

Università degli Studi di Modena e Reggio Emilia

Facoltà di ingegneria di Modena

Corso di Laurea in Ingegneria Informatica

*Progetto e sviluppo del
wrapper EDIFACT
per il sistema MOMIS*

Relatore:
Chiar.mo Prof. Sonia Bergamaschi

Candidato:
Zhu Song

Correlatore:
Chiar.mo Prof. Domenico Beneventano

Anno Accademico 2006/2007

Parole chiavi:
MOMIS
wrapper
EDIFACT
sql

Indice

<u>INTRODUZIONE</u>	<u>9</u>
<u>PRESENTAZIONE EDIFACT</u>	<u>13</u>
1.1 VANTAGGI DELL'EDI	14
1.2 SVANTAGGI DELL'EDI	15
1.3 SISTEMI DI TRASMISSIONE DELL'EDI	15
1.4 VERSIONI DELL'EDIFACT	16
<u>IL FORMATO EDIFACT</u>	<u>17</u>
2.1 SEGMENTI.....	17
2.1.1 TAG	19
2.1.2 ELEMENTI DATI	19
2.1.3 OMISSIONI E TRONCAMENTI.....	20
2.1.4 RIPETIZIONI DEGLI ELEMENTI DATI.....	20
2.2 MESSAGGI	20
2.2.1 RIPETIZIONI DEI SEGMENTI.....	28
2.2.2 GERARCHIA DEI SEGMENTI	28
2.3 LA STRUTTURA DEL FILE.....	31
2.3.1 INTERCHANGE	31
2.3.2 GRUPPO FUNZIONALE.....	32
<u>IL WRAPPER EDIFACT</u>	<u>33</u>
3.1 ESTRATTORE DATI EDIFACT	34
3.1.1 SCHEMA DEL DATABASE	34
3.1.1.1 Documentazione degli schemi relazionali per estrattore di dati EDIFACT	36
3.1.1.2 Schema logico relazionale del database EDIFACT	39
3.1.1.3 Le istruzioni sql 92 per la creazione del database EDIFACT	40
3.1.2 ESTRATTORE DATI EDIFACT: DESCRIZIONE DETTAGLIATA	42
3.1.2.1 Interfaccia grafica dell'estrattore di dati EDIFACT	44
3.1.2.2 Esempi di estrazione	44
3.1.2.3 Esempi di query sql.....	47
3.2 LE VISTE	51
3.2.1 VISTA MESSAGGIO_ORDINE	52
3.2.2 VISTA CLIENTE_ORDINE.....	53
3.2.3 VISTA DETTAGLIO_ORDINE.....	59
3.2.4 VISTA PREZZI_ORDINE	61
<u>CONCLUSIONE</u>	<u>63</u>

Elenco delle figure

FIGURA 1: ARCHITETTURA DEL SISTEMA MOMIS	10
FIGURA 2: EVOLUZIONE DEGLI STANDARD EDI	14
FIGURA 3: STRUTTURA DI UN FILE EDIFACT	31
FIGURA 4: STRUTTURA DEL WRAPPER EDIFACT	33
FIGURA 5: SCHEMA E/R DEL DATABASE EDIFACT	35
FIGURA 6: CLASSI DEL WRAPPER EDIFACT	1
FIGURA 7: INTERFACCIA DEL WRAPPER EDIFACT	44
FIGURA 8: RISULTATO DELL'ESEMPIO QUERY N1	48
FIGURA 9: RISULTATO DELL'ESEMPIO QUERY N2	49
FIGURA 10: RISULTATO DELL'ESEMPIO QUERY N3	50
FIGURA 11: RISULTATO DELL'ESEMPIO QUERY N4	50
FIGURA 12: RISULTATO DELL'ESEMPIO QUERY N5	51
FIGURA 13: RISULTATO DELLA VISTA MESSAGGIO_ORDINE	53
FIGURA 14: RISULTATO DELLA VISTA CLIENTE_ORDINE	59
FIGURA 15: RISULTATO DELLA VISTA DETTAGLIO_ORDINE.....	61
FIGURA 16: RISULTATO DELLA VISTA PREZZI_ORDINE.....	61

Introduzione

Con lo sviluppo e la diffusione delle tecnologie informatiche e delle telecomunicazioni, nasce il business elettronico, il quale sfrutta le tecnologie di comunicazioni elettronici e di elaborazione per rendere più efficienti le imprese, perciò vengono sviluppati i primi sistemi di comunicazioni elettronici per rendere più efficienti le comunicazioni tra le varie imprese.

In questa tesi si pone l'obbiettivo di studiare uno dei sistemi di comunicazioni di documenti elettronici, sistemi EDI acronimo di Elettronic Data Interchange, e soprattutto uno dei formati di file EDI, lo standard EDIFACT acronimo di EDI For Administration, Commerce and Transport, al fine di realizzare un wrapper per il progetto MOMIS, Mediator EnvirOnment for Multiple Information Sources.

Il sistema MOMIS è un progetto condotto dal Dipartimento di Ingegneria dell'Informazione di Modena e Reggio Emilia, sviluppato al fine di realizzare un tool semiautomatico per l'estrazione e l'integrazione di informazioni da sorgenti di dati strutturate e semistrutturate. Per l'estrazione dell'informazione si è introdotto un linguaggio *object-oriented*, ODL-I3, con una semantica che si basa su di una *Description Logics* [5]. L'integrazione delle informazioni viene effettuata in maniera semi-automatica, utilizzando la conoscenza presente in un *Common Thesaurus* e le descrizioni ODL-I3 degli schemi sorgenti e con tecniche di *clustering* e di *Description Logics*. Il processo di integrazione definisce una vista virtuale integrata degli schemi sottostanti nella quale vengono specificate regole di mapping e vincoli di integrità per la gestione delle eterogeneità.

Il sistema MOMIS si basa su un'architettura wrapper/mediator, e si struttura in 4 livelli:

- Il livello sorgente;
- Il livello wrapper;
- Il livello mediatore;
- Il livello utente.

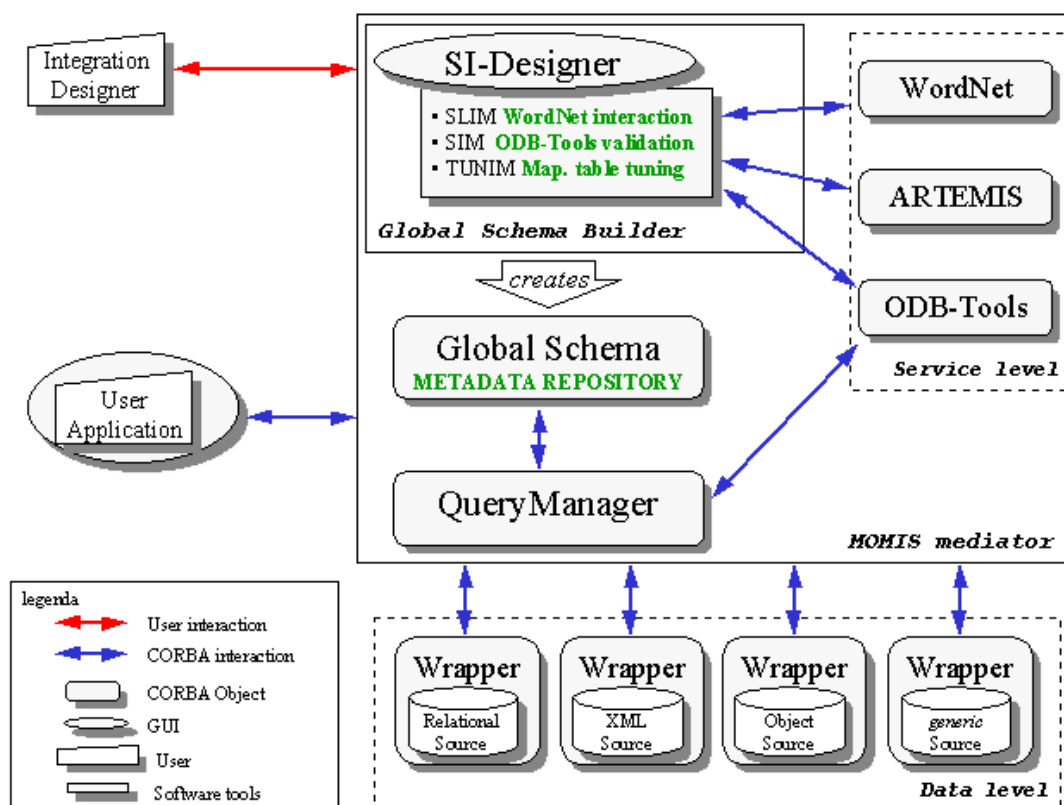


Figura 1: Architettura del sistema MOMIS

Il livello mediatore è il componente centrale che integra le varie sorgenti locali in una visione unificata.

I wrapper sono i moduli che permettono l'interfacce di sorgenti a livello di dati, quindi hanno lo scopo di estrarre le informazioni dalle varie sorgenti. Per far ciò i compiti dei wrapper sono due:

- Fornire una rappresentazione della struttura del sorgente attraverso una semantica comune che è il linguaggio ODL-13:
- Dare la possibilità di fare query locali e fornire la risposta al mediatore, il quale ha la funzione di integrare le informazioni acquisite.

Il wrapper EDIFACT ha il compito di estrarre le informazioni dai file sorgenti con formato EDIFACT, ma in questa tesi non si discuterà dell'interazione tra mediatore e il wrapper, perché il wrapper EDIFACT è articolato in due componenti. Un componente è un estrattore dati, il quale ha il compito di estrarre i dati dai file EDIFACT e creare un file che può inserire tali dati in un database dello standard sql92; un secondo componente è un wrapper sql server che ha il compito di interagire con il mediatore. Inoltre, il wrapper sql server fa già parte della architettura software di MOMIS (Wrapper JDBCcore_SQLserver) e quindi non verrà discusso ulteriormente in questa tesi.

La tesi è strutturata nel seguente modo:

- Capitolo 1 – Presentazione dell'EDIFACT, nel quale viene illustrato il sistema EDI, i suoi vantaggi, svantaggi, e le versioni delle sintassi esistenti;

- Capitolo 2 – Il formato EDIFACT, è il capitolo dedicato a descrivere il formato dei file EDIFACT;
- Capitolo 3 – Wrapper EDIFACT, in questo capitolo si descrive il wrapper realizzato, e si mostrano alcuni esempi di estrazione di dati, tra cui le query che si possono inviare al database e le viste sql.

Capitolo 1

Presentazione EDIFACT

Il forte sviluppo dei sistemi di comunicazione, e della diffusione delle tecnologie di memorizzazione e di elaborazione elettronica delle informazioni, ha favorito lo sviluppo delle trasmissioni di documenti elettronici. I primi sistemi di scambio elettronico di dati erano proprietari sviluppati dalle aziende interessate. Come conseguenza un'impresa che aveva più partner doveva utilizzare più sistemi di comunicazione. Gli standard EDI nascono per uniformare questi sistemi di comunicazione.

Sistemi EDI, acronimo di Electronic Data Interchange, sono sistemi per trasmissione di messaggi commerciali usando formati compatti e standardizzati. Gli standard dei documenti elettronici comparvero per la prima volta negli anni 70, ANSI ASC X12, uno standard EDI sviluppato negli Stati Uniti, è stato introdotto per transizione di commercio generale nel 1979, e lo standard europeo EDIFACT nasce nel 1987 con lo stesso dizionario del ANSI X12. Questi standard definisce le strutture di messaggi commerciali, con la promessa di rendere le transazioni commerciali più efficientemente.

Volendo poter trasmettere tutti i tipi di messaggi commerciali i formati EDI sono diventati man mano sempre più complessi a causa di sempre maggior tipi di messaggi e varianti, le organizzazioni internazionali hanno definito centinaia di messaggi, per poter ricoprire maggior parte possibile delle necessità delle industrie.

EDIFACT è un tipo di standard EDI, e più specificatamente EDIFACT è EDI For Administration, Commerce and Transport. Esso è una evoluzione degli standard di USA e d'Europa.

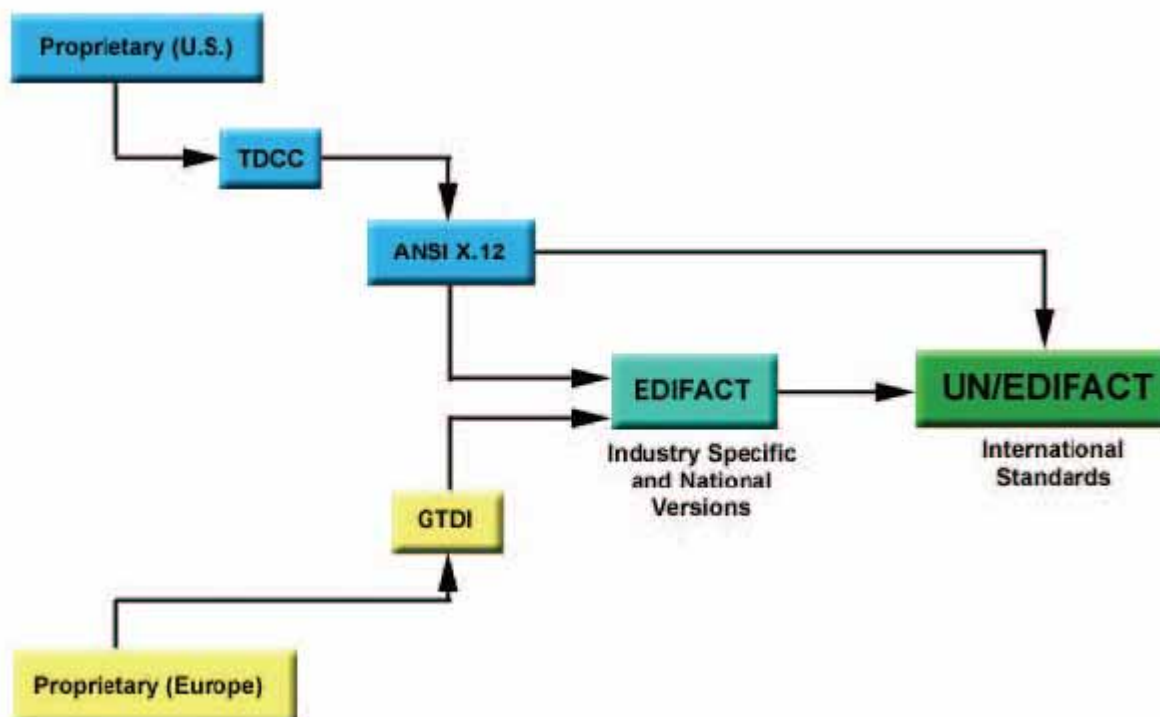


Figura 2: Evoluzione degli standard EDI

EDIFACT è un standard sviluppato e mantenuto dall'UN/ECE, United Nations/Economic Commission for Europe. Il quale provvede:

- internazionalizzazione dell'EDI;
- definire set di regole sintattiche per EDIFACT.

L'agente responsabile dell'EDIFACT è l'UN/CEFACT, il *Centre for the Facilitation of Procedures and Practices in Administration, Commerce and Transport*, il quale pubblica le revisioni successive del EDIFACT Directory.

EDIFACT Directory è l'autorità, che specifica gli tipi di elementi dati, lista di codice, segmenti e messaggi, e li definisce attraverso la pubblicazione dei vari *directory*, nei quali ci sono le strutture e definizioni dei componenti dell'EDIFACT.

1.1 Vantaggi dell'EDI

Il principale vantaggio dell'EDI è l'incremento dell'efficienza e diminuzione del costo delle comunicazioni rispetto ai metodi di comunicazione cartacei tradizionali, risparmiando i tempi ed i costi di spedizioni.

Inoltre EDIFACT riduce la perdita di tempo derivante dall'immissione ripetuta di dati. La trasmissione elettronica ed il trattamento di dati normalizzati avvengono in misura

essenzialmente più rapida rispetto alla trasmissione di documenti attraverso la via postale tradizionale, oltre a ciò i sistemi EDI velocizza l'acquisizione dei dati.

La rapidità dell'EDIFACT consente di praticare la moderna logistica Just in Time (JIT), aumentando così l'attività di trasbordo in magazzino, riducendo le scorte in magazzino e di sicurezza, diminuendo il capitale vincolato ed abbassando i costi di capitale connessi.

Gli standard EDI permettono agli utenti di definire i propri tipi di componenti di trasmissione (messaggi, segmenti e elementi dati), ciò dà agli utenti la possibilità di poter trasmettere qualsiasi messaggio di cui ha bisogno.

Usando l'EDIFACT i dati possono essere calcolati direttamente dagli archivi dati dei computer e attraverso i computer sono i dati sono più accurati dei dati immessi ripetutamente in modo manuale. Ciò significa che il numero degli interventi manuali viene ridotto ad una quantità minima e si riduce il rilevamento multiplo degli stessi dati. Inoltre mediante l'uso d'informazioni tipo e di codici uniformi a livello internazionale non si verificano più difficoltà di comprensione linguistica.

Infine i file EDIFACT hanno effetto legale, infatti dal primo gennaio 2004 è entrata in vigore la direttiva 2001/115/CE, il quale impone l'EDI o la firma digitale per l'invio della fattura trasmesse via telematica.

