significato di accordo finale sul tipo di documenti da inviare e ricevere reciprocamente.

Una volta definiti gli adattamenti degli standard BPSS, CPP e CPA al contesto MODA-ML, è stato possibile iniziare lo sviluppo delle applicazioni web.

L'applicazione sviluppata per creare i BPSS direttamente dal Dizionario si chiama BPSS-Generator e permette al MODA-ML Team Management di attivare la creazione automatica dei Business Process direttamente dal Dizionario.

Il BPSS-Generator è stato sviluppato per funzionare sul server MODA-ML e potersi interfacciare al Dizionario (che è mantenuto sempre aggiornato dal Team Management). Tramite un'interfaccia web è possibile scegliere il Business Process e generare il BPSS relativo in una delle due lingue supportate: italiano e inglese. Per ogni Business Process esisteranno dunque due BPSS (italiano e inglese) aventi il medesimo identificatore. Questo permette ad aziende di nazionalità diversa di riferirsi con facilità allo stesso Business Process e di instaurare una Business Collaboration lasciando ai Protocolli di Collaborazione il compito di fungere da interpreti dei singoli elementi della transazione.

Alla fine del processo di generazione di BPSS, viene visualizzata una rappresentazione tramite un foglio di stile XSLT che evidenzia come tutte le informazioni relative alla struttura organizzativa dei Business Process ideata da MODA-ML sia rimasta invariata nella fase di mappatura verso ebXML.

Una volta che i BPSS sono stati generati, essi sono immediatamente disponibili in una directory del server MODA-ML, pronti ad essere utilizzati dai CPP come descrizione univoca dei Business Process in formato XML.

Per permettere alle aziende aderenti al progetto MODA-ML di creare dei propri CPP, è stata progettata e implementata un'applicazione web che ha preso il nome di CPP-Editor.

Il CPP-Editor è stato sviluppato in modo tale che l'utente può inserire tutti i propri dati ed esprimere le proprie preferenze in cinque fasi principali che descrivo di seguito:

1. upload del CPP da modificare o creazione di un nuovo CPP;
2. inserimento dei dati personali e scelta del Business Process a cui fare riferimento;
3. scelta del ruolo da interpretare nel Business Process, dei documenti business supportati e dei relativi protocolli di trasporto;
4. personalizzazioni dei documenti di scambio tramite la specificazione di una preferenza per ogni entità di tipo opzionale.
5. Riepilogo e conferma completamento del CPP.

L'applicazione presenta in ogni fase un'interfaccia web che è creata dinamicamente in base alle scelte effettuate nel passo precedente. In questo modo è possibile guidare l'utente nella creazione di un CPP, controllando che non vi siano errori d'inconsistenza. Il Business Process al quale si vuole fare riferimento può essere scelto tra i BPSS relativi, generati con il BPSS-Generator e residenti sul server MODA-ML. Il CPP-Editor controlla quali BPSS sono disponibili e a creare la lista corrispondente nell'interfaccia utente. Successivamente il BPSS è utilizzato nuovamente per mostrare il Business Process all'utente e per verificare che le scelte effettuate siano coerenti.

La fase di editing forse più interessante e che ha richiesto lo sforzo maggiore per essere implementata, risulta essere quella in cui l'utente può personalizzare i documenti di scambio. Per permettere all'azienda di visualizzare tutte le entità opzionali di tutti i documenti supportati nella collaborazione, il CPP-Editor crea una rappresentazione ad albero per ogni tipo di documento MODA-ML. Per ognuna di queste rappresentazioni l'utente può decidere quali elementi e attributi opzionali vuole rendere obbligatori, non necessari o lasciare inalterati. L'applicazione gestisce di volta in volta i dati inseriti apportando le modifiche direttamente nell'elemento XML aggiunto nel CPP.

E' stato richiesto che l'applicazione supportasse non solo la creazione di CPP ma anche la successiva modifica. Nella prima pagina del CPP-Editor, prima di cominciare il processo di editing, è possibile scegliere se fare l'upload di un proprio CPP o procedere nella creazione di un nuovo Profilo. Nel caso della modifica di CPP, i dati immessi tramite l'interfaccia si riflettono in modifiche direttamente apportate sul CPP caricato. Nel caso di creazione di CPP, l'applicazione utilizza un CPP di default creato appositamente per essere utilizzato come un template.

L'architettura CP-NET si rende concreta con l'implementazione del BPSS-Generator e del CPP-Editor per supportare rispettivamente la gestione degli standard BPSS e CPP adattati al contesto MODA-ML.

Il Team Management MODA-ML, grazie al BPSS-Generator, può attivare la creazione dei BPSS relativi ai Business Process definiti nel Dizionario.

Le aziende aderenti al progetto MODA-ML possono creare, mediante il CPP-Editor, i propri Profili di Collaborazione tramite un'interfaccia web interattiva.

Allo stato dell'arte i CPA sono creati in maniera manuale ma non si esclude un'evoluzione futura di CP-NET in cui sia creata un'applicazione che guidi alla stipulazione del Contratto di Collaborazione in maniera semiautomatica.

Nei prossimi capitoli descriverò il lavoro svolto in maniera più dettagliata soffermandomi prima a descrivere il contesto del settore T/A e il progetto MODA-ML (cap. 2), poi descrivendo il framework ebXML e gli standard associati BPSS, CPP e CPA (cap. 3). In seguito descriverò la fase di progettazione (cap. 4) e di implementazione (cap. 5) dell'architettura CP-NET, delle specifiche MODA-ML definite e delle applicazioni web sviluppate.

2 MODA-ML

Scopo di **MODA-ML** (Middleware tOols and Documents to enhAncE the textile/clothing supply chain through xML, www.moda-ml.org [MML]) è dotare le aziende del settore Tessile Abbigliamento di un linguaggio comune di scambio dati. MODA-ML ha sviluppato un insieme di documenti XML standardizzati che possono essere incorporati tanto negli attuali sistemi aziendali quanto nelle nuove soluzioni offerte dai fornitori di tecnologia.

Il progetto è stato finanziato nell'ambito delle Take-Up Actions del programma IST [IST] del V Programma Quadro dell'Unione Europea.

Per meglio comprendere il contesto del settore Tessile e Abbigliamento italiano e le esigenze che hanno portato alla creazione del progetto MODA-ML descriverò gli obiettivi, i problemi che intende risolvere MODA-ML e i vantaggi derivanti dall'adozione di questa tecnologia.

Mi soffermerò in particolar modo su:

- le caratteristiche dell'attuale modello di filiera, evidenziando l'esigenza d'innovazione sia dal punto di vista organizzativo che tecnologico;
- i Business Process MODA-ML;
- il Dizionario MODA-ML.

2.1 Tessile/Abbigliamento italiano

Il settore del Tessile/Abbigliamento, è composto di tutte le imprese che operano nei diversi stadi della lavorazione delle fibre tessili ed è caratterizzato da forti interrelazioni e da una grossa interdipendenza tra le aziende. Tutto questo porta spesso a considerare il settore come un "sistema" che è identificato dalla **filiera** (figura 2.1).

Lo schema qui riportato dà una sintetica esemplificazione della complessa organizzazione strutturale del settore.



Figura 2.1: La Filiera

Il consumatore oggi è l'elemento determinante dell'attività del settore e per questo le imprese devono essere flessibili, attente al mercato e pronte a percepirne le variazioni per adeguarsi ad esse. E' quindi il momento delle continue innovazioni in cui la tecnologia gioca un ruolo basilare anche in un settore storicamente definito a bassa tecnologia.

Il modello tipico, alla cui determinazione hanno sicuramente contribuito fenomeni come il decentramento produttivo o il ricorso a terzisti specializzati, è quello di tipo reticolare. La filiera può essere vista come una rete d'impres e una serie di fornitori specializzati per fasi di lavorazione o per parti del prodotto.

La descrizione del settore porta subito in evidenza come questo, oggi, sia caratterizzato da una miriade di diversi attori, (dai confezionisti ai fornitori di tessuti, dalle tintorie ai laboratori, ai negozi e punti vendita), che costituiscono la frammentata e complessa catena produttiva. La collaborazione tra questi attori è, ormai, un'inderogabile necessità

per poter portare a termine il ciclo produttivo e rendere i prodotti pronti per essere venduti. Purtroppo però, la grossa eterogeneità in termini di dimensioni, tipologia di gestione e la cultura stessa del settore, risultano essere un ostacolo all'individuazione di un modo comune di dialogare e collaborare.

I lotti di produzione sono sempre più piccoli a seguito di una crescente personalizzazione dei capi per il consumatore finale, con vincoli di tempo sempre più serrati; le informazioni da dover scambiare crescono sempre più, sia in numero che in esigenze di dettaglio. Al tempo stesso il numero d'interlocutori è sempre più elevato, proprio a seguito di quei processi di decentramento e internazionalizzazione che hanno sempre più caratterizzato il T/A.

Diventa così necessario rivedere i modelli di business, promuovendo una cultura dell'interoperabilità, cooperazione e collaborazione di filiera e settore, piuttosto che della concorrenza tra protagonisti della stessa catena di attività; occorre inoltre cercare di risolvere il problema della difficoltà dell'interfacciamento tra sistemi differenti, riducendone sprechi ed inefficienze.

Indispensabile è agire per fornire al settore un comune linguaggio e standard di scambio grazie anche alle indiscutibili opportunità offerte dalla rete Internet e dalle nuove tecnologie della comunicazione.

2.2 Adozione di nuove tecnologie e standard

Alcuni elementi che oggi sono punti di debolezza per la filiera del Tessile/Abbigliamento potrebbero diventarne i punti di forza, grazie all'adozione di nuovi modelli organizzativi, di nuove tecnologie e standard.

La nuova opportunità di raggiungere tale scopo è oggi offerta da Internet e dallo standard XML [XML04]; iniziative europee al riguardo stanno nascendo in diversi settori.

Nel T/A il **CEN/ISSS** [CENISS02], branca del Comitato Europeo di Standardizzazione dedicato alle attività per la Società dell'Informazione, promuove il Workshop **TEXSPIN** [TEXPIN]. Questa sigla sta per "Textile Supply Chain Integrated Network" e indica un progetto coordinato da **Euratex**, l'European Apparel and Textile

Organization [EURATEX], che si dà come finalità quella di fornire un framework di modelli di scambio dati. Vengono così offerti modelli, ed un comune linguaggio di scambio, utilizzabili da aziende e fornitori di tecnologie per essere implementati nelle proprie soluzioni.

Internet e XML permettono di aumentare la condivisione delle informazioni fra tutti gli operatori, dai produttori di materie prime ai punti vendita, rendendo trasparenti la programmazione ed il controllo di avanzamento dei capi d'abbigliamento. In questo modo è possibile ridurre i tempi decisionali, amministrativi e di comunicazione tra le diverse componenti della filiera.

Attraverso la condivisione delle informazioni d'interesse comune si possono programmare le attività, controllarne puntualmente l'avanzamento ed introdurre regole di schedulazione per l'ottimizzazione dei carichi di lavoro. Il risultato è il passaggio da un'operatività di tipo "push", in cui si manda in produzione ciò che è possibile realizzare con quello che si ha in magazzino, ad una di tipo "pull", in cui il flusso è programmato in funzione delle esigenze della clientela. Questo significa che la programmazione della produzione deve operare nella direzione contraria rispetto a quella seguita tradizionalmente, partendo dalla spedizione al cliente ed arrivando alla consegna delle materie prime. Lavorando in modalità pull si avrà nei magazzini ciò che il cliente desidera ricevere prima, non ciò che sarebbe stato più semplice produrre velocemente per le aziende.

2.3 I Business Process MODA-ML

MODA-ML si è sviluppato a partire dall'analisi dei Business Process e dei documenti a loro connessi, per poter poi conseguentemente creare una struttura organizzata e più efficiente.

Ogni processo è stato analizzato e suddiviso in **attività**, all'interno delle quali sono state specificate le transazioni necessarie e i **relativi tipi di documento di scambio**. Questa classificazione funge da riferimento per mostrare quali possibili scambi si possono realizzare, ma nulla vieta di sfruttare la flessibilità di XML per implementare

sequenze parziali o diverse, sempre in conformità ad accordi tra le parti **facendo riferimento ad un Business Process comune**.

Al momento sono stati formalizzati un certo numero di Business Process MODA-ML. L'analisi del processo ha portato a poter definire specifici documenti XML corrispondenti alle differenti Transazioni, tramite uno schema di business semplice, ma al tempo stesso esaustivo.

In questo paragrafo mi limiterò a prendere in considerazione solamente il processo "Fornitura Tessuti", per poterlo utilizzare come esempio descrivendone le caratteristiche e la rappresentazione data da MODA-ML.

Sono state individuate tre sotto-attività, e, per ognuna, una serie di operazioni che richiedono lo scambio di informazioni tra le parti (figura 2.2).

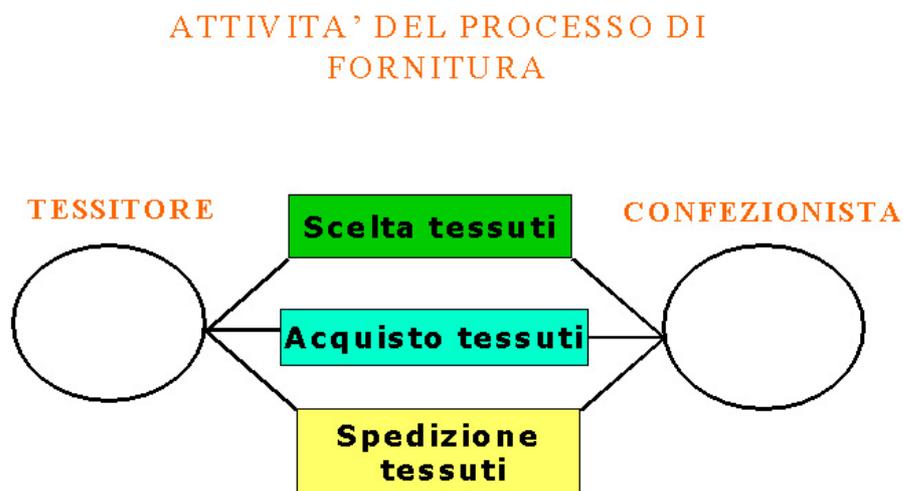


Figura 2.2 Processo MODA-ML "Fornitura Tessuti"

Vediamo nel dettaglio la seconda attività, "Acquisto Tessuti" (Figura 2.3).



Figura 2.3: Attività "Acquisto Tessuti"

Logicamente, il tipo di documento principale che regola questa fase è "Ordine al Fornitore di tessuti". Tale messaggio è usato da parte del Confezionista per inoltrare un ordine di tessuto al proprio Fornitore. All'interno del progetto questo può assumere tre differenti forme:

- un ordine standard, quindi chiuso e a sé stante,
- un ordine aperto, con requisiti ancora da definire, ma che consente l'anticipazione di alcuni aspetti contrattuali,
- una disposizione d'ordine, che evade o scarica un ordine aperto.

Ad ogni ordine fa seguito un "Risposta d'ordine del Fornitore" con cui, per ogni articolo ordinato, questi può confermare, variare o cancellare le specifiche dettate dal Confezionista, che riguardano genericamente quantità, date e luoghi di consegna, prezzi e sconti.

Per il Confezionista è prevista poi la possibilità di inviare una "Modifica", tramite cui poter eventualmente variare i termini specificati in un precedente ordine.

Forse il più interessante dei messaggi riguardanti la fase d'acquisto è il tipo di documento che sintetizza lo "Stato di Avanzamento dell'Ordine". Con questo, infatti, il Fornitore può dare al Confezionista indicazioni sullo stato in cui si trovano i tessuti da lui ordinati, e può riguardare sia tutti gli articoli, tramite un documento di tipo storico, sia i soli articoli non ancora evasi, o ancora solamente i tessuti in stato spedibile.

Tutti i tipi di documento di scambio, risultano essere pubblicamente e gratuitamente disponibili tramite Web, e incorporabili in prodotti e soluzioni commerciali.

Nel prossimo paragrafo descriverò il Dizionario di MODA-ML, struttura dati utilizzata per immagazzinare tutte le informazioni relative all'organizzazione dei Business Process e dei tipi di documento.

2.4 Il Dizionario

Uno strumento che rappresenta il cuore della soluzione MODA-ML è il **Dizionario**, supporta la creazione di schemi e guide e controlla le altre risorse ed unità fornendo le informazioni su elementi, rappresentazioni ed entità dei documenti XML.

(In questo paragrafo mi soffermerò a descrivere in maniera abbastanza dettagliata la struttura del Dizionario. Esso riveste un ruolo centrale nel funzionamento delle applicazioni create in questo progetto di tesi).

2.4.1 Il contesto e i requisiti

Uno degli obiettivi primari del progetto MODA-ML è stata la creazione di un linguaggio comune per lo scambio di messaggi nella filiera del Tessile/Abbigliamento, ovvero di un vocabolario specifico per le transazioni del settore. La definizione del vocabolario riveste un ruolo centrale nel progetto, la sua intera struttura si fonda sulle descrizioni di:

- **tipi di documento XML che sono scambiati** tra gli anelli della filiera,
- **processi commerciali** di cui fanno parte i tipi di documento di scambio,

se non fossero definiti i modelli di documenti, non avrebbe più senso parlare di collaborazione e linguaggio comune.

La definizione dei documenti, e dei processi commerciali che ne fanno uso, ha coinvolto attivamente le aziende pilota ed ha permesso di individuare i requisiti da soddisfare per garantire il successo dell'operazione.

Nel corso della modellazione dei tipi di documento si è rilevato un crescente numero di elementi ed attributi ed il fatto che diversi di questi componenti fossero comuni a più

messaggi. Il gruppo di lavoro di MODA-ML ha riscontrato, quindi, l'esigenza di un supporto capace in primo luogo di operare da repository di tutte le strutture dati create e in grado di semplificarne la manutenzione, e, in secondo luogo, che fornisca un meccanismo per il loro riutilizzo.

Altri requisiti che si sono evidenziati sono:

- esistenza di un pratico **supporto per le traduzioni** (almeno in due lingue);
- strumenti che consentono di definire le **relazioni fra i componenti**;
- **accesso al repository tramite Web**;
- la generazione automatica di derivati quali Guide Utente e XML Schema.

Questo "supporto software" è stato realizzato su base dati di tipo relazionale ed ha preso il nome di MODA-ML Dictionary o "**Dizionario di MODA-ML**".

La base dati che contiene il Dizionario soddisfa alcune esigenze, in primo luogo funzionali:

- relazionare i processi di filiera ai documenti da scambiare;
- memorizzare le strutture dati possibili in XML:
- relazioni padri-figli;
- cardinalità degli elementi;
- associazioni elementi-attributi;
- tipizzazione dei dati;
- semplicità nell'effettuare inserimenti, modifiche o cancellazioni dei dati;
- un'interfaccia in grado di guidare l'operatore attraverso tutte le opzioni e le funzionalità dell'applicazione;
- compattezza esterna dei dati in modo da rendere agevole il backup ed il trasferimento del Dizionario e permetterne in questo modo l'uso anche su altri computer.

L'ambiente di sviluppo dell'applicazione è il database Microsoft Access, su sistema operativo Windows, al quale si deve riconoscere una straordinaria diffusione ed un'ottima capacità di offrire la semplicità operativa richiesta.

2.4.2 Processi e Attività di filiera

I documenti che sono scambiati tra gli anelli di filiera presi in esame, ovvero “Fornitore di Tessuti” e “Confezionista”, sono stati organizzati secondo una struttura gerarchica formata da una serie di processi e attività commerciali: ogni processo racchiude una sequenza di attività, ognuna delle quali è composta di alcune transazioni (si veda par. 2.3).

Le transazioni commerciali esistenti hanno guidato la realizzazione della struttura di questi processi ed hanno permesso di identificare per ogni tipo di messaggio gli attori protagonisti e le legittimità delle transazioni (ad esempio non accadrà mai che l’attore “Fornitore di Tessuti” emetta una “Risposta Ordine del Fornitore di Tessuti” se prima l’attore “Confezionista” non ha inviato un “Ordine al Fornitore di tessuti”).

2.4.3 Dai processi di filiera ai documenti scambiati

Tra processi e attività e tra attività e messaggi non vi sono relazioni uno-a-molti, ma relazioni multi-a-molti poiché, se è vero che ogni processo sia composto di più attività ed ognuna di queste sia caratterizzata da un certo numero di messaggi scambiati, è stato previsto il caso in cui un’attività sia comune a più processi ed il caso in cui un tipo di documento possa essere scambiato in più attività (il diagramma entity-relationship, o E-R, di queste relazioni è presentato in figura 2.4; la visualizzazione MS Access, invece, in fig. 2.5).



Figura 2.4: Diagramma E-R Processi, Attività, Documenti

La tabella “Processi” contiene i campi necessari alla semplice enumerazione e descrizione degli stessi processi, così com’è per la tabella “Attività”.

La tabella che opera da collegamento fra le due relative ai processi e alle attività, legate da una relazione multi-a-molti, è chiamata “REL_ProAtt”.

La tabella “Documenti” contiene quanto necessario all’identificazione dei tipi di messaggi (p.e. codice, descrizione, un puntatore alla radice del documento ecc.).

La tabella che opera da collegamento fra le due tabelle relative ai documenti e alle attività, legate da una relazione multi-a-molti, è chiamata “REL_AttDoc”.

Di seguito la visualizzazione con MS Access:

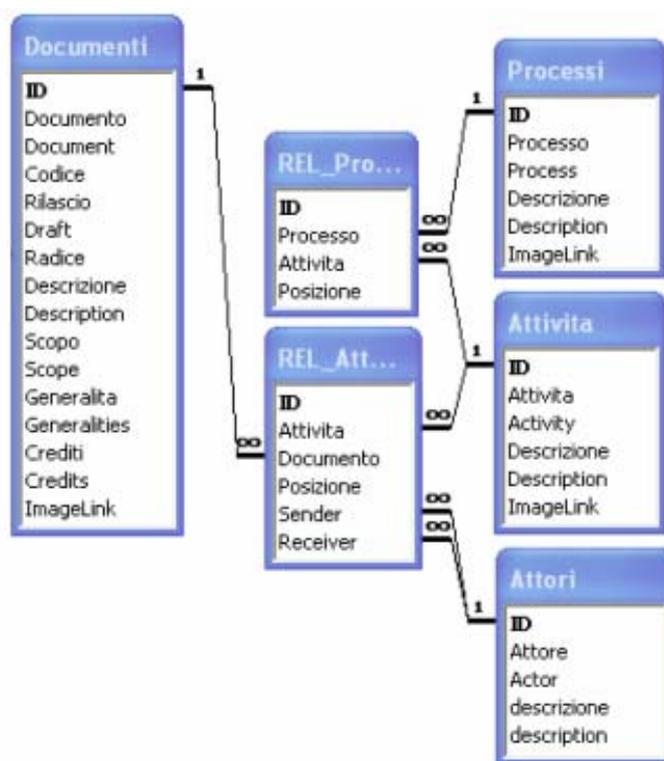


Figura 2.5: Relazioni tra Processi, Attività, Documenti.

2.4.4 Dalla radice del documento ai singoli componenti

La tipizzazione dei dati assunta nel Dizionario ha fatto sì che la relazione “Tipi” occupasse un posto centrale a livello costruttivo poiché opera da collegamento fra le diverse sezioni logiche dell’applicazione. Ogni documento, inteso come tupla della relazione “Documenti”, è, infatti, caratterizzato dalla radice del messaggio XML che

rappresenta, la quale viene selezionata dalla relazione “Tipi”; ogni elemento o attributo è sempre un’istanza di uno di questi tipi (si veda le figure 2.6 e 2.7).

In realtà le relazioni (non intese secondo la terminologia propria della teoria delle basi dati) tra i tipi e i componenti sono molteplici:

- vi è, come detto, la relazione di istanziazione;
- la relazione di assegnazione di figli, per la quale i tipi definiti come complessi posseggono uno o più elementi figli;
- la relazione di assegnazione di attributi per la quale i tipi complessi con contenuto semplice posseggono uno o più attributi.

Si noti che il fatto che un tipo possa essere assegnatario di figli, o attributi, significa che le relazioni d’assegnazione avvengono quando un componente è già stato definito come elemento o come attributo, vale a dire quando è già stato istanziato.

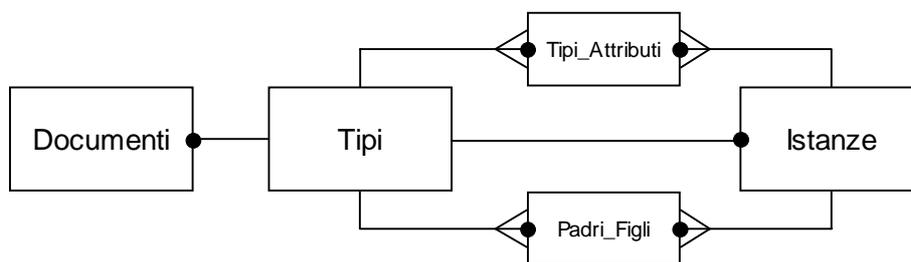


Figura 2.6: Diagramma E-R Documenti, Tipi, Istanze

Di seguito la visualizzazione MS Access:

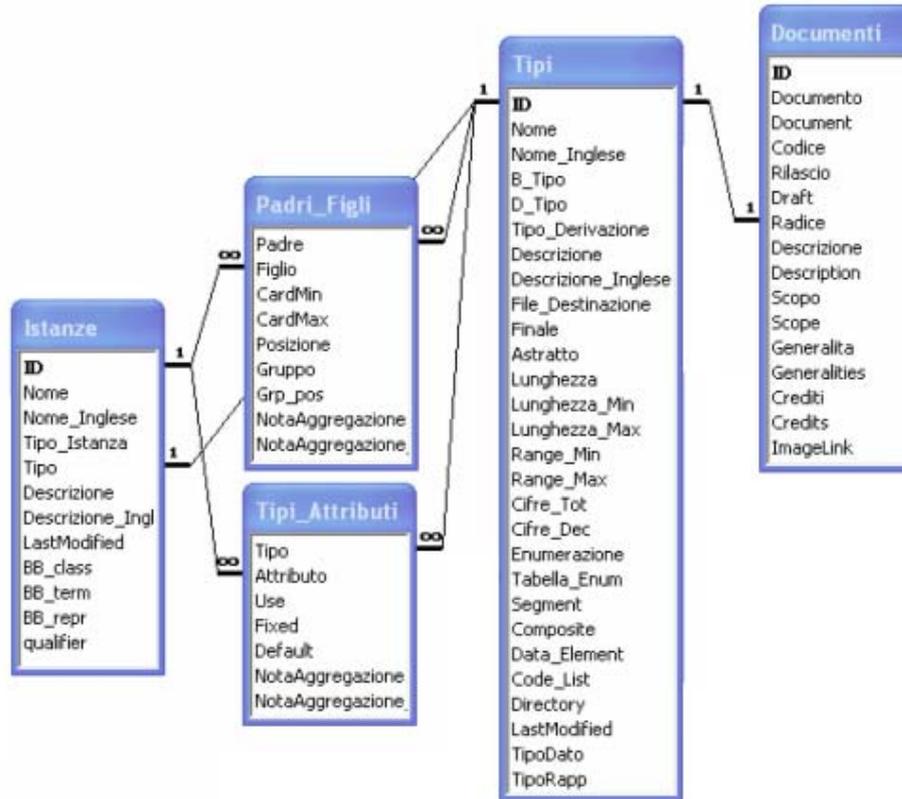


Figura 2.7: Relazioni tra Documenti, Tipi, Istanze

Non andremo oltre nel vedere le tabelle che compongono il Dizionario. Per i nostri scopi è sufficiente ricordare come il Dizionario si occupa di definire l'organizzazione dei Business Process e la struttura dei documenti, vedremo nei capitoli riguardanti il progetto di tesi (cap. 4 e 5) perché e in quale modo sfruttare tali informazioni.

3 EBXML

ebXML (Electronic Business using eXtensible Markup Language, [ebXML]) è un insieme di specifiche [ebXMLa] che congiuntamente permettono di costruire un framework modulare per l'e-business; ebXML si prefigge di fornire un'infrastruttura aperta, basata su XML, che consenta l'utilizzo globale di informazioni relative al commercio elettronico in modo interagibile, sicuro e coerente da parte di tutti i soggetti coinvolti [ebXMLb].

ebXML nasce dall'iniziativa di due grandi organizzazioni internazionali per la definizione di standard, UN/CEFACT (United Nations Center for Trade Facilitation and Electronic Business [UNCEFACT]) e OASIS (Organization for the Advancement of Structured Information Standards [OASIS]), a cui fanno riferimento un gran numero di consorzi industriali e organismi di standardizzazione (IETF, OMG, DISA, CommerceNet, ecc.), e con la collaborazione di grandi aziende private come IBM, Sun Microsystems, Commerce One ed Oracle.

ebXML nasce per venire incontro alle esigenze di aziende di tutti i settori industriali e di tutte le dimensioni, quindi anche delle piccole e medie imprese che in passato non hanno potuto partecipare al commercio elettronico a causa degli alti costi d'implementazione di EDI.

EDI, storicamente, è stato il primo tentativo di definire uno standard per lo scambio di documenti in formato elettronico e può essere descritto come lo scambio di dati riguardanti transazioni di carattere commerciale tra aziende di cui esistono due versioni: ANSI X.12 (vedi [X12]) usato negli Stati Uniti e UN/EDIFACT (vedi [EDIFACT90]) in Europa e nel resto del mondo. Il fatto di essere stato progettato prima dell'affermazione di Internet nel mondo degli affari, ha fatto sì che le sue specifiche definissero non soltanto il formato dei messaggi, ma anche i protocolli di

comunicazione e persino certi requisiti hardware. Questo ne rende più complessa l'implementazione e richiede la realizzazione di costose "Value Added Networks" (VAN). Inoltre, prevede l'uso di una sintassi estremamente concisa che rende i messaggi di difficile comprensione.

Per le grandi aziende che hanno usato EDI per anni, ebXML costituisce un'opportunità per allacciare nuove collaborazioni di e-commerce con un numero più ampio di partner. Per raggiungere questo scopo, deve essere possibile sviluppare applicazioni che le aziende più piccole possano integrare con facilità nel loro sistema informativo. Per ridurre la propria complessità, ebXML considera complementari e non alternative altre iniziative indirizzate al B2B come gli standard SOAP [SOAP00] e UDDI [UDDI01] e i framework RosettaNet [RNET99] ed OAGIS [OAGIS01].

ebXML è definito da una serie di specifiche, ciascuna delle quali definisce una singola parte del framework, (eccetto la specifica "ebXML Technical Architecture Specification Version 1.0.4" [ebTA01] che fornisce una panoramica sulla sua architettura). Ognuna di queste parti costituisce un modulo quanto più possibile autonomo, permettendo così alle aziende di adottare ebXML gradualmente, implementando inizialmente solo alcuni dei suoi servizi.

In questo capitolo descriverò l'architettura ebXML per capire in cosa consiste. Mi soffermerò in particolar modo sugli standard ebXML che sono stati utilizzati per lo sviluppo di un modulo di tale framework: i Profili di Collaborazione.

3.1 Il framework ebXML

L'architettura e-business ebXML deriva da una visione di commercio elettronico dove le imprese si incontrano in un mercato virtuale e svolgono determinati passi per poter instaurare una Business Collaboration.

1. Fase di Ricerca del partner:

- ricerca elettronica (via rete) del partner con cui realizzare una transazione.

2. Fase di progettazione della Business Collaboration:

- definizione del processo di business attraverso lo scambio di messaggi XML;
- accordo su una sequenza standard del processo di business;
- definizione di una chiara semantica di business;
- definizione di template di messaggi XML per lo scambio di informazioni;
- accordo per stabilire regole (mutue o standard) per protocolli di collaborazione tra partner.

3. Fase operativa di scambio informazioni:

- uso di programmi applicativi per lo scambio effettivo delle informazioni; queste applicazioni possono essere soluzioni proprietarie o meno, utilizzando il medesimo standard ebXML.

Illustro brevemente alcuni termini che incontreremo spesso nel corso del documento e che sono essenziali per poter capire i concetti relativi a ebXML.

Party: corrisponde all'azienda (o, come vedremo, parte di essa). E' l'entità che intende intraprendere una Business Collaboration.

Business Collaboration: è la collaborazione commerciale che si vuole instaurare tra due o più Party. Questa collaborazione consiste di un insieme di ruoli collaboranti tra loro attraverso un insieme ben organizzato e ordinato di transazioni business.

Business Process: è il processo commerciale nel quale un Party può impegnarsi all'interno della Business Collaboration. Ogni Business Process è composto da una serie di Business Transaction.

Business Process Specification Schema (BPSS): descrive in maniera formale (in versione XML e/o UMM) il Business Process in cui un Party può impegnarsi nella Business Collaboration.

Registry: è un server che immagazzina una varietà di dati necessari per descrivere e costituire l'attività/lavoro ebXML. Questo server mette a disposizione una serie di informazioni in forma XML quali:

- Business Process e modelli di metainformazioni
- Core Library
- Profili dei protocolli di comunicazione
- Business Library

Quando un Party vuole instaurare una Business Collaboration, richiede al registry di individuare un partner adatto e di cercare le informazioni riguardanti le possibili modalità di collaborazione per poter trattare con esso.

Collaboration Protocol Profile (CPP): è un documento XML associato a ciascun singolo Party; consiste di un insieme di informazioni e caratteristiche che permettono ai Party di accordarsi nella maniera più opportuna prima di cominciare la vera e propria Business Collaboration tramite lo scambio di documenti.

Possiamo raggruppare queste caratteristiche in tre insiemi principali:

- riferimenti e preferenze relativi ai Business Process,
- specificazione dei protocolli di trasporto che verranno utilizzati,
- parametri e certificati relativi alla sicurezza.

Collaboration Protocol Agreement (CPA): una volta che i Party hanno messo a disposizione i propri CPP, i due profili vengono messi a confronto per cercare di creare, possibilmente in maniera automatica, un documento XML comune ai due party, il CPA, che rappresenta e descrive l'accordo raggiunto per cominciare la Business Collaboration.

La seguente figura illustra il funzionamento del framework ebXML [ebXMLibm].

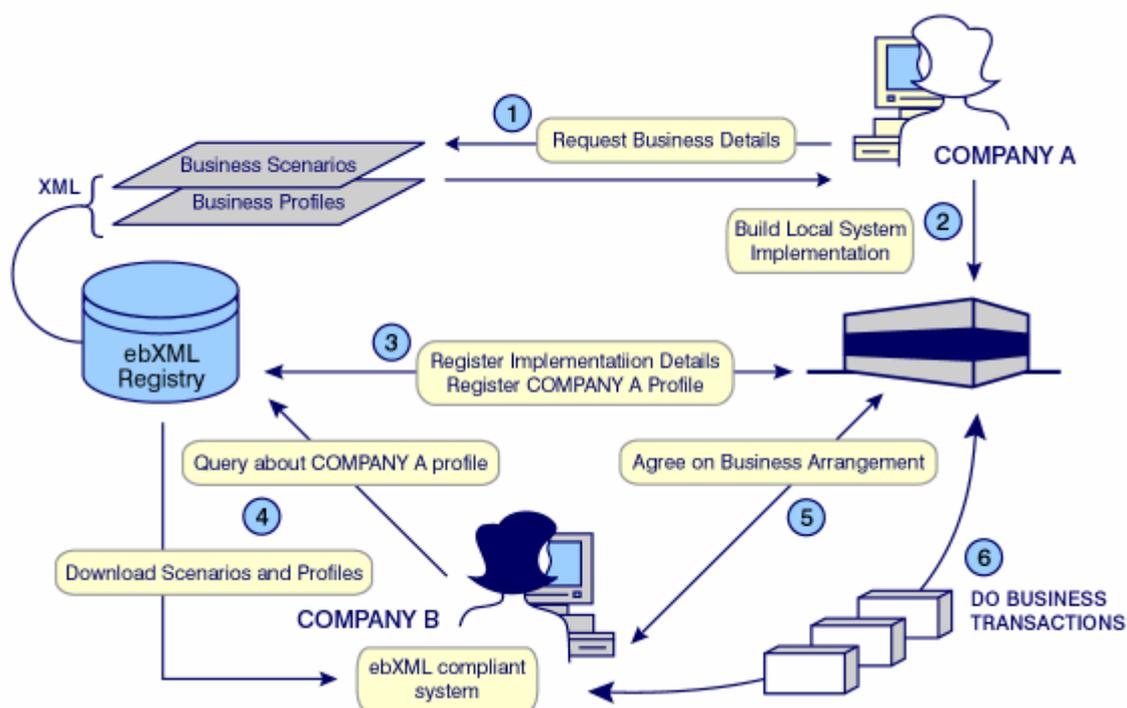


Figura 3.1 Framework ebXML

Di seguito elenco i passi principali previsti da ebXML (mostrati nella figura 3.1) per raggiungere il traguardo di creare un nuovo business tra due Party che non si conoscono ma che sono interessati a intraprendere lo stesso Business Process:

1. La compagnia A accede al Registry per ottenere le informazioni e i dettagli relativi ai componenti standard disponibili e relativi ai Business Process per poi decidere se creare o comprare un'implementazione ebXML appropriata per le proprie transazioni.

In figura si ipotizza che la compagnia A si costruisca la propria implementazione mentre la compagnia B utilizzi un'implementazione prefatta.

2. La compagnia A crea il proprio CPP, che abbiamo visto essere un documento XML rappresentante un profilo formale dell'azienda in questione, descrittore ruoli e protocolli di negoziazione dell'azienda stessa.

3. L'azienda A registra tale CPP nel Registry per rendersi disponibile a intraprendere collaborazioni con altre aziende.

In sintesi questo passo consiste nel descrivere quali Business Process la compagnia A vuole e può intraprendere e offrire tali informazioni alle altre aziende che potrebbero essere interessate a fornire o desiderare tali servizi.

4. Una compagnia B ricerca nel Registry un partner con cui realizzare una Business Collaboration. Il Registry, analizzata la domanda, risponde fornendo alla compagnia B il CPP della compagnia A che dovrebbe soddisfare tali esigenze.

5. La compagnia B verifica la compatibilità tra il proprio CPP e quello della compagnia A poi prova a instaurare una comunicazione per trovare un accordo commerciale e di comunicazione che viene raggiunto con la creazione di un nuovo documento XML, il CPA.

6. Dopo la stipulazione del CPA, le applicazioni di scambio di dati sono configurate, se possibile in maniera automatica, tramite questo documento comune e possono così cominciare la serie di transazioni automatiche che costituiranno la vera e propria Business Collaboration.

Queste transazioni sono viste come documenti XML conformi agli standard e recommendations ebXML.

3.2 ebXML BPSS

La definizione di modalità formali per descrivere Business Process ha lo scopo di garantire l'interoperabilità fra le organizzazioni, facilitando le transazioni commerciali attraverso la reciproca conoscenza dei Business Process. Per raggiungere questo obiettivo, è stato definito lo Schema di Specifica dei Processi di Business ebXML (Business Process Specification Schema, in breve ebXML BPSS).

Un BPSS permette di definire i Business Process integrando la modellazione di processi e-business con le specifiche di componenti software; è basato sul precedente lavoro dell'UN/CEFACT, specificamente sull'UMM ("UN/CEFACT Modelling Methodology" [UMM01]).

Un BPSS supporta la specifica di Business Transaction e la loro combinazione all'interno delle Business Collaborations, può venire dunque utilizzato per controllare l'ordine di scambio dei documenti e dei messaggi tra i Party e la loro consistenza per portare a termine il processo di commercio elettronico desiderato.

La specifica ebXML BPSS si concentra sulla definizione delle Business Transactions e sulla loro combinazione all'interno delle Business Collaboration e viene utilizzata come complemento per la definizione dei profili e accordi nei Protocolli di Collaborazione con i partner commerciali ebXML (ebXML CPP e CPA, si veda par. 3.3). I BPSS generati, secondo l'architettura ebXML, sono collocati nel registry ebXML.

L'architettura del Business Process Specification Schema ebXML consiste nei seguenti componenti funzionali:

- versione UML [UML01] del Business Process Specification Schema
- versione XML del Business Process Specification Schema
- regole di produzione (production rules) che definiscono il mapping fra la versione UML e la versione XML dello schema
- definizione dei segnali di business

L'insieme di tali componenti permette di specificare completamente tutti gli aspetti di run-time di un modello di Business Process, ma ogni componente può essere implementato in maniera indipendente.

3.2.1 Concetti chiave dello standard ebXML BPSS

Il seguente paragrafo descrive i concetti chiave dell'ebXML Business Process Specification Schema secondo le specifiche "Business Process Specification Schema v.1.01" [ebBPSSa].

Lo standard ebXML BPSS provvede le semantiche, gli elementi e le proprietà necessarie per definire Business Collaborations.

Nella figura seguente sono mostrati i concetti basilari di una Collaboration.

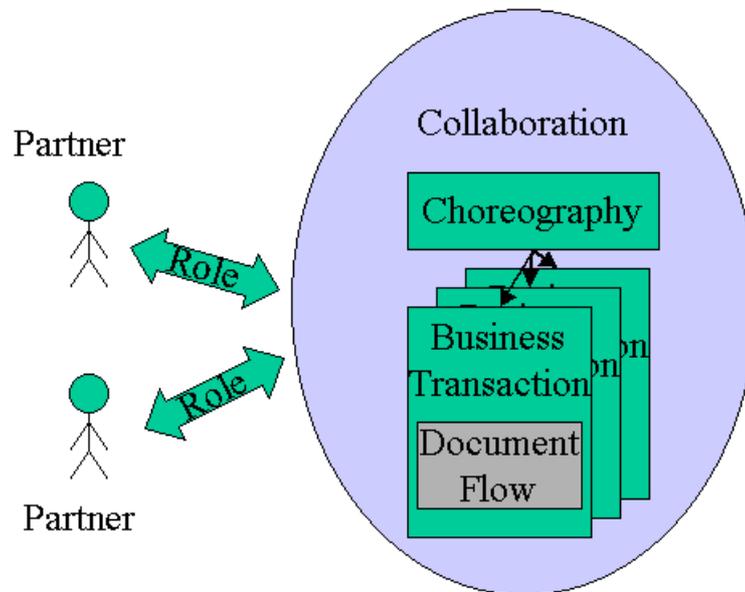


Figura 3.2: Concetti di base di una business collaboration

Due o più Party partecipano a una Business Collaboration interpretando **Ruoli**. I Ruoli interagiscono tra loro attraverso Business Transactions le quali, a loro volta, sono organizzate e ordinate secondo una determinata **Choreography** che determina il modo in cui si susseguono una con l'altra.

Descriviamo ora i concetti chiave di una Business Collaboration in modo tale da focalizzare la sua organizzazione e poter capire meglio in che modo il framework ebXML la gestisce.

3.2.1.1 Business Collaboration

Una Business Collaboration è un insieme di Business Transactions che avvengono tra due partners commerciali. Il BPSS supporta due livelli di collaborazioni business: Binary Collaborations e Multiparty Collaborations.

Le **Binary Collaborations** (Collaborazioni Binarie) si verificano solo tra due **ruoli**. Le **Multiparty Collaborations** (Collaborazioni Multi-Party) sono tra più di due ruoli ma vengono create raggruppando due o più Binary Collaborations.

Per esempio, potremmo considerare tre ruoli A, B e C che in una determinata Business Collaboration interagiscono l'uno con l'altro.

Otterremo tre Binary Collaborations interne a una Multiparty Collaboration:

1. tra A e B,
2. tra B e C,
3. tra C e A.

Nella figura 3.3 è evidenziata questa situazione.

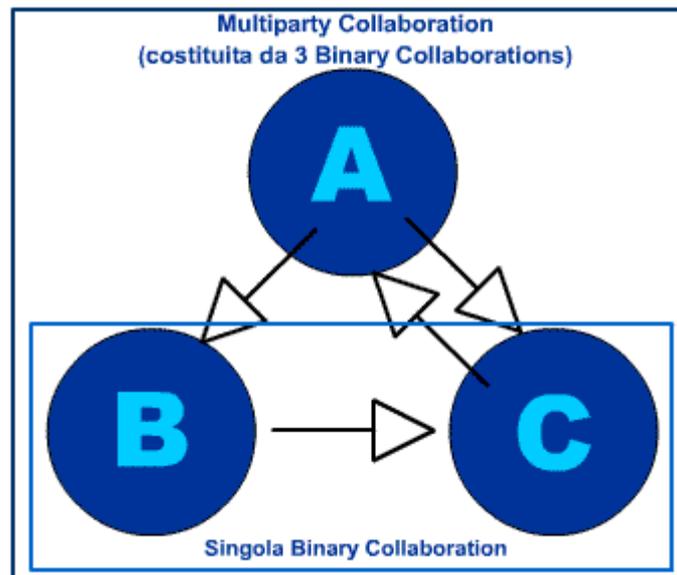


Figura 3.3: Multiparty Collaboration

3.2.1.2 Business Activitiy

Le Binary Collaborations sono espresse come un insieme di **Business Activities** tra due **ruoli**.

Ogni Business Activity rappresenta uno stato particolare al quale si è giunti nella Business Collaboration, ovvero il completamento di uno scopo prefissato per poter avvicinarsi all'espletamento totale del Business Process.

Ogni Business Activity, a sua volta, può essere vista come una **Business Transaction Activity**, cioè l'attività di condurre una singola Business Transaction, oppure come una **Collaboration Activity**, cioè l'attività di condurre un'altra Binary Collaboration.

La capacità di una Binary Collaboration di avere al suo interno attività che possono eseguire altre Binary Collaborations è la chiave per la loro composizione ricorsiva e riutilizzo.

3.2.1.3 Business Transaction

Una **Business Transaction** è l'unità atomica del lavoro in un accordo di scambio tra due Party che giocano **ruoli** opposti, **ruolo di richiesta e ruolo di risposta**, nella collaborazione; equivale dunque ad una singola azione intrapresa da parte di due partner e può comprendere uno (richiesta) o due (richiesta e risposta relativa) documenti business (Document Flow) più una serie di eventuali segnali aggiuntivi per rendere migliore la comunicazione.

Come una Binary Collaboration, una Business Transaction è un'entità riusabile tra due ruoli ma a differenza di essa, come già detto, è atomica, non si può cioè dividere una Business Transaction in un'altra entità di livello più basso.

La via per riutilizzare una Business Transaction sta nel richiamarla da una Business Transaction Activity.

Sono definiti due stati possibili per una Business Transaction: successo o fallimento. Grazie alla definizione degli stati le Business Transactions possono giungere è possibile descrivere nel BPSS i comportamenti relativi agli stati raggiunti (p.e. se la Business Transaction X ha avuto un esito positivo, posso descrivere nel BPSS che la Business Transaction seguente è la Y, altrimenti la Business Transaction Z).

3.2.1.4 Business Document Flows

Una Business Transaction è realizzata come una coppia di **Documenti Business** tra due **ruoli**, uno di richiesta e uno di risposta più una serie di eventuali segnali per una maggiore coordinazione della comunicazione (figura 3.4).

C'è sempre un documento business di richiesta e, opzionalmente, un documento business di risposta (notificazione one-way contro conversazione two-way) dipendentemente dalle semantiche desiderate.

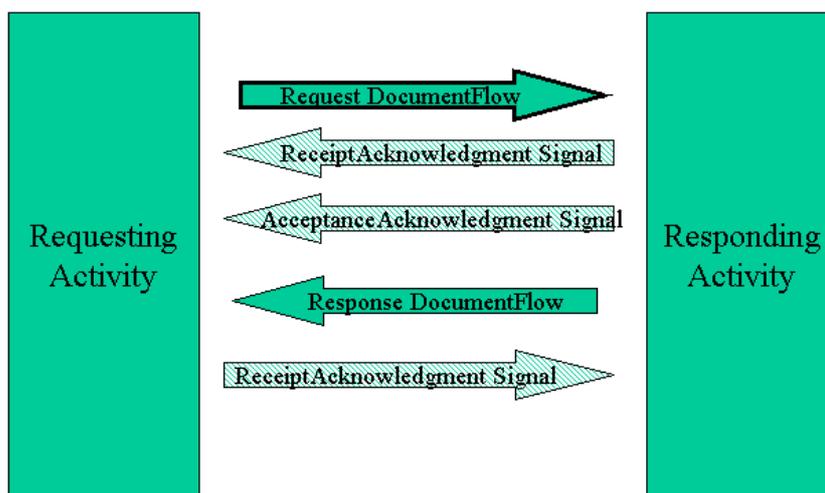


Figura 3.4: Business Transaction

La definizione di questi documenti è raggiunta usando le specifiche sulle ebXML Core Component [ebCC] oppure altre metodologie XML esterne a ebXML.

3.2.1.5 Business Transaction Choreography

La **Business Transaction Choreography** descrive l'ordine e i passaggi tra le varie Business Transactions, Business Activities e Binary Collaborations presenti nel BPSS

fornendo una vera e propria “coreografia” che i due Party dovranno seguire nell’interpretare i ruoli specificati.

In uno strumento UML questo può essere fatto usando un diagramma di attività UML dove sono presenti i concetti di stato iniziale, stato di completamento, attività, sincronizzazioni, transizioni tra attività e guardie su tali transizioni.

La Choreography è descritta nel BPSS ebXML, sia nella sua versione in UML che in quella XML, usando questi stessi concetti.

3.2.2 Struttura XML dello standard ebXML BPSS

La struttura XML del Business Process Specification Schema ebXML prevede una descrizione strutturale ben definita di un particolare Business Process tramite i concetti esposti nel precedente paragrafo.

Un Business Process può essere descritto tramite la definizione dei seguenti elementi: *BinaryCollaboration*, *BusinessTransactionActivity*, *CollaborationActivity*, *BusinessTransaction* e *BusinessDocument* (nella figura 3.5 è presentato il diagramma UML di una *BinaryCollaboration*).

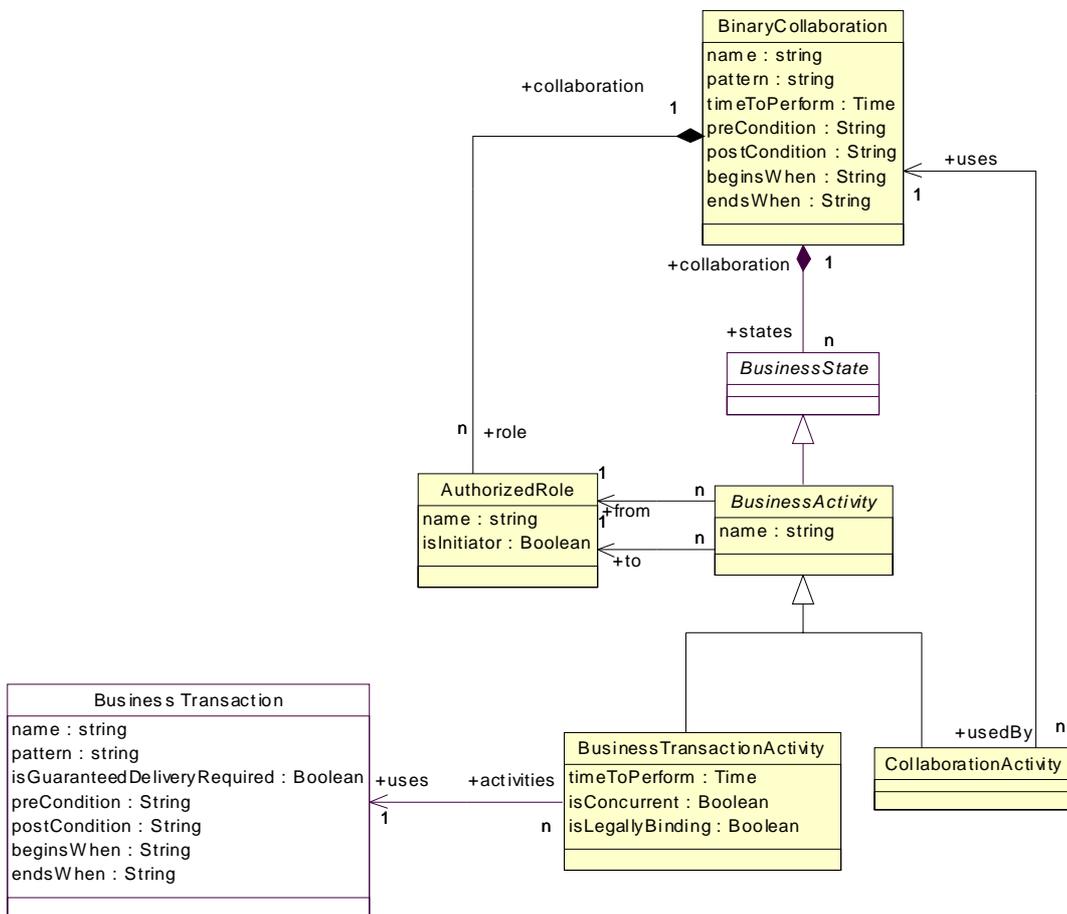


Figure 3.5: Diagramma UML di una Binary Collaboration

Il seguente frammento XML evidenzia l'uso degli elementi di cui ci è parlato e mostra la medesima organizzazione del diagramma UML.

```

<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE ProcessSpecification SYSTEM "...">

<ProcessSpecification name="..." version="..." uuid="...">

  <!-- Business Documents - one or more -->
  <BusinessDocument name="..." nameID="...">
    <Documentation>...</Documentation>
  </BusinessDocument>

  ...

```

```

<Package name="...">

    <!-- Business Transaction - one or more -->
    <BusinessTransaction name="...">
    <RequestingBusinessActivity name="">
        <DocumentEnvelope businessDocument="..."/>
    </RequestingBusinessActivity>
    <RespondingBusinessActivity name=""/>
    </BusinessTransaction>

    ...

    <!-- Binary Collaborations - one or more -->
    <BinaryCollaboration name="..." nameID="...">
    <InitiatingRole name="..."/>
    <RespondingRole name="..."/>
    <BusinessTransactionActivity
        name="..."
        businessTransaction="..."
        fromAuthorizedRole="..."
        toAuthorizedRole="..."/>
    <BusinessTransactionActivity
        name="..."
        businessTransaction="..."
        fromAuthorizedRole="..."
        toAuthorizedRole="..."/>
    <Start toBusinessState="..."/>
    </BinaryCollaboration>

    ...

</Package>

</ProcessSpecification>

```

Come si può notare una *BinaryCollaboration* specifica i ruoli che dovranno essere interpretati, definisce al suo interno alcune *BusinessTransactionActivity* che richiamano le *BusinessTransaction* definite precedentemente. Ogni *BusinessTransaction* fa riferimento a sua volta a uno o due documenti di business.

La caratteristica fondamentale di questo approccio è la composizione ricorsiva che spinge al ri-utilizzo di entità, composte o meno, già definite in precedenza.

3.3 ebXML Collaboration Protocol

Lo scambio di informazioni fra due o più Party richiede che ciascuna parte sia a conoscenza delle Business Collaborations, e quindi dei Business Process, supportate dalle altre parti, i ruoli da queste ricoperti e i dettagli tecnologici circa il modo in cui tali parti inviano e ricevono i messaggi.

Per instaurare la Business Collaboration è necessario che le controparti raggiungano un accordo su tali dettagli e quindi è necessario un dialogo preliminare.

Per facilitare questo scambio di informazioni, ebXML descrive il Collaboration Protocol Profile (CPP) ed il Collaboration Protocol Agreement (CPA). Questi due protocolli sono descritti in un'unica specifica: "Collaboration Protocol Profile and Agreement Specification Version 2.0" [ebCPP].

Sia CPP che CPA utilizzano il Business Process Specification Schema (si veda par. 3.2) per indicare quale Business Process si stia prendendo in considerazione per esprimere le proprie preferenze a riguardo (nella figura 3.6 è illustrata la situazione in un repository secondo la visione ebXML).

Repository

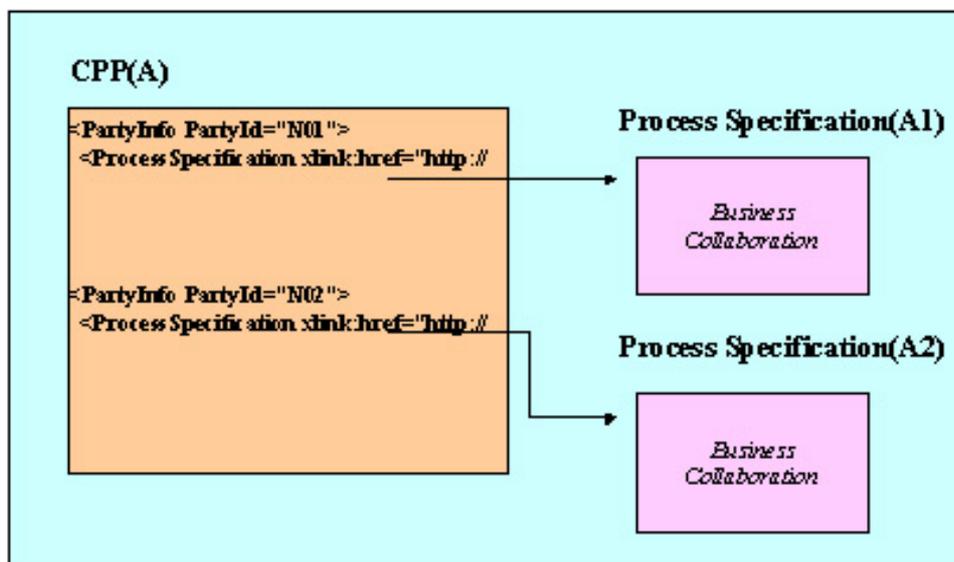


Figura 3.6: Struttura di CPP & BPSS in un ebXML Registry

3.3.1 ebXML CPP

Il Collaboration Protocol Profile (CPP, figura 3.7) definisce le potenzialità e le modalità in cui un'azienda può impegnarsi in un business elettronico con altre aziende. Queste potenzialità sono sia tecnologiche (protocolli di rappresentazione dati e trasporto supportati) sia organizzative (quali Business Collaborations supporta e in che modo vi partecipa).

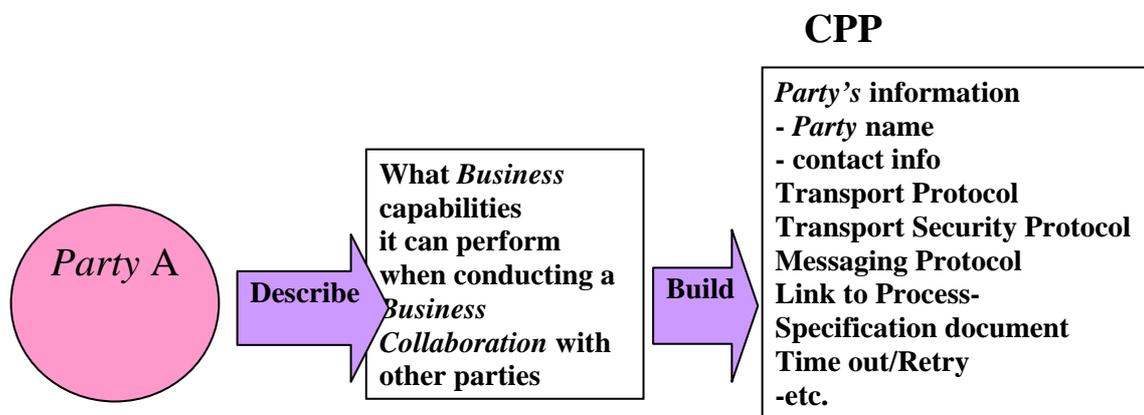


Figura 3.7: Collaboration Protocol Profile (CPP)

Per facilitare la ricerca di possibili Business Partner, i CPP possono essere immagazzinati in un apposito repository pubblico quale l'ebXML Registry.