1.2 Svantaggi dell'EDI

Lo svantaggio degli standard EDI è la sua complessità, ci sono centinaia tipi di messaggi, un numero maggiore di tipi di segmenti e ancora maggiore sono i tipi di elementi dati. Tutto ciò perché gli standard EDI sono dei standard generici, in modo da poter ricoprire tutti i possibili tipi di messaggi usati da tutte le compagnie. Differenti industrie e compagnie possono definire sottoset negli vari standard, perché non hanno bisogno di tutti i tipi di messaggi per i loro usi. Perciò quando due compagnie vogliono comunicare attraverso EDI, essi devono prima di tutto negoziare e definire esattamente sul sottoset dello standard EDI da usare.

1.3 Sistemi di trasmissione dell'EDI

La rete usata per trasmissione dei messaggi EDI possono essere sia un sistema di rete diretta, due calcolatori delle due aziende, che intendono adottare i sistemi EDI, collegati direttamente, oppure trasmettere attraverso una rete a valore aggiunto (VAN), nella quale le aziende interessate comunicano con un elaboratore centrale posseduto dal gestore del VAN, il quale si occupa della gestione dei protocolli di linea.

1.4 Versioni dell'EDIFACT

Le regole a livello di sintassi dell'EDIFACT sono contenute nello standard ISO 9735, e fino ad oggi sono stati rilasciati 4 versioni delle regole di sintassi.

La prima versione è stata pubblicata nel 1988, il quale ha il formato della versione/release delle *directory*, invece che in formato numerico (es 91.2) nella versione 1 è in formato alfa numerico (es D99B), mentre gli elementi dati che corrispondono agli attuali 0052/0054, i quali rappresentano la versione e il release del messaggio nel UNG e UNH, sono in formati numerici.

Successivamente è stata rilasciato la versione 2 nel 1990, il quale è uguale alla versione 1 tranne per il fatto che l'elemento dato 0054 è in formato alfanumerico e l'elemento dato 0051, *controlling agency* è diventato da opzionale ad obbligatorio.

La versione 3 del 1992 è la versione 2 più *Amendment 1*, il quale (*Amendment 1*) estende il set di caratteri usato, oltre ai *character set* A (ISO 646 tranne i caratteri minuscoli e alcuni caratteri grafici) e B (ISO 646 tranne alcuni caratteri grafici) usato già nelle versioni precedenti si introducono nell'EDIFACT anche i *character set* da C alla F (che coprono l'alfabeto latino, cirillico e greco).

L'ultima versione è stato approvato ad ottobre del 1998, la versione 4 contiene una significativa revisione delle regole sintattiche e non è completamente compatibile con la versione 3, inoltre nella versione 4 è stato esteso ulteriormente i set di caratteri supportando anche UNICODE.

Le regole sintattiche che verranno descritto successivamente è la versione 4.

Capitolo 2

Il formato EDIFACT

A prima vista un file EDIFACT è solo una sequenza di caratteri, con qualche stringa di dati in mezzo a un insieme di simboli, in questo capitolo cercheremo di spiegare come interpretare tali sequenze di caratteri, la struttura di tali sequenze e di come è strutturato un file EDIFACT.

2.1 Segmenti

Un file EDIFACT è un insieme di segmenti. I segmenti possono essere di due tipi i segmenti dei dati d'utente e i segmenti di servizio.

I segmenti di dati d'utente contengono gli elementi dati che sono l'argomento della trasmissione, come ad esempio gli indirizzi, i dati sui prodotti, il costo, le tasse, il totale, la modalità di pagamento ecc.

I segmenti di servizio contengono i dati di servizio come: i dati del mittente, il tipo di sintassi usato, la data della preparazione della trasmissione, i tipi di priorità e altri dati della trasmissione. Solitamente questi segmenti sono segmenti *header* e segmenti *trailer*.

Un segmento è composto da un TAG e da uno o più elementi dati e infine il segmento termina con un terminatore di segmento. Le informazioni contenute dipendono dal tipo di segmento, ogni tipo di segmento contiene diversi tipi di elementi dati. I tipi di elementi dati, il loro ordine e il numero delle volte che possono essere presenti nel segmento sono definiti nella struttura del tipo di segmento. I tipi di segmenti e la loro struttura possono essere definiti sia dagli standard internazionali che dagli utilizzatori dell'EDIFACT.

Un esempio di segmento:

NAD+SE+999::92++24h Pizza+Long Way+San-Francisco+CA+34424+US'

Le prime tre lettere **NAD** sono il TAG.

Il simbolo + è il separatore di elementi dati.

Il simbolo : è il separatore di componenti di dati.

Il simbolo ' è il separatore di segmenti.

In questo caso non c'è l'annidamento.

Struttura degli elementi dati presenti nel segmento NAD:

010	3035	PARTY FUNCTION CODE QUALIFIER	M	1	an..3
020	C082	PARTY IDENTIFICATION DETAILS	C	1	
	3039	Party identifier	M		an..35
	1131	Code list identification code	C		an..17
	3055	Code list responsible agency code	C		an..3
030	C058	NAME AND ADDRESS	C	1	
	3124	Name and address description	M		an..35
	3124	Name and address description	C		an..35
	3124	Name and address description	C		an..35
	3124	Name and address description	C		an..35
	3124	Name and address description	C		an..35
040	C080	PARTY NAME	C	1	
	3036	Party name	M		an..35
	3036	Party name	C		an..35
	3036	Party name	C		an..35
	3036	Party name	C		an..35
	3036	Party name	C		an..35
	3045	Party name format code	C		an..3
050	C059	STREET	C	1	
	3042	Street and number or post office box identifier	M		an..35
	3042	Street and number or post office box identifier	C		an..35
	3042	Street and number or post office box identifier	C		an..35
	3042	Street and number or post office box identifier	C		an..35
060	3164	CITY NAME	C	1	an..35
070	C819	COUNTRY SUBDIVISION DETAILS	C	1	
	3229	Country subdivision identifier	C		an..9
	1131	Code list identification code	C		an..17
	3055	Code list responsible agency code	C		an..3
	3228	Country subdivision name	C		an..70
080	3251	POSTAL IDENTIFICATION CODE	C	1	an..17
090	3207	COUNTRY IDENTIFIER	C	1	an..3

La prima colonna indica la posizione dell'elemento di dato nel segmento.

La seconda colonna indica il codice dell'elemento dati.

La terza colonna contiene la descrizione dell'elemento dati.

La quarta colonna indica se l'elemento dato è obbligatorio o è facoltativo, **M** per obbligatorio, **C** per facoltativo. Se un elemento di dato è obbligatorio significa che deve essere presente almeno una volta nel segmento.

La quinta colonna indica il massimo numero delle ripetizioni dell'elemento dati che possono esserci all'interno del segmento.

La sesta colonna indica il tipo di caratteri dell'elemento dati, **an** indica che l'elemento dati contiene caratteri alfanumerici, e il numero successivo indica il massimo numero di caratteri.

2.1.1 TAG

Un TAG è composto da 3 lettere che indicano il tipo di segmento e le informazioni contenute nel segmento, inoltre, dopo le tre lettere, ci sono le indicazioni di eventuali ripetizioni e la gerarchia esplicita, di cui si vedrà successivamente, separati dalle tre lettere del TAG da un separatore di componenti di elementi dati.

Ogni tipo di segmento contiene determinati tipi di elementi dati.

Nell'esempio precedente il TAG **NAD** indica che nel segmento sono contenuti i dati di un nome e di un indirizzo.

I segmenti di servizio hanno i TAG che iniziano con le lettere **UN**, e i TAG dei segmenti di dati d'utente non possono iniziare con **UN**.

2.1.2 Elementi dati

Gli elementi dati sono la parte che contiene le informazioni del file, e possono essere distinti in:

- Elementi dati semplici;
- Elementi dati composti.

Gli elementi dati semplici hanno solo un componente di dato all'interno, mentre gli elementi dati composti sono composti da più elementi dati semplici separati dai separatori dei componenti di elementi dati.

Ogni tipo di elemento dati ha un suo codice, una descrizione, il tipo di carattere usato e il numero massimo di carattere.

I tipi di elementi dati possono essere definiti da standard internazionali o dagli utenti.

Nella struttura degli elementi dati presenti nel segmento **NAD** gli elementi dati composti hanno i codici che iniziano con la lettera **C**, e di seguito, sono indicati gli elementi dati semplici che lo compongono.

Nell'esempio di segmento precedente, l'elemento dati "SE" è il tipo di dato 3035, il "999::92" è il tipo di dato C082, con "999" è il tipo di dato 3039, mentre il tipo di dato 1131 è vuoto (facoltativo), il "92" è il tipo di dato 3055, il tipo di dati C058 è vuoto, il tipo di dato C080 è "24h Pizza" che è anche il tipo di dato 3036, con altri elementi dati semplici vuoti perciò sono stati troncati, ecc...

2.1.3 Omissioni e troncamenti

Quando un elemento di dato semplice in un elemento dati composti è vuoto, ma di seguito ha un elemento dati non nullo che fa parte dello stesso elemento dati composti, in questi casi si omette l'elemento vuoto, mentre se tutti gli elementi dati di seguito sono vuoti si usa il troncamento. L'operazione di omissione si fa per mantenere la posizione degli dati nel loro ordine.

Esempio:

TAG+ED+ED:ED+ED:::ED' I due elementi dati sono omessi.



DE sono gli elementi dati

Se nell'esempio anche l'ultimo ED fosse vuoto, il segmento diventerebbe:

TAG+ED+ED:ED+ED' I tre elementi dati sono stati troncati

Un discorso analogo di omissioni e troncamenti si può fare per gli elementi dati presenti nei segmenti.

Esempio:

TAG+ED+ED+ED+ED+++ED' I due elementi dati sono omessi.



TAG+ED+ED+ED+ED' I tre elementi dati sono stati troncati.

Nell'esempio del segmento NAD, i tipi di dati 1131 e C058 sono stati omessi, inoltre ci sono stati dei troncamenti negli elementi dati composti con i codici C080, C059, C819.

2.1.4 Ripetizioni degli elementi dati

I tipi elementi dati hanno un numero massimo di ripetizioni all'interno del segmento, se il numero degli elementi dati è minore, gli elementi dati vuoti sono omessi o troncati.

2.2 Messaggi

I messaggi dei file EDIFACT sono un insieme di segmenti. Ogni tipo di messaggio ha una struttura definita, cioè ogni messaggio di un determinato tipo di messaggio è

formato da specifici tipi di segmenti, che nei messaggi possono essere obbligatori o facoltativi e con un numero massimo di ripetizioni.

Gli elementi dati dei segmenti possono assumere significati differenti a seconda della posizione del segmento all'interno del messaggio. Ad esempio, un elemento dati dell'indirizzo in un segmento NAD può assumere il significato di indirizzo di un cliente o il significato dell'indirizzo di un fornitore a seconda della posizione del segmento NAD nel messaggio.

Un messaggio inizia con un segmento header UNH e termina con un segmento trailer. Nel header del messaggio è indicato il tipo di messaggio e la sua versione, oltre al message reference number (identificativo del messaggio).

Il segmento terminatore del messaggio è UNT, nel quale è contenuto il numero di segmenti nel messaggio compresi UNH ed UNT, inoltre è contenuto il message reference number.

I tipi standard di messaggi pubblicati dall'UN/CEFACT sono:

<u>APERAK</u>	<u>AUTHOR</u>	<u>BALANC</u>	<u>BANSTA</u>	<u>BAPLIE</u>	<u>BERMAN</u>	<u>BMISRM</u>
<u>BOPBNK</u>	<u>BOPCUS</u>	<u>BOPDIR</u>	<u>BOPINF</u>	<u>BUSCRD</u>	<u>CALINF</u>	<u>CASINT</u>
<u>CASRES</u>	<u>CHACCO</u>	<u>CLASET</u>	<u>CNTCND</u>	<u>COACSU</u>	<u>COARRI</u>	<u>CODECO</u>
<u>CODENO</u>	<u>COEDOR</u>	<u>COHAOR</u>	<u>COLREQ</u>	<u>COMDIS</u>	<u>CONAPW</u>	<u>CONDPV</u>
<u>CONDRA</u>	<u>CONDRO</u>	<u>CONEST</u>	<u>CONITT</u>	<u>CONPVA</u>	<u>CONQVA</u>	<u>CONRPW</u>
<u>CONTEN</u>	<u>CONWQD</u>	<u>COPARN</u>	<u>COPAYM</u>	<u>COPINO</u>	<u>COPRAR</u>	<u>COREOR</u>
<u>COSTCO</u>	<u>COSTOR</u>	<u>CREADV</u>	<u>CREEXT</u>	<u>CREMUL</u>	<u>CUSCAR</u>	<u>CUSDEC</u>
<u>CUSEXP</u>	<u>CUSPED</u>	<u>CUSREP</u>	<u>CUSRES</u>	<u>DAPLOS</u>	<u>DEBADV</u>	<u>DEBMUL</u>
<u>DEBREC</u>	<u>DELFOR</u>	<u>DELJIT</u>	<u>DESADV</u>	<u>DESTIM</u>	<u>DGRECA</u>	<u>DIRDEB</u>
<u>DIRDEF</u>	<u>DMRDEF</u>	<u>DMSTAT</u>	<u>DOCADV</u>	<u>DOCAMA</u>	<u>DOCAMI</u>	<u>DOCAMR</u>
<u>DOCAPP</u>	<u>DOCARE</u>	<u>DOCINF</u>	<u>ENTREC</u>	<u>FINCAN</u>	<u>FINPAY</u>	<u>FINSTA</u>
<u>GENRAL</u>	<u>GESMES</u>	<u>HANMOV</u>	<u>ICASRP</u>	<u>ICSOLI</u>	<u>IFCSUM</u>	<u>IFTCCA</u>
<u>IFTDGN</u>	<u>IFTFCC</u>	<u>IFTICL</u>	<u>IFTMAN</u>	<u>IFTMBC</u>	<u>IFTMBF</u>	<u>IFTMBP</u>
<u>IFTMCA</u>	<u>IFTMCS</u>	<u>IFTMIN</u>	<u>IFTRIN</u>	<u>IF TSAI</u>	<u>IFTSTA</u>	<u>IFTSTQ</u>
<u>IMPDEF</u>	<u>INFCON</u>	<u>INFENT</u>	<u>INSDDES</u>	<u>INSPRE</u>	<u>INSREQ</u>	<u>INSRPT</u>
<u>INVOIC</u>	<u>INVRPT</u>	<u>IPPOAD</u>	<u>IPPOMO</u>	<u>ISENDS</u>	<u>ITRRPT</u>	<u>JAPRES</u>

JINFDE	JOBAPP	JOBCON	JOBMOD	JOBOFF	JUPREQ	LEDGER
LREACT	LRECLM	MEDPID	MEDPRE	MEDREQ	MEDRPT	MEDRUC
MEQPOS	MOVINS	MSCONS	ORDCHG	ORDERS	ORDRSP	OSTENQ
OSTRPT	PARTIN	PAXLST	PAYDUC	PAYEXT	PAYMUL	PAYORD
PRICAT	PRIHIS	PROCST	PRODAT	PRODEX	PROINQ	PROSRV
PROTAP	PRPAID	QUALITY	QUOTES	RDRMES	RECADV	RECECO
RECORD	REGENT	RELIST	REMADV	REQDOC	REQOTE	RESMSG
RETANN	RETINS	RPCALL	SAFHAZ	SANCRT	SLSFCT	SLSRPT
SOCADE	SSIMOD	SSRECH	SSREGW	STATAC	STLRPT	SUPCOT
SUPMAN	SUPRES	TANSTA	TAXCON	TPFREP	UTILMD	UTILTS
VATDEC	VESDEP	WASDIS	WKGRDC	WKGRRE		

Oltre a questi tipi di messaggi l'utente e le organizzazioni che utilizzano l'EDIFACT possono definire i nuovi tipi di messaggi.