Tramite un processo di ricerca ed individuazione, codificato nelle specifiche del repository [ebRS], una parte commerciale può trovare adeguati Business Partner.

3.3.2 ebXML CPA

Il Collaboration Protocol Agreement (CPA) esprime le modalità di attivazione ed esecuzione dei servizi messi a disposizione dalle due parti, sui quali esse devono trovarsi in accordo.

Un CPA è creato mediante calcoli e negoziazioni derivanti dalla comparazione di due CPP. Se un CPP descrive le caratteristiche di un Party, un CPA invece descrive le

caratteristiche di più Party coinvolti in una Business Collaboration, in modo tale che queste caratteristiche siano complementari tra loro (figura 3.8).

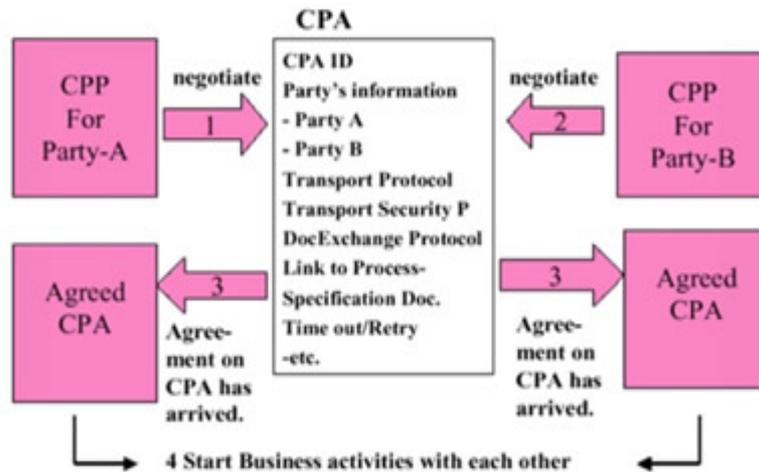


Figura 3.8: Collaboration Protocol Agreement (CPA)

Il processo di creazione di un CPA può dunque essere lungo e difficoltoso se i CPP contengono informazioni troppo eterogenee.

Un esempio semplice e immediato in cui vi sia inconsistenza tra CPP per la creazione di CPA è la situazione in cui due Party presentino i propri Profili di Collaborazione contenenti informazioni su Business Process differenti (i CPP richiamano al loro interno BPSS diversi); è palese il fatto che, per partecipare al medesimo Business Process, diversi CPP devono esprimere le preferenze e i parametri relativi allo stesso processo, quindi, al medesimo BPSS che lo descrive.

L'utilizzo finale del CPA consiste nel poter configurare in maniera automatica i sistemi gestionali dei Party per rendere possibile lo scambio di messaggi durante lo svolgimento delle Business Collaborations desiderate. Il software che implementa lo scambio dei messaggi, ed in generale tutte le interazioni fra le controparti, è uno strato *middleware* che supporta ogni Business Collaboration possibile. Uno dei componenti di questo middleware può essere l'ebXML Message Service Handler [ebMS]. Eventuali

termini e condizioni legali di un accordo commerciale sono al di là degli scopi delle specifiche ebXML.

Le specifiche ebXML, inoltre, non si occupano di spiegare in quale modo la negoziazione atta a formare un CPA avviene, lo sviluppo di tale processo è lasciato agli sviluppatori del software che utilizzerà questi standard.

3.3.3 Struttura XML dello standard ebXML CPP e CPA

La struttura XML del Collaboration Protocol Profile (CPP) e del Collaboration Protocol Agreement (CPA) sono simili, si eviterà quindi di descrivere entrambi gli standard separatamente poiché parti della medesima specifica.

E' possibile suddividere un CPP o un CPA in quattro livelli principali:

- **Process Specification** costituisce il cuore dell'accordo commerciale tra le due parti coinvolte, poiché specifica i servizi che ciascuna di esse può fornire e l'ordine in cui devono essere effettuate le richieste e le risposte; le informazioni si riferiscono al Business Process descritto nel BPSS a cui faranno riferimento sia i CPP che il CPA.
- **Delivery Channels** descrive il modo in cui un attore della comunicazione desidera spedire e ricevere ogni singolo messaggio, attraverso l'indicazione di un elemento *Transport* e di un elemento *DocExchange*, che definiscono rispettivamente le modalità di trasporto e di scambio dei documenti; deve inoltre contenere l'elemento *Characteristics* che descrive le caratteristiche del canale di comunicazione, fra cui quelle riguardanti la sua sicurezza (autenticazione, autorizzazione, riservatezza). Un CPP può contenere più elementi *DeliveryChannel* per aumentare le probabilità di successo nella costruzione di un CPA.
- **Document Exchange** fornisce le informazioni necessarie a configurare correttamente il Message Service in base a quanto specificato dal Delivery Channel, permette di abilitare e configurare la comunicazione affidabile dei messaggi e, se necessario, si occupa di criptare e/o firmare il documento che deve essere trasmesso oppure di decrittare il documento ricevuto. Se i requisiti

previsti dal Delivery Channel non possono essere soddisfatti dal Message Service, allora dovranno essere garantiti dal layer Transport.

- **Transport** è responsabile della trasmissione dei messaggi attraverso il protocollo di comunicazione prescelto; i protocolli supportati sono HTTP [HTTP99], SMTP [SMTP82] e FTP [FTP85].

Ora descriverò alcuni dettagli di un ebXML CPP in termini di individuali elementi XML in modo tale da comprenderne le funzionalità salienti; tuttavia non mi soffermerò a descrivere la totalità degli elementi XML e tutti gli attributi relativi disponibili ma mi limiterò a evidenziare la struttura principale dello standard. Il mio scopo è descrivere un CPP in modo da comprendere i successivi capitoli riguardanti la progettazione e l'implementazione del progetto di tesi che trattano del medesimo standard ripreso, adattato, esteso ed applicato al contesto MODA-ML (cap. 4 e 5). Tralascierò quindi volutamente quegli elementi e attributi che non sono stati utilizzati nella specifica MODA-ML poiché non utili ai nostri fini e poiché, per definizione, sono opzionali.

Un CPP deve avere la seguente struttura e gli elementi XML devono comparire in questo ordine:

```
<?xml version="1.0"?>
<tp:CollaborationProtocolProfile
  xmlns:tp="http://www.oasis-open.org/committees/ebxml-cppa/schema/cpp-
  cpa-2_0.xsd"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xlink="http://www.w3.org/1999/xlink"
  xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xsi:schemaLocation="http://www.oasis-open.org/committees/ebxml-
  cppa/schema/cpp-cpa-2_0.xsd cpp-cpa-2_0b.xsd"
  tp:cppid="uri:companyA-cpp" tp:version="2_0b">

  <tp:PartyInfo> <!--one or more-->
  ...
</tp:PartyInfo>

  <tp:SimplePart id="..."> <!-- one or more -->
  ...
</tp:SimplePart>

  <tp:Packaging id="ID"> <!--one or more-->
  ...
</tp:Packaging>
</tp:CollaborationProtocolProfile>
```

L'elemento *CollaborationProtocolProfile* è la radice dell'istanza CPP, gli attributi relativi definiscono i vari namespace utilizzati all'interno del documento XML, lo XML-Schema per la convalidazione, l'identificatore e la versione dello standard CPP a cui fa riferimento l'istanza.

L'elemento *CollaborationProtocolProfile* deve contenere i seguenti elementi figli:

- *PartyInfo*: comprende tutte le informazioni riguardanti un singolo Party.
- *SimplePart*: elementi che descrivono i costituenti usati per la creazione dei messaggi di scambio (ebMS).
- *Packaging*: elementi che descrivono come i costituenti payload e Message Header sono strutturati per la trasmissione (ebMS).

Descriverò di seguito la struttura dell'elemento *PartyInfo* il cui scopo è permettere, all'azienda che vorrà definire un proprio CPP, di descrivere le proprie capacità e i dettagli riguardanti la Business Collaboration.

Un CPP è creato da un'organizzazione commerciale o azienda che potrebbe voler descrivere separatamente diverse sottosezioni che la compongono. Per questo motivo la specifica dell'ebXML CPP definisce l'elemento *PartyInfo* in modo tale che possa essere presente una o più volte. In questo modo può rappresentare la totalità di un'azienda o più parti della stessa (e essere utilizzato sia dalle grandi aziende che da quelle medie e piccole).

```
<tp:PartyInfo
  tp:partyName="..."
  tp:defaultMshChannelId="..."
  tp:defaultMshPackageId="...">

  <tp:PartyId tp:type="...">
  ...
  </tp:PartyId>

  <tp:PartyRef xlink:href="..." />

  <tp:CollaborationRole> <!-- one or more -->
  ...
  </tp:CollaborationRole>

  <tp:Certificate> <!-- one or more -->
  ...
  </tp:Certificate>

  <tp:SecurityDetails> <!-- one or more -->
```

```

...
</tp:SecurityDetails>

<tp:DeliveryChannel> <!-- one or more -->
...
</tp:DeliveryChannel>

<tp:Transport> <!-- one or more -->
...
</tp:Transport>

<tp:DocExchange> <!-- one or more -->
...
</tp:DocExchange>

</tp:PartyInfo>

```

Associato ad ogni Party vi sono caratteristiche quali:

- un nome;
- un identificatore;
- un link a eventuali informazioni aggiuntive relative al Party (come nomi, numeri di telefono, indirizzi, locazione geografica dell'azienda o informazioni più strettamente tecniche riguardanti, per esempio, i Business Process ai quali il Party può partecipare);
- le Business Collaboration alle quali ha la capacità di poter partecipare e le modalità in cui si rende capace di interpretare un ruolo;
- certificati utilizzati per la sicurezza e l'eventuale crittografia dei messaggi;
- i protocolli che il Party può usare per spedire e/o ricevere i messaggi e i parametri per le modalità di scambio di dati.

Per i nostri scopi ci soffermeremo solamente sull'elemento *CollaborationRole* relativo alle Business Collaborations supportate.

Di seguito la struttura dell'elemento *CollaborationRole*:

```

<tp:CollaborationRole>

  <tp:ProcessSpecification
    tp:version="2.0"
    tp:name="..."
    xlink:type="simple"
    xlink:href="..."
    tp:uuid="..."/>

  <tp:Role
    tp:name="..."
    xlink:type="..."

```

```

        xlink:href="..." />
        <tp:ApplicationCertificateRef tp:certId="..." />
        <tp:ServiceBinding>...</tp:ServiceBinding>
</tp:CollaborationRole>

```

L'elemento *CollaborationRole* contiene i seguenti elementi figli:

- *ProcessSpecification*: elemento che definisce quale Business Process Specification Schema, descrivente un particolare Business Process, si sta considerando in una particolare Business Collaboration,
- *Role*: il ruolo “interpretato” in questo Business Process
- *ApplicationCertificateRef*: parametro relativo alla sicurezza
- *ServiceBinding*: elemento che specifica quali messaggi di scambio, relativi al Business Process, il Party può spedire o ricevere. All'interno di questo elemento ogni messaggio è considerato singolarmente per poterne specificare i vari parametri.

Di seguito viene fornito un esempio dell'elemento *ServiceBinding* in cui sono specificati solamente due messaggi, uno per l'invio e uno per la ricezione:

```

<tp:ServiceBinding>
    <tp:Service>...</tp:Service>
    <tp:CanSend>
        <tp:ThisPartyActionBinding tp:id="..." tp:action="...">
            <tp:BusinessTransactionCharacteristics ... />
            <tp:ActionContext
                tp:binaryCollaboration="..."
                tp:businessTransactionActivity="..."
                tp:requestOrResponseAction="..." />
            <tp:ChannelId>...</tp:ChannelId>
        </tp:ThisPartyActionBinding>
    </tp:CanSend>
    <tp:CanReceive>
        <tp:ThisPartyActionBinding tp:id="..." tp:action="...">
            <tp:BusinessTransactionCharacteristics ... />
            <tp:ActionContext
                tp:binaryCollaboration="Request Purchase Order"
                tp:businessTransactionActivity="..."
                tp:requestOrResponseAction="..."
            <tp:ChannelId>...</tp:ChannelId>
        </tp:ThisPartyActionBinding>
    </tp:CanReceive>
</tp:ServiceBinding>

```

Ne descrivo brevemente gli elementi più significativi:

- *ThisPartyActionBinding*: l'azione che il Party è in grado di spedire o ricevere. E' identificata da un identificatore, id, e un nome, action;
- *ActionContext*: in questo elemento vengono definite alcune caratteristiche per definire l'azione nel contesto del BPSS (*binaryCollaboration*, *businessTransactionActivity*, *requestOrResponseAction*);
- *ChannelId*: identifica il canale di trasporto utilizzato per l'azione.

Non mi dilungherò oltre nella spiegazione degli elementi XML che compongono il CPP. Nei prossimi capitoli riprenderemo questa struttura per descrivere l'adattamento effettuato al contesto MODA-ML e l'estensione allo XML Schema che si è stata eseguita per rispondere ad alcune esigenze del progetto.

4 PROGETTAZIONE

4.1 Il Problema

Il progetto MODA-ML ha raggiunto, allo stato attuale, il traguardo di fornire un formato XML per lo scambio di documenti corrispondenti alle più comuni Business Transaction.

Uno standard comune per la rappresentazione delle informazioni di business tra le aziende del settore tessile/abbigliamento è essenziale per poter velocizzare il processo di una filiera che ha, come suo punto critico, quello della comunicazione tra i vari attori della catena produttiva (si veda a questo proposito il capitolo 2).

Lo scambio di documenti tra le applicazioni delle aziende che vogliono utilizzare lo standard MODA-ML, non è ancora completamente automatizzato. L'invio e la ricezione di ogni tipo di documento XML devono venire precedentemente concordati tra i partner coinvolti, tramite canali di comunicazione "primitivi" quali sono il telefono e il fax, con la conseguente perdita di tempo per espletare l'intero processo di allineamento tra i sistemi.

Un approccio per ridurre i tempi di avvio della comunicazione di business è quello di rendere semi-automatico il raggiungimento di un accordo tra le aziende; a questo scopo occorre:

- individuare il Business Process MODA-ML che si intende realizzare;
- scegliere quali messaggi del Business Process utilizzare;
- dichiarare il tipo di protocollo di trasporto usato per lo scambio di ogni messaggio;
- decidere i parametri relativi alla sicurezza;
- configurare opportunamente il software gestionale.

Per risolvere questi punti tramite l'ausilio di applicazioni software, le aziende devono poter specificare le proprie capacità e preferenze in maniera formale giungendo alla

definizione di un Profilo di Collaborazione. Successivamente i Profili delle diverse aziende verranno confrontati per raggiungere l'accordo finale. Una metodologia per poter implementare questa soluzione è fornita dal framework proposto da ebXML e dalle relative specifiche sui Profili di Collaborazione (CPP), Contratti di Collaborazione (CPA) e Business Process Specification Schema (BPSS).

I Profili di Collaborazione sono stati ideati per permettere alle aziende di descrivere le proprie capacità e preferenze; i Contratti di Collaborazione servono per descrivere l'accordo a cui si è giunti tra le aziende dopo il confronto tra i CPP; i Business Process Specification Schema sono una metodologia per descrivere in maniera formale i Business Process a cui fanno riferimento CPP e CPA (si veda il capitolo 3).

4.2 Architettura CP-NET

Nella fase di progettazione si è cercato, come prima cosa, di ideare e sviluppare un'architettura, il più possibile in linea con il framework ebXML, che consentisse la gestione dei Business Process e dei Profili di Collaborazione nel contesto MODA-ML. Idealmente si sarebbe potuto implementare il framework ebXML esattamente come viene definito dalle specifiche dello standard (vedi par. 3.1), ma diversi motivi mi hanno spinto (in accordo con il gruppo di ricerca ENEA) a scegliere una soluzione diversa.

L'architettura ebXML è centralizzata, il Registry è in questo caso indispensabile per instaurare una qualsiasi Business Collaboration. (Ricordiamo che il Registry ebXML si preoccupa di fornire alle aziende gli scenari di business e i componenti standard per permettere loro di definire localmente i CPP. Successivamente avviene l'upload dei CPP sul Registry per la ricerca di un partner commerciale).

L'approccio utilizzato per la creazione dei Profili di Collaborazione secondo la visione di CP-NET è diverso. Non è nelle intenzioni di ENEA sviluppare il framework ebXML in modo tale che un server centrale supporti la ricerca dei partner commerciali, per questo motivo si è sviluppata l'architettura CP-NET in modo tale che si limitasse a supportare la creazione dei Profili di Collaborazione ebXML.

I Profili di Collaborazione vogliono quindi essere utilizzati solo per formalizzare la descrizione delle aziende e successivamente per raggiungere il traguardo di stipulare un Contratto di Collaborazione formale. In questo modo ogni azienda (o fornitore di servizi) è libero di poter gestire i propri CPP e CPA.

I BPSS devono essere forniti alle aziende da MODA-ML in modo tale da fungere da punto di riferimento univoco nella descrizione dei Business Process (la presenza di Business Process differenti porterebbe a situazioni non coerenti di ambiguità e incomprensione). Per questo motivo si è scelto di creare un'applicazione web che permettesse al gruppo MODA-ML di creare in maniera semplice, automatica e immediata i BPSS MODA-ML relativi ai Business Process descritti nel Dizionario e di renderli immediatamente disponibili sul server. Questa applicazione ha preso il nome di **BPSS-Generator**.

Diversamente dal BPSS, un Profilo (CPP) deve invece essere creato o modificato dall'azienda che intende utilizzarlo.

Poiché gran parte delle informazioni contenute nel Profilo di Collaborazione sono inerenti ai Business Process (facendo riferimento ai BPSS relativi) e alle informazioni contenute nel Dizionario MODA-ML, si è scelto di creare, anche in questo caso, un'applicazione web: il **CPP-Editor**.

Grazie a questa soluzione MODA-ML fornisce un supporto alle aziende per la gestione dei Profili di Collaborazione garantendo un servizio sempre aggiornato tramite le eventuali revisioni effettuate al Dizionario e ai BPSS.

D'altro canto, essendo BPSS e specifiche dei CPP pubblici, ogni azienda è libera di scegliere come creare il proprio CPP, anche senza usare il CPP-Editor.

L'insieme delle due applicazione web, BPSS-Generator e CPP-Editor, unitamente alle specifiche BPSS, CPP e CPA adattate a MODA-ML ed estese, ha preso il nome di CP-NET (Collaboration Profile Networking Enterprise Technology).

Mostriamo qui di seguito una figura che illustra l'architettura progettata di CP-NET e le varie fasi che compongono il processo di creazione dei Protocolli di Collaborazione.

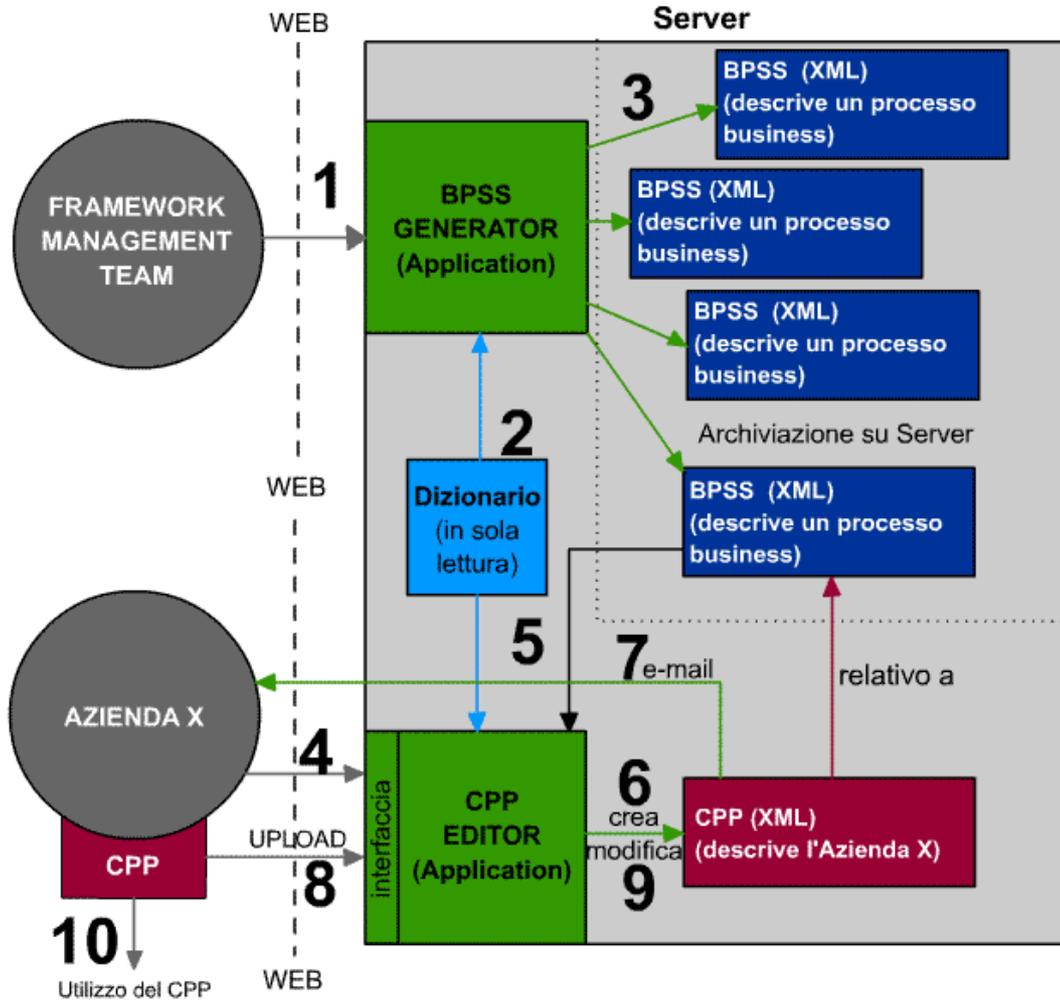


Figura 4.1 Architettura di CP-NET

Di seguito è descritto ogni singolo passo del processo di creazione di BPSS e di un Profilo di Collaborazione secondo l'architettura CP-NET (mostrata in figura 4.1) applicata al contesto MODA-ML:

1. lo staff di MODA-ML accede al server e, tramite l'interfaccia web del BPSS-Generator, sceglie quale Business Process intende esplicitare sottoforma di BPSS (di norma, tutti saranno generati e disponibili sul sito);
2. il BPSS-Generator accede al Dizionario MODA-ML e ricerca le informazioni che permettano di descrivere il Business Process scelto;
3. il BPSS-Generator crea in maniera automatica il BPSS MODA-ML corrispondente al Business Process scelto. Una volta generati, tutti i Business Process sono archiviati sul server e resi disponibili nel web;
4. un'azienda X, partecipante al progetto MODA-ML, decide di creare un proprio CPP e accede al CPP-Editor tramite l'interfaccia web relativa. Seguendo le indicazioni offerte dall'interfaccia, l'azienda X inserisce le informazioni descrittive delle proprie capacità e preferenze relative alla Business Collaboration che intende instaurare;
5. durante il processo di creazione del CPP, il CPP-Editor accede al Dizionario MODA-ML e al BPSS relativo al Business Process scelto dall'utente per verificare la consistenza dei dati e reperire ulteriori informazioni relative ai documenti di scambio;
6. il CPP-Editor genera il file XML contenente il CPP;
7. il CPP viene inviato come allegato e-mail all'azienda X;
8. l'azienda X ha ora a disposizione un CPP immagazzinato nel proprio sistema gestionale. Per modificare il proprio Profilo di Collaborazione, l'azienda può utilizzare nuovamente il CPP-Editor caricando il documento tramite la medesima interfaccia web;
9. per modificare il CPP si ripete il medesimo meccanismo utilizzato nel processo di creazione. Per fare questo si accede nuovamente sia al Dizionario MODA-ML sia al BPSS a cui fa riferimento il CPP. (Una volta completato il processo di modifica di quest'ultimo, esso viene inviato nuovamente come allegato e-mail);
10. una volta che l'azienda ritiene che il proprio CPP sia completo ed esaurientemente chiaro a descrivere le proprie capacità e preferenze, può confrontare il proprio CPP, in maniera manuale o semiautomatica, con quello di

un'altra azienda, allo scopo di concordare una sorta di contratto di business (la via suggerita per fare questo è la formazione del CPA, argomento trattato in questa tesi ma non supportato dalle applicazioni del progetto)

Come si può notare dal framework descritto, mentre il BPSS viene archiviato sul server e reso disponibile in rete, il CPP viene invece inviato all'azienda che l'ha creato o modificato. Il motivo di questa diversità è banale: un BPSS è una descrizione univoca di un Business Process MODA-ML, per questo motivo deve esserne una sola istanza per ogni Business Process per evitare situazioni di ambiguità tra le aziende; un CPP è invece una descrizione di un'azienda ed è da questa utilizzato per raggiungere un accordo commerciale con terze parti. Per questo motivo il CPP deve riceverlo e possibilmente immagazzinarlo nel proprio sistema gestionale la sola azienda descritta in esso.

4.3 Specifiche MODA-ML

Le applicazioni che compongono CP-NET sono state realizzate per gestire i BPSS e CPP in formato XML (per i BPSS è possibile definire anche la versione UML, si veda par. 3.2).

E' indispensabile soffermarsi a descrivere le specifiche MODA-ML BPSS e CPP che si basano su gli omonimi standard ebXML (si veda il capitolo 3) e le scelte eseguite per ricavare tali specifiche.

4.3.1 MODA-ML BPSS

Per raggiungere il traguardo di ottenere una specifica MODA-ML BPSS, è stato necessario mappare i Business Process MODA-ML descritti nel Dizionario con la semantica XML dei BPSS ebXML e applicare tale mappatura nel BPSS-Generator (si veda la figura 4.2).



Figura 4.2

Per spiegare la relazione tra Business Process in chiave MODA-ML e Business Process secondo le specifiche ebXML, farò riferimento, poiché inevitabile in questo contesto, a concetti di cui si è parlato nei precedenti capitoli:

- i Business Process MODA-ML (vedi par.2.3),
- il Dizionario MODA-ML (vedi par.2.4),
- la struttura XML di un ebXML BPSS (vedi par.3.2.2).

In MODA-ML non vi sono informazioni sulle possibili coppie di messaggi “richiesta-risposta” che vengono invece considerate, in ebXML, l’astrazione delle Business Transaction di base, unità atomiche per la completa definizione del Business Process nel BPSS. I processi MODA-ML sono stati definiti dando loro una struttura che potesse facilitare la comprensione dei processi stessi: suddividendo il processo in *attività* composte a loro volta da singole *transazioni commerciali*. La struttura dei Business Process, visti secondo la visione ebXML, è leggermente diversa: concetti come *attività* e *transazioni* permangono anche se con un significato diverso da quello attribuito da MODA-ML.

Lo sforzo che si è fatto è stato quindi quello di mappare la struttura di Business Process MODA-ML con la struttura dello standard ebXML BPSS.

Tutti i processi MODA-ML sono collaborazioni binarie; verrà quindi presa in considerazione solo la collaborazione tra due ruoli ignorando la complessità che ne deriva dai processi che prevedono la partecipazione di più di due Partner commerciali

(Party). Nel BPSS ebXML, il caso in cui un processo abbia al suo interno più di due attori in gioco, viene gestito attraverso l'elemento XML *MultiPartyCollaboration* che, allo stato attuale del progetto MODA-ML e dei processi in esso definiti, non è di nostro interesse conoscere in maniera approfondita (se ne è comunque parlato nel par. 3.2).

Dobbiamo inoltre considerare l'esigenza del progetto MODA-ML di consentire alla singola azienda di specificare quali messaggi del processo spedire per via elettronica e quali ignorare, fino al caso estremo di spedire anche un solo documento XML del Business Process.

Per questi motivi si è scelto di associare ogni documento di business a un'unica Business Transaction ebXML, composta dalla sola Requesting Business Activity (scelta implementativa chiamata "notificazione one-way". Se ne è parlato nel par. 3.2.1.4).

Si è poi scelto di mantenere l'informazione riguardante il raggruppamento in attività interne ai processi MODA-ML definendole come Binary Collaborations di secondo livello all'interno della Binary Collaboration principale rappresentante il processo nel suo complesso (scelta anche questa permessa dalle specifiche ebXML).

Per capire meglio questa relazione osserviamo lo schema seguente:

MODA-ML		ebXML
1 Processo	>	1 Binary Collaboration principale (di primo livello)
m Attività	>	m Binary Collaboration (di secondo livello, richiamate all'interno della principale)
n Documenti XML	>	n Business Transaction Activity
	>	n Business Transaction

Schema1: Mappatura dei Business Process tra MODA-ML e ebXML

Ogni Binary Collaboration, rappresentante un'attività MODA-ML, richiamerà al suo interno le varie Business Transaction.

Nella seguente figura è illustrato chiaramente questa struttura ad albero.

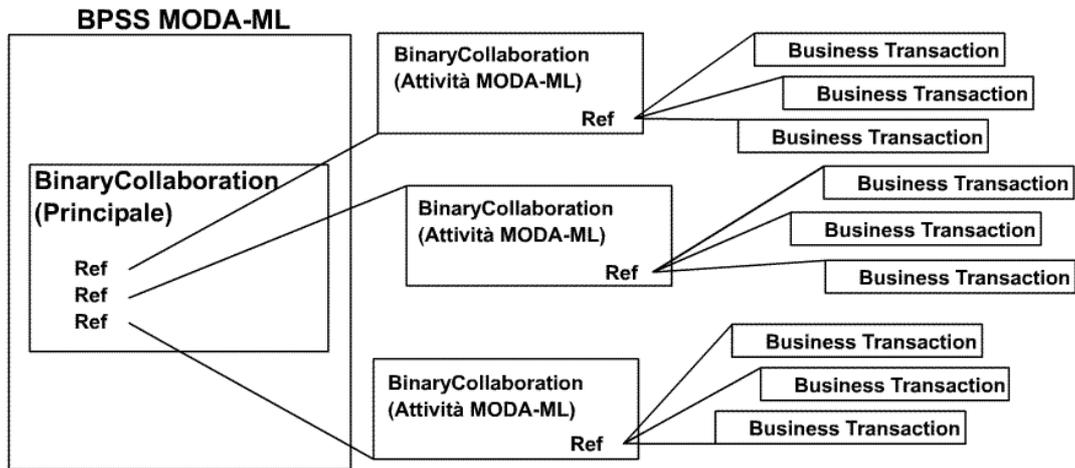


Figura 4.3 Struttura di un BPSS MODA-ML

Ogni entità MODA-ML (processo, attività, transazione) è stata rappresentata con la semantica ebXML; per ottenere una descrizione di Business Process priva di ambiguità si sono utilizzati alcuni attributi definiti dallo standard ebXML relativi alla specificazione di identificatori associati alle entità. Utilizzando le enumerazioni e alcuni codici del Dizionario MODA-ML si sono creati tali identificatori.

Di seguito un esempio (relativo alla versione 2003-1 del Dizionario e al processo “Produzione Tessuto”):

MODA-ML		Identificatore corrispondente
Processo 3	>	2003-1_Proc-3
Attività 12	>	Att-12
Tipo Documento 1	>	OF51-013
Tipo Documento 2	>	OF51-014

Schema2: Identificatori

Osservando questo esempio possiamo intuire:

- il processo 3 nella tabella “Processi” del Dizionario (versione 2003-1) MODA-ML è stato mappato con l’identificatore *2003-1_Proc-3*
- la sola attività presente nel processo “Produzione Tessuto”, dodicesima nella tabella delle attività del Dizionario, è stata mappata con l’identificatore *Att-12*
- i due documenti business sono stati mappati con il codice relativo presente nella tabella “Documenti”, *OF51-013* e *OF51-014*

In questo modo, tutte le informazioni relative al processo MODA-ML vengono conservate facilitando la manipolazione dei BPSS (sia essa manuale o elettronica).

(Non vogliamo dilungarci ulteriormente descrivendo nel dettaglio la struttura XML di un BPSS MODA-ML poiché non differisce in maniera sostanziale con quella descritta per il BPSS ebXML nel capitolo 3. Nelle Appendici A e B sono riportati, come esempio esplicativo, i listati di due istanze BPSS relative rispettivamente ai processi MODA-ML “Produzione Tessuto” e “Fornitura Tessuti”, per la natura stessa del linguaggio XML, sono facilmente comprensibili e si possano individuare immediatamente i concetti descritti in questo paragrafo).

Per ogni Business Process MODA-ML è possibile creare due BPSS MODA-ML, uno per ogni linguaggio supportato (italiano e inglese), aventi il medesimo identificatore. Questa scelta progettuale è stata fatta per permettere a due aziende che utilizzano linguaggi diversi di riferirsi al medesimo Business Process descritto nei BPSS.

(Nella bibliografia sono riportati i riferimenti [ebBPSSxml], [ebBPSSdtd], [ebBPSSxsd] agli esempi proposti da ebxml.org e che sono stati di supporto nell’applicazione dello standard ebXML BPSS).

4.3.2 MODA-ML CPP e CPA

Il MODA-ML CPP è stato realizzato tenendo ben presente la sua funzionalità ultima, ovvero il raggiungimento di un accordo commerciale tra due Party che vede la sua concretizzazione nella stipulazione del CPA.

La struttura XML degli standard ebXML CPP e CPA è la medesima; lo XML-Schema per validarne le istanze è lo stesso. Tratteremo l'argomento dei Protocolli di Collaborazione facendo riferimento al solo CPP ma dobbiamo tener bene presente che gli adattamenti e le estensioni effettuate sul CPP si ripercuotono anche sul CPA.

La specifica MODA-ML per il CPP è inscindibile da quella riguardante il CPA, così come è per ebXML.

Il CPP, nella versione elaborata per MODA-ML, si basa in maniera consistente su quello definito dalle specifiche ebXML ma per i nostri scopi ci si è dedicati maggiormente alla parte relativa alla logica dei Business Process. La parte relativa al trasporto e alla sicurezza, non essendo tra le priorità del progetto, è stata trattata in maniera minimale al fine unico di rendere il CPP MODA-ML un CPP conforme a ebXML.

La creazione della specifica MODA-ML CPP si è raggiunta in due fasi:

- applicazione dello standard ebXML al caso MODA-ML
- estensione dello standard ebXML CPP e CPA

4.3.2.1 Adattamento dello standard ebXML CPP a MODA-ML

L'adattamento dello standard ebXML al caso MODA-ML ha presentato una serie di scelte da effettuare quali la rinuncia a molti elementi e attributi opzionali che, per i nostri fini, non erano indispensabili e avrebbero portato solamente ad un appesantimento inutile del documento CPP finale poiché non effettivamente utilizzati.

Oltre a decidere quali elementi e attributi utilizzare per delineare la specifica MODA-ML CPP, si sono dovute compiere altre scelte, sull'uso stesso di alcuni elementi, che riassumiamo di seguito:

- il CPP ebXML prevede l'utilizzo di più elementi *PartyInfo* per descrivere più sottosezioni della stessa azienda, il CPP MODA-ML si limita a descrivere un solo *PartyInfo* relativo all'azienda nella sua totalità (scelta permessa dalle specifiche ebXML). Questa scelta è stata compiuta perché le aziende partecipanti al progetto MODA-ML sono di natura medio piccola, si è dunque preferito un CPP più semplice e comprensibile;
- in un CPP ebXML si possono specificare più Business Process, un solo Business Process invece può essere specificato in un CPP MODA-ML. Lo scopo di CP-NET è di agevolare la creazione di un Profilo di Collaborazione relativo a un solo Business Process. Allo stato dell'arte, infatti, il progetto MODA-ML si basa su collaborazioni binarie prendendo un Business Process alla volta;
- nel CPP ebXML l'identificatore del CPP viene delineato essere un URN registrato, in MODA-ML, per semplicità, l'identificatore di CPP è stato specificato come la composizione della sigla del proprio paese (IT, EN) più il numero di partita iva dell'azienda;
- nel CPP ebXML viene dato largo spazio alla specificazione dei parametri relativi a Trasporto, alla Sicurezza alla gestione della busta ebXML. Nel CPP MODA-ML, momentaneamente, i soli parametri che si possono modificare sono quelli relativi al tipo di protocollo di Trasporto ed endpoint relativi per ogni singolo documento di business che si è intenzionati a spedire o ricevere.

4.3.2.2 Il nuovo elemento aggiunto al CPP

I documenti di scambio realizzati da MODA-ML tramite il Dizionario (par. 2.4.4) constano di **numerosi elementi opzionali**. Questo numero così elevato di entità opzionali se da un lato porta alcuni vantaggi, dall'altro invece comporta alcuni problemi. Il vantaggio principale è quello di permettere una molteplicità di varianti e soluzioni sicuramente utili per uno standard che vuole essere utilizzato dalle aziende più disparate del settore. Lo svantaggio che si è riscontrato è invece la presenza di situazioni ambigue, in cui entrambi i party descrivono la medesima cosa in modi

diversi; inoltre l'insieme, molte volte consistente, degli elementi non utilizzati, appesantisce i documenti di scambio e le procedure di integrazione.

Come prima soluzione si era pensato di apportare, in maniera automatica, delle modifiche agli XML Schema dei tipi di documento di scambio, dipendentemente dalle scelte espresse dagli utenti, per poi farvi riferimento nei CPP. Questo approccio è stato scartato quasi immediatamente: ogni nuova Business Collaboration avrebbe introdotto tanti nuovi XML Schema quanti i tipi di documento utilizzati in tale Business Collaboration. L'architettura MODA-ML è pensata come non centralizzata (ovvero l'utilizzo di un repository non è essenziale per instaurare una collaborazione tra due aziende), di conseguenza la creazione in un numero così consistente di nuovi XML Schema da immagazzinare nei sistemi gestionali delle aziende, avrebbe portato solamente confusione e situazioni di ambiguità.

Con l'adozione dei Protocolli di Collaborazione si vuole quindi dare l'opportunità a ogni azienda di poter non solo descrivere quali documenti di quali processi intende supportare, ma anche specificare, tramite la scelta degli elementi opzionali, quali parti dei documenti utilizzare e quali invece trascurare. Lo standard CPP ebXML, per come è definito, non ha questa capacità; per questo motivo la struttura dei Profili di Collaborazione ebXML è stata estesa.

La soluzione adottata consiste nell'estendere il CPP con un elemento apposito per immagazzinare le informazioni relative agli elementi opzionali dei tipi di documento. Nel CPA viene immagazzinato invece l'accordo raggiunto su tali informazioni. Il CPA, successivamente, viene utilizzato dal sistema gestionale per creare i documenti di scambio secondo le preferenze espresse. In questo modo, tramite i Protocolli di Collaborazione, le aziende sanno esattamente cosa aspettarsi nella ricezione dei documenti.

Vediamo in cosa consiste l'elemento aggiunto alla struttura ebXML di CPP e CPA.

L'elemento aggiuntivo al CPP, chiamato *DocumentOptionalElements*, tiene traccia delle personalizzazioni effettuate sui documenti scelti per il Business Process di riferimento.

All'interno dell'elemento *DocumentOptionalElements* saranno immagazzinate tutte le informazioni necessarie. Nessuna altra estensione è stata introdotta nello standard di CPP e CPA.

Per entrambi i protocolli l'elemento creato per l'estensione è stato inserito come ultimo nella sequenza dei figli della radice. Di seguito la definizione di *DocumentOptionalElements* aggiunta nello XML Schema:

```
<element ref="tns:DocumentOptionalElements" minOccurs="0" maxOccurs="1"/>
```

L'elemento *DocumentOptionalElements* può comparire una sola volta, o non comparire affatto, sia nel CPP che nel CPA. Nel CPP esso è relativo alle personalizzazioni di un solo Party. Nel CPA è l'espressione dell'accordo raggiunto tra i due Party relativamente agli elementi opzionali (se ne parlerà più dettagliatamente alla fine di questo paragrafo). La presenza di questo elemento nel CPP ne implica la presenza nel CPA conseguente.

Durante la definizione della struttura gerarchica dell'elemento aggiuntivo e dei sottoelementi che ne fanno parte, si è dovuta considerare la necessità di immagazzinare le seguenti informazioni:

- quali documenti di business sto considerando;
- quali sono le entità opzionali, per ogni documento, in cui il Party ha espresso la sua preferenza (Optional, Required o Rejected).

Oltre a queste informazioni, altre, di carattere più tecnico e relative alla gestione dei Profili da parte del software gestionale, sono state aggiunte:

- a quale Business Process sto facendo riferimento (identificatore);
- come posso rintracciare l'elemento opzionale all'interno dello XML-Schema relativo al documento che lo contiene, tramite XPath [XPAT99].

Descriverò ora la struttura XML dell'elemento *DocumentOptionalElements*, e quella degli elementi inclusi in esso soffermandomi ad evidenziare le motivazioni che mi hanno spinto a crearlo in questa maniera.

Esaminiamo il frammento aggiunto allo XML Schema ebXML e un esempio, frammento di un'istanza di CPP creato con il CPP-Editor.

Frammento XML-Schema per la definizione dell'elemento

DocumentOptionalElements:

```
<element name="DocumentOptionalElements">
  <complexType>
    <sequence>
      <element ref="tns:Doc" maxOccurs="unbounded" minOccurs="1"/>
    </sequence>
    <attribute name="bpssuuid" type="tns:non-empty-string" use="required"/>
  </complexType>
</element>
<element name="Doc">
  <complexType>
    <sequence>
      <element ref="tns:Entity" minOccurs="0" maxOccurs="unbounded"/>
    </sequence>
    <attribute name="name" type="tns:non-empty-string" use="required"/>
    <attribute name="position" type="tns:non-empty-string" use="required"/>
  </complexType>
</element>
<element name="Entity">
  <complexType>
    <attribute name="name" type="tns:non-empty-string" use="required"/>
    <attribute name="count" type="tns:non-empty-string" use="required"/>
    <attribute name="state" type="tns:preference" use="required"/>
    <attribute name="xpath" type="tns:non-empty-string" use="required"/>
  </complexType>
</element>
<simpleType name="preference">
  <restriction base="tns:non-empty-string">
    <enumeration value="Required"/>
    <enumeration value="Rejected"/>
  </restriction>
</simpleType>
```

Come si può facilmente intuire, l'elemento *DocumentOptionalElements* contiene altri elementi. *DocumentOptionalElements* contiene *Doc* che a sua volta contiene *Entity*. Si è scelta questa struttura per meglio organizzare le entità opzionali dei diversi documenti facenti parte del Business Process scelto.

Il concetto appena esposto diviene ancora più evidente se prendiamo in considerazione un'istanza di CPP, descrivente un'ipotetica situazione reale.

Frammento di un'istanza di CPP, creata con il CPP-Editor, contenente l'elemento *DocumentOptionalElements*:

```
<tp:DocumentOptionalElements tp:bpssuuid="v2003-1_Proc-1">
  <tp:Doc tp:name="Ordine al fornitore di tessuti" tp:position="3">
    <tp:Entity
      tp:name="msgfunction"
      tp:count="0"
      tp:state="Required"
      tp:xpath="TEXOrder/@msgfunction" />
    <tp:Entity
      tp:name="logo"
      tp:count="1"
      tp:state="Rejected"
      tp:xpath="TEXOrder/TOheader/supplier/@logo" />
  </tp:Doc>
  <tp:Doc tp:name="Risposta ordine del fornitore di tessuti" tp:position="4">
    <tp:Entity
      tp:name="msgfunction"
      tp:count="0"
      tp:state="Required"
      tp:xpath="TEXOrdResponse/@msgfunction" />
  </tp:Doc>
</tp:DocumentOptionalElements>
```

Considerando entrambi i frammenti descriverò di seguito ogni elemento ed attributo relativo.

- Elemento *DocumentOptionalElements*

E' l'elemento che racchiude l'estensione apportata all'ebXML CPP atta a immagazzinare le preferenze del Party relativamente agli elementi opzionali dei documenti di scambio disponibili; racchiude elementi di tipo *Doc*, uno per ogni documento di scambio scelto.

DocumentOptionalElements ha un attributo obbligatorio:

bpssuuid: identificatore del Business Process Specification Schema descrivente il Business Process di cui i documenti di business fanno parte.

- Elemento *Doc*

Racchiude le informazioni sugli elementi opzionali di un singolo documento. Può non essere presente se il Party non ha espresso alcuna preferenza su gli elementi opzionali

dei documenti, di conseguenza l'elemento *DocumentOptionalElements* sarà vuoto e non comparirà.

Per ogni elemento opzionale in cui il Party ha espresso una preferenza, *Doc* avrà un corrispondente figlio *Entity*.

Due attributi obbligatori sono stati definiti per *Doc*:

name: il nome del documento a cui fa riferimento;

position: un intero indicante la "posizione" del documento nel Business Process. Ogni Business Process è composto da un certo numero di documenti di scambio in un ordine prestabilito. Questa informazione è utile sia al CPP-Editor per la manipolazione di questo tipo di elementi sia, si presume, al futuro software gestionale che dovrà interfacciare queste informazioni con i documenti di scambio veri e propri.

- Elemento *Entity*

Questo elemento si riferisce a un elemento opzionale su cui il Party ha espresso una preferenza (valori "Required" o "Rejected").

Nel caso in cui un elemento opzionale rimanga tale (valore "Optional"), il relativo elemento *Entity* non comparirà al fine di limitare al minimo indispensabile la quantità di informazioni necessarie e non "stravolgere" la natura originale del CPP.

L'elemento *Entity* ha quattro attributi obbligatori:

name: nome dell'elemento opzionale;

count: determina univocamente l'entità opzionale nella sequenza delle entità opzionali di quel particolare documento. Questo attributo risulta essere utile nella manipolazione da parte del CPP-Editor e si presume possa essere utile al SW gestionale che dovrà utilizzare il CPP per la configurazione semi-automatica;

state: preferenza espressa relativamente all'elemento opzionale. Può assumere solamente due valori: "Required" (elemento obbligatorio) o "Rejected" (elemento rifiutato);

xpath: xpath dell'elemento opzionale per poterlo individuare nello XML-Schema del documento di cui fa parte.

Riconsideriamo il frammento di istanza di CPP, mostrato precedentemente in questo paragrafo come esempio, e traiamo alcune considerazioni per meglio comprenderne il funzionamento:

1. grazie all'attributo *bpssuuid* possiamo dedurre che stiamo prendendo in considerazione un determinato Business Process Specification Schema relativo al processo "Fornitura Tessuti";
2. di tale processo solo due sono i documenti disponibili per questo Party, "Ordine al fornitore di tessuti" e "Risposta ordine del fornitore di tessuti". Per ognuno sono state espresse alcune preferenze sugli elementi opzionali:
 - a) in entrambi i documenti l'attributo *msgfunction*, attributo della radice dei documenti, è richiesto;
 - b) l'attributo *logo* nel primo documento è rifiutato;
 - c) per ogni elemento è indicato lo XPath per essere identificato univocamente nella istanza del documento relativo.

Lo XML-Schema ebXML originale [ebCPPAxsd] è stato dunque esteso e giustamente rinominato poiché utilizzato come namespace interno dei CPP e CPA.

Il nuovo XML-Schema ha preso il nome di *modaml_cpp-cpa-2_0b.xsd* [CPNETc].

4.3.2.3 Gestione del nuovo elemento nel CPA

Non è tra gli scopi del progetto di tesi implementare nel progetto la creazione semiautomatica del CPA; tuttavia ritengo sia opportuno soffermarci sulla fase di contrattazione dei CPP MODA-ML, relativamente all'elemento aggiuntivo, per i seguenti motivi:

- l'elemento *DocumentOptionalElements* aggiunto nel MODA-ML CPP è stato pensato per essere presente anche nel CPA;
- il MODA-ML CPP è stato definito considerando quale sarebbe stato il suo successivo utilizzo una volta ultimato, ovvero la creazione di un MODA-ML CPA;

- la creazione del CPA è lo stadio ultimo del processo ebXML riguardante i Protocolli di Collaborazione, per questo motivo soffermarmi nuovamente sul CPA non può che essere di aiuto nella comprensione di questa tesi;
- questa breve spiegazione sarà di supporto per eventuali sviluppi futuri del progetto CP-NET.

Si è discusso riguardo la creazione del CPA da due CPP nel capitolo 3; di seguito riportiamo un semplice esempio che spieghi la fase di matching tra due CPP relativamente al solo elemento *DocumentOptionalElements*:

- un Party A dichiara nel proprio CPP che l'attributo opzionale *msgfunction* interno al documento "Ordine al fornitore di tessuti", è richiesto. Specifica in questo modo che, per poter utilizzare tale documento di business con un'altra azienda, esige l'utilizzo di tale entità;
- un Party B, invece, nel proprio CPP non esprime invece nessuna preferenza riguardo lo stesso attributo opzionale, *msgfunction*. In questo modo l'utilizzo dell'attributo rimane per il Party B opzionale;
- nella fase di stipulazione del CPA si metteranno a confronto le diverse preferenze dei due Party relativamente ai medesimi documenti e si raggiungerà un accordo finale (in caso di scelte incompatibili, queste saranno segnalate e richiederanno un intervento da parte degli utenti);
- l'attributo *msgfunction*, diviene infine obbligatorio per entrambi poiché richiesto dal Party A e indifferente per il Party B.

Di seguito la figura 4.4 illustra la situazione appena descritta.

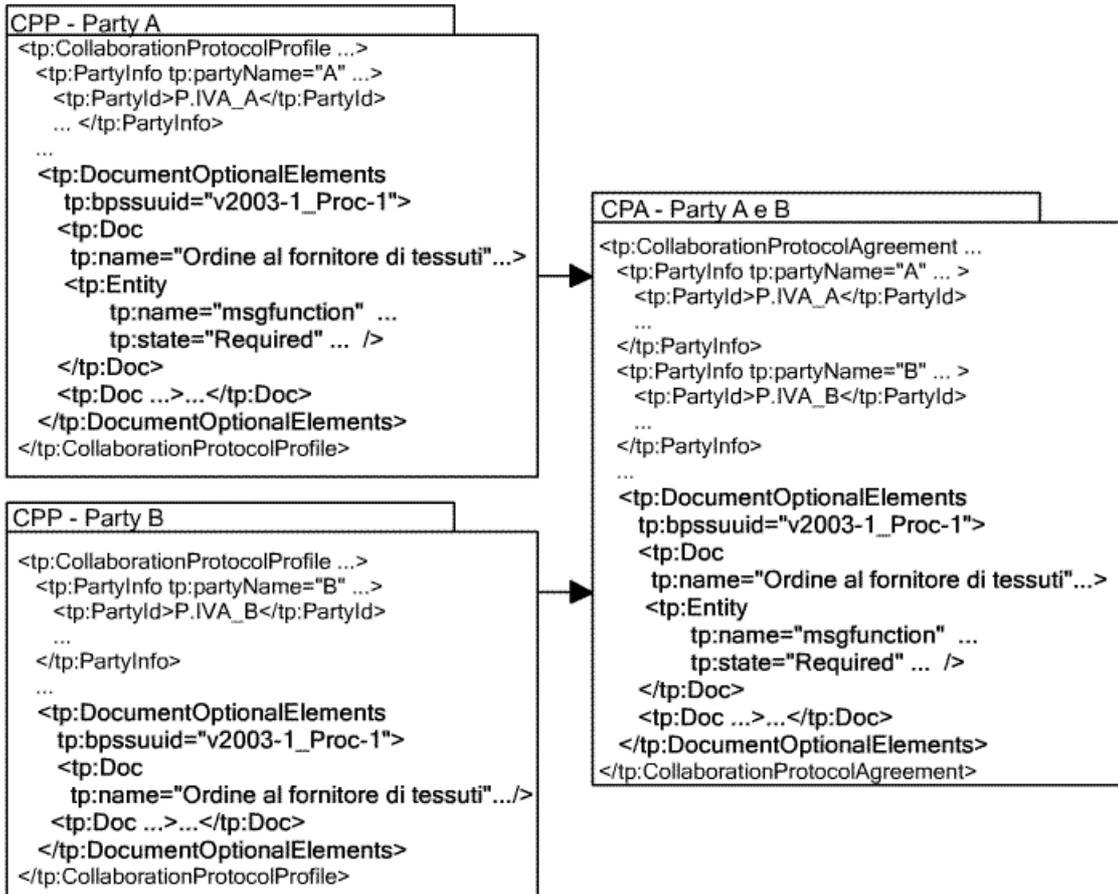


Figura 4.4 Elemento DocumentOptionalElements nel CPP e CPA

(Nella bibliografia sono riportati i riferimenti [ebCPPxml], [ebCPAxml], [ebCPPAxsd] agli esempi proposti da oasis-open.org e che sono stati di supporto nell'applicazione dello standard ebXML CPP).

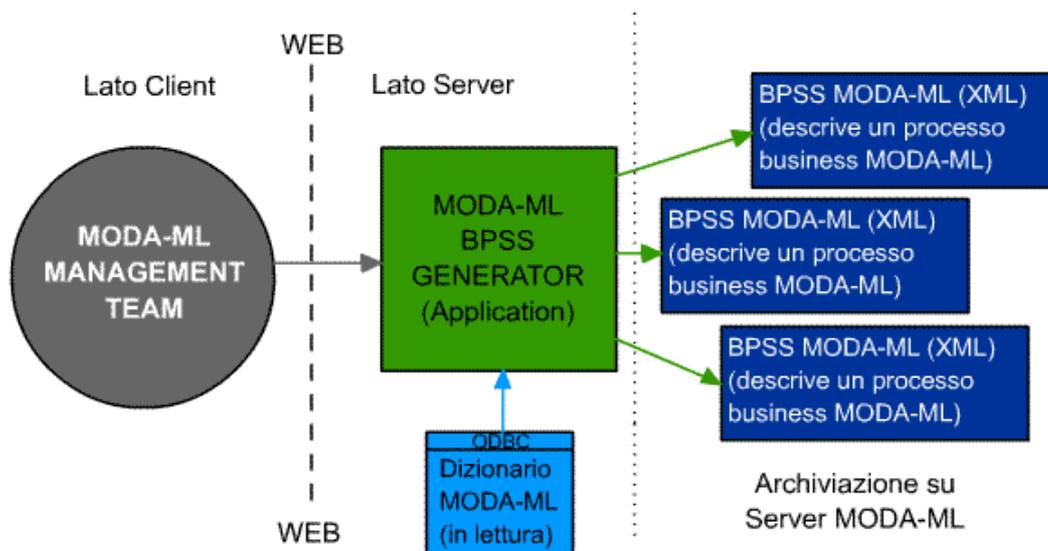
4.4 Progettazione del BPSS-Generator

L'obiettivo che si prefigge il MODA-ML BPSS-Generator è quello di creare in maniera automatica BPSS ebXML (in versione XML) descrittivi Business Process MODA-ML. Le informazioni relative la struttura dei Business Process MODA-ML vengono prelevate tramite interrogazioni al Dizionario. Ogni BPSS generato è automaticamente depositato in una directory del server e dunque riferibile tramite URL dai CPP creati o modificati con il CPP-Editor.

E' stato richiesto che l'applicazione supportasse sia la lingua italiana che quella inglese; come conseguenza il BPSS-Generator è stato progettato per creare BPSS in entrambe le lingue.

Le informazioni relative alle strutture di Business Process e documenti di scambio sono già state inserite nel Dizionario dal Management Team MODA-ML. L'applicazione BPSS-Generator richiede solamente la scelta di quale Business Process, tra quelli disponibili nel Dizionario, si voglia prendere in considerazione per generare il BPSS relativo, nessuna altra forma di interattività è prevista.

Nella seguente figura è illustrato il funzionamento del BPSS-Generator.



4.5 Funzionamento del MODA-ML BPSS-Generator

Nel successivo capitolo (cap. 5) descriverò la fase di implementazione, le funzioni e l'interfaccia del BPSS-Generator.

4.5 Progettazione del CPP-Editor

4.5.1 Obiettivo del CPP-Editor

La motivazione principale per cui si è intrapreso lo sviluppo di un CPP-Editor da integrare nel progetto MODA-ML è l'opportunità di poter fornire ad ogni azienda che utilizzi lo standard MODA-ML un supporto per descrivere le proprie potenzialità in un "Profilo di Collaborazione" secondo un formato standard. Successivamente questo documento potrà essere utilizzato nella definizione (manuale o semiautomatica) di un "Contratto di Collaborazione" (CPA) tra due aziende.

Il motivo principale per cui si è scelto di sviluppare il CPP-Editor come un'applicazione web è da attribuire alla presenza sul server MODA-ML del Dizionario e dei BPSS MODA-ML, sempre aggiornati. Poiché il CPP contiene molte informazioni relative ai Business Process e ai documenti di scambio, la soluzione web permette a MODA-ML di aiutare le aziende in maniera più diretta con il risultato di evitare situazioni di incoerenza tra CPP. Inoltre, la soluzione web, permette di ottenere un'applicazione facilmente accessibile e che non richiede fasi di installazione, configurazione, manutenzione ed aggiornamenti da parte dell'utente.

4.5.2 Creazione o Modifica

Poiché il CPP-Editor deve permettere sia di creare un CPP MODA-ML partendo da zero che consentirne anche le successive modifiche, l'applicazione è stata sviluppata considerando entrambe le situazioni.

Nella fase di modifica, l'utente, carica sul server MODA-ML il proprio CPP, dopodiché attua le modifiche necessarie.

Nella fase di creazione, l'utente non carica nessun CPP, il CPP-Editor carica automaticamente un proprio CPP di Default permettendo all'utente di modificarlo per ottenere un nuovo CPP.

Il CPP di default è un'istanza completa di MODA-ML CPP ed è stato scelto come "template" di costruzione di un nuovo CPP perché è apparsa essere la soluzione migliore per integrare le fasi di creazione e modifica in maniera trasparente per l'utente (in Appendice C il listato del CPP di default in versione italiana).

4.5.3 Funzionamento del CPP-Editor

Dopo aver caricato il CPP (sia esso quello dell'utente che quello di default), l'editor guida l'utente nella creazione o modifica del documento:

1. permettendogli tramite alcuni form di inviare le informazioni per personalizzare il CPP
2. prelevando e controllando le informazioni relative alla struttura dei Business Process direttamente dal BPSS corrispondente al processo scelto e le informazioni relative alla struttura dei documenti direttamente dal Dizionario MODA-ML; una volta completato il CPP, esso viene inviato tramite e-mail all'utente e cancellato dal server (figura 4.6).