Struttura del messaggio ORDERS:

0010	UNH	Message header	M	1	
0020	BGM	Beginning of message	M	1	
0030	DTM	Date/time/period	M	35	
0040	PAI	Payment instructions	C	1	
0050	ALI	Additional information	C	5	
0060	IMD	Item description	C	999	
0070	FTX	Free text	C	99	
0080	GIR	Related identification numbers	C	10	
0090		----- Segment group 1 -----	C	9999	-----+
0100	RFF	Reference	M	1	
0110	DTM	Date/time/period	C	5	-----+
0120		----- Segment group 2 -----	C	99	-----+
0130	NAD	Name and address	M	1	
0140	LOC	Place/location identification	C	99	
0150	FII	Financial institution information	C	5	
0160		----- Segment group 3 -----	C	99	-----+
0170	RFF	Reference	M	1	
0180	DTM	Date/time/period	C	5	-----+
0190		----- Segment group 4 -----	C	5	-----+
0200	DOC	Document/message details	M	1	
0210	DTM	Date/time/period	C	5	-----+
0220		----- Segment group 5 -----	C	5	-----+
0230	CTA	Contact information	M	1	

<u>0240</u>	<u>COM</u>	Communication contact	C	5-----++
<u>0250</u>		----- Segment group 6 -----	C	5-----++
<u>0260</u>	<u>TAX</u>	Duty/tax/fee details	M	1
<u>0270</u>	<u>MOA</u>	Monetary amount	C	1
<u>0280</u>	<u>LOC</u>	Place/location identification	C	9-----++
<u>0290</u>		----- Segment group 7 -----	C	5-----++
<u>0300</u>	<u>CUX</u>	Currencies	M	1
<u>0310</u>	<u>PCD</u>	Percentage details	C	5
<u>0320</u>	<u>DTM</u>	Date/time/period	C	5-----++
<u>0330</u>		----- Segment group 8 -----	C	10-----++
<u>0340</u>	<u>PYT</u>	Payment terms	M	1
<u>0350</u>	<u>DTM</u>	Date/time/period	C	5
<u>0360</u>	<u>PCD</u>	Percentage details	C	1
<u>0370</u>		----- Segment group 9 -----	C	9999-----++
<u>0380</u>	<u>MOA</u>	Monetary amount	M	1
<u>0390</u>	<u>GIR</u>	Related identification numbers	C	9
<u>0400</u>	<u>RJL</u>	Accounting journal identification	C	99-----++
<u>0410</u>		----- Segment group 10 -----	C	10-----++
<u>0420</u>	<u>TDT</u>	Transport information	M	1
<u>0430</u>		----- Segment group 11 -----	C	10-----++
<u>0440</u>	<u>LOC</u>	Place/location identification	M	1
<u>0450</u>	<u>DTM</u>	Date/time/period	C	5-----++
<u>0460</u>		----- Segment group 12 -----	C	5-----++
<u>0470</u>	<u>TOD</u>	Terms of delivery or transport	M	1
<u>0480</u>	<u>LOC</u>	Place/location identification	C	2-----++
<u>0490</u>		----- Segment group 13 -----	C	99-----++
<u>0500</u>	<u>PAC</u>	Package	M	1
<u>0510</u>	<u>MEA</u>	Measurements	C	5
<u>0520</u>		----- Segment group 14 -----	C	5-----++
<u>0530</u>	<u>PCI</u>	Package identification	M	1
<u>0540</u>	<u>RFF</u>	Reference	C	1
<u>0550</u>	<u>DTM</u>	Date/time/period	C	5
<u>0560</u>	<u>GIN</u>	Goods identity number	C	10-----++
<u>0570</u>		----- Segment group 15 -----	C	10-----++
<u>0580</u>	<u>EQD</u>	Equipment details	M	1
<u>0590</u>	<u>HAN</u>	Handling instructions	C	5
<u>0600</u>	<u>MEA</u>	Measurements	C	5
<u>0610</u>	<u>FTX</u>	Free text	C	5-----++
<u>0620</u>		----- Segment group 16 -----	C	10-----++
<u>0630</u>	<u>SCC</u>	Scheduling conditions	M	1
<u>0640</u>	<u>FTX</u>	Free text	C	5
<u>0650</u>	<u>RFF</u>	Reference	C	5
<u>0660</u>		----- Segment group 17 -----	C	10-----++
<u>0670</u>	<u>QTY</u>	Quantity	M	1
<u>0680</u>	<u>DTM</u>	Date/time/period	C	5-----++

<u>0690</u>		----- Segment group 18 -----	C	25-----+
<u>0700</u>	<u>APR</u>	Additional price information	M	1
<u>0710</u>	<u>DTM</u>	Date/time/period	C	5
<u>0720</u>	<u>RNG</u>	Range details	C	1-----+
<u>0730</u>		----- Segment group 19 -----	C	99-----+
<u>0740</u>	<u>ALC</u>	Allowance or charge	M	1
<u>0750</u>	<u>ALI</u>	Additional information	C	5
<u>0760</u>	<u>DTM</u>	Date/time/period	C	5
<u>0770</u>		----- Segment group 20 -----	C	1-----+
<u>0780</u>	<u>QTY</u>	Quantity	M	1
<u>0790</u>	<u>RNG</u>	Range details	C	1-----+
<u>0800</u>		----- Segment group 21 -----	C	1-----+
<u>0810</u>	<u>PCD</u>	Percentage details	M	1
<u>0820</u>	<u>RNG</u>	Range details	C	1-----+
<u>0830</u>		----- Segment group 22 -----	C	2-----+
<u>0840</u>	<u>MOA</u>	Monetary amount	M	1
<u>0850</u>	<u>RNG</u>	Range details	C	1-----+
<u>0860</u>		----- Segment group 23 -----	C	1-----+
<u>0870</u>	<u>RTE</u>	Rate details	M	1
<u>0880</u>	<u>RNG</u>	Range details	C	1-----+
<u>0890</u>		----- Segment group 24 -----	C	5-----+
<u>0900</u>	<u>TAX</u>	Duty/tax/fee details	M	1
<u>0910</u>	<u>MOA</u>	Monetary amount	C	1-----+
<u>0920</u>		----- Segment group 25 -----	C	999-----+
<u>0930</u>	<u>RCS</u>	Requirements and conditions	M	1
<u>0940</u>	<u>RFF</u>	Reference	C	5
<u>0950</u>	<u>DTM</u>	Date/time/period	C	5
<u>0960</u>	<u>FTX</u>	Free text	C	99999-----+
<u>0970</u>		----- Segment group 26 -----	C	999-----+
<u>0980</u>	<u>DGS</u>	Dangerous goods	M	1
<u>0990</u>	<u>FTX</u>	Free text	C	5
<u>1000</u>		----- Segment group 27 -----	C	99-----+
<u>1010</u>	<u>CTA</u>	Contact information	M	1
<u>1020</u>	<u>COM</u>	Communication contact	C	5-----+
<u>1030</u>		----- Segment group 28 -----	C	99-----+
<u>1040</u>	<u>EFI</u>	External file link identification	M	1
<u>1050</u>	<u>CED</u>	Computer environment details	C	99
<u>1060</u>	<u>COM</u>	Communication contact	C	9
<u>1070</u>	<u>RFF</u>	Reference	C	9
<u>1080</u>	<u>DTM</u>	Date/time/period	C	9
<u>1090</u>	<u>QTY</u>	Quantity	C	9-----+
<u>1100</u>		----- Segment group 29 -----	C	200000-----+
<u>1110</u>	<u>LIN</u>	Line item	M	1
<u>1120</u>	<u>PIA</u>	Additional product id	C	25
<u>1130</u>	<u>IMD</u>	item description	C	99
<u>1140</u>	<u>MEA</u>	Measurements	C	99
<u>1150</u>	<u>QTY</u>	Quantity	C	99

<u>1160</u>	<u>PCD</u>	Percentage details	C	5	
<u>1170</u>	<u>ALI</u>	Additional information	C	5	
<u>1180</u>	<u>DTM</u>	Date/time/period	C	35	
<u>1190</u>	<u>MOA</u>	Monetary amount	C	10	
<u>1200</u>	<u>GEI</u>	Processing information	C	99	
<u>1210</u>	<u>GIN</u>	Goods identity number	C	1000	
<u>1220</u>	<u>GIR</u>	Related identification numbers	C	1000	
<u>1230</u>	<u>QVR</u>	Quantity variances	C	1	
<u>1240</u>	<u>DOC</u>	Document/message details	C	99	
<u>1250</u>	<u>PAI</u>	Payment instructions	C	1	
<u>1260</u>	<u>MTD</u>	Maintenance operation details	C	99	
<u>1270</u>	<u>FTX</u>	Free text	C	99	
<u>1280</u>		----- Segment group 30 -----	C	999	-----+
<u>1290</u>	<u>CCI</u>	Characteristic/class id	M	1	
<u>1300</u>	<u>CAV</u>	Characteristic value	C	10	
<u>1310</u>	<u>MEA</u>	Measurements	C	10	-----+
<u>1320</u>		----- Segment group 31 -----	C	10	-----+
<u>1330</u>	<u>PYT</u>	Payment terms	M	1	
<u>1340</u>	<u>DTM</u>	Date/time/period	C	5	
<u>1350</u>	<u>PCD</u>	Percentage details	C	1	
<u>1360</u>		----- Segment group 32 -----	C	9999	-----+
<u>1370</u>	<u>MOA</u>	Monetary amount	M	1	
<u>1380</u>	<u>GIR</u>	Related identification numbers	C	9	-----+
<u>1390</u>		----- Segment group 33 -----	C	25	-----+
<u>1400</u>	<u>PRI</u>	Price details	M	1	
<u>1410</u>	<u>CUX</u>	Currencies	C	1	
<u>1420</u>	<u>APR</u>	Additional price information	C	99	
<u>1430</u>	<u>RNG</u>	Range details	C	1	
<u>1440</u>	<u>DTM</u>	Date/time/period	C	5	-----+
<u>1450</u>		----- Segment group 34 -----	C	9999	-----+
<u>1460</u>	<u>RFF</u>	Reference	M	1	
<u>1470</u>	<u>DTM</u>	Date/time/period	C	5	
<u>1480</u>	<u>GEI</u>	Processing information	C	99	
<u>1490</u>	<u>MOA</u>	Monetary amount	C	99	-----+
<u>1500</u>		----- Segment group 35 -----	C	99	-----+
<u>1510</u>	<u>PAC</u>	Package	M	1	
<u>1520</u>	<u>MEA</u>	Measurements	C	5	
<u>1530</u>	<u>QTY</u>	Quantity	C	5	
<u>1540</u>	<u>DTM</u>	Date/time/period	C	5	
<u>1550</u>		----- Segment group 36 -----	C	1	-----+
<u>1560</u>	<u>RFF</u>	Reference	M	1	
<u>1570</u>	<u>DTM</u>	Date/time/period	C	5	-----+
<u>1580</u>		----- Segment group 37 -----	C	5	-----+
<u>1590</u>	<u>PCI</u>	Package identification	M	1	
<u>1600</u>	<u>RFF</u>	Reference	C	1	
<u>1610</u>	<u>DTM</u>	Date/time/period	C	5	
<u>1620</u>	<u>GIN</u>	Goods identity number	C	10	-----+
<u>1630</u>		----- Segment group 38 -----	C	9999	-----+
<u>1640</u>	<u>LOC</u>	Place/location identification	M	1	

<u>1650</u>	<u>QTY</u>	Quantity	C	1	
<u>1660</u>	<u>PCD</u>	Percentage details	C	1	
<u>1670</u>	<u>DTM</u>	Date/time/period	C	5	-----+
<u>1680</u>		----- Segment group 39 -----	C	10	-----+
<u>1690</u>	<u>TAX</u>	Duty/tax/fee details	M	1	
<u>1700</u>	<u>MOA</u>	Monetary amount	C	1	
<u>1710</u>	<u>LOC</u>	Place/location identification	C	5	-----+
<u>1720</u>		----- Segment group 40 -----	C	999	-----+
<u>1730</u>	<u>NAD</u>	Name and address	M	1	
<u>1740</u>	<u>LOC</u>	Place/location identification	C	5	
<u>1750</u>	<u>FII</u>	Financial institution information	C	5	
<u>1760</u>		----- Segment group 41 -----	C	99	-----+
<u>1770</u>	<u>RFF</u>	Reference	M	1	
<u>1780</u>	<u>DTM</u>	Date/time/period	C	5	-----+
<u>1790</u>		----- Segment group 42 -----	C	5	-----+
<u>1800</u>	<u>DOC</u>	Document/message details	M	1	
<u>1810</u>	<u>DTM</u>	Date/time/period	C	5	-----+
<u>1820</u>		----- Segment group 43 -----	C	5	-----+
<u>1830</u>	<u>CTA</u>	Contact information	M	1	
<u>1840</u>	<u>COM</u>	Communication contact	C	5	-----+
<u>1850</u>		----- Segment group 44 -----	C	99	-----+
<u>1860</u>	<u>ALC</u>	Allowance or charge	M	1	
<u>1870</u>	<u>ALI</u>	Additional information	C	5	
<u>1880</u>	<u>DTM</u>	Date/time/period	C	5	
<u>1890</u>		----- Segment group 45 -----	C	1	-----+
<u>1900</u>	<u>QTY</u>	Quantity	M	1	
<u>1910</u>	<u>RNG</u>	Range details	C	1	-----+
<u>1920</u>		----- Segment group 46 -----	C	1	-----+
<u>1930</u>	<u>PCD</u>	Percentage details	M	1	
<u>1940</u>	<u>RNG</u>	Range details	C	1	-----+
<u>1950</u>		----- Segment group 47 -----	C	2	-----+
<u>1960</u>	<u>MOA</u>	Monetary amount	M	1	
<u>1970</u>	<u>RNG</u>	Range details	C	1	-----+
<u>1980</u>		----- Segment group 48 -----	C	1	-----+
<u>1990</u>	<u>RTE</u>	Rate details	M	1	
<u>2000</u>	<u>RNG</u>	Range details	C	1	-----+
<u>2010</u>		----- Segment group 49 -----	C	5	-----+
<u>2020</u>	<u>TAX</u>	Duty/tax/fee details	M	1	
<u>2030</u>	<u>MOA</u>	Monetary amount	C	1	-----+
<u>2040</u>		----- Segment group 50 -----	C	10	-----+
<u>2050</u>	<u>TDI</u>	Transport information	M	1	
<u>2060</u>		----- Segment group 51 -----	C	10	-----+
<u>2070</u>	<u>LOC</u>	Place/location identification	M	1	
<u>2080</u>	<u>DTM</u>	Date/time/period	C	5	-----+

<u>2090</u>		----- Segment group 52 -----	C	5-----+
<u>2100</u>	<u>TOD</u>	Terms of delivery or transport	M	1
<u>2110</u>	<u>LOC</u>	Place/location identification	C	2-----+
<u>2120</u>		----- Segment group 53 -----	C	10-----+
<u>2130</u>	<u>EQD</u>	Equipment details	M	1
<u>2140</u>	<u>HAN</u>	Handling instructions	C	5
<u>2150</u>	<u>MEA</u>	Measurements	C	5
<u>2160</u>	<u>FTX</u>	Free text	C	5-----+
<u>2170</u>		----- Segment group 54 -----	C	100-----+
<u>2180</u>	<u>SCC</u>	Scheduling conditions	M	1
<u>2190</u>	<u>FTX</u>	Free text	C	5
<u>2200</u>	<u>RFF</u>	Reference	C	5
<u>2210</u>		----- Segment group 55 -----	C	10-----+
<u>2220</u>	<u>QTY</u>	Quantity	M	1
<u>2230</u>	<u>DTM</u>	Date/time/period	C	5-----++
<u>2240</u>		----- Segment group 56 -----	C	999-----+
<u>2250</u>	<u>RCS</u>	Requirements and conditions	M	1
<u>2260</u>	<u>RFF</u>	Reference	C	5
<u>2270</u>	<u>DTM</u>	Date/time/period	C	5
<u>2280</u>	<u>FTX</u>	Free text	C	99999-----+
<u>2290</u>		----- Segment group 57 -----	C	10-----+
<u>2300</u>	<u>STG</u>	Stages	M	1
<u>2310</u>		----- Segment group 58 -----	C	3-----+
<u>2320</u>	<u>QTY</u>	Quantity	M	1
<u>2330</u>	<u>MOA</u>	Monetary amount	C	1-----++
<u>2340</u>		----- Segment group 59 -----	C	999-----+
<u>2350</u>	<u>DGS</u>	Dangerous goods	M	1
<u>2360</u>	<u>FTX</u>	Free text	C	5
<u>2370</u>		----- Segment group 60 -----	C	99-----+
<u>2380</u>	<u>CTA</u>	Contact information	M	1
<u>2390</u>	<u>COM</u>	Communication contact	C	5-----++
<u>2400</u>		----- Segment group 61 -----	C	99-----+
<u>2410</u>	<u>EFI</u>	External file link identification	M	1
<u>2420</u>	<u>CED</u>	Computer environment details	C	99
<u>2430</u>	<u>COM</u>	Communication contact	C	99
<u>2440</u>	<u>RFF</u>	Reference	C	9
<u>2450</u>	<u>DTM</u>	Date/time/period	C	9
<u>2460</u>	<u>QTY</u>	Quantity	C	9-----++
<u>2470</u>	<u>UNS</u>	Section control	M	1
<u>2480</u>	<u>MOA</u>	Monetary amount	C	99
<u>2490</u>	<u>CNT</u>	Control total	C	10
<u>2500</u>		----- Segment group 62 -----	C	10-----+
<u>2510</u>	<u>ALC</u>	Allowance or charge	M	1
<u>2520</u>	<u>ALI</u>	Additional information	C	1
<u>2530</u>	<u>MOA</u>	Monetary amount	M	2-----+
<u>2540</u>	<u>UNT</u>	Message trailer	M	1

Lo schema indica la struttura dei segmenti presenti nel messaggio dell'ordine, i raggruppamenti dei segmenti.

La prima colonna è una numerazione degli segmenti e dei gruppi di segmenti.

La seconda colonna è il TAG del segmento.

La terza colonna c'è la descrizione del segmento.

La quarta colonna indica se il segmento o il gruppo di segmento è obbligatorio (M), o facoltativo (C), se un segmento è obbligatorio e il gruppo di segmento è opzionale, significa che il segmento è obbligatorio solo se è presente il gruppo di segmento.