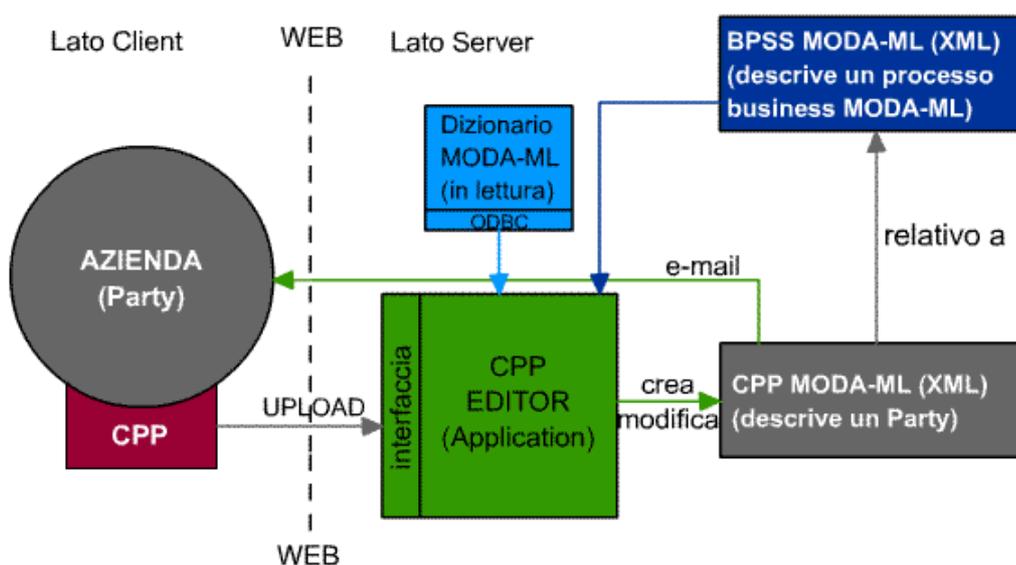


Figura 4.6 Funzionamento del MODA-ML CPP-Editor

L'interfaccia utente progettata consiste in cinque fasi consecutive:

1. Upload del CPP (o caricamento del CPP di default)
2. Scelta del Business Process a cui fare riferimento nel CPP
3. Scelta dei documenti di scambio relativi al Business Process scelto
4. Specificazione delle preferenze relative agli elementi e attributi opzionali contenuti nei documenti di scambio scelti
5. Riepilogo e conferma. Invio del CPP all'azienda

La prima fase, coincidente con la pagina iniziale dell'applicazione, è dedicata alla scelta del CPP da caricare sul server per la modifica o, in mancanza di esso, per la creazione di un nuovo CPP; le quattro fasi successive consistono invece nel processo di editing vero e proprio.

Ogni fase è rappresentata da un'interfaccia disegnata dinamicamente in base alle scelte effettuate nel passo precedente (p.es. non si potrebbero presentare i documenti relativi a un Business Process senza aver specificato quest'ultimo precedentemente).

Nella figura seguente è riassunta la situazione appena descritta.

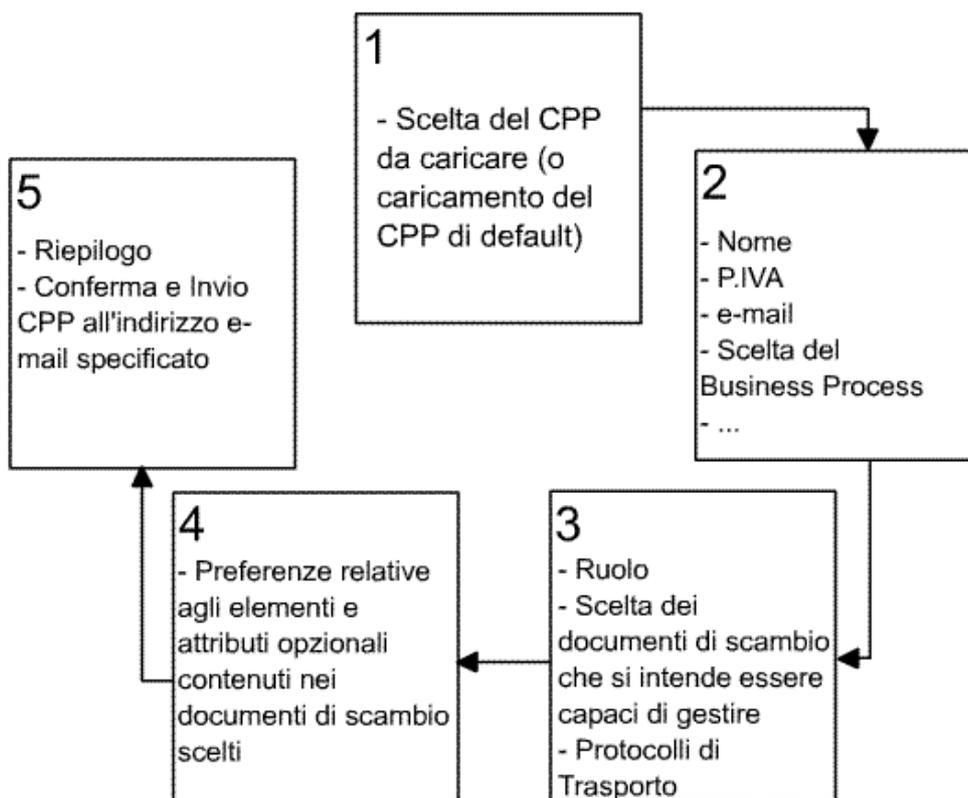


Figura 4.7 Le cinque fasi del CPP-Editor

4.5.4 Preferenze sulle entità opzionali dei documenti

Ritengo sia necessario soffermarmi su una fase di editing particolare che ha richiesto uno sforzo sicuramente maggiore rispetto alle altre per essere sviluppata: la fase di immissione delle preferenze riguardanti le entità opzionali dei documenti di scambio (la fase numero 4 nella figura 4.7).

Questa fase del CPP-Editor è necessaria per creare o modificare le informazioni che verranno immagazzinate nell'elemento aggiuntivo del CPP: *DocumentOptionalElements* (par. 4.3.2.2).

Si è affrontato dunque il problema di presentare all'utente una grande mole di informazioni poiché, lo ricordiamo, ogni Business Process può essere formato da molti documenti e ogni documento di scambio può contenere molti elementi e attributi opzionali.

La soluzione che è sembrata apparire più funzionale e “usabile”, da parte dell’utente, può essere descritta in 3 punti fondamentali:

1. Viene presentato un menù le cui voci corrispondono ai documenti scelti dall’utente nel passo precedente (la fase numero 3 nella figura 4.7).
2. Selezionando da questo menù una delle voci, vengono prelevate dal Dizionario le informazioni relative alla struttura del tipo di documento scelto e successivamente visualizzate nell’interfaccia. Si è cercato di fornire un’organizzazione visiva delle informazioni con una rappresentazione simile a quella ad albero utilizzata nelle guide MODA-ML [MMLG], di cui mostriamo un frammento da esempio:

TEXOrder

@TOtype [Required]

@msgfunction [Optional] [Default= OR]

TOheader 1-1

msgN 1-1

msgDate 1-1

msgCurrency 0-1

otherCurrency 0-9

@currencyUseQualifier [Required]

refDoc 0-9

@docType [Required]

docID 1-1

docDate 0-1

season 0-1

itemID 0-1

buyer 1-1

@logo [Optional]

...

Le entità opzionali, siano esse elementi o attributi, sono le uniche a poter essere modificate specificando uno dei tre valori ammessi (Optional, Required o Rejected). Eventuali preferenze espresse in precedenza vengono visualizzate.

3. Una volta espresse le preferenze nel documento selezionato, esse vengono immagazzinate nell’elemento aggiuntivo del CPP: *DocumentOptionalElements*.

4. Si ripete il procedimento dal punto 1. Una volta concluso di specificare le proprie scelte relative alle entità opzionali dei documenti, si può passare alla fase successiva (la numero 5 nella figura 4.7).

Se si fosse permesso di visualizzare tutte le rappresentazioni ad albero di tutti i documenti in una sola pagina si sarebbero avuti i seguenti svantaggi:

- carico di lavoro per il server eccessivo nella fase di creazione dell'interfaccia per tutti i documenti in una sola operazione. La generazione della rappresentazione ad albero di un solo tipo di documento comporta il reperimento dal Dizionario di tutte le informazioni relative alla struttura XML del documento stesso e la successiva elaborazione per la visualizzazione nell'interfaccia. Questo carico di lavoro è già molto consistente, anche se accettabile, per un solo documento;
- conseguente tempo di attesa troppo elevato per l'utente;
- presentazione di un'interfaccia poco usabile:
 - La quantità eccessiva di informazioni presentate in una sola volta all'utente, anche se ben organizzate, avrebbero avuto come conseguenza una difficoltà maggiore per esprimere le proprie preferenze. Ricercare l'entità opzionale da rendere obbligatoria tra tutte quelle presenti in tutti i tipi di documenti scelti sarebbe stato troppo oneroso da parte dell'utente che, come caso estremo ma non improbabile, potrebbe voler modificare la preferenza ad una sola entità opzionale;
 - Rischio di perdita di tutte le informazioni immesse per inattività da parte dell'utente (scadenza delle variabili dell'oggetto session).
- carico di lavoro per il server eccessivo nella fase di inserimento delle preferenze espresse su tutte le entità opzionali di tutti i documenti scelti dall'utente nel CPP.

Nel prossimo capitolo descriveremo dettagliatamente le cinque fasi presentate dall'interfaccia del CPP-Editor, descrivendo l'approccio e le scelte effettuate, le funzioni fondamentali e l'interfaccia.

5 IMPLEMENTAZIONE

L'architettura CP-NET si è concretizzata nello sviluppo di due applicazioni web di supporto a MODA-ML:

- **BPSS-Generator**: permette al team management di MODA-ML di attivare la creazione di Business Process Specification Schema. I BPSS sono generati prelevando le informazioni necessarie dal Dizionario MODA-ML (par. 2.4) e organizzate secondo la struttura MODA-ML BPSS (par. 3.2.2 e par. 4.3.1);

- **CPP-Editor**: permette ad ogni azienda che adotta lo standard MODA-ML di poter creare o modificare un proprio Collaboration Protocol Profile. Un CPP descrive le capacità di un'azienda relative ad un determinato Business Process e permette di specificare le proprie preferenze relative alla Business Collaboration che si desidera instaurare. I CPP creati e modificati con il CPP-Editor sono organizzati secondo le specifiche MODA-ML CPP (par. 3.3.3 e par. 4.3.2).

Nel precedente capitolo si è descritta la fase di progettazione delle due applicazioni web, BPSS-Generator (par. 4.4) e CPP-Editor (par. 4.5); scopo di questo capitolo è invece descriverne l'implementazione a un più basso livello.

Descriverò il funzionamento di ognuna delle applicazioni, tramite:

- la presentazione dell'interfaccia
- la definizione delle azioni e funzioni principali implementate

Le applicazioni consistono in pagine ASP (Active Server Pages) scritte in linguaggio Visual Basic Script. La piattaforma di sviluppo e testing è il sistema operativo Windows con tecnologia NT (Windows 2000, Windows XP) integrato con il Web Server IIS (Internet Information Server). Il Dizionario MODA-ML, che, ricordiamo, contiene le informazioni sulla struttura dei Processi e dei documenti di scambio, è realizzato con MS Access e ad esso ci si interfaccia tramite driver ODBC e query SQL [SQL92]. Gli inserimenti, le modifiche e le cancellazioni d'informazioni effettuate nel

CPP sono state implementate ricorrendo all'uso del Document Object Model [DOM98].

5.1 Organizzazione di Files e Directory

Le due applicazioni, destinate a funzionare sul server MODA-ML, sono state sviluppate assegnando loro una struttura gerarchica delle directory e dei files ben definita. Per ognuna è stata definita una directory principale di partenza.

I files che compongono il BPSS-Generator sono raggruppati e organizzati nelle seguenti sottodirectory della directory principale *bpss*:

- *bpss/dtdxslt/* : contenente il DTD e il foglio di stile XSLT relativi ai BPSS creati;
- *bpss/images/* : contenente le immagini utilizzate nell'interfaccia web;
- *bpss/generator/* : contenente tutti gli script ASP che compongono l'applicazione vera e propria;
- *bpss/v[VersioneDizionario]/[lingua]/* : nel processo di generazione dei BPSS vengono create ulteriori directory per suddividere i risultati a seconda della versione del Dizionario di origine. Per ogni versione del Dizionario utilizzata viene dunque creata nella directory principale *bpss* una nuova sottodirectory che prende il nome della versione. All'interno di ogni directory corrispondente a una versione del Dizionario vengono a loro volta create due directory, *it* e *en*, per suddividere i BPSS a seconda della lingua scelta. (Un BPSS generato in lingua italiana, dalla versione 2003-1 del Dizionario, verrà depositato nella directory *bpss/v2003-1/it/*).

Il codice dell'applicazione (interno alla directory *generator*) è stato organizzato in quattro livelli distinti ma interconnessi:

- Applicazione: i files sono *default.asp*, *bpssgenerator.asp* e *funzioni.inc.asp*, quest'ultimo racchiude tutte le procedure e funzioni sviluppate e utilizzate dall'applicazione;
- Configurazione: *costanti.inc.asp* è il file di configurazione nel quale apportare le uniche modifiche necessarie nel processo di installazione dell'applicazione;

- Gestione degli errori: *error.asp* e *error.inc.asp*;
- Query SQL: *sql1.inc.asp* e *sql2.inc.asp* definiscono tutte le query SQL utilizzate per interrogare il Dizionario MODA-ML. In questo modo, a eventuali modifiche sulla banca dati corrisponderà un facile aggiornamento delle query SQL.

I files che compongono il CPP-Editor sono raggruppati e organizzati, in modo simile al BPSS-Generator, nelle seguenti sottodirectory della directory principale *cpp*:

- *cpp/dtdxslt/* : contenente lo XML Schema e il foglio di stile XSLT relativi ai BPSS creati;
- *cpp/images/* : contenente le immagini utilizzate nell'interfaccia web;
- *cpp/editor/* : contenente tutti gli script ASP che compongono l'applicazione vera e propria.

Sono state inoltre create le seguenti due sottodirectory della directory *editor*:

- *cpp/editor/temp*: questa directory viene utilizzata per creare i CPP temporanei sui quali apportare le modifiche dell'utente durante la fase di creazione o modifica del CPP. Al termine della procedura guidata il file temporaneo viene cancellato;
- *cpp/editor/genericuser/*: contenente i CPP di default (uno in lingua italiana e uno in lingua inglese).

I files che compongono l'applicazione CPP-Editor (interni alla directory *editor*) sono stati sviluppati e organizzati in modo analogo a quello del BPSS-Generator:

1. Applicazione: Il codice è stato suddiviso in base alle cinque fasi che compongono il funzionamento del CPP-Editor:

- 1a. *default.asp* e *funzioni0.inc.asp* definiscono la pagina iniziale del CPP-Editor;
- 1b. *cppeditor_step1.asp* e *funzioni1.inc.asp* definiscono rispettivamente l'interfaccia del primo passo di editing e le procedure e funzioni utilizzate;
- 1c. *cppeditor_step2.asp* e *funzioni2.inc.asp* per il secondo passo di editing più il file *FunUpload.inc.asp* dedicato alla sola gestione dell'upload del CPP;
- 1d. *cppeditor_step3.asp* e *funzioni3.inc.asp* per il terzo passo di editing. In questo caso si è preferito raggruppare tutte le funzioni e procedure relative alla sola

generazione della rappresentazione ad albero dei tipi di documento in un unico file:

funzioni3_MakeTree.inc.asp;

1e. *cppeditor_step4.asp* e *mmmail.inc.asp* per gestire il quarto e ultimo passo di editing e definire la funzione per l'invio della e-mail all'azienda con allegato il CPP completato;

3. Presentazione: Il file *menu.inc.asp* definisce un menù dinamico;

4. Configurazione: *costanti.inc.asp* è il file di configurazione (analogo a quello del BPSS-Generator);

5. Gestione degli errori: *error.asp* e *error.inc.asp*;

6. Query SQL: *sqlcpp.inc.asp* definisce tutte le query SQL utilizzate per interrogare il Dizionario MODA-ML.

5.2 Implementazione di MODA-ML BPSS-Generator

Il BPSS-Generator permette di generare un BPSS di un Business Process MODA-ML.

L'applicazione è stata quindi sviluppata per permettere allo staff di MODA-ML di:

1. scegliere il Business Process da descrivere attraverso il relativo BPSS
2. visualizzare il BPSS generato tramite un foglio di stile XSLT

Descriverò di seguito come queste due azioni sono state implementate.

Nella prima pagina del MODA-ML BPSS-Generator (figura 5.1), tramite un'interfaccia web si ha la possibilità di scegliere:

- quale lingua utilizzare tra quelle supportate (italiano o inglese)
- quale Business Process si intende prendere in considerazione, tra quelli disponibili nel Dizionario, per attivare il meccanismo di generazione del relativo BPSS MODA-ML.

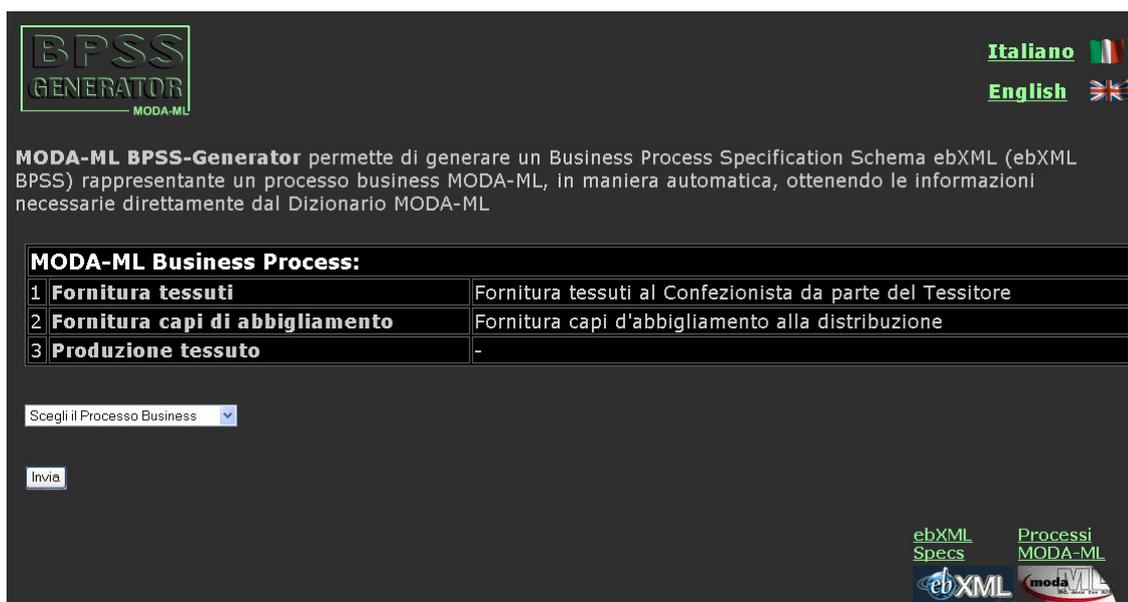


Figura 5.1 MODA-ML BPSS-Generator, scelta del Business Process

In questa prima pagina viene creata una tabella dei Business Process disponibili con relativa descrizione; da un menù a cascata è possibile effettuare la scelta del Business Process. La lista dei Business Process disponibili viene ricavata tramite interrogazione del Dizionario sulla tabella “Processi”, e inserita in un form HTML che si preoccupa di inviare l’informazione di scelta alla successiva pagina.

La seconda pagina ASP, che è il vero cuore del generatore, esegue tre operazioni fondamentali:

- estrae dal Dizionario le informazioni relative alla struttura del Processo;
- elabora le informazioni;
- genera la directory destinazione, se non presente;
- genera il BPSS finale, relativo al processo scelto.

L’accesso al Dizionario va a interrogare solamente le tabelle concernenti la definizione della struttura del Business Process scelto (par. 2.4.3). Ricavate le informazioni concernenti la struttura del Business Process MODA-ML, tali informazioni sono

riorganizzate secondo la semantica ebXML per generare il BPSS finale (si è parlato della mappatura tra MODA-ML e ebXML nel par. 4.3.1).

Per eseguire questa riorganizzazione sono state definite delle funzioni apposite, ognuna delle quali genera gli elementi XML relativi a *BusinessDocument*, *BusinessTransaction* e *BinaryCollaboration* (nelle Appendici A e B sono riportate due istanze complete di BPSS MODA-ML).

Una volta generato il BPSS relativo al Business Process scelto, esso viene visualizzato tramite un foglio di stile XSLT che presenta, tramite una tabella HTML, la struttura del processo MODA-ML preso in considerazione (si veda la figura 5.2).

BPSS GENERATOR
MODA-ML

Versione Dizionario MODA-ML: v2003-1
Directory Destinazione: F:\inetpub\wwwroot\moda-ml\repository\bpss\v2003-1\it\

XML Document View

MODA-ML Process: Fornituratessuti [v2003-1_Proc-1]

MODA-ML Activity: Scelta tessuti [Att-7]

Fornitore di tessuti	-->	Scheda Anagrafica Tessuto [OF51-009]	-->	Confezionista
	-->	Listino tessuti [OF51-010]	-->	
	<--	Previsione Collezione Tessuti [OF51-011]	<--	

MODA-ML Activity: Acquisto tessuti [Att-8]

Confezionista	-->	Ordine al fornitore di tessuti [OF51-001]	-->	Fornitore di tessuti
	<--	Risposta ordine del fornitore di tessuti [OF51-002]	<--	
	<--	Avanzamento ordine del fornitore di tessuti [OF51-003]	<--	
	-->	Modifica ordine al fornitore di tessuti [OF51-004]	-->	

MODA-ML Activity: Spedizione tessuti [Att-6]

Confezionista	-->	Richiesta spedizione tessuto [OF51-005]	-->	Fornitore di tessuti
	<--	Avviso spedizione del tessuto [OF51-006]	<--	
	<--	Report Qualità pezza [OF51-007]	<--	
	<--	Fattura Tessitore [OF51-012]	<--	

Figura 5.2 Visualizzazione XSLT del BPSS MODA-ML

La visualizzazione XSLT permette di rappresentare in maniera immediata i due ruoli che partecipano al processo, la suddivisione in attività, e i vari documenti che vengono scambiati, con indicato il verso di scambio tra un partner commerciale e l'altro.

Il BPSS creato, relativo alla versione del Dizionario MODA-ML desiderata, viene depositato in una cartella del server; successivamente ad esso potranno fare riferimento tutti i CPP relativi al Business Process descritto (tramite l'elemento *ProcessSpecification*, si veda par. 3.3). La presenza dei BPSS MODA-ML sul server è, infatti, indispensabile per l'utilizzo del CPP-Editor.

La scelta da parte dell'utente della lingua italiana o inglese comporta, da parte dell'applicazione:

- la generazione di un'interfaccia nel linguaggio desiderato
- l'adozione delle query SQL corrispondenti alla lingua scelta
- la generazione di un BPSS nel linguaggio prescelto (possono esistere due BPSS, descrittivi lo stesso Business Process MODA-ML, in due diversi linguaggi; gli identificatori di BPSS e dei suoi elementi rimangono gli stessi per entrambi).

5.3 Implementazione di MODA-ML CPP-Editor

Il CPP-Editor, a differenza del BPSS-Generator, è un'applicazione interattiva il cui obiettivo è quello di supportare un'azienda che abbia adottato lo standard MODA-ML durante il processo di creazione o modifica di un CPP. Durante la fase di progettazione (si veda par. 4.5) si è giunti ad una soluzione che prevede una pagina iniziale per caricare il CPP sul server più quattro passi fondamentali di editing dello stesso CPP.

Il funzionamento di ogni passo di editing può essere riassunto come segue:

1. elaborazione dei dati inviati dal passo precedente;
2. inserimento o modifica dei dati ricevuti (ed eventualmente elaborati) nel CPP;
3. costruzione e presentazione dell'interfaccia web per l'inserimento dei dati relativi a questo passo ripartendo dal CPP così ottenuto.

Descriverò l'implementazione del CPP-Editor esaminando ognuna delle cinque fasi implementate in altrettante pagine ASP.

5.3.1 Pagina iniziale del CPP-Editor

Nella prima pagina dell'editor *default.asp* (mostrata nella figura 5.3) viene presentata all'utente la possibilità di compiere l'upload di un proprio CPP per effettuare la modifica oppure di procedere nella creazione di un nuovo Profilo di Collaborazione da zero. (Per effettuare la creazione di un nuovo CPP è sufficiente richiedere l'upload senza selezionare nessun tipo di file).

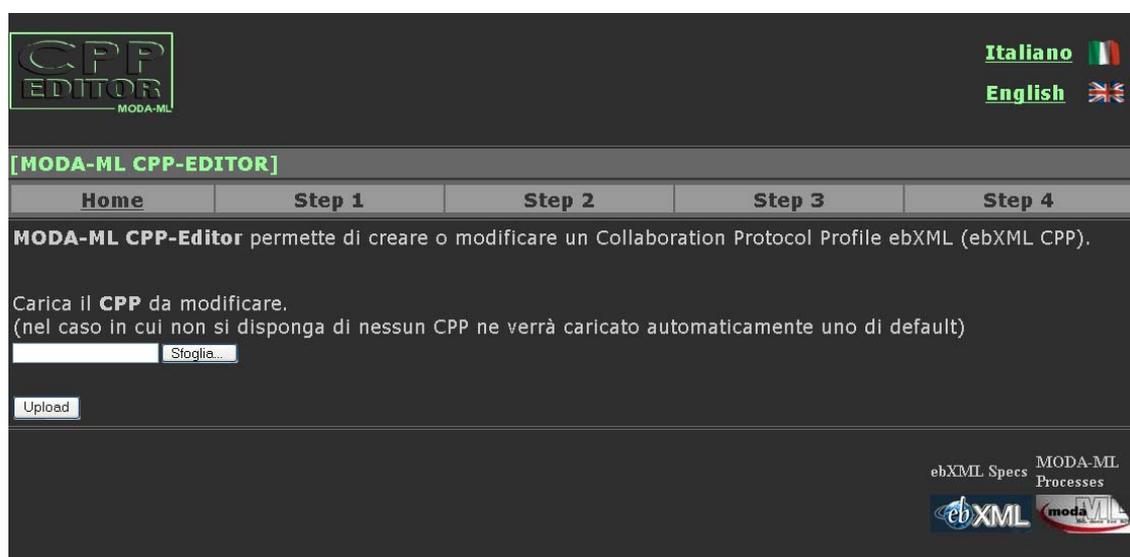


Figura 5.3 Home CPP-Editor

In entrambi i casi, modifica o creazione, il CPP su cui l'applicazione inserirà o aggiornerà i dati sarà un file temporaneo che verrà cancellato una volta ultimato il processo di editing.

In questa prima fase sono eseguite delle scelte fondamentali:

- creazione o modifica di un CPP,
- quale lingua utilizzare tra quelle supportate (italiano o inglese),

scelte che andranno a incidere in maniera consistente nelle successive fasi dell'applicazione.

5.3.2 CPP-Editor Step 1

Due sono i casi fondamentali gestiti in questo primo passo:

- modifica di CPP (l'utente ha caricato un CPP sul server)
- creazione di CPP (l'utente non ha caricato nessun CPP sul server)

Nel primo caso, il CPP-Editor apporterà le successive modifiche direttamente nel CPP caricato. Nel secondo caso sarà caricato, al posto del CPP dell'utente, un CPP di default che verrà utilizzato come template per la creazione del nuovo Profilo (si è parlato del CPP di default nel capitolo di progettazione, par. 4.5.2).

In questo primo passo il CPP prescelto è convalidato per poi ricavarne l'oggetto DOM; vengono subito dopo presentati nell'interfaccia, internamente ad un form HTML, i primi campi che l'utente può modificare o lasciare inalterati (figura 5.4).

CPP EDITOR
MODA-ML

[MODA-ML CPP-EDITOR] STEP 1

Home	Step 1	Step 2	Step 3	Step 4
------	--------	--------	--------	--------

- Identificatore del CPP (la propria partita IVA)
cppid

- Nome del Party (il nome dell'azienda)
PartyName

- URL della pagina informativa del Party (es: <http://www.tessilepiero.it/chisono.html>)
PartyRef

- **BPSS ebXML** (Processo Business)
(Attenzione! Una volta modificato il Processo Business andranno perdute tutte le informazioni sul precedente Processo)

- v2003-1_Fornituratessuti.xml
- v2003-1_Produzionetessuto.xml

Inserire un indirizzo e-mail al quale sarà inviato il CPP una volta completato

Figura 5.4 CPP-Editor Step 1

Nel primo passo di editing vengono richiesti:

- un identificatore del CPP, univoco, corrispondente alla partita IVA del Party preceduta da un identificatore del paese, “IT” o “EN”;
- il nome dell’Azienda a cui il CPP fa riferimento (ovvero il nome del Party);
- un URL a una pagina informativa del Party (l’indirizzo del proprio sito o di una specifica pagina che tratti delle capacità del Party relativamente ai processi MODA-ML, anche in maniera discorsiva);
- il BPSS descrivente il Business Process a cui si intende fare riferimento nel proprio CPP;
- un indirizzo e-mail al quale il CPP sarà inviato al termine del processo di editing (il sito MODA-ML [MML] permette di riconoscere gli utenti registrati ed assegnare loro un livello di protezione opportuno al fine di evitare qualsiasi fenomeno di spamming).

La scelta del BPSS è affrontata scegliendo da una lista dei BPSS disponibili. Tale lista è presentata nell’interfaccia con dei radio button per limitare l’utente ad un’unica selezione (figura 5.5).

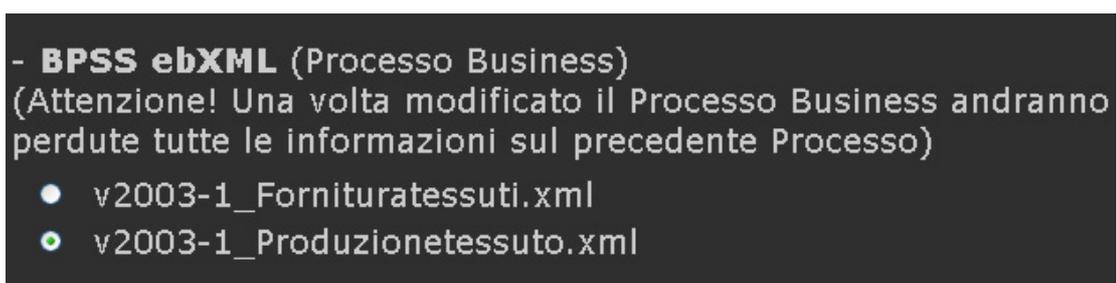


Figura 5.5 Lista dei BPSS disponibili sul Server MODA-ML

La lista mostrata in figura è generata leggendo dalla directory del server MODA-ML contenente i BPSS creati con il BPSS-Generator. Nel caso in cui sia stato caricato dall’utente un CPP che si riferisce a un BPSS non presente in tale directory, viene generato e visualizzato il messaggio di errore corrispondente.

5.3.3 CPP-Editor Step 2

Questo secondo step, implementato nel file *cppeditor_step2.asp*, si preoccupa prima di tutto di verificare i dati immessi nel passo precedente. Se è stato caricato il CPP di default, infatti, l'utente deve aver necessariamente compilato tutti i campi presentati nell'interfaccia allo step 1.

L'applicazione a questo punto crea un nuovo file temporaneo, di tipo XML, assegnandogli come nome il *cppid* inserito e inserendovi il CPP dell'utente. Questa azione è effettuata per evitare situazioni di errore che si avrebbero con l'accesso contemporaneo di più utenti allo stesso CPP di default.

Le informazioni relative a: *cppid*, nome del Party, URL del Party e e-mail vengono facilmente gestite modificando le informazioni negli elementi o attributi XML corrispondenti (elementi che si sono descritti nel par. 3.3.3) o, nel caso della e-mail, modificando la variabile di sessione appropriata.

La decisione di cambiare il BPSS al quale fare riferimento provoca invece modifiche consistenti nel CPP poiché tutte le informazioni concernenti il Business Process e i documenti che lo compongono verranno modificate di conseguenza.

La funzione *SetBPSS* si occupa di rendere concrete queste modifiche nel CPP:

- cambiando tutti i riferimenti al BPSS con il nuovo BPSS scelto (nell'elemento *CollaborationRole*);
- cancellando tutte le informazioni relative ai documenti di scambio che compongono il Business Process descritto nel BPSS precedente (negli elementi *ServiceBinding*, *SimplePart* e *Packaging*, si veda par. 3.3.3 per una loro breve descrizione);
- cancellando tutte le informazioni relative alle entità opzionali dei documenti (nell'elemento *DocumentOptionalElements*. Nel par. 4.3.2.2 è descritta la struttura e il funzionamento di questo elemento).

Le funzioni sviluppate per apportare questi aggiornamenti sono raggruppate nel file *funzioni2.inc.asp*, operano sul documento (CPP) secondo il modello DOM e utilizzano i metodi e le proprietà definite dall'oggetto Microsoft.XMLDOM.

Una volta gestite le informazioni immesse nel passo precedente, il secondo passo di editing presenta nell'interfaccia una rappresentazione del Business Process MODA-ML scelto. Questo è visualizzato nella propria interezza tramite una tabella HTML, mostrando la suddivisione in attività e i messaggi corrispondenti alle transazioni commerciali (figura 5.6).

La visualizzazione del Business Process mantiene la struttura dei Business Process MODA-ML (si provi a confrontare la figura seguente con quelle del par. 2.3) a dimostrazione dello sforzo fatto per cercare di non perdere nessuna delle informazioni potenzialmente utili (par. 4.3.1).

Home	Step 1	Step 2	Step 3	Step 4		
MODA-ML Process: Fornituratessuti [v2003-1_Proc-1]						
MODA-ML Activity: Scelta tessuti						
Fornitore di tessuti	Send	-->	<input checked="" type="checkbox"/> SMTP	Scheda Anagrafica Tessuto	-->	Confezionista
	Send	-->	<input checked="" type="checkbox"/> FTP	Listino tessuti	-->	
	Receive	<--	<input checked="" type="checkbox"/> SMTP	Previsione Collezione Tessuti	<--	
MODA-ML Activity: Acquisto tessuti						
Fornitore di tessuti	Receive	<--	<input checked="" type="checkbox"/> SMTP	Ordine al fornitore di tessuti	<--	Confezionista
	Send	-->	<input checked="" type="checkbox"/> SMTP	Risposta ordine del fornitore di tessuti	-->	
	Send	-->	<input checked="" type="checkbox"/> SMTP	Avanzamento ordine del fornitore di tessuti	-->	
	Receive	<--	<input checked="" type="checkbox"/> SMTP	Modifica ordine al fornitore di tessuti	<--	
MODA-ML Activity: Spedizione tessuti						
Fornitore di tessuti	Receive	<--	<input checked="" type="checkbox"/> SMTP	Richiesta spedizione tessuto	<--	Confezionista
	Send	-->	<input type="checkbox"/> SMTP	Avviso spedizione del tessuto	-->	
	Send	-->	<input type="checkbox"/> FTP	Report Qualità pezza	-->	
	Send	-->	<input type="checkbox"/> FAX	Fattura Tessitore	-->	
Ruolo:		Specificare gli endpoint dei Protocolli scelti (e scegliere, con il radio button, quello di default)				
• Fornitore di tessuti		SMTP: <input checked="" type="radio"/> novelli@mioendpoint.it	SMTP2: <input type="radio"/> novellcristiano@mioen			
• Confezionista		FTP: <input type="radio"/> ftp://cristianonovelli.it	FTP2: <input type="radio"/> ftp://genericuser.it			
		HTTP: <input type="radio"/> http://www.cristianonove	FAX: <input type="radio"/> 000000000			
Next						

Figura 5.6 CPP-Editor Step 2

La visualizzazione del Business Process, in questa fase, è molto simile alla visualizzazione XSLT dei BPSS (figura 5.2). In questo caso però nessun foglio di stile è stato applicato. Il motivo è da rapportarsi al fatto che questa visualizzazione, interna ad un form HTML, contiene alcuni componenti quali check box e menù a cascata, creati dinamicamente di volta in volta; i nomi stessi e i valori delle variabili associate a questi componenti sono dipendenti dal processo selezionato e di conseguenza dal numero di documenti presenti in esso.

Per ognuna delle transazioni commerciali è data la possibilità di specificare:

1. se la transazione è supportata o meno (tramite check box)
2. il tipo di protocollo di trasporto che si intende usare per tale transazione (tramite menù a tendina)

MODA-ML Activity: Acquisto tessuti							
Fornitore di tessuti	Receive	<--	<input checked="" type="checkbox"/>	SMTP	Ordine al fornitore di tessuti	<--	Confezionista
	Send	-->	<input checked="" type="checkbox"/>	FTP	Risposta ordine del fornitore di tessuti	-->	
	Send	-->	<input checked="" type="checkbox"/>	HTTP	Avanzamento ordine del fornitore di tessuti	-->	
	Receive	<--	<input checked="" type="checkbox"/>	FAX	Modifica ordine al fornitore di tessuti	<--	

Figura 5.7

E' dato all'utente scegliere quale dei due ruoli disponibili intende interpretare nel Business Process e specificare gli endpoint dei protocolli di trasporto scelti per lo scambio dei messaggi (p.es. se per uno specifico messaggio in ricezione ho scelto il protocollo di trasporto SMTP, allora dovrò specificare un indirizzo e-mail al quale desidero ricevere tale messaggio nella Business Collaboration. Nella figura 5.8 sono mostrati più dettagliatamente i campi in cui specificare gli endpoint).

Specificare gli endpoint dei Protocolli scelti (e scegliere, con il radio button, quello di default)			
SMTP:	cnovelli@mioendpoint.it	<input checked="" type="radio"/>	
SMTP2:	novellicristiano@mioen	<input type="radio"/>	
FTP:	ftp://cristianonovelli.it	<input type="radio"/>	
FTP2:	ftp://genericuser.it	<input type="radio"/>	
HTTP:	http://www.cristianonove	<input type="radio"/>	
FAX:	000000000	<input type="radio"/>	

Figura 5.8

Il CPP-Editor, come già detto in precedenza, non si limita a permettere all'utente di immettere le informazioni per creare un nuovo CPP, ma deve anche fornire la possibilità di modificare i valori già contenuti in un proprio Profilo precedentemente creato. L'interfaccia deve allora presentare nei campi del form tali valori dopo una lettura del CPP sul quale si sta lavorando per poterne effettuare i cambiamenti.

5.3.4 CPP-Editor Step 3

Il terzo passo di editing è implementato nel file *cppeditor_step3.asp* ed è probabilmente la fase che ha richiesto il maggiore sforzo per essere concretizzata.

Nella prima parte di questo terzo step sono gestite le informazioni, inviate dal passo precedente, relative a:

- documenti di scambio selezionati;
- tipi di protocollo di trasporto selezionati per ogni documento scelto;
- endpoint mittente e/o destinatario per i tipi di protocollo scelti;
- ruolo scelto nel Business Process.

Le funzioni utilizzate in questa prima parte sono raggruppate nel file *funzioni3.inc.asp*.

I documenti selezionati (o deselezionati) comportano i seguenti aggiornamenti nel CPP:

- aggiornamento dell'elemento *ServiceBinding*: sono aggiunti (o eliminati) gli elementi figli *CanSend* e *CanReceive* rispettivamente descriventi quali documenti si intende spedire e ricevere nella Business Collaboration;

- inserimento (o eliminazione) degli elementi *SimplePart*, e *Packaging* per ogni messaggio;
- aggiornamento dell'elemento *DocumentOptionalElement*: sono aggiunti (o eliminati) gli elementi *Doc*.

La seconda parte di questo step si occupa di gestire le preferenze espresse relativamente alle entità opzionali dei documenti di scambio. Nella fase di progettazione è stato descritto l'approccio scelto per presentare all'utente la rappresentazione ad albero dei tipi di documento (si veda il capitolo precedente, par. 4.5.4), di seguito descriverò l'implementazione di tale approccio.

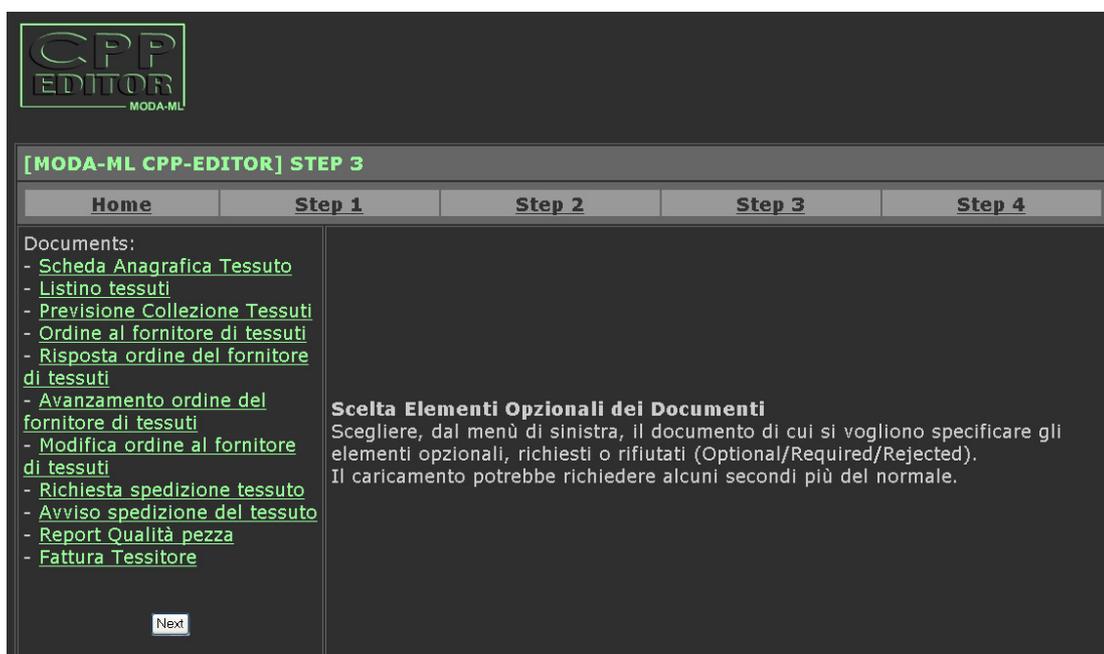


Figura 5.9 CPP-Editor Step 3

L'interfaccia di questo passo (figura 5.9) è stata organizzata in due parti fondamentali:

1. un menù sulla sinistra, sempre presente, le cui voci corrispondono ai documenti supportati. Tali voci richiamano, nel riquadro alla destra del menù, la visualizzazione dell'albero del tipo di documento corrispondente;

2. nel riquadro alla destra del menù viene visualizzata, tramite una tabella HTML contenuta in un form, la rappresentazione ad albero del tipo di documento scelto (mostrata in figura 5.10).

TEXOrder		Radice del documento 'Ordine al fornitore di tessuti'
@TOtype		tipo di ordine di acquisto
@msgfunction	Optional ▾	funzione rivestita dal presente messaggio in relazione alla sua trasmissione
- TOheader		testata del documento ORDINE DI TESSUTO
- - otherCurrency	Optional ▾	altra valuta specificata nel documento
- - @currencyUseQualifier		attribuzione d'uso di un'altra valuta nel documento
- - supplier		dati del Fornitore
- - @logo	Optional ▾	logo aziendale (URL dell'immagine in formato jpg o gif). Si raccomanda che l'immagine non ecceda i 640 pixel di larghezza.
- - msgN		numero identificativo assegnato al messaggio da chi lo emette
- - thirdParty	Optional ▾	dati di una Terza Parte, diversa da Cliente o Fornitore
- - @role		qualificatore della Terza Parte
- - buyer		dati del Cliente
- - @logo	Optional ▾	logo aziendale (URL dell'immagine in formato jpg o gif). Si raccomanda che l'immagine non ecceda i 640 pixel di larghezza.
- - refDoc	Optional ▾	identificazione di un documento di riferimento ed eventualmente di un suo item (riga)
- - @docType		tipo del documento
- - - itemID	Optional ▾	numero che contraddistingue la riga/item del documento in oggetto
- - - docID		numero identificativo assegnato al documento in oggetto da chi lo emette
- - - docDate	Optional ▾	data di emissione del documento in oggetto

Figura 5.10 Visualizzazione struttura del tipo di documento

La prima colonna contiene tutti i nomi delle entità; nel caso tali nomi corrispondano ad attributi, sono preceduti dal simbolo @. Nella seconda colonna è visualizzato, solo per le entità di tipo opzionale, un menù a tendina che permette all'utente di esprimere la propria preferenza (Opzionale, Richiesto o Rifiutato). Nella terza colonna vi è una breve descrizione testuale dell'entità.

Il terzo passo di editing presenta una situazione diversa dai precedenti.

Un utente può decidere di voler visualizzare più rappresentazioni ad albero per ogni tipo di documento sul quale vuole esprimere le proprie preferenze; tuttavia può essere visualizzata solo una rappresentazione ad albero per volta (se ne sono spiegate le motivazioni nel par. 4.5.4). Ne consegue la necessità che l'utente "rimanga" nel terzo passo finché non ha terminato di esprimere le proprie preferenze riguardanti gli elementi opzionali di tutti i documenti scelti (o deciso di non esprimerne alcuna).

Per questo motivo *cppeditor_step3.asp* non invia i dati inseriti al passo successivo (*cppeditor_step4.asp*), li invia invece a sé stesso, in modo tale che vengano elaborati e inseriti nel CPP. Ogni volta che l'inserimento delle preferenze di un documento vengono inviate, l'applicazione ripresenta l'interfaccia del terzo passo; solo quando l'utente deciderà di abbandonare questa fase, allora l'editor passerà alla successiva.

La visualizzazione del tipo di documento secondo una struttura ad albero, segue lo stesso approccio di quello utilizzato per la creazione delle guide e degli Schema dal Dizionario MODA-ML. Sono state sviluppate molte funzioni dedicate alla sola generazione della rappresentazione ad albero; si è deciso di raggrupparle separatamente dalle altre nel file *funzioni3_MakeTree.inc.asp*. Di seguito illustriamo le principali funzioni per la generazione dell'albero del tipo di documento.

La funzione di partenza per la generazione della visualizzazione ad albero è la *MakeTree* che, ricevuto il nome del documento, trova la corrispondente radice nella tabella documenti, la stampa insieme alla descrizione corrispondente e infine la trasmette alla funzione ricorsiva *MakeSubTree*.

La funzione *MakeSubTree*, ricevuto come parametro un elemento si preoccupa di:

1. chiamare la funzione *ViewAttribute* per visualizzare gli attributi corrispondenti all'elemento oggetto;
2. Trovare la lista degli eventuali figli dell'elemento oggetto;
3. Per ognuno dei figli la *MakeSubTree* compie le seguenti azioni in successione:

- A. stampa, nella prima colonna della tabella, il nome del figlio, con la giusta indentatura per la visualizzazione ad albero,
- B. se l'elemento figlio in esame è di tipo opzionale, verifica, nell'elemento *DocumentOptionalElements* del CPP, eventuali precedenti preferenze espresse per tale elemento e visualizza, nella seconda colonna della tabella, il menù a tendina per esprimere la propria scelta o modificarne una fatta in precedenza,
- C. stampa la descrizione relativa a questo elemento nella terza colonna della tabella,
- D. richiama sé stessa, ricorsivamente, per generare il sottoalbero del figlio preso in considerazione.

La funzione *ViewAttribute*, chiamata nel punto 1, si comporta nel medesimo modo della *MakeSubTree* visualizzando:

- A. il nome dell'attributo nella prima colonna della tabella,
- B. un menù a tendina, nella seconda colonna della tabella, nel caso sia opzionale per esprimere la preferenza o modificarne la precedente,
- C. la descrizione dell'attributo nella terza colonna della tabella.

(La funzione *ViewAttribute*, naturalmente, si limita a questi tre punti; essendo gli attributi XML privi di figli non vi sono chiamate ricorsive).

Tutte le informazioni ricavate per generare l'albero del tipo di documento sono estrapolate dal Dizionario MODA-ML (si veda il par. 2.4.4 "Dalla radice del documento ai singoli componenti" e le figure in esso riportate).

5.3.5 CPP-Editor Step 4

Nell'ultimo passo, implementato nel file *cppeditor_step3.asp*, viene presentato un breve riepilogo di alcune scelte effettuate. Un link al CPP, ancora residente sul server, permette di poter visualizzarne il codice XML.

Come ultima azione l'utente può confermare di aver concluso la creazione o la modifica del CPP e attivare così la spedizione del file tramite e-mail (si veda la figura 5.11).

Una volta data la conferma:

- il CPP viene inviato all'indirizzo e-mail specificato dall'utente nel primo passo di editing;
- il file temporaneo, sul quale sono state apportate le modifiche viene, cancellato dal server;
- vengono rilasciate tutte le variabili di sessione utilizzate.

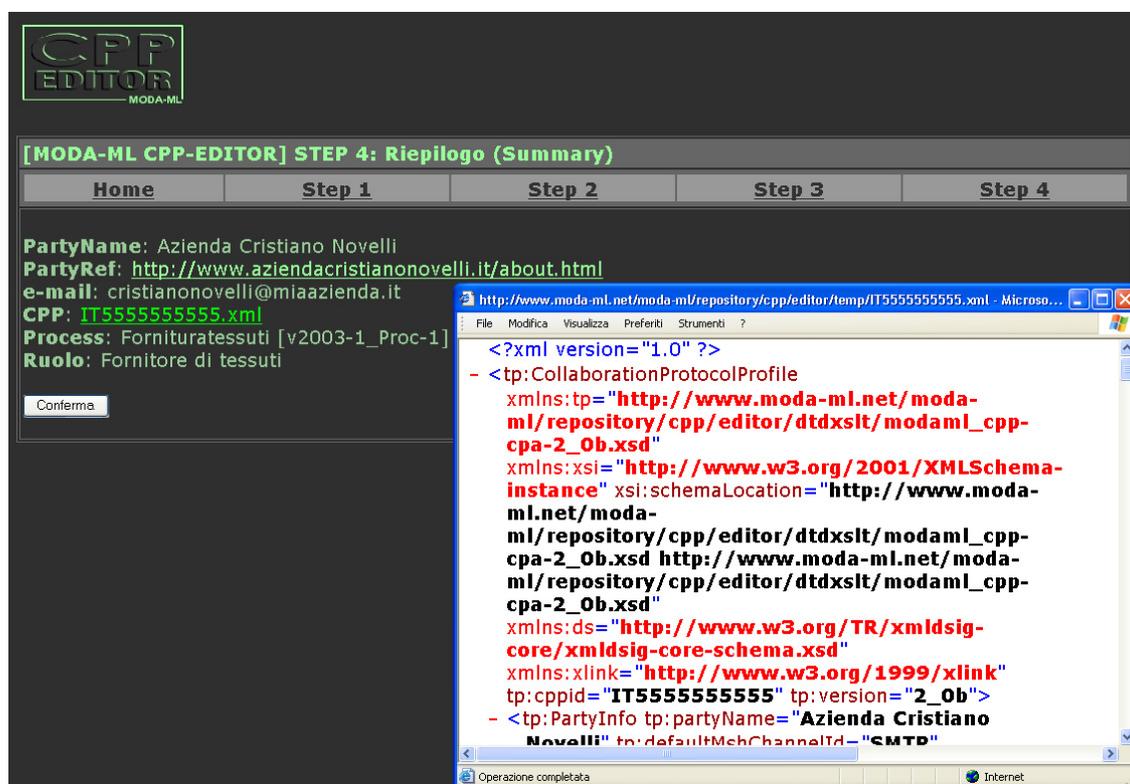


Figura 5.11 CPP-Editor Step 4

5.3.6 Libertà di navigazione nelle cinque fasi del CPP-Editor

E' stato fortemente richiesto, dal gruppo di ricerca ENEA, che l'editor non si limitasse a presentare le cinque fasi in successione. Una volta che l'utente avesse affrontato le scelte principali, fondamentali per quelle successive, l'applicazione avrebbe dovuto

dare completa libertà di navigare tra le cinque pagine dell'editor per eseguire ulteriori cambiamenti al CPP.

Il problema che si è presentato è stato quello di fornire all'applicazione una metodologia per memorizzare temporaneamente la grande quantità di informazioni che caratterizza il Profilo di Collaborazione.

Una prima soluzione individuata è l'utilizzo delle variabili di sessione del linguaggio ASP VBScript. L'oggetto DOM corrispondente al CPP non può essere però immagazzinato nella sessione ASP, poiché tale azione non è supportata dal linguaggio. Utilizzare le sole variabili di sessione sarebbe equivalso ad emulare il CPP nella sua interezza, la mole di informazioni da immagazzinare è consistente e con un'organizzazione dei dati non banale.

Il problema è stato risolto, infine, utilizzando un file temporaneo di supporto: il CPP stesso caricato dall'utente sul server (o quello di default opportunamente rinominato).

Ad ogni passo di editing:

1. viene letto il CPP temporaneo e creato l'oggetto DOM corrispondente;
2. vengono inserite le informazioni e attuate le modifiche dipendenti dalle eventuali scelte espresse dall'utente nel passo precedente;
3. vengono lette le informazioni per creare dinamicamente la nuova interfaccia da presentare all'utente in questo passo.

Questa soluzione (unitamente all'utilizzo di poche variabili di sessione) è risultata ottimale e funzionale. In questo modo si ha la certezza che l'allocazione di memoria del server non oltrepassi la soglia critica.

Un altro sforzo che si è dovuto compiere è stato quello di limitare all'utente i permessi di navigazione tra le cinque pagine al fine di evitare situazioni di inconsistenza. L'interfaccia di ogni passo di editing viene creata in base a scelte (fondamentali e obbligatorie) effettuate nel passo precedente. Raggiungere una fase di editing senza aver prima effettuato le scelte nella precedente, porterebbe a situazioni di incoerenza, errore e malfunzionamento. Le uniche pagine che devono poter essere accessibili in un dato momento sono quindi quelle "attivate" dalle informazioni già inserite.

Per poter limitare la navigazione dell'utente alle sole pagine rese "attive" dalle sue scelte, è stato reso dinamico il menù di navigazione (fig. 5.12).



Figura 5.12 Menù di navigazione dinamico

Le voci relative alle pagine successive a quella corrente vengono rese "linkabili" o meno a seconda che siano o meno stati caricati i dati che costituiscono un prerequisito della pagina. Lo scopo è quello di evitare che si vengano a creare situazioni inconsistenti e che porterebbero ad errori certi (l'esempio più banale di errore che potrebbe occorrere è il seguente: se mi trovo al passo 1 e non scelgo quale Business Process gestire, è impossibile "saltare" al passo 3 per scegliere gli elementi opzionali relativi ai documenti di un Business Process non ancora specificato. Non si è ritenuto adeguata l'adozione di valori di default poiché la caratteristica principale dell'editor, la modifica di CPP, sarebbe andata perduta).

5.4 Gestione degli errori

BPSS-Generator e CPP-Editor possono incontrare diverse tipologie di errori durante il loro funzionamento. Per evitare situazioni non gestite e possibile perdita dei dati immessi (in particolar modo nel caso del CPP-Editor) per ognuna delle due applicazioni sono stati creati due file, `error.asp` e `error.inc.asp`, appositamente per gestire l'occorrenza degli errori riconosciuti. Al loro interno sono stati definiti:

1. una funzione per ricevere e gestire l'errore: *GetErrorDescription*
2. una matrice che racchiude le seguenti informazioni per ogni errore catalogato:
 - codice errore
 - descrizione dell'errore in lingua italiana
 - descrizione dell'errore in lingua inglese

Il funzionamento di questo file avviene come segue:

- se un errore viene generato durante l'esecuzione del codice ASP VB Script, viene chiamata la funzione *GetErrorDescription* passandole come parametro il codice errore relativo, il tipo di lingua in uso più, eventualmente, il messaggio di errore restituito dal linguaggio ASP;
- la funzione ricerca nella matrice il codice e restituisce il messaggio nel linguaggio prescelto;
- la pagina di gestione degli errori visualizza un messaggio presentandone la causa, una possibile soluzione, l'eventuale messaggio di errore ASP e un link "Riprendi" per ritornare al punto in cui l'errore è occorso (nella figura 5.13 è visualizzato un esempio).

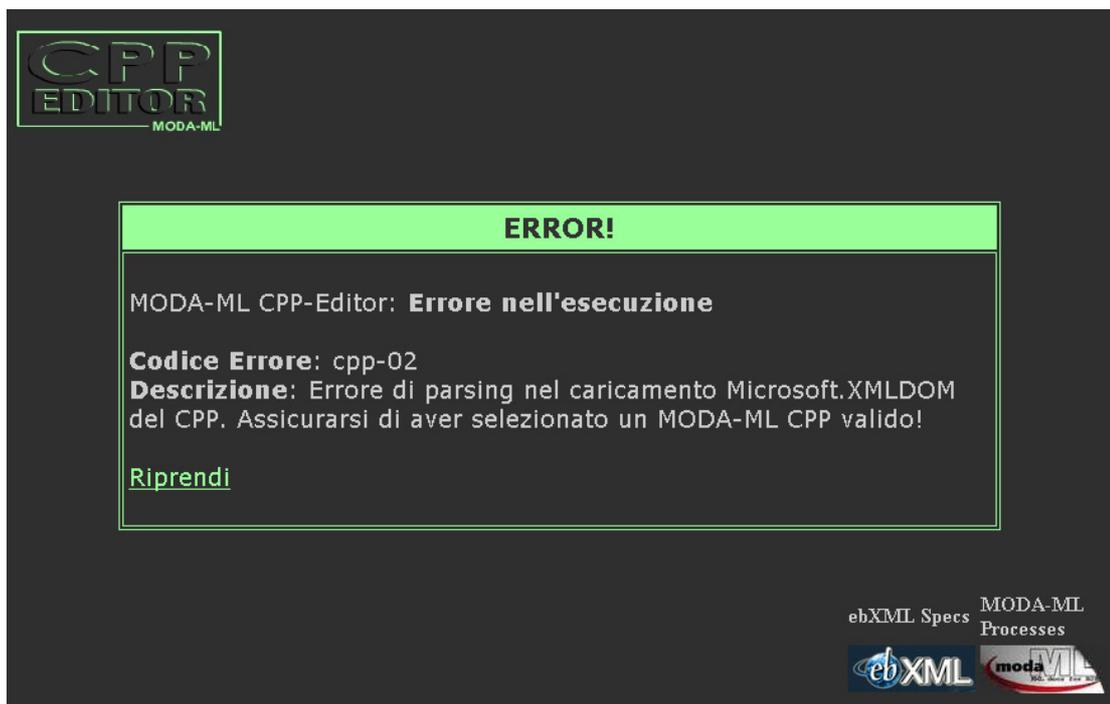


Figura 5.13 Messaggio di errore

In questo modo la gestione degli errori è completamente divisa dall'applicazione vera e propria e la manutenzione dei vari tipi di messaggi in entrambe le lingue diventa più semplice e organizzata.

5.5 Installazione delle applicazioni sul server MODA-ML

Le applicazioni sono state installate sul server MODA-ML senza riscontrare particolari difficoltà (gli URL sono indicati nella bibliografia, [CPNETa] e [CPNETb]). Durante l'implementazione si è cercato di raggruppare tutti i parametri di configurazione (p.es. il nome DNS assegnato al Dizionario durante la configurazione ODBC, il mittente delle e-mail inviate agli utenti finali del CPP-Editor ecc.) nel file *costanti.inc.asp*, presente con lo stesso nome sia nel BPSS-Generator che nel CPP-Editor. Nella fase di installazione si sono cambiati solamente i parametri contenuti in tale file (una spiegazione relativa è contenuta nel file *Install.txt*).

Il BPSS-Generator non ha dato nessun tipo di complicazione nella fase di testing; i BPSS creati in entrambe le lingue sono risultati essere corretti e immediatamente disponibili nelle directory di destinazione.