La quinta colonna indica la ripetizione de gli segmenti e dei gruppi si segmenti.

Se un segmento è opzionale può essere omesso dal messaggio, se non ha i dati significativi (cioè ha tutti gli elementi dati del segmento nullo).

Se un gruppo di segmento è condizionale significa che l'intero gruppo può essere omesso, cioè non esiste nessun segmento del gruppo nel messaggio.

Se un segmento è obbligatorio in un gruppo di segmenti condizionale, significa che il segmento è obbligatorio solo se il gruppo è presente nel messaggio, cioè esiste almeno un segmento del gruppo.

Come si può notare il messaggio inizia con un segmento UNH, e finisce con un segmento UNT. Inoltre si può notare che alcuni segmenti sono raggruppati in gruppi di segmenti che possono a loro volta avere dei sottogruppi di segmenti, da ciò genera una gerarchia di segmenti che si vedrà successivamente.

2.2.1 Ripetizioni dei segmenti

Dato un tipo di messaggio, la tecnica della ripetizione di segmenti può essere esplicita o implicita. La decisione di quale tecnica usare è decisa quando si progetta il messaggio, non si possono usare tutte e due le tecniche nello stesso messaggio.

Quando si adotta la tecnica della ripetizione esplicita l'ultimo componente del TAG indica il numero della ripetizione.

Nella ripetizione implicita, la ripetizione del segmento non viene indicata, e la ripetizione è comprensibile dall'ordine dei segmenti perché segue la struttura del tipo di messaggio.

2.2.2 Gerarchia dei segmenti

Come mostrato nell'esempio della struttura del messaggio si può osservare che i segmenti sono raggruppati in vari gruppi di segmenti, inoltre si può notare che all'interno di un gruppo di segmento può esserci un altro gruppo di segmento, da ciò si ricava una gerarchia dei segmenti. La gerarchia dei segmenti può essere indicata esplicitamente o implicitamente, le due tecniche di indicazione vengono decise nella

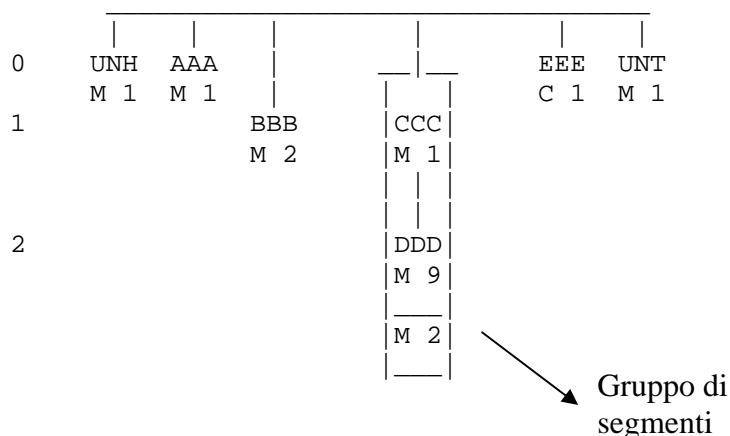
fase di progettazione del messaggio, e non si possono mischiare le due tecniche nello stesso messaggio.

Con la tecnica della gerarchia esplicita si indica la gerarchia come componenti di elementi dati nel TAG. Quando si adotta la tecnica della ripetizione esplicita, i componenti del TAG successivo alle tre lettere del TAG indicano sia la gerarchia sia le ripetizioni del segmento. Il numero di componenti che indicano la gerarchia e le ripetizioni indica il livello di gerarchia. Ogni componente prima dell'ultimo componente (escluso le prime tre lettere), indicano il numero del gruppo a cui fa parte nei vari livelli, mentre l'ultimo componente indica il numero di ripetizione del segmento.

Quando la gerarchia è implicita la gerarchia si capisce dall'ordine dei segmenti perché segue la struttura del tipo di messaggio.

Due esempi di indicazione esplicita:

Livello Struttura del messaggio



Il gruppo di segmento è obbligatorio e ha il numero massimo di ripetizioni 2.

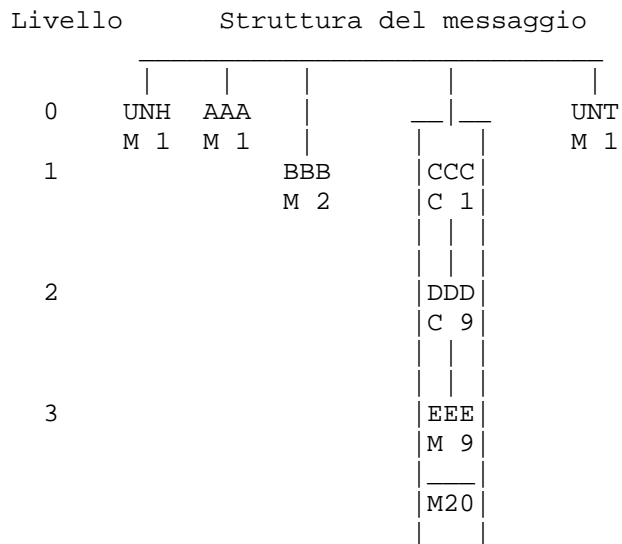
All'interno del gruppo di segmento il segmento CCC è obbligatorio e con ripetizioni massimo è 2, inoltre c'è un ulteriore sottogruppo formato solo dal segmento DDD è obbligatorio con ripetizione massimo è 9. Ciò significa che ci sono 2 livelli nel gruppo.

Segmenti

- UNH+data'
- AAA+data'
- BBB:1+data' Oggetto 1 del BBB
- BBB:2+data' Oggetto 2 del BBB
- CCC:1+data' Oggetto 1 del CCC
- DDD:1:1+data' Oggetto 1 del DDD in CCC(1)
- DDD:1:2+data' Oggetto 2 del DDD in CCC(1)
- CCC:2+data' Oggetto 2 del CCC
- DDD:2:1+data' Oggetto 1 del DDD in CCC(2)
- EEE+data'
- UNT+data'

In un'unica stringa:

```
UNH+dat'AAA+data'BBB:1+data'BBB:2+data'CCC:1+data'DDD:1:1+data'DDD:1:2
+data'CCC:2+data'DDD:2:1+data'EEE+data'UNT+data'
```



Segmenti

UNH+data'

AAA+data'

BBB:1+data' Oggetto 1 del BBB

BBB:2+data' Oggetto 2 del BBB

EEE::1+data' Oggetto 1 del EEE senza DDD e CCC

EEE::2+data' Oggetto 2 del EEE senza DDD e CCC

CCC:1+data' primo CCC

DDD:1:1+data' primo DDD nel CCC(1)

EEE:1:1:1+data' EEE(1) nel DDD(1) nel CCC(1)

EEE:1:1:2+data' EEE(2) nel DDD(1) nel CCC(1)

DDD:1:2+data' DDD(2) nel CCC(1)

EEE:1:2:1+data' EEE(1) nel DDD(2) nel CCC(1)

CCC:2+data' CCC(2)

EEE:2::1+data' EEE(1) nel CCC(2) senza DDD

UNT+data'

In un'unica stringa:

```
UNH+data'AAA+data'BBB:1+data'BBB:2+data'EEE::1+data'EEE::2+data'CCC:
1+data'DDD:1:1+data'EEE:1:1:1+data'EEE:1:1:2+data'DDD:1:2+data'EEE:1:2
:1+data'CCC:2+data'EEE:2::1+data'UNT+data'
```

2.3 La struttura del file

In un file EDIFACT può esserci più di un messaggio, e tali messaggi possono essere raggruppati in uno o più gruppi funzionali, un gruppo di messaggi può avere lo stesso “sottodestinatario”, ad esempio una divisione specifica all’interno dell’impresa destinatario, o un “sottomittente”, oppure può essere utile raggruppare i messaggi per tipi di messaggi. Perciò i livelli di un file EDIFACT possono essere 3, visto che il livello del gruppo è opzionale, il livello più esterno è l’interchange, il quale può contenere all’interno uno o più gruppi funzionali, oppure può contenere uno o più messaggi. Nel primo caso all’interno dei gruppi funzionali ci sono poi uno o più messaggi.

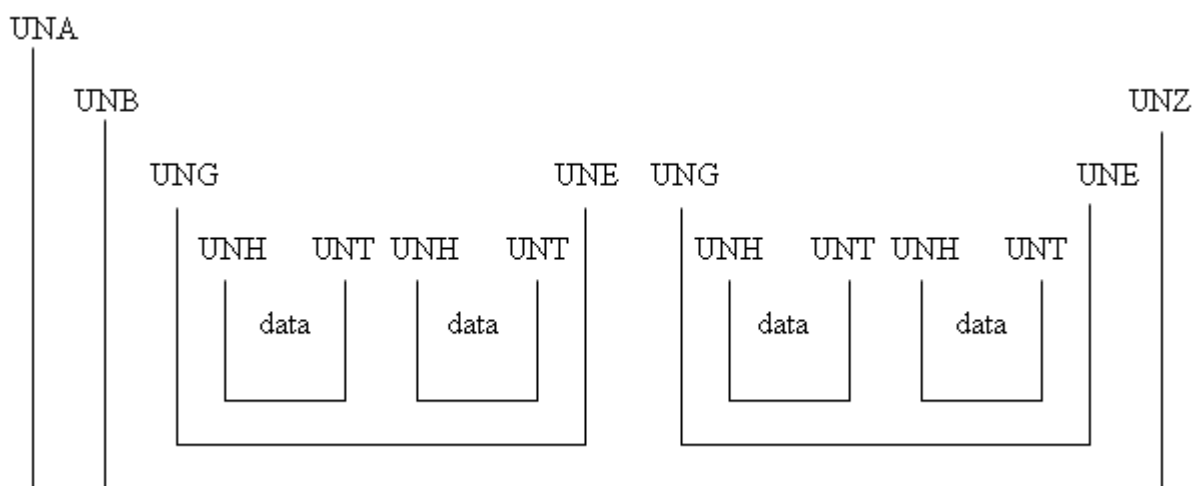


Figura 3: Struttura di un file EDIFACT

2.3.1 Interchange

Un’interchange è un insieme di gruppi funzionali o un insieme di messaggi, nel caso in cui non c’è il livello di gruppi.

Gli interchange possono avere 2 header, UNA e UNB.

UNA è un segmento header opzionale in cui sono indicati gli operatori della comunicazione. Se il segmento non è presente si usano gli operatori di default.

Gli operatori di default sono:

Caratteri nel UNA	Operatori di default	Funzione
Primo carattere	:	separatore di componenti di elementi dati
Secondo carattere	+	separatore di elementi dati

Terzo carattere	.	punto decimale(negli elementi dati numerici)
Quarto carattere	?	carattere di rilascio
Quinto carattere	(spazio)	riservato per usi futuri
Sesto carattere	'	terminatore segmento

UNB è un segmento header obbligatorio, in cui sono indicati il mittente e il destinatario dell'interchange, oltre all'identificativo e data della preparazione dell'interchange, inoltre ci sono altre informazioni relativo alla comunicazione.

Nel segmento UNB sono obbligatori i seguenti dati e sono ordinati nel seguente ordine:

- . identificatore di sintassi e numero di versione;
- . mittente di interchange;
- . destinatario di interchange, più un sottoindirizzo (normalmente codificato) opzionale per onward routing;
- . data e tempo di preparazione;
- . interchange control reference (il codice dell'interchange).

UNZ è il segmento trailer dell'interchange, negli elementi dati del segmento è contenuto il numero dei messaggi o dei gruppi funzionali presenti nell'interchange e l'interchange control reference.

2.3.2 Gruppo funzionale

Un gruppo funzionale è un insieme di messaggi, inizia con il segmento header UNG, e un segmento trailer UNT.

Il livello di gruppo funzionale è opzionale nell'interchange, se non c'è significa che nell'interchange ci sono direttamente i messaggi.

Il vantaggio di avere il livello del gruppo funzionale è che per le grandi organizzazioni si può avere suddivisi i messaggi per i tipi di messaggi e/o si può raggruppare i messaggi per i vari divisioni dell'organizzazione (sia dalla parte del mittente che dalla parte del destinatario).

Nel segmento UNG è contenuto il functional group reference number del gruppo funzionale all'interno dell'interchange, il functional group reference number è univoco nell'interchange.

Il segmento trailer del gruppo funzionale è UNE, il quale ha come elementi dati il numero dei messaggi contenuti nel gruppo funzionale e il functional group reference number.

Capitolo 3

Il Wrapper EDIFACT

Come si è potuto osservare dal capitolo precedente, l'EDIFACT è più un protocollo di comunicazione che un formato di database. Perciò, per realizzare il wrapper EDIFACT per il sistema MOMIS bisogna prima realizzare un estrattore di dati dai file EDIFACT e una loro trasformazione in un formato relazionale, e si è scelto il formato standard sql 92 per la creazione del database relazionale che memorizza il contenuto dei file EDIFACT. Il wrapper sql server del sistema MOMIS viene poi utilizzato per inserire i dati nel sistema stesso. In questo elaborato parleremo dell'estrattore di dati EDIFACT in quanto il wrapper sql server esiste già.

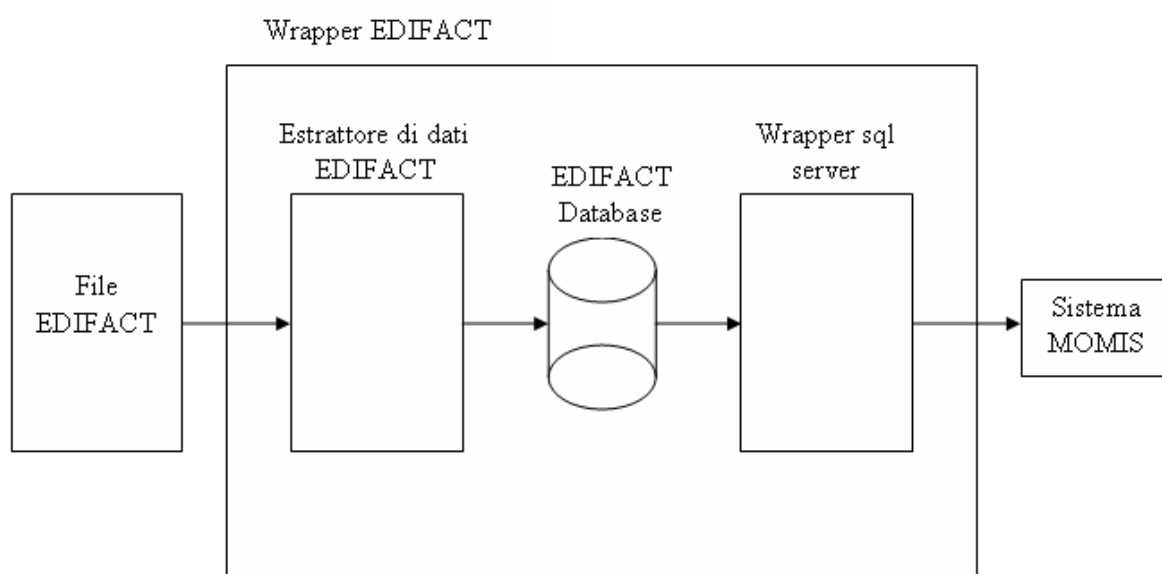


Figura 4: Struttura del wrapper EDIFACT

Per estrarre i dati da un file EDIFACT bisogna innanzitutto comprendere la sua struttura dati sia in termini sintattici che in forma semantici. Inoltre, oltre a estrarre i dati bisogna dare un significato in modo da poterli inserire in un database relazionale. Ma come si è visto in precedenza nel file EDIFACT il significato di un elemento dati non dipende solo dalla sua posizione nel segmento ma anche dal tipo di messaggio e dalla posizione del segmento nel messaggio. L'operazione di trasformazione/interpretazione è problematica in quanto ci sono quasi 200 tipi di

messaggi e un numero ancora maggiore di tipi di segmenti e di elementi dati. Inoltre, per aumentare la complessità gli utenti possono definire i propri tipi di messaggi, di segmenti e di elementi dati. Oltre a ciò bisognerebbe usare diversi schemi, uno per ogni tipo di messaggio usato.

Si è scelto di progettare ed usare un unico schema generale di database per contenere tutti i dati estratti dai file EDIFACT, mantenendo memorizzate le informazioni relative alle posizioni degli elementi dati. In questo schema generale tutti gli elementi dati sono trattati allo stesso modo, e per capire il loro significato bisogna cercare le loro posizioni nel messaggio e nel segmento, poi controllare il significato di ogni elemento dato dallo schema del tipo di messaggio. Per facilitare la comprensione del significato degli elementi dati e le loro relazioni si possono creare delle viste sql per ogni tipo di messaggio.

3.1 Estrattore dati EDIFACT

L'estrattore dati EDIFACT deve estrarre elementi dati da un file EDIFACT e creare un file di input di tali dati per il database EDIFACT. Perciò si ha bisogno di uno schema di database generale in cui inserire gli elementi dati.

Nell'estrazione dei dati bisogna anche memorizzare le posizione degli elementi dati nei segmenti, la posizione dei segmenti e i loro TAG, altrimenti si perde l'informazione sul loro significato. Oltre alle informazioni sulla posizione dei dati è indispensabile ricavare anche identificatori dell'*interchange*, dei *gruppi funzionali* e dei *messaggi*.

3.1.1 Schema del database

Lo schema E/R dello database EDIFACT è mostrato in figura 5.

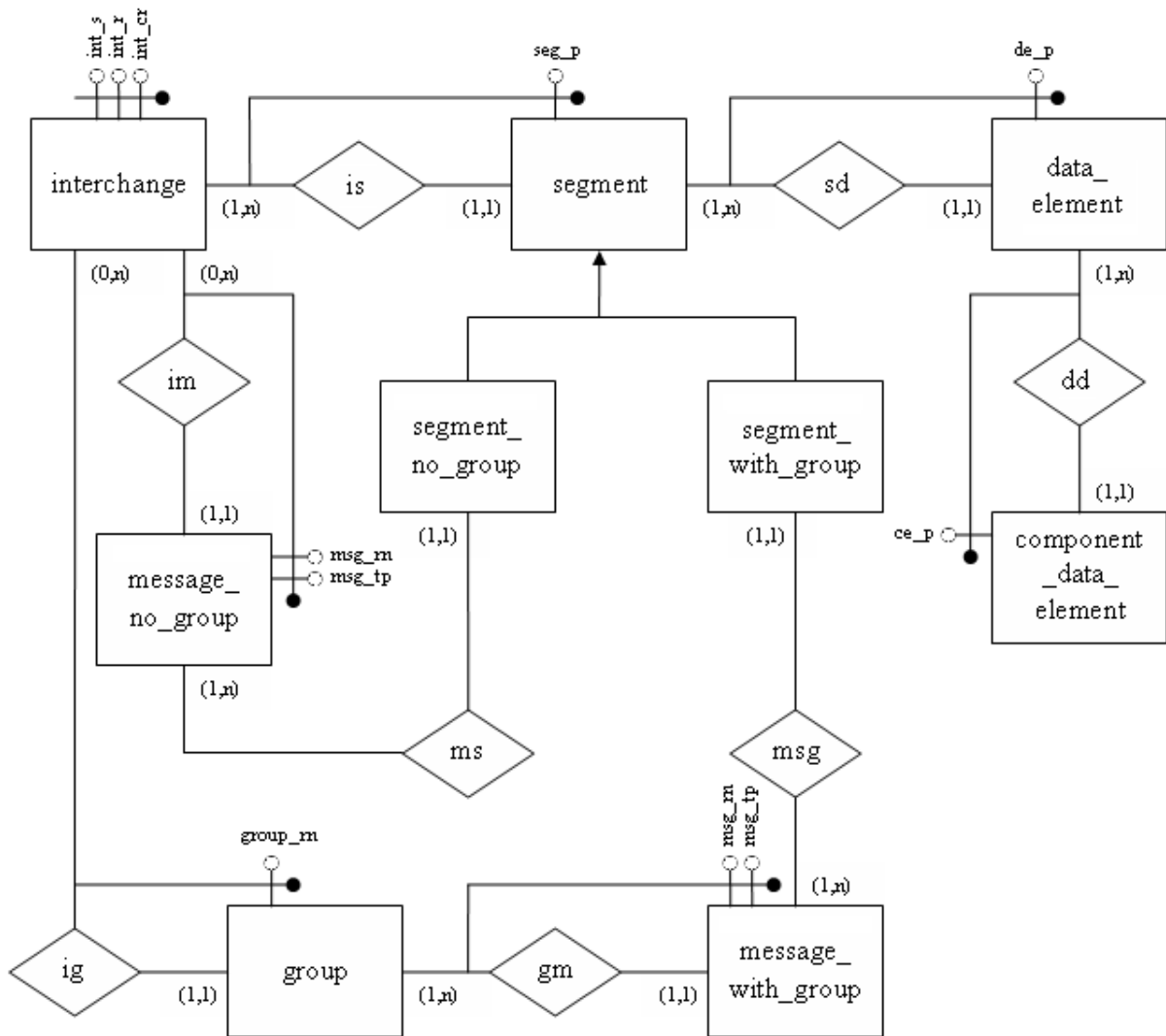


Figura 5: Schema E/R del database EDIFACT

Nello schema sono stati messi in evidenza solo gli attributi identificatori per maggior chiarezza; l'elenco completo degli attributi è nella Documentazione degli schemi relazionali per estrattore di dati EDIFACT riportato di seguito.

Con i data_element si considerano sia elementi dati semplici sia quelli complessi; gli elementi dati semplici hanno un solo componente di elementi dati (component_data_element), mentre elementi dati complessi hanno più di un componente di elementi dati.

3.1.1.1 Documentazione degli schemi relazionali per estrattore di dati EDIFACT

Entità	Descrizione	Attributi	Identificatore
interchange	Primo livello del messaggio	<ul style="list-style-type: none"> • int_s • int_r • int_cr • syn_id • int_dtp 	<ul style="list-style-type: none"> • int_s • int_r • int_cr
group	Secondo livello del messaggio	<ul style="list-style-type: none"> • group_rn • appl_s • appl_r • group_dtp 	<ul style="list-style-type: none"> • int_s • int_r • int_cr • group_rn
message_with_group	Terzo livello del messaggio	<ul style="list-style-type: none"> • msg_rn • msg_tp 	<ul style="list-style-type: none"> • int_s • int_r • int_cr • group_rn • msg_rn • msg_tp
message_no_group	Il secondo livello del messaggio quando nel messaggio non c'è il livello group	<ul style="list-style-type: none"> • msg_rn • msg_tp 	<ul style="list-style-type: none"> • int_s • int_r • int_cr • msg_rn • msg_tp
segment	Segmento del messaggio	<ul style="list-style-type: none"> • seg_p • TAG • seg_nr 	<ul style="list-style-type: none"> • int_s • int_r • int_cr • seg_p
segment_with_group	Specializzazione del segmento per messaggi con livello group	<ul style="list-style-type: none"> • seg_pm • group_rn • msg_rn • msg_tp 	<ul style="list-style-type: none"> • int_s • int_r • int_cr • seg_p
segment_no_group	Specializzazione del segmento per messaggi senza livello group	<ul style="list-style-type: none"> • seg_pm • msg_rn • msg_tp 	<ul style="list-style-type: none"> • int_s • int_r • int_cr • seg_p
data_element	Gruppo di dati all'interno del segmento, può essere composto da 1 o più componenti di dati	<ul style="list-style-type: none"> • de_p 	<ul style="list-style-type: none"> • int_s • int_r • int_cr • seg_p • de_p
component_data_element	Dato indivisibile	<ul style="list-style-type: none"> • ce_p • ce_v 	<ul style="list-style-type: none"> • int_s • int_r

			<ul style="list-style-type: none"> • int_cr • seg_p • de_p • ce_p
--	--	--	---

Relazione	Descrizione	Entità coinvolte
ig	Associa a un'interchange ai propri group	<ul style="list-style-type: none"> • interchange (0,n) • group (1,1)
gm	Associa a un group ai propri message	<ul style="list-style-type: none"> • group (1,n) • message_with_group (1,1)
im	Associa a un'interchange ai propri message_no_group	<ul style="list-style-type: none"> • interchange (0,n) • message_no_group (1,1)
is	Associa a un'interchange ai propri segment	<ul style="list-style-type: none"> • interchange (1,n) • segment (1,1)
msg	Associa a un message_with_group ai propri segmenti (with group)	<ul style="list-style-type: none"> • message_with_group (1,n) • segment_with_group (1,1)
ms	Associa a un message senza gruppo ai propri segmenti (no group)	<ul style="list-style-type: none"> • message_no_group (1,n) • segment_no_group (1,1)
sd	Associa a un segment ai propri data element	<ul style="list-style-type: none"> • segment (1,n) • data_group(1,1)
dd	Associa a un data element ai propri componenti	<ul style="list-style-type: none"> • data_group (1,n) • data_element (1,1)

La seguente tabella indica la posizione da cui si ricavano i vari attributi dello schema, oltre alla loro descrizione.

Attributo	Descrizione	Nome dato in EDIFACT	Posizione del dato
int_s	Interchange sender, identifica il mittente dell'interchange.	S002	UNB:020
int_r	Interchange recipient, indica la destinazione dell'interchange.	S003	UNB:030
int_cr	Interchange control reference, riferimento univoco assegnato dal mittente all'interchange	0020	UNB:050
syn_id	Syntax identifier, indica la sintassi usato nell'interchange attraverso il codice	0001;0002 (S001)	UNB:010

	del agency controlling e il numero della versione		
int_dtp	Date and time of preparation of the interchange.	S004	UNB:040
group_rn	Group reference number, numero univoco all'interno del group.	0048	UNG:50
appl_s	Application sender, identifica il mittente del group.	S006	UNG:020
appl_r	Application recipient, indica la destinazione del group.	S007	UNG:030
group_dtp	Date and time of preparation of the group. (Se non c'è significa che è uguale a quello dell'interchange)	S004	UNG:40
msg_rn	Message reference number, riferimento univoco del messaggio assegnato dal mittente	0062	UNH:010
msg_id	Message identifier, identifica il tipo, la versione e altri dati riguardati al tipo del messaggio.	S009	UNH:020
seg_p	La posizione del segmento all'interno dell'interchange		
TAG	L'inizio del segmento, formato da 3 lettere e identifica il tipo di segmento.		
seg_nr	Informazioni sulla gerarchia e ripetizioni		
de_p	La posizione del data element nel segmento.		
ce_p	La posizione del component data element all'interno del data group.		
ce_v	Il valore del component data element.		

Con lo schema E/R presentato si mantiene la struttura del file EDIFACT, i vari livelli del file, inoltre si tiene conto che il livello di gruppo funzionale è opzionale.

Nello schema E/R la posizione degli elementi dati sono memorizzate con de_p e ce_p. Oltre agli identificatori, le posizioni degli elementi dati e i TAG, può essere utile conoscere la sintassi utilizzato attraverso syn_id, la data e l'orario della preparazione, e la gerarchia e ripetizioni dei segmenti.

3.1.1.2 Schema logico relazionale del database EDIFACT

Dallo schema E/R precedente si può ricavare, seguendo la traduzione di progetto logico in **Progetto di Basi di Dati Relazionale** [1], il seguente schema logico:

interchange(int_s, int_r, int_cr, syn_id, int_dtp)

group(int_s, int_r, int_cr, group_rn, appl_s, appl_r, group_dtp)

FK: int_s, int_r, int_cr **references** interchange

message_with_group(int_s, int_r, int_cr, group_rn, msg_rn, msg_tp)

FK: int_s, int_r, int_cr, appl_s, appl_r, group_rn **references** group

message_no_group(int_s, int_r, int_cr, msg_rn, msg_tp)

FK: int_s, int_r, int_cr **references** interchange

segment(int_s, int_r, int_cr, seg_p, TAG, seg_nr)

FK: int_s, int_r, int_cr **references** interchange

segment_with_group(int_s, int_r, int_cr, seg_p, group_rn, msg_rn, msg_id, seg_pm)

FK: int_s, int_r, int_cr, seg_p **references** segment

FK: group_rn, msg_rn, msg_id **references** message_with_group

AK: int_s, int_r, int_cr, group_rn, msg_rn, msg_id, seg_pm

segment_no_group(int_s, int_r, int_cr, msg_rn, msg_id, seg_p, seg_pm)

FK: int_s, int_r, int_cr, seg_p **references** segment

FK: msg_rn, msg_id **references** message_no_group

AK: int_s, int_r, int_cr, group_rn, msg_rn, msg_id, seg_pm

data_element(int_s, int_r, int_cr, seg_p, de_p)

FK: int_s, int_r, int_cr, seg_p **references** segment

component_data_element(int_s, int_r, int_cr, seg_p, de_p, ce_p, ce_v)

FK: int_s, int_r, int_cr, seg_p, dg pos **references** data group

Nella traduzione allo schema logico relazionale è stato possibile eliminare tutte le associazioni, semplificando notevolmente lo schema.

L'entità *message* viene tradotta attraverso la tecnica del collasso verso il basso nelle due entità figlie *message_with_group* e *message_no_group*.

3.1.1.3 Le istruzioni sql 92 per la creazione del database EDIFACT

Infine dallo schema logico è stato possibile ricavare le seguenti istruzioni per creare le tabelle dal database EDIFACT; una volta inserite le tabelle è possibile inserire i file di input creati dall'estrattore.

```
create table interchange (  
int_s varchar(112) not null,  
int_r varchar(112) not null,  
int_cr varchar(14) not null,  
syn_id varchar(17),  
int_dtp datetime,  
primary key (int_s, int_r, int_cr)  
)  
go
```

```
create table [group] (  
int_s varchar(112),  
int_r varchar(112),  
int_cr varchar(14),  
group_rn varchar(14) not null,  
appl_s varchar(40),  
appl_r varchar(40),  
group_dtp datetime,  
foreign key (int_s, int_r, int_cr) references interchange (int_s, int_r, int_cr),  
primary key (int_s, int_r, int_cr, group_rn)  
)  
go
```

```
create table message_with_group (  
int_s varchar(112),  
int_r varchar(112),  
int_cr varchar(14),  
group_rn varchar(14),  
msg_rn varchar(14) not null,  
msg_id varchar(39) not null,  
foreign key (int_s, int_r, int_cr, group_rn) references [group] (int_s, int_r, int_cr,  
group_rn),  
primary key (int_s, int_r, int_cr, group_rn, msg_rn, msg_id)  
)  
go
```



```

create table message_no_group (
int_s varchar(112),
int_r varchar(112),
int_cr varchar(14),
msg_rn varchar(14) not null,
msg_id varchar(39) not null,
foreign key (int_s, int_r, int_cr) references interchange (int_s, int_r, int_cr),
primary key (int_s, int_r, int_cr, msg_rn, msg_id)
)
go

```

```

create table segment (
int_s varchar(112),
int_r varchar(112),
int_cr varchar(14),
seg_p int not null,
TAG char(3),
seg_nr varchar(64),
foreign key (int_s, int_r, int_cr) references interchange (int_s, int_r, int_cr),
primary key (int_s, int_r, int_cr, seg_p)
)
go

```

```

create table segment_with_group (
int_s varchar(112),
int_r varchar(112),
int_cr varchar(14),
group_rn varchar(14),
msg_rn varchar(14),
msg_id varchar(39),
seg_p int,
seg_pm int not null,
foreign key (int_s, int_r, int_cr, seg_p) references segment (int_s, int_r, int_cr, seg_p),
foreign key (int_s, int_r, int_cr, group_rn, msg_rn, msg_id) references
message_with_group (int_s, int_r, int_cr, group_rn, msg_rn, msg_id),
primary key (int_s, int_r, int_cr, seg_p),
unique (int_s, int_r, int_cr, group_rn, msg_rn, msg_id, seg_pm)
)
go

```

```

create table segment_no_group (
int_s varchar(112),
int_r varchar(112),
int_cr varchar(14),
msg_rn varchar(14),
msg_id varchar(39),
seg_p int,

```

```

seg_pm int not null,
foreign key (int_s, int_r, int_cr, seg_p) references segment (int_s, int_r, int_cr, seg_p),
foreign key (int_s, int_r, int_cr, msg_rn, msg_id) references message_no_group (int_s,
int_r, int_cr, msg_rn, msg_id),
primary key (int_s, int_r, int_cr, seg_p),
unique (int_s, int_r, int_cr, msg_rn, msg_id, seg_pm)
)
go

```

```

create table data_element (
int_s varchar(112),
int_r varchar(112),
int_cr varchar(14),
seg_p int,
de_p int not null,
foreign key (int_s, int_r, int_cr, seg_p) references segment (int_s, int_r, int_cr, seg_p),
primary key (int_s, int_r, int_cr, seg_p, de_p)
)
go

```

```

create table component_data_element (
int_s varchar(112),
int_r varchar(112),
int_cr varchar(14),
seg_p int,
de_p int,
ce_p int not null,
ce_v varchar(64),
foreign key (int_s, int_r, int_cr, seg_p, de_p) references data_element (int_s, int_r,
int_cr, seg_p, de_p),
primary key (int_s, int_r, int_cr, seg_p, de_p, ce_p)
)
go

```

3.1.2 Estrattore dati EDIFACT: descrizione dettagliata

L'estrattore dati deve estrarre i dati dal file EDIFACT e produrre un file di input per database EDIFACT.