La fase di testing del CPP-Editor è stata più lunga e non priva di difficoltà. La varietà di possibili informazioni da inserire e di scelte da eseguire durante le cinque fasi ha dato luogo ad altrettanti casi in cui verificare eventuali problemi. L'ostacolo più consistente, e che ha richiesto lo sforzo maggiore per essere superato, è stato dato dalla creazione della rappresentazione ad albero dei tipi di documento; mentre nel funzionamento su computer locale nessun problema era stato riscontrato, sul server MODA-ML si è notato un incremento del tempo di elaborazione eccessivo che provocava l'errore che mostriamo di seguito:

Active Server Pages error 'ASP 0113'

Script timed out

/moda-ml/repository/cpp/editor/cppeditor_step3.asp

The maximum amount of time for a script to execute was exceeded. You can change this limit by specifying a new value for the property Server.ScriptTimeout or by changing the value in the IIS administration tools.

Come conseguenza di questo errore, per alcuni tipi di documento non era possibile visualizzare la rappresentazione ad albero. Una soluzione poteva essere quella proposta dallo stesso messaggio di errore, ovvero aumentare la soglia di tempo oltre la quale l'esecuzione di uno script viene interrotta; si è preferito rivedere le funzioni sviluppate per la generazione dell'albero ed individuare i punti deboli dell'algoritmo ricorsivo creato.

Dopo un'attenta analisi del problema si è cercato di:

- ridurre il più possibile il numero di oggetti di tipo recordset aperti sulla connessione ODBC al Dizionario (con maggior attenzione sull'utilizzo delle funzioni ricorsive);
- ridurre le interrogazioni al Dizionario MODA-ML "fondendo" tra loro alcune query SQL;
- riutilizzando alcune variabili, quando possibile, per diminuire l'utilizzo di memoria allocata.

Dopo aver apportato queste modifiche il CPP-Editor è risultato funzionante nella sua totalità e i CPP creati e successivamente modificati hanno risposto bene alle esigenze richieste.

6 CONCLUSIONI

Scopo di questa tesi è stato progettare un'architettura, basata su ebXML, che fosse di supporto alle aziende del settore Tessile e Abbigliamento italiano che hanno aderito alle iniziative del progetto MODA-ML.

Lo scopo ultimo dell'architettura sviluppata è la creazione di Profili di Collaborazione (CPP) per favorire le Business Collaborations nell'e-commerce.

La tesi si è sviluppata in quattro parti distinte:

1. studio del contesto MODA-ML e individuazione del problema (cap. 2);
2. studio del framework ebXML e degli standard utilizzati per gestire i Protocolli di Collaborazione (cap. 3);
3. adattamento degli standard ebXML BPSS, CPP e CPA al contesto MODA-ML e progettazione di un'architettura in grado di utilizzarli (cap. 4);
4. implementazione di due applicazioni web atte a gestire le specifiche MODA-ML dei Profili di Collaborazione (cap. 5).

Nella **prima parte** del progetto di tesi si è cercato di individuare il problema da risolvere. Il progetto MODA-ML, allo stato attuale, ha definito uno standard XML per lo scambio di dati al fine di permettere una comunicazione più veloce e meno ambigua tra le aziende del tessile e abbigliamento italiano. Tali aziende, infatti, per la natura stessa del settore commerciale nel quale operano, sono molto numerose, diversificate per scopi e dimensioni, con conseguenti capacità e metodologie diverse per implementare i propri sistemi informatici gestionali. Conseguenza di questa situazione è che il tempo speso per comunicare tra i vari anelli della filiera supera, di fatto, il tempo effettivo del processo produttivo. Lo standard MODA-ML è il primo passo per definire un linguaggio comune a tutte le aziende per lo scambio di dati.

Il progetto MODA-ML ha descritto le strutture dei documenti di scambio e alcuni processi commerciali in una banca dati chiamata Dizionario, tramite la quale è possibile generare Guide Utente e Schemi XML dei documenti. Allo stato dell'arte, le aziende partecipanti al progetto utilizzano i documenti di scambio forniti da MODA-ML, ma devono prima sapere esattamente come avverrà la Business Collaboration, quali documenti verranno spediti, in quale ordine e in quali modalità. Per effettuare questo allineamento dei sistemi gestionali elettronici, si ricorre a mezzi di comunicazione quali telefono, fax o e-mail, che rendono tale processo lento e costoso in termini di risorse umane. Inoltre la definizione XML dei tipi di documento di scambio comprende un gran numero di entità opzionali che, se da un lato permettono una descrizione particolareggiata delle diverse situazioni business, dall'altro sono a volte dispersive, e in alcuni rari casi necessitano di un accordo tra le parti per eliminare le ambiguità.

Una soluzione per rendere automatico gran parte di questo meccanismo di ricerca dell'accordo è fornito dai **Profili di Collaborazione (CPP)**. Il framework ebXML ne fornisce una possibile implementazione e ha rilasciato delle specifiche basate su XML per definirli.

Nella **seconda parte** della tesi è stata descritta l'architettura e gli standard ebXML. Il framework ebXML nasce per favorire l'evoluzione dell'e-business, ovvero per facilitare e rendere più veloce la creazione di Business Collaborations tramite il web. Una parte consistente di tale architettura sono i Protocolli di Collaborazione che si dividono in :

- CPP (Collaboration Protocol Profile): sono Profili di Collaborazione che permettono alle aziende di descrivere in maniera formale le proprie capacità relativamente alle Business Collaborations che intendono supportare;
- CPA (Collaboration Protocol Agreement): sono Contratti di Collaborazione che descrivono in maniera formale l'accordo raggiunto tra le aziende per affrontare insieme Business Collaborations.

Indispensabile per rendere completa l'implementazione dei Protocolli di Collaborazione è la definizione degli Schemi di Specificazione di Business Process (BPSS) che permettono di descrivere i Business Process.

La **terza parte** della tesi è stata incentrata su un possibile adattamento del framework ebXML al contesto MODA-ML per attuare la creazione di Profili di Collaborazione. La fase di progettazione è consistita nei seguenti **obiettivi**:

1. **ideazione di un'architettura**, basata sul framework ebXML, per la creazione dei Profili di Collaborazione;
2. definizione delle specifiche MODA-ML BPSS, CPP e CPA basate sugli standard ebXML. In particolare si resa necessaria un'**estensione dello standard CPP**;
3. progettazione di due applicazioni web per MODA-ML:
 - **BPSS-Generator**: genera in maniera automatica MODA-ML BPSS relativi ai Business Process MODA-ML,
 - **CPP-Editor**: permette alle aziende di creare o modificare un proprio MODA-ML CPP.

L'architettura da me progettata è stata chiamata **CP-NET** (Collaboration Profile Networking Enterprises Technology). Si è scelto di sviluppare le due applicazioni come strumenti web, residenti sul server MODA-ML, per i seguenti motivi:

- i BPSS sono creati direttamente dal Dizionario dal solo Team Management di MODA-ML. Il Dizionario, sempre aggiornato, è residente sul server MODA-ML;
- i BPSS devono essere immediatamente resi disponibili nel web per essere usati nei CPP come riferimento univoco dei Business Process MODA-ML;
- il CPP-Editor MODA-ML utilizza, oltre alle informazioni sui Business Process contenute nei BPSS, anche informazioni sulla struttura dei documenti di scambio del progetto MODA-ML. Tali informazioni sono presenti nel Dizionario, sul server MODA-ML;

Il funzionamento dell'architettura CP-NET può essere descritto brevemente come segue:

- A. Il MODA-ML Team Management attiva il BPSS-Generator per creare in maniera automatica i BPSS relativi ai Business Process MODA-ML descritti

nel Dizionario. I BPSS creati sono immediatamente disponibili sul server MODA-ML;

- B. Ogni azienda partecipante al progetto MODA-ML può accedere al CPP-Editor situato sul server MODA-ML. Tramite un'interfaccia web l'editor di CPP guida l'utente nella creazione o nella modifica di un CPP. Le preferenze espresse nel CPP riguardano un determinato Business Process; queste preferenze sono dunque espresse facendo riferimento ad un particolare BPSS;
- C. Il Profilo di Collaborazione creato, o modificato, è inviato tramite e-mail all'azienda che lo ha creato.

Durante la fase di adattamento degli standard ebXML al contesto MODA-ML, si è riscontrata l'esigenza di **estendere lo standard CPP** per poter fornire ai Profili di Collaborazione MODA-ML capacità descrittive non supportate. In particolare, il CPP MODA-ML vuole essere capace di descrivere le preferenze relativamente alle entità opzionali presenti nei documenti di scambio per ottenere una comunicazione priva d'ambiguità.

Per fare questo è stato introdotto un **nuovo elemento XML** composto, *DocumentOptionaElements* (argomento che è stato oggetto di un articolo, "Use and extension of ebXML business profiles for Textile/Clothing firms" [GNBV04]).

La **quarta** e ultima parte del progetto di tesi è stata dedicata all'implementazione delle due applicazioni web progettate.

Le applicazioni sviluppate, una volta installate sul server, hanno risposto in maniera positiva alle esigenze per cui sono state create.

CP-NET fornisce alle aziende del T/A un supporto per creare Profili di Collaborazione unitamente ad una metodologia per definire formalmente i Business Process.

Tra i **progetti futuri** correlati a questa tesi vi sono numerose iniziative legate alla collaborazione e supporto dell'ENEA.

1. Traferimento Tecnologico dell'architettura CP-NET in un contesto concreto: la SW House MAPS ha espressamente richiesto questo tipo di tecnologia per migliorare e rendere più competitivo il proprio prodotto software, in uso presso l'azienda tessile Milar.
2. In parallelo allo sviluppo di CP-NET, un progetto del Politecnico di Milano ha sviluppato sempre per MODA-ML la creazione dei Contratti di Collaborazione tramite un CPA-Editor, trascurando la descrizione dei Business Process e dedicandosi al livello di Trasporto. Il progetto di questa tesi supporta invece la creazione di Profili di Collaborazione ebXML che descrivono in maniera completa le capacità relative alla logica dei Business Process considerando solo in minima parte i livelli relativi a trasporto e sicurezza. Uno sviluppo futuro di CP-NET potrebbe essere dunque la fusione delle due esperienze per ottenere dei Protocolli di Collaborazioni in grado di supportare ebXML in un modo più completo.
3. Allo stato dell'arte la creazione di CPA avviene in maniera manuale mentre il fine ultimo è utilizzare uno strumento che permetta la creazione di un CPA in maniera semiautomatica. L'architettura ebXML non indica come la negoziazione per formare il CPA avvenga, la soluzione è lasciata agli sviluppatori dell'architettura. La soluzione di questo problema potrebbe essere una terza linea di sviluppo futuro. Due approcci potrebbero essere utilizzati per implementare la negoziazione e formazione del CPA:
 - a. Soluzione centralizzata: si tratterebbe di un'applicazione web, residente sul server del framework, che permetta di confrontare due CPP di due Party e individuare i parametri che risultino incompatibili. Ottenuta la segnalazione delle incompatibilità, l'applicazione dovrebbe poi supportare i Party nella risoluzione.
 - b. Soluzione P2P: in questo tipo di approccio i due Party comunicerebbero direttamente uno con l'altro tramite strumenti locali

con protocollo di scambio comune. In questo modo la formazione del CPA sarebbe indipendente da un server centrale. La difficoltà sarebbe nel coordinare le applicazioni comunicanti e stabilire i criteri che guidano la modifica dei CPP in un contesto di lavoro collaborativo.

4. Un quarto progetto futuro di più ampio respiro consiste nell'integrare il CPA nel MSH e/o nel sistema gestionale aziendale per consentire una sorta di autoconfigurazione dei processi di scambio di documenti sulla base degli agreement in vigore con i diversi partner.

APPENDICE A: BPSS “Produzione Tessuto”

Istanza XML di un Business Process Specification Schema MODA-ML.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE ProcessSpecification SYSTEM "ebBPSS.dtd">

<ProcessSpecification name="Produzionetessuto" version="1.1" uuid="v2003-1_Proc-3">

  <!-- Business Documents -->
  <BusinessDocument name="Disposizione rammendo tessuto" nameID="OF51-013">
    <Documentation>Documento che il Produttore del tessuto invia al Rammendatore
per richiedere il servizio di rammendo</Documentation>
  </BusinessDocument>

  <BusinessDocument name="Ritorno Rammendo Tessuto" nameID="OF51-014">
    <Documentation>Documento che il Produttore del tessuto riceve dal Rammendatore
a conclusione del servizio di rammendo</Documentation>
  </BusinessDocument>

  <Package name="moda-ml-v2003-1">
    <!-- Business Transaction -->
    <BusinessTransaction name="Disposizione rammendo tessuto">
      <RequestingBusinessActivity name="">
        <DocumentEnvelope businessDocument="Disposizione rammendo tessuto"/>
      </RequestingBusinessActivity>
      <RespondingBusinessActivity name=""/>
    </BusinessTransaction>

    <BusinessTransaction name="Ritorno Rammendo Tessuto">
      <RequestingBusinessActivity name="">
        <DocumentEnvelope businessDocument="Ritorno Rammendo Tessuto"/>
      </RequestingBusinessActivity>
      <RespondingBusinessActivity name=""/>
    </BusinessTransaction>

    <!-- Binary Collaborations - Att. MODA-ML -->
    <BinaryCollaboration name="Rammendo tessuto" nameID="Att-12">
      <InitiatingRole name="Fornitore di tessuti"/>
      <RespondingRole name="Terzista rammendo"/>
      <BusinessTransactionActivity
name="Disposizione rammendo tessuto"
businessTransaction="Disposizione rammendo tessuto"
fromAuthorizedRole="Fornitore di tessuti"
toAuthorizedRole="Terzista rammendo"/>
      <BusinessTransactionActivity
name="Ritorno Rammendo Tessuto"
businessTransaction="Ritorno Rammendo Tessuto"
fromAuthorizedRole="Terzista rammendo"
toAuthorizedRole="Fornitore di tessuti"/>
      <Start toBusinessState="Disposizione rammendo tessuto"/>
    </BinaryCollaboration>

    <!-- Main Binary Collaboration - MODA-ML Process -->
    <BinaryCollaboration name="Produzionetessuto">
      <InitiatingRole name="Fornitore di tessuti"/>
      <RespondingRole name="Terzista rammendo"/>
      <CollaborationActivity
name="Rammendo tessuto"
binaryCollaboration="Rammendo tessuto"
fromAuthorizedRole="Fornitore di tessuti"
toAuthorizedRole="Terzista rammendo"/>
    </BinaryCollaboration>

  </Package>
</ProcessSpecification>
```


APPENDICE B: BPSS “Fornitura Tessuti”

Istanza XML di un Business Process Specification Schema MODA-ML.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE ProcessSpecification SYSTEM "ebBPSS.dtd">

<ProcessSpecification name="Fornituratessuti" version="1.1" uuid="v2003-1_Proc-1">

  <!-- Business Documents -->
  <BusinessDocument name="Ordine al fornitore di tessuti" nameID="OF51-001">
    <Documentation>Ordine al fornitore di tessuti</Documentation>
  </BusinessDocument>

  <BusinessDocument name="Risposta ordine del fornitore di tessuti" nameID="OF51-
002">
    <Documentation>Risposta all'ordine emessa dal fornitore di
tessuti</Documentation>
  </BusinessDocument>

  <BusinessDocument name="Avanzamento ordine del fornitore di tessuti"
nameID="OF51-003">
    <Documentation>Stato di avanzamento dell'ordine</Documentation>
  </BusinessDocument>

  <BusinessDocument name="Modifica ordine al fornitore di tessuti" nameID="OF51-
004">
    <Documentation>Modifica ordine al fornitore di tessuti</Documentation>
  </BusinessDocument>

  <BusinessDocument name="Richiesta spedizione tessuto" nameID="OF51-005">
    <Documentation>Richiesta spedizione tessuto</Documentation>
  </BusinessDocument>

  <BusinessDocument name="Avviso spedizione del tessuto" nameID="OF51-006">
    <Documentation>Avviso spedizione del tessuto</Documentation>
  </BusinessDocument>

  <BusinessDocument name="Report Qualit&agrave; pezza" nameID="OF51-007">
    <Documentation>Rapporto di qualit&agrave; della pezza</Documentation>
  </BusinessDocument>

  <BusinessDocument name="Scheda Anagrafica Tessuto" nameID="OF51-009">
    <Documentation>Documento tecnico che riporta le caratteristiche commerciali e
costruttive e le propriet&agrave; chimico-fisiche dell'articolo
tessile.</Documentation>
  </BusinessDocument>

  <BusinessDocument name="Listino tessuti" nameID="OF51-010">
    <Documentation>Documento commerciale che riporta i prezzi e le caratteristiche
tecniche degli articoli tessili che compongono l'offerta del
produttore.</Documentation>
  </BusinessDocument>

  <BusinessDocument name="Previsione Collezione Tessuti" nameID="OF51-011">
    <Documentation>Documento con cui il Confezionista anticipa al Tessitore la
conoscenza degli articoli a cui &egrave; interessato ovvero su cui potrebbe emettere
ordini di acquisto</Documentation>
  </BusinessDocument>

  <BusinessDocument name="Fattura Tessitore" nameID="OF51-012">
    <Documentation>Fattura inviata dal Fornitore di tessuto ai suoi
Clienti</Documentation>
  </BusinessDocument>

  <Package name="moda-ml-v2003-1">
    <!-- Business Transaction -->
    <BusinessTransaction name="Ordine al fornitore di tessuti">
      <RequestingBusinessActivity name="">
```

```

        <DocumentEnvelope businessDocument="Ordine al fornitore di tessuti"/>
    </RequestingBusinessActivity>
    <RespondingBusinessActivity name=""/>
</BusinessTransaction>

<BusinessTransaction name="Risposta ordine del fornitore di tessuti">
    <RequestingBusinessActivity name="">
        <DocumentEnvelope businessDocument="Risposta ordine del fornitore di
tessuti"/>
    </RequestingBusinessActivity>
    <RespondingBusinessActivity name=""/>
</BusinessTransaction>

<BusinessTransaction name="Avanzamento ordine del fornitore di tessuti">
    <RequestingBusinessActivity name="">
        <DocumentEnvelope businessDocument="Avanzamento ordine del fornitore di
tessuti"/>
    </RequestingBusinessActivity>
    <RespondingBusinessActivity name=""/>
</BusinessTransaction>

<BusinessTransaction name="Modifica ordine al fornitore di tessuti">
    <RequestingBusinessActivity name="">
        <DocumentEnvelope businessDocument="Modifica ordine al fornitore di
tessuti"/>
    </RequestingBusinessActivity>
    <RespondingBusinessActivity name=""/>
</BusinessTransaction>

<BusinessTransaction name="Richiesta spedizione tessuto">
    <RequestingBusinessActivity name="">
        <DocumentEnvelope businessDocument="Richiesta spedizione tessuto"/>
    </RequestingBusinessActivity>
    <RespondingBusinessActivity name=""/>
</BusinessTransaction>

<BusinessTransaction name="Avviso spedizione del tessuto">
    <RequestingBusinessActivity name="">
        <DocumentEnvelope businessDocument="Avviso spedizione del tessuto"/>
    </RequestingBusinessActivity>
    <RespondingBusinessActivity name=""/>
</BusinessTransaction>

<BusinessTransaction name="Report Qualit&agrave; pezza">
    <RequestingBusinessActivity name="">
        <DocumentEnvelope businessDocument="Report Qualit&agrave; pezza"/>
    </RequestingBusinessActivity>
    <RespondingBusinessActivity name=""/>
</BusinessTransaction>

<BusinessTransaction name="Scheda Anagrafica Tessuto">
    <RequestingBusinessActivity name="">
        <DocumentEnvelope businessDocument="Scheda Anagrafica Tessuto"/>
    </RequestingBusinessActivity>
    <RespondingBusinessActivity name=""/>
</BusinessTransaction>

<BusinessTransaction name="Listino tessuti">
    <RequestingBusinessActivity name="">
        <DocumentEnvelope businessDocument="Listino tessuti"/>
    </RequestingBusinessActivity>
    <RespondingBusinessActivity name=""/>
</BusinessTransaction>

<BusinessTransaction name="Previsione Collezione Tessuti">
    <RequestingBusinessActivity name="">
        <DocumentEnvelope businessDocument="Previsione Collezione Tessuti"/>
    </RequestingBusinessActivity>
    <RespondingBusinessActivity name=""/>
</BusinessTransaction>

<BusinessTransaction name="Fattura Tessitore">
    <RequestingBusinessActivity name="">
        <DocumentEnvelope businessDocument="Fattura Tessitore"/>

```

```

    </RequestingBusinessActivity>
    <RespondingBusinessActivity name="" />
</BusinessTransaction>

<!-- Binary Collaborations - Att. MODA-ML -->
<BinaryCollaboration name="Scelta tessuti" nameID="Att-7">
  <InitiatingRole name="Fornitore di tessuti"/>
  <RespondingRole name="Confezionista"/>
  <BusinessTransactionActivity
    name="Listino tessuti"
    businessTransaction="Listino tessuti"
    fromAuthorizedRole="Fornitore di tessuti"
    toAuthorizedRole="Confezionista"/>
  <BusinessTransactionActivity
    name="Scheda Anagrafica Tessuto"
    businessTransaction="Scheda Anagrafica Tessuto"
    fromAuthorizedRole="Fornitore di tessuti"
    toAuthorizedRole="Confezionista"/>
  <BusinessTransactionActivity
    name="Previsione Collezione Tessuti"
    businessTransaction="Previsione Collezione Tessuti"
    fromAuthorizedRole="Confezionista"
    toAuthorizedRole="Fornitore di tessuti"/>
  <Start toBusinessState="Listino tessuti"/>
</BinaryCollaboration>

<BinaryCollaboration name="Acquisto tessuti" nameID="Att-8">
  <InitiatingRole name="Confezionista"/>
  <RespondingRole name="Fornitore di tessuti"/>
  <BusinessTransactionActivity
    name="Ordine al fornitore di tessuti"
    businessTransaction="Ordine al fornitore di tessuti"
    fromAuthorizedRole="Confezionista"
    toAuthorizedRole="Fornitore di tessuti"/>
  <BusinessTransactionActivity
    name="Risposta ordine del fornitore di tessuti"
    businessTransaction="Risposta ordine del fornitore di tessuti"
    fromAuthorizedRole="Fornitore di tessuti"
    toAuthorizedRole="Confezionista"/>
  <BusinessTransactionActivity
    name="Modifica ordine al fornitore di tessuti"
    businessTransaction="Modifica ordine al fornitore di tessuti"
    fromAuthorizedRole="Confezionista"
    toAuthorizedRole="Fornitore di tessuti"/>
  <BusinessTransactionActivity
    name="Avanzamento ordine del fornitore di tessuti"
    businessTransaction="Avanzamento ordine del fornitore di tessuti"
    fromAuthorizedRole="Fornitore di tessuti"
    toAuthorizedRole="Confezionista"/>
  <Start toBusinessState="Ordine al fornitore di tessuti"/>
</BinaryCollaboration>

<BinaryCollaboration name="Spedizione tessuti" nameID="Att-6">
  <InitiatingRole name="Confezionista"/>
  <RespondingRole name="Fornitore di tessuti"/>
  <BusinessTransactionActivity
    name="Richiesta spedizione tessuto"
    businessTransaction="Richiesta spedizione tessuto"
    fromAuthorizedRole="Confezionista"
    toAuthorizedRole="Fornitore di tessuti"/>
  <BusinessTransactionActivity
    name="Avviso spedizione del tessuto"
    businessTransaction="Avviso spedizione del tessuto"
    fromAuthorizedRole="Fornitore di tessuti"
    toAuthorizedRole="Confezionista"/>
  <BusinessTransactionActivity
    name="Report Qualit&agrave; pezza"
    businessTransaction="Report Qualit&agrave; pezza"
    fromAuthorizedRole="Fornitore di tessuti"
    toAuthorizedRole="Confezionista"/>
  <BusinessTransactionActivity
    name="Fattura Tessitore"
    businessTransaction="Fattura Tessitore"

```

```

        fromAuthorizedRole="Fornitore di tessuti"
        toAuthorizedRole="Confezionista"/>
    <Start toBusinessState="Richiesta spedizione tessuto"/>
</BinaryCollaboration>

<!-- Main Binary Collaboration - MODA-ML Process -->
<BinaryCollaboration name="Fornituratessuti">
    <InitiatingRole name="Fornitore di tessuti"/>
    <RespondingRole name="Confezionista"/>
    <CollaborationActivity
        name="Scelta tessuti"
        binaryCollaboration="Scelta tessuti"
        fromAuthorizedRole="Fornitore di tessuti"
        toAuthorizedRole="Confezionista"/>
    <CollaborationActivity
        name="Acquisto tessuti"
        binaryCollaboration="Acquisto tessuti"
        fromAuthorizedRole="Confezionista"
        toAuthorizedRole="Fornitore di tessuti"/>
    <CollaborationActivity
        name="Spedizione tessuti"
        binaryCollaboration="Spedizione tessuti"
        fromAuthorizedRole="Confezionista"
        toAuthorizedRole="Fornitore di tessuti"/>
</BinaryCollaboration>

</Package>

</ProcessSpecification>

```

APPENDICE C: CPP di default

Istanza XML nella versione in lingua italiana.

```
<?xml version="1.0"?>
<tp:CollaborationProtocolProfile
  xmlns:tp="http://www.moda-ml.net/moda-ml/repository/cpp/editor/dtdxslt/modaml_cpp-
cpa-2_0b.xsd"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.moda-ml.net/moda-
ml/repository/cpp/editor/dtdxslt/modaml_cpp-cpa-2_0b.xsd
  modaml_cpp-cpa-2_0b.xsd"
  xmlns:ds="http://www.w3.org/TR/xmldsig-core/xmldsig-core-schema.xsd"
  xmlns:xlink="http://www.w3.org/1999/xlink"
  tp:cppid="IT0987654321" tp:version="2_0b">
  <tp:PartyInfo
    tp:partyName="Azienda Mario Bianchi"
    tp:defaultMshChannelId="SMTP"
    tp:defaultMshPackageId="MSH_Handler">
    <tp:PartyId>IT0987654321</tp:PartyId>
    <tp:PartyRef xlink:href="http://www.moda-ml.net/moda-
ml/repository/cpp/editor/genericuser/about.html" />
    <tp:CollaborationRole>
      <tp:ProcessSpecification
        tp:version="1.1"
        tp:name="Produzionetessuto"
        xlink:type="simple"
        xlink:href="http://www.moda-ml.net/moda-ml/repository/bpss/v2003-1/it/v2003-
1_Produzionetessuto.xml"
        tp:uuid="v2003-1_Proc-3" />
      <tp:Role
        tp:name="Fornitore di tessuti"
        xlink:type="simple"
        xlink:href="http://www.moda-ml.net/moda-ml/repository/bpss/v2003-1/it/v2003-
1_Produzionetessuto.xml#Fornitore di tessuti" />
      <tp:ServiceBinding>
        <tp:Service>v2003-1_Proc-3</tp:Service>
        <tp:CanSend>
          <tp:ThisPartyActionBinding
            tp:id="OF51-013"
            tp:action="Disposizione rammendo tessuto"
            tp:packageId="p-OF51-013">
            <tp:BusinessTransactionCharacteristics
              tp:isNonRepudiationRequired="true"
              tp:isNonRepudiationReceiptRequired="true"
              tp:isConfidential="transient"
              tp:isAuthenticated="persistent"
              tp:isTamperProof="persistent"
              tp:isAuthorizationRequired="true"
              tp:timeToAcknowledgeReceipt="PT2H"
              tp:timeToPerform="PLD" />
            <tp:ActionContext
              tp:binaryCollaboration="Rammendo tessuto"
              tp:businessTransactionActivity="Disposizione rammendo tessuto"
              tp:requestOrResponseAction="Disposizione rammendo tessuto" />
            <tp:ChannelId>SMTP</tp:ChannelId>
          </tp:ThisPartyActionBinding>
        </tp:CanSend>
        <tp:CanReceive>
          <tp:ThisPartyActionBinding
            tp:id="OF51-014"
            tp:action="Ritorno Rammendo Tessuto"
            tp:packageId="p-OF51-014">
            <tp:BusinessTransactionCharacteristics
              tp:isNonRepudiationRequired="true"
              tp:isNonRepudiationReceiptRequired="true"
              tp:isConfidential="transient"
              tp:isAuthenticated="persistent"
              tp:isTamperProof="persistent"
              tp:isAuthorizationRequired="true"
              tp:timeToAcknowledgeReceipt="PT5M" />
          </tp:CanReceive>
        </tp:ServiceBinding>
    </tp:PartyRef>
  </tp:PartyInfo>
</tp:CollaborationRole>
</tp:ProcessSpecification>
</tp:Role>
</tp:ServiceBinding>
</tp:CanSend>
</tp:CanReceive>
</tp:PartyInfo>
</tp:CollaborationProfile>
```

```

        <tp:ActionContext
            tp:binaryCollaboration="Rammendo tessuto"
            tp:businessTransactionActivity="Ritorno Rammendo Tessuto"
            tp:requestOrResponseAction="Ritorno Rammendo Tessuto"/>
        <tp:ChannelId>FAX</tp:ChannelId>
    </tp:ThisPartyActionBinding>
</tp:CanReceive>
</tp:ServiceBinding>
</tp:CollaborationRole>
<!-- one or more -->
<tp:DeliveryChannel
    tp:channelId="SMTP"
    tp:transportId="transportSMTP"
    tp:docExchangeId="docEx">
    <tp:MessagingCharacteristics
        tp:syncReplyMode="none"
        tp:ackRequested="perMessage"
        tp:ackSignatureRequested="perMessage" />
</tp:DeliveryChannel>
<tp:DeliveryChannel
    tp:channelId="SMTP2"
    tp:transportId="transportSMTP2"
    tp:docExchangeId="docEx">
    <tp:MessagingCharacteristics
        tp:syncReplyMode="none"
        tp:ackRequested="perMessage"
        tp:ackSignatureRequested="perMessage" />
</tp:DeliveryChannel>
<tp:DeliveryChannel
    tp:channelId="FTP"
    tp:transportId="transportFTP"
    tp:docExchangeId="docEx">
    <tp:MessagingCharacteristics
        tp:syncReplyMode="none"
        tp:ackRequested="perMessage"
        tp:ackSignatureRequested="perMessage" />
</tp:DeliveryChannel>
<tp:DeliveryChannel
    tp:channelId="FTP2"
    tp:transportId="transportFTP2"
    tp:docExchangeId="docEx">
    <tp:MessagingCharacteristics
        tp:syncReplyMode="none"
        tp:ackRequested="perMessage"
        tp:ackSignatureRequested="perMessage" />
</tp:DeliveryChannel>
<tp:DeliveryChannel
    tp:channelId="HTTP"
    tp:transportId="transportHTTP"
    tp:docExchangeId="docEx">
    <tp:MessagingCharacteristics
        tp:syncReplyMode="none"
        tp:ackRequested="perMessage"
        tp:ackSignatureRequested="perMessage" />
</tp:DeliveryChannel>
<tp:DeliveryChannel
    tp:channelId="FAX"
    tp:transportId="transportFAX"
    tp:docExchangeId="docEx">
    <tp:MessagingCharacteristics
        tp:syncReplyMode="none"
        tp:ackRequested="perMessage"
        tp:ackSignatureRequested="perMessage" />
</tp:DeliveryChannel>
<!-- one or more -->
<tp:Transport tp:transportId="transportSMTP">
    <tp:TransportSender>
        <tp:TransportProtocol tp:version="1.0">SMTP</tp:TransportProtocol>
    </tp:TransportSender>
    <tp:TransportReceiver>
        <tp:TransportProtocol tp:version="1.0">SMTP</tp:TransportProtocol>
        <tp:Endpoint tp:uri="mailto:genericuser@moda-ml.org" tp:type="allPurpose"/>
    </tp:TransportReceiver>
</tp:Transport>

```

```

<tp:Transport tp:transportId="transportSMTP2">
  <tp:TransportSender>
    <tp:TransportProtocol tp:version="1.0">SMTP</tp:TransportProtocol>
  </tp:TransportSender>
  <tp:TransportReceiver>
    <tp:TransportProtocol tp:version="1.0">SMTP</tp:TransportProtocol>
    <tp:Endpoint tp:uri="mailto:genericuser@moda-ml.org" tp:type="allPurpose"/>
  </tp:TransportReceiver>
</tp:Transport>
<tp:Transport tp:transportId="transportFTP">
  <tp:TransportSender>
    <tp:TransportProtocol tp:version="1.0">FTP</tp:TransportProtocol>
  </tp:TransportSender>
  <tp:TransportReceiver>
    <tp:TransportProtocol tp:version="1.0">FTP</tp:TransportProtocol>
    <tp:Endpoint tp:uri="ftp://genericuser.it" tp:type="allPurpose"/>
  </tp:TransportReceiver>
</tp:Transport>
<tp:Transport tp:transportId="transportFTP2">
  <tp:TransportSender>
    <tp:TransportProtocol tp:version="1.0">FTP</tp:TransportProtocol>
  </tp:TransportSender>
  <tp:TransportReceiver>
    <tp:TransportProtocol tp:version="1.0">FTP</tp:TransportProtocol>
    <tp:Endpoint tp:uri="ftp://genericuser.it" tp:type="allPurpose"/>
  </tp:TransportReceiver>
</tp:Transport>
<tp:Transport tp:transportId="transportHTTP">
  <tp:TransportSender>
    <tp:TransportProtocol tp:version="1.0">HTTP</tp:TransportProtocol>
  </tp:TransportSender>
  <tp:TransportReceiver>
    <tp:TransportProtocol tp:version="1.0">HTTP</tp:TransportProtocol>
    <tp:Endpoint tp:uri="http://genericuser.it" tp:type="allPurpose"/>
  </tp:TransportReceiver>
</tp:Transport>
<tp:Transport tp:transportId="transportFAX">
  <tp:TransportSender>
    <tp:TransportProtocol tp:version="1.0">FAX</tp:TransportProtocol>
  </tp:TransportSender>
  <tp:TransportReceiver>
    <tp:TransportProtocol tp:version="1.0">FAX</tp:TransportProtocol>
    <tp:Endpoint tp:uri="fax:000000000" tp:type="allPurpose"/>
  </tp:TransportReceiver>
</tp:Transport>
<!-- one or more -->
<tp:DocExchange tp:docExchangeId="docEx"/>
</tp:PartyInfo>
<!-- Schema per validare la busta ebXML e un SimplePart per ogni messaggio -->
<tp:SimplePart tp:id="MSH_ErrorHdr" tp:mimetype="text/xml">
  <tp:NamespaceSupported
    tp:location="http://www.moda-ml.net/moda-
ml/repository/soap/modamlschema2002.xsd"
    tp:version="1.0">
    http://www.moda-ml.net/moda-ml/repository/soap/modamlschema2002.xsd
  </tp:NamespaceSupported>
  <tp:NamespaceSupported
    tp:location="http://ebxml.org/project_teams/transport/xmldsig-core-schema.xsd"
    tp:version="0.1">
    http://www.w3.org/2000/09/xmldsig#
  </tp:NamespaceSupported>
</tp:SimplePart>
<tp:SimplePart tp:id="TEXDarnOrder" tp:mimetype="text/xml">
  <tp:NamespaceSupported
    tp:location="http://www.moda-ml.net/moda-ml/repository/schema/v2003-
1/TEXDarnOrder.xsd"
    tp:version="v2003-1">
    http://www.moda-ml.net/moda-ml/repository/schema/v2003-1/TEXDarnOrder.xsd
  </tp:NamespaceSupported>
</tp:SimplePart>
<tp:SimplePart tp:id="TEXDarnReturn" tp:mimetype="text/xml">
  <tp:NamespaceSupported
    tp:location="http://www.moda-ml.net/moda-ml/repository/schema/v2003-
1/TEXDarnReturn.xsd"

```

```

        tp:version="v2003-1">
            http://www.moda-ml.net/moda-ml/repository/schema/v2003-1/TEXDarnReturn.xsd
        </tp:NamespaceSupported>
    </tp:SimplePart>
    <!-- Un Packaging per il SOAP Envelope e uno per ogni messaggio -->
    <tp:Packaging tp:id="MSH_Handler">
        <tp:ProcessingCapabilities tp:parse="true" tp:generate="true"/>
        <tp:CompositeList>
            <tp:Composite
                tp:id="Envelope"
                tp:mimetype="multipart/related"
                tp:mimeparameters="type=text/xml">
                <tp:Constituent tp:idref="MSH_ErrorHdr"/>
            </tp:Composite>
        </tp:CompositeList>
    </tp:Packaging>
    <tp:Packaging tp:id="p-OF51-013">
        <tp:ProcessingCapabilities tp:parse="true" tp:generate="true"/>
        <tp:CompositeList>
            <tp:Composite
                tp:id="Env-OF51-013"
                tp:mimetype="multipart/related"
                tp:mimeparameters="type=text/xml">
                <tp:Constituent tp:idref="MSH_ErrorHdr"/>
                <tp:Constituent tp:idref="TEXDarnOrder"/>
            </tp:Composite>
        </tp:CompositeList>
    </tp:Packaging>
    <tp:Packaging tp:id="p-OF51-014">
        <tp:ProcessingCapabilities tp:parse="true" tp:generate="true"/>
        <tp:CompositeList>
            <tp:Composite
                tp:id="Env-OF51-014"
                tp:mimetype="multipart/related"
                tp:mimeparameters="type=text/xml">
                <tp:Constituent tp:idref="MSH_ErrorHdr"/>
                <tp:Constituent tp:idref="TEXDarnReturn"/>
            </tp:Composite>
        </tp:CompositeList>
    </tp:Packaging>
    <!-- Estensione al CPP -->
    <tp:DocumentOptionalElements tp:partyId="IT0987654321" tp:bpssuuid="v2003-1_Proc-3">
        <tp:Doc tp:name="Disposizione rammendo tessuto" tp:position="0"/>
        <tp:Doc tp:name="Ritorno Rammendo Tessuto" tp:position="1"/>
    </tp:DocumentOptionalElements>
</tp:CollaborationProtocolProfile>

```

APPENDICE D: CPP relativo al BPSS “Fornitura Tessuti”

Istanza XML di un CPP creato con MODA-ML CPP-Editor.

```
<?xml version="1.0"?>
<tp:CollaborationProtocolProfile xmlns:tp="http://www.moda-ml.net/moda-
ml/repository/cpp/editor/dtdxslt/modaml_cpp-cpa-2_0b.xsd"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.moda-ml.net/moda-
ml/repository/cpp/editor/dtdxslt/modaml_cpp-cpa-2_0b.xsd
modaml_cpp-cpa-2_0b.xsd" xmlns:ds="http://www.w3.org/TR/xmldsig-
core/xmldsig-core-schema.xsd" xmlns:xlink="http://www.w3.org/1999/xlink"
tp:cppid="IT0987654321" tp:version="2_0b">
  <tp:PartyInfo tp:partyName="Azienda Cristiano Novelli"
tp:defaultMshChannelId="SMTP" tp:defaultMshPackageId="MSH_Handler">
    <tp:PartyId>IT0987654321</tp:PartyId>
    <tp:PartyRef xlink:href="http://www.miosito.com/about.html"/>
    <tp:CollaborationRole>
      <tp:ProcessSpecification tp:version="1.1"
tp:name="Fornituratessuti" xlink:type="simple" xlink:href="http://www.moda-ml.net/moda-
ml/repository/bpss/v2003-1/it/v2003-1_Fornituratessuti.xml" tp:uuid="v2003-1_Proc-1"/>
        <tp:Role tp:name="Fornitore di tessuti" xlink:type="simple"
xlink:href="http://www.moda-ml.net/moda-ml/repository/bpss/v2003-1/it/v2003-
1_Fornituratessuti.xml#Fornitore di tessuti"/>
          <tp:ServiceBinding>
            <tp:Service>v2003-1_Proc-1</tp:Service>
            <tp:CanSend>
              <tp:ThisPartyActionBinding tp:id="OF51-002"
tp:action="Risposta ordine del fornitore di tessuti" tp:packageId="p-OF51-002">
                <tp:BusinessTransactionCharacteristics
tp:isNonRepudiationRequired="true" tp:isNonRepudiationReceiptRequired="true"
tp:isConfidential="transient" tp:isAuthenticated="persistent"
tp:isTamperProof="persistent" tp:isAuthorizationRequired="true"
tp:timeToAcknowledgeReceipt="PT2H" tp:timeToPerform="P1D"/>
                  <tp:ActionContext
tp:binaryCollaboration="Acquisto tessuti" tp:businessTransactionActivity="Risposta
ordine del fornitore di tessuti" tp:requestOrResponseAction="Risposta ordine del
fornitore di tessuti"/>
                    <tp:ChannelId>SMTP</tp:ChannelId>
                  </tp:ThisPartyActionBinding>
                </tp:CanSend>
              <tp:CanReceive>
                <tp:ThisPartyActionBinding tp:id="OF51-001"
tp:action="Ordine al fornitore di tessuti" tp:packageId="p-OF51-001">
                  <tp:BusinessTransactionCharacteristics
tp:isNonRepudiationRequired="true" tp:isNonRepudiationReceiptRequired="true"
tp:isConfidential="transient" tp:isAuthenticated="persistent"
tp:isTamperProof="persistent" tp:isAuthorizationRequired="true"
tp:timeToAcknowledgeReceipt="PT5M"/>
                    <tp:ActionContext
tp:binaryCollaboration="Acquisto tessuti" tp:businessTransactionActivity="Ordine al
fornitore di tessuti" tp:requestOrResponseAction="Ordine al fornitore di tessuti"/>
                    <tp:ChannelId>SMTP</tp:ChannelId>
                  </tp:ThisPartyActionBinding>
                </tp:CanReceive>
              </tp:ServiceBinding>
            </tp:CollaborationRole>
          <!-- one or more -->
          <tp:DeliveryChannel tp:channelId="SMTP" tp:transportId="transportSMTP"
tp:docExchangeId="docEx">
            <tp:MessagingCharacteristics tp:syncReplyMode="none"
tp:ackRequested="perMessage" tp:ackSignatureRequested="perMessage"/>
          </tp:DeliveryChannel>
          <tp:DeliveryChannel tp:channelId="SMTP2" tp:transportId="transportSMTP2"
tp:docExchangeId="docEx">
            <tp:MessagingCharacteristics tp:syncReplyMode="none"
tp:ackRequested="perMessage" tp:ackSignatureRequested="perMessage"/>
          </tp:DeliveryChannel>
        </tp:PartyInfo>
      </tp:CollaborationProtocolProfile>
    </tp:PartyInfo>
  </tp:PartyInfo>
</tp:CollaborationProtocolProfile>
```

```

        <tp:DeliveryChannel tp:channelId="FTP" tp:transportId="transportFTP"
tp:docExchangeId="docEx">
        <tp:MessagingCharacteristics tp:syncReplyMode="none"
tp:ackRequested="perMessage" tp:ackSignatureRequested="perMessage"/>
    </tp:DeliveryChannel>
    <tp:DeliveryChannel tp:channelId="FTP2" tp:transportId="transportFTP2"
tp:docExchangeId="docEx">
        <tp:MessagingCharacteristics tp:syncReplyMode="none"
tp:ackRequested="perMessage" tp:ackSignatureRequested="perMessage"/>
    </tp:DeliveryChannel>
    <tp:DeliveryChannel tp:channelId="HTTP" tp:transportId="transportHTTP"
tp:docExchangeId="docEx">
        <tp:MessagingCharacteristics tp:syncReplyMode="none"
tp:ackRequested="perMessage" tp:ackSignatureRequested="perMessage"/>
    </tp:DeliveryChannel>
    <tp:DeliveryChannel tp:channelId="FAX" tp:transportId="transportFAX"
tp:docExchangeId="docEx">
        <tp:MessagingCharacteristics tp:syncReplyMode="none"
tp:ackRequested="perMessage" tp:ackSignatureRequested="perMessage"/>
    </tp:DeliveryChannel>
    <!-- one or more -->
    <tp:Transport tp:transportId="transportSMTP">
        <tp:TransportSender>
            <tp:TransportProtocol
tp:version="1.0">SMTP</tp:TransportProtocol>
            </tp:TransportSender>
        <tp:TransportReceiver>
            <tp:TransportProtocol
tp:version="1.0">SMTP</tp:TransportProtocol>
            <tp:Endpoint
tp:uri="mailto:cristianonovelli@miaazienda.org" tp:type="allPurpose"/>
            </tp:TransportReceiver>
        </tp:Transport>
    <tp:Transport tp:transportId="transportSMTP2">
        <tp:TransportSender>
            <tp:TransportProtocol
tp:version="1.0">SMTP</tp:TransportProtocol>
            </tp:TransportSender>
        <tp:TransportReceiver>
            <tp:TransportProtocol
tp:version="1.0">SMTP</tp:TransportProtocol>
            <tp:Endpoint tp:uri="genericuser@moda-ml.org"
tp:type="allPurpose"/>
            </tp:TransportReceiver>
        </tp:Transport>
    <tp:Transport tp:transportId="transportFTP">
        <tp:TransportSender>
            <tp:TransportProtocol
tp:version="1.0">FTP</tp:TransportProtocol>
            </tp:TransportSender>
        <tp:TransportReceiver>
            <tp:TransportProtocol
tp:version="1.0">FTP</tp:TransportProtocol>
            <tp:Endpoint tp:uri="ftp://genericuser.it"
tp:type="allPurpose"/>
            </tp:TransportReceiver>
        </tp:Transport>
    <tp:Transport tp:transportId="transportFTP2">
        <tp:TransportSender>
            <tp:TransportProtocol
tp:version="1.0">FTP</tp:TransportProtocol>
            </tp:TransportSender>
        <tp:TransportReceiver>
            <tp:TransportProtocol
tp:version="1.0">FTP</tp:TransportProtocol>
            <tp:Endpoint tp:uri="ftp://genericuser.it"
tp:type="allPurpose"/>
            </tp:TransportReceiver>
        </tp:Transport>
    <tp:Transport tp:transportId="transportHTTP">
        <tp:TransportSender>
            <tp:TransportProtocol
tp:version="1.0">HTTP</tp:TransportProtocol>
            </tp:TransportSender>

```

```

        <tp:TransportReceiver>
            <tp:TransportProtocol
tp:version="1.0">HTTP</tp:TransportProtocol>
            <tp:Endpoint tp:uri="http://genericuser.it"
tp:type="allPurpose"/>
        </tp:TransportReceiver>
    </tp:Transport>
    <tp:Transport tp:transportId="transportFAX">
        <tp:TransportSender>
            <tp:TransportProtocol
tp:version="1.0">FAX</tp:TransportProtocol>
        </tp:TransportSender>
        <tp:TransportReceiver>
            <tp:TransportProtocol
tp:version="1.0">FAX</tp:TransportProtocol>
            <tp:Endpoint tp:uri="fax:000000000" tp:type="allPurpose"/>
        </tp:TransportReceiver>
    </tp:Transport>
    <!-- one or more -->
    <tp:DocExchange tp:docExchangeId="docEx"/>
</tp:PartyInfo>
<!-- Schema per validare la busta ebXML e un SimplePart per ogni messaggio -->
<tp:SimplePart tp:id="MSH_ErrorHdr" tp:mimetype="text/xml">
    <tp:NamespaceSupported tp:location="http://www.moda-ml.net/moda-
ml/repository/soap/modamlschema2002.xsd" tp:version="1.0">
        http://www.moda-ml.net/moda-ml/repository/soap/modamlschema2002.xsd
    </tp:NamespaceSupported>
    <tp:NamespaceSupported
tp:location="http://ebxml.org/project_teams/transport/xmldsig-core-schema.xsd"
tp:version="0.1">
        http://www.w3.org/2000/09/xmldsig#
    </tp:NamespaceSupported>
    </tp:SimplePart>
    <tp:SimplePart tp:id="TEXOrder" tp:mimetype="text/xml">
        <tp:NamespaceSupported tp:location="http://www.moda-ml.net/moda-
ml/repository/schema/v2003-1/TEXOrder.xsd" tp:version="1.0">http://www.moda-
ml.net/moda-ml/repository/schema/v2003-1/TEXOrder.xsd</tp:NamespaceSupported>
        <tp:NamespaceSupported
tp:location="http://ebxml.org/project_teams/transport/xmldsig-core-schema.xsd"
tp:version="0.1">
            http://www.w3.org/2000/09/xmldsig#
        </tp:NamespaceSupported>
    </tp:SimplePart>
    <tp:SimplePart tp:id="TEXOrdResponse" tp:mimetype="text/xml">
        <tp:NamespaceSupported tp:location="http://www.moda-ml.net/moda-
ml/repository/schema/v2003-1/TEXOrdResponse.xsd" tp:version="1.0">http://www.moda-
ml.net/moda-ml/repository/schema/v2003-1/TEXOrdResponse.xsd</tp:NamespaceSupported>
        <tp:NamespaceSupported
tp:location="http://ebxml.org/project_teams/transport/xmldsig-core-schema.xsd"
tp:version="0.1">
            http://www.w3.org/2000/09/xmldsig#
        </tp:NamespaceSupported>
    </tp:SimplePart>
    <!-- Un Packaging per il SOAP Envelope e uno per ogni messaggio -->
    <tp:Packaging tp:id="MSH_Handler">
        <tp:ProcessingCapabilities tp:parse="true" tp:generate="true"/>
        <tp:CompositeList>
            <tp:Composite tp:id="Envelope" tp:mimetype="multipart/related"
tp:mimeparameters="type=text/xml">
                <tp:Constituent tp:idref="MSH_ErrorHdr"/>
            </tp:Composite>
        </tp:CompositeList>
    </tp:Packaging>
    <tp:Packaging tp:id="p-OF51-001">
        <tp:ProcessingCapabilities tp:parse="true" tp:generate="true"/>
        <tp:CompositeList>
            <tp:Composite tp:id="Env-OF51-001"
tp:mimetype="multipart/related" tp:mimeparameters="type=text/xml">
                <tp:Constituent tp:idref="MSH_ErrorHdr"/>
                <tp:Constituent tp:idref="TEXOrder"/>
            </tp:Composite>
        </tp:CompositeList>
    </tp:Packaging>
    <tp:Packaging tp:id="p-OF51-002">

```

```

        <tp:ProcessingCapabilities tp:parse="true" tp:generate="true"/>
        <tp:CompositeList>
            <tp:Composite tp:id="Env-OF51-002"
tp:mimetype="multipart/related" tp:mimeparameters="type=text/xml">
                <tp:Constituent tp:idref="MSH_ErrorHdr"/>
                <tp:Constituent tp:idref="TEXOrdResponse"/>
            </tp:Composite>
        </tp:CompositeList>
    </tp:Packaging>
    <!-- Estensione al CPP -->
    <tp:DocumentOptionalElements
        tp:partyId="IT0987654321"
        tp:bpssuuid="v2003-1_Proc-1">
        <tp:Doc
            tp:name="Ordine al fornitore di tessuti"
            tp:position="3">
            <tp:Entity
                tp:name="msgfunction"
                tp:count="0"
                tp:state="Required" tp:xpath="TEXOrder/@msgfunction"/>
            <tp:Entity
                tp:name="logo"
                tp:count="1"
                tp:state="Rejected"
                tp:xpath="TEXOrder/TOheader/supplier/@logo"/>
        </tp:Doc>
        <tp:Doc
            tp:name="Risposta ordine del fornitore di tessuti"
            tp:position="4">
            <tp:Entity
                tp:name="msgfunction"
                tp:count="0"
                tp:state="Required"
                tp:xpath="TEXOrdResponse/@msgfunction"/>
        </tp:Doc>
    </tp:DocumentOptionalElements>
</tp:CollaborationProtocolProfile>

```

BIBLIOGRAFIA

[CENISS02] “CEN/ISSS Electronic Commerce Workshop”, 2002,
<http://www.eeurope-standards.org/Docs/ECommerce2002.pdf>

[CPNETa] Novelli C., MODA-ML BPSS-Generator Application, 2004,
<http://www.moda-ml.net/moda-ml/repository/bpss/generator/>

[CPNETb] Novelli C., MODA-ML CPP-Editor Application, 2004,
<http://www.moda-ml.net/moda-ml/repository/cpp/editor/>

[CPNETc] XML Schema ebXML CPP e CPA esteso per il progetto MODA-ML,
Download: http://www.moda-ml.net/moda-ml/repository/cpp/editor/dtdxslt/modaml_cpp-cpa-2_0b.xsd

[DOM98] Apparao V., Byrne S., Champion M., Isaacs S., Jacobs I., Le Hors A., Nicol G., Robie J., Sutor R., Wilson C., Wood L., “W3C Recommendation: Document Object Model (DOM) Level 1 Specification”, 1998, <http://www.w3.org/TR/REC-DOM-Level-1/>

[ebCC] ebXML Core Components, “Core Component Overview v1.05”, 2001,
<http://www.ebxml.org/specs/ccOVER.pdf>

[ebBPSSa] Business Process Project Team,
“Business Process Specification Schema v.1.01”, 2001,
<http://www.ebxml.org/specs/ebBPSS.pdf>

[ebBPSSdtd]
DTD ebXML BPSS (da ebxml.org)
Download: <http://www.ebxml.org/specs/ebBPSS.dtd>

[ebBPSSxml]
Esempio di istanza di un ebXML BPSS (da ebxml.org)
Download: <http://www.ebxml.org/specs/ebBPSS.xml>

[ebBPSSxsd]
XML-Schema ebXML BPSS (da ebxml.org)
Download: <http://www.ebxml.org/specs/ebBPSS.xsd>

[ebCPP] OASIS ebXML Collaboration Protocol Profile and Agreement Technical Committee, “Collaboration-Protocol Profile and Agreement Specification v2.0”, 2002,
Download: <http://www.oasis-open.org/committees/download.php/224/ebcpp-2.0c.doc>

[ebCPPxml]

Istanza ebXML CPP (da oasis-open.org)

Download: http://www.oasis-open.org/committees/download.php/238/cpp-example-companyA-2_0b.xml

[ebCPAxml]

Istanza ebXML CPA (da oasis-open.org)

Download: http://www.oasis-open.org/committees/download.php/253/cpa-example-2_0b.xml

[ebCPPAxsd]

XML Schema ebXML CPP e CPA (da oasis-open.org)

Download: http://www.oasis-open.org/committees/download.php/240/cpp-cpa-2_0b.xsd

[ebGLOSS] Technical Architecture Team, “ebXML Glossary Version 0.99“, 2001, <http://www.ebxml.org/specs/ebGLOSS.pdf>

[ebMS] OASIS ebXML Messaging Services Technical Committee, “Message Service Specification Version 2.0”, 2002, <http://www.ebxml.org/specs/ebMS2.pdf>

[ebRS] OASIS/ebXML Registry Technical Committee, “OASIS/ebXML Registry Services Specification v2.0”, 2001, <http://www.ebxml.org/specs/ebrs2.pdf>

[ebTA01] ebXML Technical Architecture Project Team, “ebXML Technical Architecture Specification Version 1.0.4”, 2001, <http://www.ebxml.org/specs/ebTA.pdf>

[ebXML] ebXML (Electronic Business using eXtensible Markup Language), “Enabling A Global Electronic Market”, <http://www.ebxml.org/>

[ebXMLa] ebXML SPECS, <http://www.ebxml.org/specs/>

[ebXMLb] CEN/ISSS W/S eBES, “A proposito di ebXML”, http://www.ebxml.eu.org/It/cosa_e'ebXML.htm

[ebXMLibm] Mertz D., “Understanding ebXML”, <http://www-106.ibm.com/developerworks/xml/library/x-ebxml/>

[EDIFACT90] UNECE, “ISO 9735 - Electronic data interchange for administration, commerce and transport (EDIFACT) - Application level syntax rules”, 1990, http://www.unece.org/trade/untdid/texts/d422_d.htm

[EURATEX] European Apparel and Textile Organization, <http://www.euratex.org/>

[FTP85] Postel, J., Reynolds, J., “RFC 959 - File Transfer Protocol (FTP)”, 1985, <http://www.ietf.org/rfc/rfc0959.txt>

[GNBV04] Gessa N., Novelli C., Busuoli M., Vitali F., “Use and extension of ebXML business profiles for Textile/Clothing firms”, 2004

[HTTP99] Fielding, R., Gettys, J., Mogul, J., Frystyk, H., Masinter, L., Leach, P. and T. Berners-Lee, “RFC 2616 - Hypertext Transfer Protocol, HTTP/1.1”, 1999, <http://www.ietf.org/rfc/rfc2616.txt>

[IST] Informations Society Technologies, “Computing, communications and networks take-up measures”, <http://www.cordis.lu/ist/>

[MML]

MODA-ML, Middleware tOols and Documents to enhAnce the textile/clothing supply chain through xML, <http://www.moda-ml.org>

[MMLg] “MODA-ML 2003-1, Anello tessitore confezionista della filiera, messaggi scambiati”, <http://www.moda-ml.org/moda-ml/imple/SupportedDocuments2003-1.asp?nomenu=1&lingua=it>

[OASIS] OASIS, “Organization for the Advancement of Structured Information Standards”, <http://www.oasis-open.org/home/index.php>

[RNET99] “RosettaNet Implementation Framework Specification Version 1.0”, 1999, <http://www.rosettanet.org>

[SMTP82]

Postel, J. B., “RFC 821 - Simple Mail Transfer Protocol”, 1982, <http://www.ietf.org/rfc/rfc0821.txt>

[SOAP00] Box, D., Ehnebuske, D., Kakivaya, G., Layman, A., Frystyk Nielsen, H., Thatte, S., Mendelsohn, N., Winer, D., “W3C Note Simple Object Access Protocol (SOAP) v1.1”, 2000, <http://www.w3.org/TR/SOAP>

[SQL92] ANSI, "ANSI X3.135-1992 - Database Language SQL", 1992

[TEXPIN] Textile Supply Chain Integrated Network, “CEN/ISSS Workshop su TEXTILE SUPPLY CHAIN INTEGRATED NETWORK (TEX.SPIN)”, <http://www.moda-ml.org/moda-ml/imple/texspin.asp?lingua=it>

[UDDI01] McKee, B., Ehnebuske, D., Rogers, D., “UDDI Version 2.0 API Specification”, 2001, <http://www.uddi.org/pubs/ProgrammersAPI-V2.00-Open-20010608.pdf>

[UML01] OMG, “Unified Modeling Language Specification”, 2001, <ftp://ftp.omg.org/pub/docs/formal/01-09-67.pdf>

[UMM01] Techniques and Methodologies Working Group, “UN/CEFACT Modeling Methodology”, 2001, <http://www.unece.org/cefact/docum/download/01cf26.pdf>

[UNCEFACT] United Nations Centre for Trade Facilitation and Electronic Business,
<http://www.unece.org/cefact/index.htm>

[X12] Accredited Standards Committee X12, “ASCX12 Home Page”,
<http://www.x12.org/>

[XML04] Bray T., Paoli J., C. M. Sperberg-McQueen, Maler E., Yergeau F.,
“Extensible Markup Language (XML) 1.0 (Third Edition)”, 2004,
<http://www.w3.org/TR/REC-xml/>

[XPath99] Clark, J., DeRose, S., “W3C Recommendation XML Path Language
(XPath) Version 1.0”, 1999, <http://www.w3.org/TR/1999/REC-xpath-19991116>

Ringraziamenti

Voglio ringraziare il prof. Fabio Vitali per avermi dato la possibilità di svolgere questa tesi e per la sua disponibilità.

Ringrazio Piero De Sabbata e Nicola Gessa per l'infinita pazienza, disponibilità e gentilezza che mi hanno sempre dimostrato durante lo sviluppo di questo progetto e gli amici del X-Lab dell'Enea per il supporto ricevuto.

Un ringraziamento particolare va ai miei genitori, per essermi sempre stati vicini e per la fiducia che hanno sempre riposto in me.