Il programma è stato sviluppato in linguaggio JAVA, e le classi creati sono:

- estrattoreEDIFACT nel quale è contenuto il main del programma che chiama l'oggetto finestraEDIFACT;
- finestraEDIFACT è l'interfaccia grafica dell'estrattore, inoltre contiene gli ActionListener dell'interfaccia;

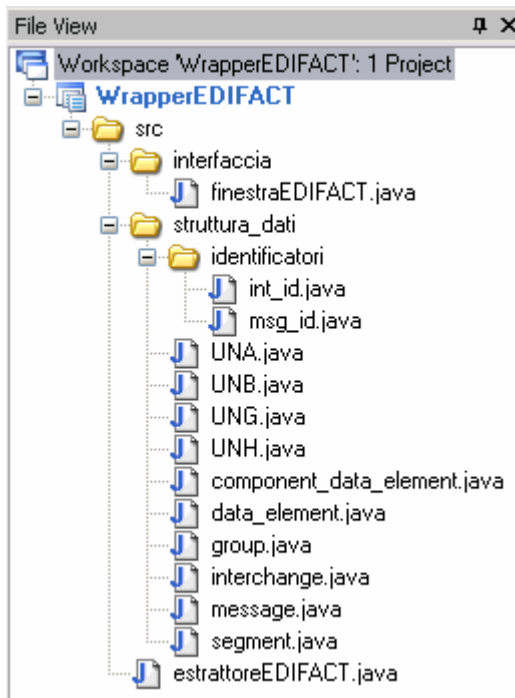


Figura 6: Classi del wrapper EDIFACT

msg_rn e msg_tp; msg_id deve essere usati insieme agli attributi di int_id ed eventualmente l'attributo identificatore del gruppo (se c'è il livello gruppo) per essere usato come identificatore dei messaggi.

- interchange, group, e message sono le classi che hanno il compito di analizzare i vari livelli del file EDIFACT;
- segment è la classe che analizza la struttura dei singoli segmenti;
- data_element è la classe che analizza gli elementi dati;
- component_data_element è la classe che estrae i valori dei componenti degli elementi dati;
- UNA, UNB, UNG e UNH sono classi derivate dalla classe segment, e servono ad analizzare i segmenti header del file EDIFACT;
- int_id e msg_id sono classi che contengono gli attributi identificatori estratti dai segmenti header. Si noti che nel msg_id ci sono attributi identificatori estratti dal UNH, perciò contiene solo

3.1.2.1 Interfaccia grafica dell'estrattore di dati EDIFACT

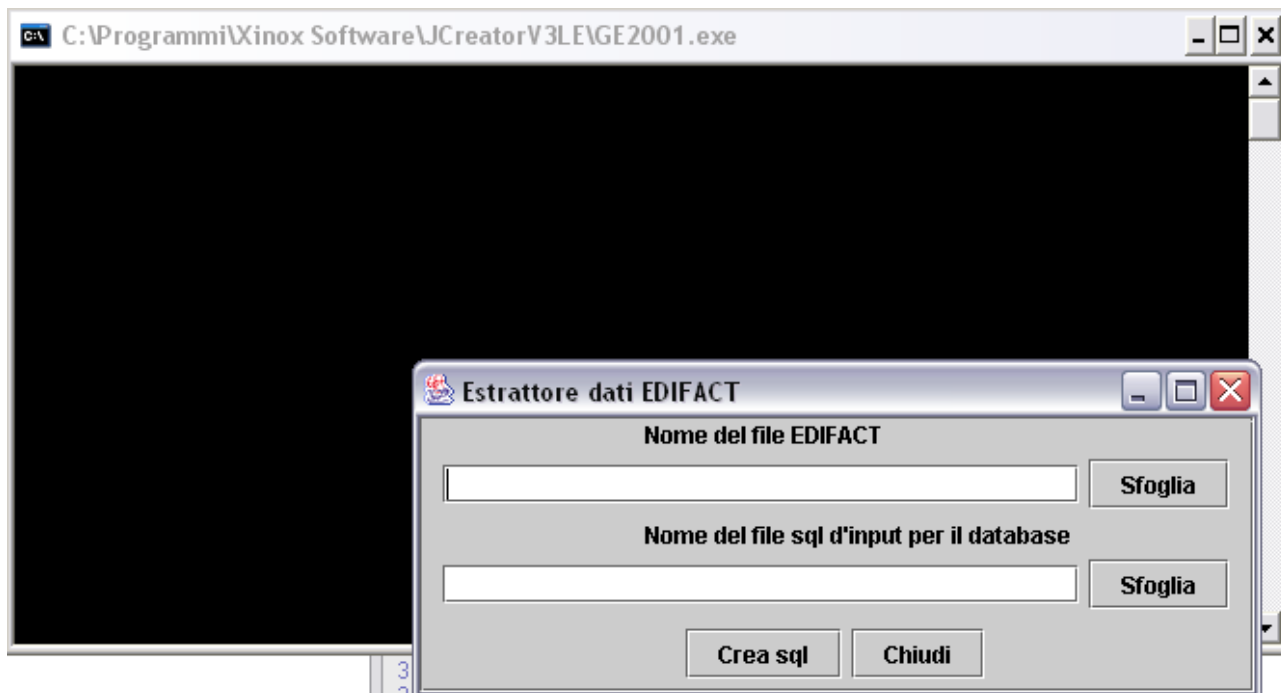


Figura 7: Interfaccia del wrapper EDIFACT

Una volta inserito i nomi dei file basta premere “Crea sql” per creare un file sql, nel quale sono contenuti i dati d’input per database.

3.1.2.2 Esempi di estrazione

Esempio 1:

Traduzione dei header.

```
UNB+UNOB:1+003897733:01:MFGB+PARTNER ID:ZZ:ROUTING  
ADDR+970101:1230+0000000000001++ORDERS++++1  
UNH+0001+ORDERS:D:04B:UN
```

Questi due segmenti sono i due header di un file EDIFACT senza il livello di gruppo, e viene estratto nel seguente modo:

```
insert into [interchange] values('003897733:01:MFGB:', 'PARTNER  
ID:ZZ:ROUTING ADDR:', '00000000000001', 'UNOB:1::', '970101 12:30')  
Go
```

--Inserimento dell'interchange

```
insert into [segment] values('003897733:01:MFGB:', 'PARTNER  
ID:ZZ:ROUTING ADDR:', '00000000000001', 1, 'UNB', '')  
insert into [data_element] values('003897733:01:MFGB:', 'PARTNER  
ID:ZZ:ROUTING ADDR:', '00000000000001', 1, 1)
```

```

insert into [component_data_element]
values('003897733:01:MFGB:', 'PARTNER ID:ZZ:ROUTING
ADDR:', '00000000000001', 1, 1, 1, 'UNOB')
insert into [component_data_element]
values('003897733:01:MFGB:', 'PARTNER ID:ZZ:ROUTING
ADDR:', '00000000000001', 1, 1, 2, '1')
insert into [data_element] values('003897733:01:MFGB:', 'PARTNER
ID:ZZ:ROUTING ADDR:', '00000000000001', 1, 2)
insert into [component_data_element]
values('003897733:01:MFGB:', 'PARTNER ID:ZZ:ROUTING
ADDR:', '00000000000001', 1, 2, 1, '003897733')
insert into [component_data_element]
values('003897733:01:MFGB:', 'PARTNER ID:ZZ:ROUTING
ADDR:', '00000000000001', 1, 2, 2, '01')
insert into [component_data_element]
values('003897733:01:MFGB:', 'PARTNER ID:ZZ:ROUTING
ADDR:', '00000000000001', 1, 2, 3, 'MFGB')
insert into [data_element] values('003897733:01:MFGB:', 'PARTNER
ID:ZZ:ROUTING ADDR:', '00000000000001', 1, 3)
insert into [component_data_element]
values('003897733:01:MFGB:', 'PARTNER ID:ZZ:ROUTING
ADDR:', '00000000000001', 1, 3, 1, 'PARTNER ID')
insert into [component_data_element]
values('003897733:01:MFGB:', 'PARTNER ID:ZZ:ROUTING
ADDR:', '00000000000001', 1, 3, 2, 'ZZ')
insert into [component_data_element]
values('003897733:01:MFGB:', 'PARTNER ID:ZZ:ROUTING
ADDR:', '00000000000001', 1, 3, 3, 'ROUTING ADDR')
insert into [data_element] values('003897733:01:MFGB:', 'PARTNER
ID:ZZ:ROUTING ADDR:', '00000000000001', 1, 4)
insert into [component_data_element]
values('003897733:01:MFGB:', 'PARTNER ID:ZZ:ROUTING
ADDR:', '00000000000001', 1, 4, 1, '970101')
insert into [component_data_element]
values('003897733:01:MFGB:', 'PARTNER ID:ZZ:ROUTING
ADDR:', '00000000000001', 1, 4, 2, '1230')
insert into [data_element] values('003897733:01:MFGB:', 'PARTNER
ID:ZZ:ROUTING ADDR:', '00000000000001', 1, 5)
insert into [component_data_element]
values('003897733:01:MFGB:', 'PARTNER ID:ZZ:ROUTING
ADDR:', '00000000000001', 1, 5, 1, '00000000000001')
insert into [data_element] values('003897733:01:MFGB:', 'PARTNER
ID:ZZ:ROUTING ADDR:', '00000000000001', 1, 6)
insert into [component_data_element]
values('003897733:01:MFGB:', 'PARTNER ID:ZZ:ROUTING
ADDR:', '00000000000001', 1, 6, 1, '')
insert into [data_element] values('003897733:01:MFGB:', 'PARTNER
ID:ZZ:ROUTING ADDR:', '00000000000001', 1, 7)
insert into [component_data_element]
values('003897733:01:MFGB:', 'PARTNER ID:ZZ:ROUTING
ADDR:', '00000000000001', 1, 7, 1, 'ORDERS')
insert into [data_element] values('003897733:01:MFGB:', 'PARTNER
ID:ZZ:ROUTING ADDR:', '00000000000001', 1, 8)
insert into [component_data_element]
values('003897733:01:MFGB:', 'PARTNER ID:ZZ:ROUTING
ADDR:', '00000000000001', 1, 8, 1, '')
insert into [data_element] values('003897733:01:MFGB:', 'PARTNER
ID:ZZ:ROUTING ADDR:', '00000000000001', 1, 9)

```

```

insert into [component_data_element]
values('003897733:01:MFGB:', 'PARTNER ID:ZZ:ROUTING
ADDR:', '00000000000001', 1, 9, 1, '')
insert into [data_element] values('003897733:01:MFGB:', 'PARTNER
ID:ZZ:ROUTING ADDR:', '00000000000001', 1, 10)
insert into [component_data_element]
values('003897733:01:MFGB:', 'PARTNER ID:ZZ:ROUTING
ADDR:', '00000000000001', 1, 10, 1, '')
insert into [data_element] values('003897733:01:MFGB:', 'PARTNER
ID:ZZ:ROUTING ADDR:', '00000000000001', 1, 11)
insert into [component_data_element]
values('003897733:01:MFGB:', 'PARTNER ID:ZZ:ROUTING
ADDR:', '00000000000001', 1, 11, 1, '1')
Go

```

--Inserimento del segmento UNB con i suoi elementi dati.

```

insert into [segment] values('003897733:01:MFGB:', 'PARTNER
ID:ZZ:ROUTING ADDR:', '00000000000001', 2, 'UNB', '')
insert into [data_element] values('003897733:01:MFGB:', 'PARTNER
ID:ZZ:ROUTING ADDR:', '00000000000001', 2, 1)
insert into [component_data_element]
values('003897733:01:MFGB:', 'PARTNER ID:ZZ:ROUTING
ADDR:', '00000000000001', 2, 1, 1, '0001')
insert into [data_element] values('003897733:01:MFGB:', 'PARTNER
ID:ZZ:ROUTING ADDR:', '00000000000001', 2, 2)
insert into [component_data_element]
values('003897733:01:MFGB:', 'PARTNER ID:ZZ:ROUTING
ADDR:', '00000000000001', 2, 2, 1, 'ORDERS')
insert into [component_data_element]
values('003897733:01:MFGB:', 'PARTNER ID:ZZ:ROUTING
ADDR:', '00000000000001', 2, 2, 2, 'D')
insert into [component_data_element]
values('003897733:01:MFGB:', 'PARTNER ID:ZZ:ROUTING
ADDR:', '00000000000001', 2, 2, 3, '04B')
insert into [component_data_element]
values('003897733:01:MFGB:', 'PARTNER ID:ZZ:ROUTING
ADDR:', '00000000000001', 2, 2, 4, 'UN')
GO

```

--Inserimento del segmento UNH con i suoi elementi dati.

```

insert into [message_no_group] values('003897733:01:MFGB:', 'PARTNER
ID:ZZ:ROUTING ADDR:', '00000000000001', '0001', 'ORDERS:D:04B:UN:::')
Go

```

--Inserimento del messaggio senza gruppo

Si può notare che le informazioni del UNB vengono usate come identificatori, inoltre negli attributi identificatori del messaggio senza gruppo si usano gli elementi dati del UNH.

Esempio 2:

Traduzione di un segmento del messaggio.

DTM+4.200404301742PDT:303'

```

insert into [segment] values('003897733:01:MFGB:', 'PARTNER
ID:ZZ:ROUTING ADDR:', '00000000000001', 4, 'DTM', '')

```

```

insert into [data_element] values('003897733:01:MFGB:', 'PARTNER
ID:ZZ:ROUTING ADDR:', '00000000000001', 4, 1)
insert into [component_data_element]
values('003897733:01:MFGB:', 'PARTNER ID:ZZ:ROUTING
ADDR:', '00000000000001', 4, 1, 1, '4')
insert into [component_data_element]
values('003897733:01:MFGB:', 'PARTNER ID:ZZ:ROUTING
ADDR:', '00000000000001', 4, 1, 2, '200404301742PDT')
insert into [component_data_element]
values('003897733:01:MFGB:', 'PARTNER ID:ZZ:ROUTING
ADDR:', '00000000000001', 4, 1, 3, '303')
insert into [segment_no_group] values('003897733:01:MFGB:', 'PARTNER
ID:ZZ:ROUTING ADDR:', '00000000000001', '0001', 'ORDERS:D:04B:UN::', 4, 2)
Go

```

Visto che il segmento appartiene a un file EDIFACT senza gruppi il segmento viene messo in relazione al *messaggio senza gruppo* attraverso `segment_no_group`.

Esempio 3:

Traduzione del header UNA.

UNA:+.?

```

insert into [segment]
values('7349734757:12::', '5033075000007:14::', '000000000000627', 1, 'UNA',
, '')
insert into [data_element]
values('7349734757:12::', '5033075000007:14::', '000000000000627', 1, 1)
insert into [component_data_element]
values('7349734757:12::', '5033075000007:14::', '000000000000627', 1, 1, 1, '
:+.?'')
GO

```

--Nell'estrazione è presente " " a causa della simbologia del sql server

3.1.2.3 Esempi di query sql

Una volta eseguita la creazione del file e l'inserimento nel database, è possibile effettuare le query al database.

Ecco alcuni esempi:

--Query (n1) per leggere i TAG di un determinato interchange:

```

select TAG from segment
where int_s = '003897733:01:MFGB:' and int_r = 'PARTNER ID:ZZ:ROUTING
ADDR:'
      and int_cr = '00000000000001'
order by seg_p
GO

```

	TAG
1	UNB
2	UNH
3	BGM
4	DTM
5	FTX
6	RFF
7	RFF
8	NAD
9	CTA
10	NAD
11	CTA
12	NAD
13	TAX
14	CUX
15	PCD
16	TDT
17	LOC
18	TOD
19	LIN
20	PIA
21	IMD
22	QTY

Figura 8: Risultato dell'esempio query n1

--Query (n2) per leggere i TAG di un determinato messaggio all'interno di un determinato interchange (senza livello group) e la posizione del segmento all'interno del messaggio:

```

select s.TAG,sg.seg_pm
from segment s left join segment_no_group sg on (s.int_s = sg.int_s and s.int_r =
sg.int_r and s.int_cr = sg.int_cr and s.seg_p = sg.seg_p)
where sg.int_s = '003897733:01:MFGB:' and sg.int_r = 'PARTNER ID:ZZ:ROUTING
ADDR:'
      and sg.int_cr = '00000000000001'
      and sg.msg_rn = '0001' and sg.msg_id = 'ORDERS:D:04B:UN:::'
order by s.seg_p
GO

```


	TAG	seg_pm
1	BGM	1
2	DTM	2
3	FTX	3
4	RFF	4
5	RFF	5
6	NAD	6
7	CTA	7
8	NAD	8
9	CTA	9
10	NAD	10
11	TAX	11
12	CUX	12
13	PCD	13
14	TDT	14
15	LOC	15
16	TOD	16
17	LIN	17
18	PIA	18
19	IMD	19
20	QTY	20
21	DTM	21

Figura 9: Risultato dell'esempio query n2

A differenza della query precedente, in questa non ci sono i header.

--Query (n3) per leggere i TAG di un determinato messaggio all'interno di un determinato interchange (senza livello group), la posizione del segmento all'interno del messaggio e i valori dei componenti dati con la relativa posizione:

```

select s.TAG,sg.seg_pm,de.de_p,ce.ce_p,ce.ce_v
from ((segment s left join segment_no_group sg on (s.int_s = sg.int_s and s.int_r =
sg.int_r and s.int_cr = sg.int_cr and s.seg_p = sg.seg_p))
      join data_element de on (s.int_s = de.int_s and s.int_r = de.int_r and s.int_cr =
de.int_cr and s.seg_p = de.seg_p))
      join component_data_element ce on (s.int_s = ce.int_s and s.int_r = ce.int_r and
s.int_cr = ce.int_cr and s.seg_p = ce.seg_p and de.de_p = ce.de_p)
where s.int_s = '003897733:01:MFGB:' and s.int_r = 'PARTNER ID:ZZ:ROUTING
ADDR:' and s.int_cr = '00000000000001'
      and sg.msg_rn = '0001' and sg.msg_id = 'ORDERS:D:04B:UN:::'
order by s.seg_p,de.de_p,ce.ce_p
GO

```

	TAG	seg_pm	de_p	ce_p	ce_v
1	BGM	1	1	1	221
2	BGM	1	2	1	ABC123456XYZ
3	BGM	1	3	1	9
4	DTM	2	1	1	4
5	DTM	2	1	2	200404301742PDT
6	DTM	2	1	3	303
7	FTX	3	1	1	PUR
8	FTX	3	2	1	3
9	FTX	3	3	1	
10	FTX	3	4	1	Pizza purchase order
11	RFF	4	1	1	CT
12	RFF	4	1	2	123-456
13	RFF	5	1	1	CR
14	RFF	5	1	2	1122
15	NAD	6	1	1	SE
16	NAD	6	2	1	999
17	NAD	6	2	2	
18	NAD	6	2	3	92

Figura 10: Risultato dell'esempio query n3

--Query (n4) per leggere il TAG e il valore dei componenti dei dati dei segmenti all'interno di un determinato messaggio con lo stesso TAG:

```

select s.TAG,s.seg_p,sg.seg_pm,de.de_p,ce.ce_p,ce.ce_v
from ((segment s left join segment_no_group sg on (s.int_s = sg.int_s and s.int_r =
sg.int_r and s.int_cr = sg.int_cr and s.seg_p = sg.seg_p))
      join data_element de on (s.int_s = de.int_s and s.int_r = de.int_r and s.int_cr =
de.int_cr and s.seg_p = de.seg_p))
      join component_data_element ce on (s.int_s = ce.int_s and s.int_r = ce.int_r and
s.int_cr = ce.int_cr and s.seg_p = ce.seg_p and de.de_p = ce.de_p)
where s.int_s = '003897733:01:MFGB:' and s.int_r = 'PARTNER ID:ZZ:ROUTING
ADDR:'
      and s.int_cr = '000000000000001'
      and s.TAG = 'BGM'
      and sg.msg_rn = '0001' and sg.msg_id = 'ORDERS:D:04B:UN:::'
order by s.seg_p,de.de_p,ce.ce_p
GO

```

	TAG	seg_p	seg_pm	de_p	ce_p	ce_v
1	BGM	3	1	1	1	221
2	BGM	3	1	2	1	ABC123456XYZ
3	BGM	3	1	3	1	9

Figura 11: Risultato dell'esempio query n4

--Query (n5) per leggere il TAG e il valore dei componenti dei dati di un determinato segmento all'interno di un determinato messaggio dando la posizione del segmento all'interno del messaggio:

```

select s.TAG,de.de_p,ce.ce_p,ce.ce_v
from ((segment s left join segment_no_group sg on (s.int_s = sg.int_s and s.int_r =
sg.int_r and s.int_cr = sg.int_cr and s.seg_p = sg.seg_p))
      join data_element de on (s.int_s = de.int_s and s.int_r = de.int_r and s.int_cr =
de.int_cr and s.seg_p = de.seg_p))
      join component_data_element ce on (s.int_s = ce.int_s and s.int_r = ce.int_r and
s.int_cr = ce.int_cr and s.seg_p = ce.seg_p and de.de_p = ce.de_p)
where s.int_s = '003897733:01:MFGB:' and s.int_r = 'PARTNER ID:ZZ:ROUTING
ADDR:' and s.int_cr = '00000000000001'
      and sg.msg_rn = '0001' and sg.msg_id = 'ORDERS:D:04B:UN:::'
      and sg.seg_pm = 18
order by de.de_p,ce.ce_p
GO

```

	TAG	de_p	ce_p	ce_v
1	PIA 1	1	1	
2	PIA 2	1	Pizza	
3	PIA 2	2	EF	
4	PIA 3	1	42	
5	PIA 3	2	VP	

Figura 12: Risultato dell'esempio query n5

Si noti come la possibilità di eseguire query renda effettivamente fruibile l'informazione contenuta (che è praticamente illeggibile) nel file EDIFACT.

3.2 Le viste

Attraverso l'estrattore dei dati si può estrarre elementi dati conservando le informazioni sulla loro posizione, conservare le informazioni sulla struttura del file ed estrarre alcune informazioni dei header del file. Però l'estrattore non dà significato agli elementi dati nè le loro associazioni se non osservando le posizioni degli elementi dati.

Per avere una migliore visione degli elementi dati e le loro associazioni si possono usare le viste sql, i quali permettono di dare un significato agli elementi dati.

Però i significati degli elementi dati possono essere differenti a seconda dei tipi di messaggi, che sono circa 200, oltre ai tipi di messaggi creati dagli utenti.

Inoltre gli utenti possono dare un proprio significato ad un elemento dato, ad esempio il segmento FTX, nel quale l'utente può inserire le informazioni scelte da egli, perciò le viste devono quindi essere create appositamente per ogni diversa applicazione dell'utente.

Di seguito si farà un semplice esempio di viste applicato ai messaggi del tipo ORDERS. Nell'esempio si farà riferimento ad un unico mittente dell'interchange, cioè per utenti diversi si possono usare viste sql diverse. Naturalmente utenti diversi con lo stesso tipo di messaggio si possono accordarsi per condividere la stessa vista.

Le prime tre viste cercano di ricavare i dati dagli elementi dati presenti nella database, per fare ciò bisogna specificare le posizioni dei componenti di elementi dati. La quarta vista è un'elaborazione dei dati ricavati dalla terza vista.

3.2.1 Vista messaggio_ordine

Questa vista consente all'utente di visualizzare il numero dell'ordine e la data in cui è stato effettuato l'ordine.

```
create view messaggio_ordine
as select msng.int_s, msng.int_r, msng.int_cr, msng.msg_id, msng.msg_rn,
--numero ordine
(select ce.ce_v
from ((segment s left join segment_no_group sg on (s.int_s = sg.int_s and s.int_r =
sg.int_r and s.int_cr = sg.int_cr and s.seg_p = sg.seg_p))
join data_element de on (s.int_s = de.int_s and s.int_r = de.int_r and s.int_cr = de.int_cr
and s.seg_p = de.seg_p))
join component_data_element ce on (s.int_s = ce.int_s and s.int_r = ce.int_r and
s.int_cr = ce.int_cr and s.seg_p = ce.seg_p and de.de_p = ce.de_p)
where s.int_s = msng.int_s
and s.int_r = msng.int_r
and s.int_cr = msng.int_cr
and msg_id = msng.msg_id
and msg_rn = msng.msg_rn
and s.TAG = 'BGM'
and ce.de_p = 2 and ce.ce_p = 1)
as 'numero ordine',
--data
(select ce.ce_v
from ((segment s left join segment_no_group sg on (s.int_s = sg.int_s and s.int_r =
sg.int_r and s.int_cr = sg.int_cr and s.seg_p = sg.seg_p))
join data_element de on (s.int_s = de.int_s and s.int_r = de.int_r and s.int_cr = de.int_cr
and s.seg_p = de.seg_p))
join component_data_element ce on (s.int_s = ce.int_s and s.int_r = ce.int_r and
s.int_cr = ce.int_cr and s.seg_p = ce.seg_p and de.de_p = ce.de_p)
```

```

where s.int_s = msng.int_s
      and s.int_r = msng.int_r
      and s.int_cr = msng.int_cr
      and msg_id = msng.msg_id
      and msg_rn = msng.msg_rn
      and s.TAG = 'DTM'
      and ce.de_p = 1 and ce.ce_p = 2
      and s.seg_p in (
          select ce.seg_p
          from ((segment s left join segment_no_group sg on
                (s.int_s = sg.int_s and s.int_r = sg.int_r and s.int_cr =
                 sg.int_cr and s.seg_p = sg.seg_p))
               join data_element de on (s.int_s = de.int_s and s.int_r =
                de.int_r and s.int_cr = de.int_cr and s.seg_p = de.seg_p))
               join component_data_element ce on (s.int_s = ce.int_s
                and s.int_r = ce.int_r and s.int_cr = ce.int_cr and s.seg_p
                = ce.seg_p and de.de_p = ce.de_p)
          where s.int_s = msng.int_s
                and s.int_r = msng.int_r
                and s.int_cr = msng.int_cr
                and msg_id = msng.msg_id
                and msg_rn = msng.msg_rn
                and s.TAG = 'DTM'
                and ce.de_p = 1 and ce.ce_p = 1
                and ce.ce_v = '4'))
as data

from message_no_group msng
where msng.int_s = '003897733:01:MFGB:'
      and msng.msg_id = 'ORDERS:D:04B:UN:::'
GO

```

	int_s	int_r	int_cr	msg_id	msg_rn	numero ordine	data
1	003897733:01:MFGB:	PARTNER ID:ZZ:ROUTING ADDR:	000000000000001	ORDERS:D:04B:UN:::	0001	ABC123456XYZ	200404301742PDT
2	003897733:01:MFGB:	PARTNER ID:ZZ:ROUTING ADDR:	000000000000002	ORDERS:D:04B:UN:::	0001	ABC123457XYZ	200405301742PDT

Figura 13: Risultato della vista messaggio_ordine

In questa vista si può notare che per il segmento DTM si è dovuto specificare una ulteriore condizione, perché nel messaggio è presente più di un segmento DTM, con significati diversi.

3.2.2 Vista cliente_ordine

La seguente vista associa un cliente ad un messaggio dell'ordine, le informazioni del cliente vengono ricavate dal messaggio EDIFACT.

```

create view cliente_ordine
as select msng.int_s, msng.int_r, msng.int_cr, msng.msg_id, msng.msg_rn,

```

```

--numero cliente
(select ce.ce_v
from ((segment s left join segment_no_group sg on (s.int_s = sg.int_s and s.int_r =
sg.int_r and s.int_cr = sg.int_cr and s.seg_p = sg.seg_p))
join data_element de on (s.int_s = de.int_s and s.int_r = de.int_r and s.int_cr = de.int_cr
and s.seg_p = de.seg_p))
join component_data_element ce on (s.int_s = ce.int_s and s.int_r = ce.int_r and
s.int_cr = ce.int_cr and s.seg_p = ce.seg_p and de.de_p = ce.de_p)
where s.int_s = msng.int_s
      and s.int_r = msng.int_r
      and s.int_cr = msng.int_cr
      and msg_id = msng.msg_id
      and msg_rn = msng.msg_rn
      and s.TAG = 'NAD'
      and ce.de_p = 2 and ce.ce_p = 1
      and s.seg_p in (      select ce.seg_p
                        from ((segment s left join segment_no_group sg on
(s.int_s = sg.int_s and s.int_r = sg.int_r and s.int_cr =
sg.int_cr and s.seg_p = sg.seg_p))
join data_element de on (s.int_s = de.int_s and s.int_r =
de.int_r and s.int_cr = de.int_cr and s.seg_p = de.seg_p))
join component_data_element ce on (s.int_s = ce.int_s
and s.int_r = ce.int_r and s.int_cr = ce.int_cr and s.seg_p
= ce.seg_p and de.de_p = ce.de_p)
where s.int_s = msng.int_s
      and s.int_r = msng.int_r
      and s.int_cr = msng.int_cr
      and msg_id = msng.msg_id
      and msg_rn = msng.msg_rn
      and s.TAG = 'NAD'
      and ce.de_p = 1 and ce.ce_p = 1
      and ce.ce_v = 'BY'))

```

as 'numero cliente',

--nome contatto

```

(select ce.ce_v
from ((segment s left join segment_no_group sg on (s.int_s = sg.int_s and s.int_r =
sg.int_r and s.int_cr = sg.int_cr and s.seg_p = sg.seg_p))
join data_element de on (s.int_s = de.int_s and s.int_r = de.int_r and s.int_cr = de.int_cr
and s.seg_p = de.seg_p))
join component_data_element ce on (s.int_s = ce.int_s and s.int_r = ce.int_r and
s.int_cr = ce.int_cr and s.seg_p = ce.seg_p and de.de_p = ce.de_p)
where s.int_s = msng.int_s
      and s.int_r = msng.int_r
      and s.int_cr = msng.int_cr
      and msg_id = msng.msg_id
      and msg_rn = msng.msg_rn
      and s.TAG = 'CTA'

```

```

and ce.de_p = 2 and ce.ce_p = 2
and s.seg_p in (      select ce.seg_p
                    from ((segment s left join segment_no_group sg on
(s.int_s = sg.int_s and s.int_r = sg.int_r and s.int_cr =
sg.int_cr and s.seg_p = sg.seg_p))
join data_element de on (s.int_s = de.int_s and s.int_r =
de.int_r and s.int_cr = de.int_cr and s.seg_p = de.seg_p))
join component_data_element ce on (s.int_s = ce.int_s
and s.int_r = ce.int_r and s.int_cr = ce.int_cr and s.seg_p
= ce.seg_p and de.de_p = ce.de_p)
where s.int_s = msng.int_s
      and s.int_r = msng.int_r
      and s.int_cr = msng.int_cr
      and msg_id = msng.msg_id
      and msg_rn = msng.msg_rn
      and s.TAG = 'CTA'
      and ce.de_p = 1 and ce.ce_p = 1
      and ce.ce_v = 'PD'))

as 'nome contatto',
--nome compagnia
(select ce.ce_v
from ((segment s left join segment_no_group sg on (s.int_s = sg.int_s and s.int_r =
sg.int_r and s.int_cr = sg.int_cr and s.seg_p = sg.seg_p))
join data_element de on (s.int_s = de.int_s and s.int_r = de.int_r and s.int_cr = de.int_cr
and s.seg_p = de.seg_p))
join component_data_element ce on (s.int_s = ce.int_s and s.int_r = ce.int_r and
s.int_cr = ce.int_cr and s.seg_p = ce.seg_p and de.de_p = ce.de_p)
where s.int_s = msng.int_s
      and s.int_r = msng.int_r
      and s.int_cr = msng.int_cr
      and msg_id = msng.msg_id
      and msg_rn = msng.msg_rn
      and s.TAG = 'NAD'
      and ce.de_p = 4 and ce.ce_p = 1
      and s.seg_p in (      select ce.seg_p
                    from ((segment s left join segment_no_group sg on
(s.int_s = sg.int_s and s.int_r = sg.int_r and s.int_cr =
sg.int_cr and s.seg_p = sg.seg_p))
join data_element de on (s.int_s = de.int_s and s.int_r =
de.int_r and s.int_cr = de.int_cr and s.seg_p = de.seg_p))
join component_data_element ce on (s.int_s = ce.int_s
and s.int_r = ce.int_r and s.int_cr = ce.int_cr and s.seg_p
= ce.seg_p and de.de_p = ce.de_p)
where s.int_s = msng.int_s
      and s.int_r = msng.int_r
      and s.int_cr = msng.int_cr
      and msg_id = msng.msg_id

```

```

and msg_rn = msng.msg_rn
and s.TAG = 'NAD'
and ce.de_p = 1 and ce.ce_p = 1
and ce.ce_v = 'BY'))

as 'nome compagnia',
--indirizzo
(select ce.ce_v
from ((segment s left join segment_no_group sg on (s.int_s = sg.int_s and s.int_r =
sg.int_r and s.int_cr = sg.int_cr and s.seg_p = sg.seg_p))
join data_element de on (s.int_s = de.int_s and s.int_r = de.int_r and s.int_cr = de.int_cr
and s.seg_p = de.seg_p))
join component_data_element ce on (s.int_s = ce.int_s and s.int_r = ce.int_r and
s.int_cr = ce.int_cr and s.seg_p = ce.seg_p and de.de_p = ce.de_p)
where s.int_s = msng.int_s
and s.int_r = msng.int_r
and s.int_cr = msng.int_cr
and msg_id = msng.msg_id
and msg_rn = msng.msg_rn
and s.TAG = 'NAD'
and ce.de_p = 5 and ce.ce_p = 1
and s.seg_p in (
select ce.seg_p
from ((segment s left join segment_no_group sg on
(s.int_s = sg.int_s and s.int_r = sg.int_r and s.int_cr =
sg.int_cr and s.seg_p = sg.seg_p))
join data_element de on (s.int_s = de.int_s and s.int_r =
de.int_r and s.int_cr = de.int_cr and s.seg_p = de.seg_p))
join component_data_element ce on (s.int_s = ce.int_s
and s.int_r = ce.int_r and s.int_cr = ce.int_cr and s.seg_p
= ce.seg_p and de.de_p = ce.de_p)
where s.int_s = msng.int_s
and s.int_r = msng.int_r
and s.int_cr = msng.int_cr
and msg_id = msng.msg_id
and msg_rn = msng.msg_rn
and s.TAG = 'NAD'
and ce.de_p = 1 and ce.ce_p = 1
and ce.ce_v = 'BY'))

as 'indirizzo',
--città
(select ce.ce_v
from ((segment s left join segment_no_group sg on (s.int_s = sg.int_s and s.int_r =
sg.int_r and s.int_cr = sg.int_cr and s.seg_p = sg.seg_p))
join data_element de on (s.int_s = de.int_s and s.int_r = de.int_r and s.int_cr = de.int_cr
and s.seg_p = de.seg_p))
join component_data_element ce on (s.int_s = ce.int_s and s.int_r = ce.int_r and
s.int_cr = ce.int_cr and s.seg_p = ce.seg_p and de.de_p = ce.de_p)
where s.int_s = msng.int_s

```



```

and s.int_r = msng.int_r
and s.int_cr = msng.int_cr
and msg_id = msng.msg_id
and msg_rn = msng.msg_rn
and s.TAG = 'NAD'
and ce.de_p = 6 and ce.ce_p = 1
and s.seg_p in (
    select ce.seg_p
    from ((segment s left join segment_no_group sg on
(s.int_s = sg.int_s and s.int_r = sg.int_r and s.int_cr =
sg.int_cr and s.seg_p = sg.seg_p))
join data_element de on (s.int_s = de.int_s and s.int_r =
de.int_r and s.int_cr = de.int_cr and s.seg_p = de.seg_p))
join component_data_element ce on (s.int_s = ce.int_s
and s.int_r = ce.int_r and s.int_cr = ce.int_cr and s.seg_p
= ce.seg_p and de.de_p = ce.de_p)
where s.int_s = msng.int_s
and s.int_r = msng.int_r
and s.int_cr = msng.int_cr
and msg_id = msng.msg_id
and msg_rn = msng.msg_rn
and s.TAG = 'NAD'
and ce.de_p = 1 and ce.ce_p = 1
and ce.ce_v = 'BY'))
as 'città',
--stato
(select ce.ce_v
from ((segment s left join segment_no_group sg on (s.int_s = sg.int_s and s.int_r =
sg.int_r and s.int_cr = sg.int_cr and s.seg_p = sg.seg_p))
join data_element de on (s.int_s = de.int_s and s.int_r = de.int_r and s.int_cr = de.int_cr
and s.seg_p = de.seg_p))
join component_data_element ce on (s.int_s = ce.int_s and s.int_r = ce.int_r and
s.int_cr = ce.int_cr and s.seg_p = ce.seg_p and de.de_p = ce.de_p)
where s.int_s = msng.int_s
and s.int_r = msng.int_r
and s.int_cr = msng.int_cr
and msg_id = msng.msg_id
and msg_rn = msng.msg_rn
and s.TAG = 'NAD'
and ce.de_p = 7 and ce.ce_p = 1
and s.seg_p in (
    select ce.seg_p
    from ((segment s left join segment_no_group sg on
(s.int_s = sg.int_s and s.int_r = sg.int_r and s.int_cr =
sg.int_cr and s.seg_p = sg.seg_p))
join data_element de on (s.int_s = de.int_s and s.int_r =
de.int_r and s.int_cr = de.int_cr and s.seg_p = de.seg_p))

```

```

join component_data_element ce on (s.int_s = ce.int_s
and s.int_r = ce.int_r and s.int_cr = ce.int_cr and s.seg_p
= ce.seg_p and de.de_p = ce.de_p)
where s.int_s = msng.int_s
      and s.int_r = msng.int_r
      and s.int_cr = msng.int_cr
      and msg_id = msng.msg_id
      and msg_rn = msng.msg_rn
      and s.TAG = 'NAD'
      and ce.de_p = 1 and ce.ce_p = 1
      and ce.ce_v = 'BY'))
as 'stato',
--ZIP
(select ce.ce_v
from ((segment s left join segment_no_group sg on (s.int_s = sg.int_s and s.int_r =
sg.int_r and s.int_cr = sg.int_cr and s.seg_p = sg.seg_p))
join data_element de on (s.int_s = de.int_s and s.int_r = de.int_r and s.int_cr = de.int_cr
and s.seg_p = de.seg_p))
join component_data_element ce on (s.int_s = ce.int_s and s.int_r = ce.int_r and
s.int_cr = ce.int_cr and s.seg_p = ce.seg_p and de.de_p = ce.de_p)
where s.int_s = msng.int_s
      and s.int_r = msng.int_r
      and s.int_cr = msng.int_cr
      and msg_id = msng.msg_id
      and msg_rn = msng.msg_rn
      and s.TAG = 'NAD'
      and ce.de_p = 8 and ce.ce_p = 1
      and s.seg_p in (
select ce.seg_p
from ((segment s left join segment_no_group sg on
(s.int_s = sg.int_s and s.int_r = sg.int_r and s.int_cr =
sg.int_cr and s.seg_p = sg.seg_p))
join data_element de on (s.int_s = de.int_s and s.int_r =
de.int_r and s.int_cr = de.int_cr and s.seg_p = de.seg_p))
join component_data_element ce on (s.int_s = ce.int_s
and s.int_r = ce.int_r and s.int_cr = ce.int_cr and s.seg_p
= ce.seg_p and de.de_p = ce.de_p)
where s.int_s = msng.int_s
      and s.int_r = msng.int_r
      and s.int_cr = msng.int_cr
      and msg_id = msng.msg_id
      and msg_rn = msng.msg_rn
      and s.TAG = 'NAD'
      and ce.de_p = 1 and ce.ce_p = 1
      and ce.ce_v = 'BY'))
as 'ZIP'

from message_no_group msng

```

```

where msng.int_s = '003897733:01:MFGB:'
      and msng.msg_id = 'ORDERS:D:04B:UN:::'
GO

```

	r	int_cr	msg_id	msg_rn	numero cliente	nome contatto	nome compagnia	indirizzo	città	stato	ZIP
1	..	0000000000000001	OR...	0001	123	Michelle Butler	Nanonull, Inc.	119 Oakstreet Suite 4876	Vereno	CA	29213
2	..	0000000000000002	OR...	0001	123	Michelle Butler	Nanonull, Inc.	119 Oakstreet Suite 4876	Vereno	CA	29213

Figura 14: Risultato della vista cliente_ordine

3.2.3 Vista dettaglio_ordine

Questa vista mostra gli ordini contenuti nei messaggi.

```

create view dettaglio_ordine
as select seg.int_s, seg.int_r, seg.int_cr, sng.msg_id, sng.msg_rn, seg.seg_p, cel.ce_v as
numero_articolo,
--nome prodotto
(select TOP 1 ce.ce_v
from component_data_element ce join segment s on (s.int_s = ce.int_s and s.int_r =
ce.int_r and s.int_cr = ce.int_cr and s.seg_p = ce.seg_p)
where seg.int_s = s.int_s and seg.int_r = s.int_r and seg.int_cr = s.int_cr
      and TAG = 'IMD'
      and ce.de_p = 3
      and ce.ce_p = 4
      and ce.seg_p > seg.seg_p
      and ce.seg_p < all (select s1.seg_p
                          from segment s1
                          where s1.TAG = 'LIN'
                          and seg.int_s = s1.int_s and seg.int_r = s1.int_r and
seg.int_cr = s1.int_cr
                          and s1.seg_p > seg.seg_p)
) as nome_prodotto,
--prezzo singolo
convert(money,(select TOP 1 ce.ce_v
from component_data_element ce join segment s on (s.int_s = ce.int_s and s.int_r =
ce.int_r and s.int_cr = ce.int_cr and s.seg_p = ce.seg_p)
where seg.int_s = s.int_s and seg.int_r = s.int_r and seg.int_cr = s.int_cr
      and TAG = 'PRI'
      and ce.de_p = 1
      and ce.ce_p = 2
      and ce.seg_p > seg.seg_p
      and ce.seg_p < all (select s1.seg_p
                          from segment s1
                          where s1.TAG = 'LIN'
                          and seg.int_s = s1.int_s and seg.int_r = s1.int_r and
seg.int_cr = s1.int_cr

```

```

                                and s1.seg_p > seg.seg_p)
)) as prezzo_singolo,
--quantità
convert(int,(select TOP 1 ce.ce_v
from component_data_element ce join segment s on (s.int_s = ce.int_s and s.int_r =
ce.int_r and s.int_cr = ce.int_cr and s.seg_p = ce.seg_p)
where seg.int_s = s.int_s and seg.int_r = s.int_r and seg.int_cr = s.int_cr
and TAG = 'QTY'
and ce.de_p = 1
and ce.ce_p = 2
and ce.seg_p > seg.seg_p
and ce.seg_p < all (select s1.seg_p
from segment s1
where s1.TAG = 'LIN'
and seg.int_s = s1.int_s and seg.int_r = s1.int_r and
seg.int_cr = s1.int_cr
and s1.seg_p > seg.seg_p)
)) as quantità,
--percentuale tasse
convert(float,(select TOP 1 ce.ce_v
from component_data_element ce join segment s on (s.int_s = ce.int_s and s.int_r =
ce.int_r and s.int_cr = ce.int_cr and s.seg_p = ce.seg_p)
where seg.int_s = s.int_s and seg.int_r = s.int_r and seg.int_cr = s.int_cr
and TAG = 'TAX'
and ce.de_p = 5
and ce.ce_p = 4
and ce.seg_p > seg.seg_p
and ce.seg_p < all (select s1.seg_p
from segment s1
where s1.TAG = 'LIN'
and seg.int_s = s1.int_s and seg.int_r = s1.int_r and
seg.int_cr = s1.int_cr
and s1.seg_p > seg.seg_p)
)) as percentuale_tasse

from (segment seg join segment_no_group sng on (seg.int_s = sng.int_s and seg.int_r =
sng.int_r and seg.int_cr = sng.int_cr and seg.seg_p = sng.seg_p))
join component_data_element cel on (seg.int_s = cel.int_s and seg.int_r = cel.int_r and
seg.int_cr = cel.int_cr and seg.seg_p = cel.seg_p)
where seg.int_s = '003897733:01:MFGB:'
and sng.msg_id = 'ORDERS:D:04B:UN:::'
and seg.TAG = 'LIN'
and cel.de_p = 3
and cel.ce_p = 1
GO

```

	int_s	int_r	int_cr	msg_id	msg_rn	seg_p	numero_articolo	nome_prodotto	prezzo_singolo	quantità	percentuale_tasse
1	O...	P...	0000000000000001	OR...	0001	19	42	Pizza Pepperoni	7.2000	1	9.0
2	O...	P...	0000000000000001	OR...	0001	27	105	Lasagne al Forno	6.6000	2	9.0
3	O...	P...	0000000000000001	OR...	0001	34	66	Chianti DOCG	7.5000	1	18.0
4	O...	P...	0000000000000002	OR...	0001	19	37	Pizza Margherita	6.2000	4	6.0
5	O...	P...	0000000000000002	OR...	0001	27	99	Spaghetti al ragù	5.6000	3	9.0
6	O...	P...	0000000000000002	OR...	0001	34	66	Chianti DOCG	7.5000	3	18.0

Figura 15: Risultato della vista dettaglio_ordine

In questa vista si è dovuto estrarre più di una riga da ogni messaggio, perciò si è dovuto mettere in evidenza la posizione del segmento LIN col quale inizia il gruppo di segmenti nel quale sono contenuti gli elementi dati che servono (nel messaggio ci sono ripetizioni di questo gruppo, perciò si è dovuto estrarre più di una riga da ogni messaggio), per estrarre i dati dai segmenti successivi al segmento LIN, si è dovuto mettere la condizione che la posizione del segmento interessato sia minore della posizione del LIN attuale ed minore di tutti gli altri LIN successivi al LIN attuale.

3.2.4 Vista prezzi_ordine

In questa vista si elabora i dati ricavati dalla vista precedente.

```
create view prezzi_ordine
as select int_s, int_r, int_cr, msg_id, msg_rn, numero_articolo, prezzo = prezzo_singolo
* quantità, convert(money, percentuale_tasse * prezzo_singolo * quantità / 100) as
tasse

from dettaglio_ordine
GO
```

	int_s	int_r	int_cr	msg_id	msg_rn	numero_articolo	prezzo	tasse
1	003897733:01:MFGB:	PARTNER ID:ZZ:ROUTING ADDR:	0000000000000001	ORDERS:D:04B:UN:::	0001	42	7.2000	.6480
2	003897733:01:MFGB:	PARTNER ID:ZZ:ROUTING ADDR:	0000000000000001	ORDERS:D:04B:UN:::	0001	105	13.2000	1.1880
3	003897733:01:MFGB:	PARTNER ID:ZZ:ROUTING ADDR:	0000000000000001	ORDERS:D:04B:UN:::	0001	66	7.5000	1.3500
4	003897733:01:MFGB:	PARTNER ID:ZZ:ROUTING ADDR:	0000000000000002	ORDERS:D:04B:UN:::	0001	37	24.8000	1.4880
5	003897733:01:MFGB:	PARTNER ID:ZZ:ROUTING ADDR:	0000000000000002	ORDERS:D:04B:UN:::	0001	99	16.8000	1.5120
6	003897733:01:MFGB:	PARTNER ID:ZZ:ROUTING ADDR:	0000000000000002	ORDERS:D:04B:UN:::	0001	66	22.5000	4.0500

Figura 16: Risultato della vista prezzi_ordine

Conclusione

In questa tesi è stata presentata ed analizzata la struttura di un formato di comunicazione per messaggi commerciali elettronici, EDIFACT, inoltre si è realizzato un wrapper di tale formato per il sistema MOMIS per permettere l'acquisizione da parte di MOMIS i dati in formati EDIFACT. Dal punto di vista della realizzazione software l'output dell'estrattore dati costituisce l'input del database sql EDIFACT. Il wrapper realizzato permette di superare le limitazioni dei file EDIFACT, infatti il formato EDIFACT non è un formato di database, ma un formato file per trasmettere i dati attraverso i sistemi EDI. Ciò significa che non è possibile eseguire le interrogazioni direttamente su un file EDIFACT. È stato necessario progettare un database relazionale per caricare il contenuto dei file EDIFACT e realizzare l'estrattore di dati EDIFACT per popolare tale database. Il wrapper realizzato può essere usato per qualunque tipo di messaggio EDIFACT, avendo uno schema generale valido per tutti i tipi di messaggi.

Nella tesi si è visto com'è complesso il formato EDIFACT a causa del suo obiettivo, dover coprire tutti i possibili messaggi commerciali e comprimere il più possibile il file da inviare.

La difficoltà maggiore nella realizzazione del wrapper EDIFACT sta nel dare significato agli elementi dati, perciò si è pensato di progettare e realizzare delle viste per facilitare tale compito. La realizzazione delle viste non è obbligatoria ma è fondamentale per semplificare la comprensione dei significati dei dati ed avere una visione d'insieme semplificata dei elementi dati e delle loro associazioni. A causa della complessità e della flessibilità dell'EDIFACT, la realizzazione delle viste deve essere manuale e non può essere completamente automatica.

Per aumentare l'efficienza del wrapper invece di realizzare le viste, che sono pesanti da gestire a causa della quantità di dati presenti nel database dell'EDIFACT, si può inserire elementi dati in tabelle simili alle viste, ciò implica un impiego maggiore dello spazio all'interno del database e la necessità di aggiornare le tabelle ad ogni nuovo file EDIFACT inserito, mentre nelle viste l'aggiornamento è automatico.

Bibliografia

- [1] **Domenico Beneventano, Sonia Bergamaschi, Maurizio Vicini**
“Progetto di Basi di Dati Relazionali”

- [2] **Domenico Beneventano, Sabina El Haoum, Daniele Montanari**
“Mapping of heterogeneous schemata, business structures, and terminologies”
Database and Expert Systems Applications, 2007. DEXA apos;07. 18th International Conference on Volume , Issue , 3-7 Sept. 2007 Page(s):412 - 418

- [3] **Domenico Beneventano, Daniele Montanari**
“An OWL representation of EDI/EDIFACT documents”
STASIS Technical Report March 2007

- [4] **Douglas Foxvog, and Christoph Bussler**
“Ontologizing EDI Semantics ER (Workshops)”
Lecture Notes in Computer Science, Vol. 4231, pp. 301-311, Springer, 2006

- [5] <http://dbgroup.unimo.it/Momis/>

- [6] <http://www.dbgroup.unimo.it/>

- [7] <http://www.gefeg.com/jswg/index.html>

- [8] <http://www.unece.org/trade/untdid/welcome.html>

- [9] http://www.gxs.com/pdfs/Tutorial/Tutor_EDIFACT_GXS.pdf

- [10] http://www.altova.com/products/mapforce/edi_to_xml_mapping.html

- [11] http://en.wikipedia.org/wiki/Electronic_Data_Interchange

- [12] <http://ecommerce.hostip.info/pages/384/Electronic-Data-Interchange-EDI.html>

- [13] http://www.sbbcargo.com/it/index/cao_edifact.htm

- [14] http://www.nwi.it/nwi_arretrati/ap120001.htm

- [15] http://www.cumminsfiltration.com/it/order_products/it_order_edi.shtml

- [16] <http://www.editeur.org/>

- [17] http://www.wpc-edi.com/edi_standards
- [18] <http://www.stylusstudio.com/edifact/>
- [19] <http://www.geocities.com/siliconvalley/platform/4871/photo.htm>
- [20] <http://www.xml-edifact.org/EX/>
- [21] <http://www.interlex.it/docdigit/zanini1.htm>
- [22] <http://www.eprosper.co.uk/eprosper/dedi.html>
- [23] <http://www.wsmo.org/TR/d27/v0.1/20050621/>