klase

Caso di studio: integrazione di gestione e riscossione di tributi comunali.

DeltaDator

Abstract. In questo documento si descrive un caso di studio per reale per applicare le ricerche e le tecnologie esistenti sulle tematiche di integrazione e coordinamento di sistemi eterogenei distribuiti.
Il caso di studio proposto nasce dall'esigenza di integrare i servizi di calcolo e pagamento delle tasse comunali con le entità deputate alla riscossione dei pagamenti e con la contabilità dell'ente.

Document Identifier	Deliverable D8.2
Project	MIUR-FIRB project RBAU01P5SS "Knowledge Level Automated Software Engineering"
Version	v1.0
Date	October 4, 2006
State	Final
Distribution	Public

Acknowledgements.

This document is part of a research project funded by the FIRB 2001 Programme of the "Ministero dell'Istruzione, dell'Universita' e della Ricerca" as project number RBAU01P5SS.

The partners in this project are: Istituto Trentino di Cultura (Coordinator), Universita' degli Studi di Trento, Universita' degli Studi di Genova, Universita' degli Studi di Roma "La Sapienza", DeltaDator S.p.A..

Executive summary

DeltaDator SpA rappresenta oggi uno dei principali fornitori su mercato della Pubblica Amministrazione. Oltre 400 Enti Locali (Comuni, Province, Comunità Montane, Università, ASL, Ospedali, ecc.) utilizzano i sistemi informativi basati su architettura client-server o Web che assicurano una copertura completa delle loro esigenze gestionali.

DeltaDator è anche uno storico fornitore di soluzioni sw per supportare l'attività bancaria nelle singole aree operative, come ad esempio la Tesorerie Enti per la quale gestisce tutte le funzioni di tesoreria e cassa per gli Enti Pubblici, integrandosi con il sistema informativo centrale della Banca.

In questo ambito esistono tuttora delle necessità e delle potenzialità di integrazione ed automazione di funzionalità tra i diversi domini dell'ente pubblico e anche nei confronti di chi fornisce servizi all'Ente come ad esempio la Banca Tesoriere.

Lo stesso progetto di e-Government del Ministero per l'Innovazione e le Tecnologie indica il paradigma della cooperazione applicativa realizzato tramite WebServices come il meccanismo standard di scambio dati e servizi tra applicazioni.

Nell'ambito del progetto KLASE e della collaborazione con i partner di ricerca, si è individuato un caso di studio reale per applicare le ricerche e le tecnologie esistenti sulle tematiche di integrazione e coordinamento di sistemi eterogenei distribuiti.

Il caso di studio proposto nasce dall'esigenza di integrare i servizi di calcolo e pagamento delle tasse comunali con le entità deputate alla riscossione dei pagamenti e con la contabilità dell'ente, nel nostro caso la banca tesoriere.

L'obiettivo che si intende raggiungere è di automatizzare il processo di riscossione integrando sistemi tecnologicamente eterogenei e nel contempo definendo delle specifiche di interfacciamento che permettano di sostituire gli attori con altri semanticamente equivalenti. Non meno importante è la necessità di poter variare la logica di processo che governa l'interazione tra i sistemi senza modificare i sistemi stessi ma agendo ad un livello di astrazione di processo.

1. Introduzione

Obiettivo

Il seguente documento intende descrivere il caso di studio individuato da DeltaDator nell'intento di applicare i risultati delle attività di ricerca di IRST relativamente a “WebServices Composition and Execution” e più in generale alle problematiche dei workflow distribuiti.

Come premessa è importante presentare DeltaDator quale fornitore di una suite completa di prodotti rivolti alle esigenze della Pubblica Amministrazione Locale e degli Istituti Bancari.

In questo contesto esistono tuttora delle necessità di integrazione ed automazione di funzionalità tra i diversi prodotti che compongono la suite e più in generale con prodotti di terze parti già presenti nelle realtà del clienti.

Il caso di studio individuato nasce dall'esigenza di integrare i servizi di calcolo e pagamento delle tasse comunali con le entità deputate alla riscossione dei pagamenti e con la contabilità dell'ente.

L'idea è di arrivare ad applicare nella realtà il prototipo che sarà realizzato; sarà necessario quindi identificare un ente che già dispone delle soluzioni sw di DeltaDator coinvolte nel progetto.

Organizzazione del documento

La struttura del documento seguente inizia con la descrizione della situazione attuale nel contesto del prototipo sopra menzionato, individuando gli attori in gioco e come attualmente interagiscono tra loro.

Si passerà successivamente alla definizione dello Use Case che si intende realizzare nel prototipo, focalizzando gli aspetti di competenza e responsabilità di ciascun attore e sulle interazioni che sono necessarie al fine di automatizzare il processo.

Si descriveranno poi gli aspetti di “web-servizzazione” partendo dalla definizione delle interfacce via WSDL fina ad arrivare alle problematiche ed aspetti implementativi degli stessi.

2. Il contesto attuale

Come si è detto, il contesto che si è scelto per questa sperimentazione è il pagamento di un tributo da parte del contribuente. Gli attori coinvolti, oltre al cittadino, sono i seguenti:

Ufficio Tributi dell'Ente

- Ufficio Ragioneria dell'Ente
- Banca Tesoriere dell'ente
- Banca di appoggio per la domiciliazione dell'utenza

Parlando di gestione tributi in carico all'Ente, si intende quell'insieme di tasse come TARSU (Tassa sui Rifiuti Solidi Urbani), ICI (Imposta Comunale sugli Immobili), TOSAP (Tassa Occupazione Spazi ed Aree Pubbliche), COSAP (Canone per l'occupazione di spazi ed aree pubbliche), ACQUEDOTTO, ..

Nel nostro caso si è scelta la TARSU, scelta che nulla toglie alla generalità dell'approccio e della soluzione e che quindi rimane applicabile anche alle altre tipologie di tributi.

Vediamo ora più specificamente lo Use Case che descrive la funzionalità.

Lo Use Case

Gli attori coinvolti sono:

- **Ufficio Tributi** dell'ente
- **Banca Tesoriere**
- **Ufficio Ragioneria** dell'ente

Per semplicità non abbiamo esplicitato come attore la banca sulla quale si appoggia la domiciliazione dell'utenza.

L'ufficio tributi deve richiedere al cittadino il pagamento di una tassa (ad esempio la TARSU). Il caso contemplato è quello per il quale il cittadino ha autorizzato la propria banca per il pagamento automatico via RID, questo per poter sfruttare l'automatismo implicito nel pagamento via RID.

Il processo lato **Ufficio Tributi**:

- calcola la tassa per il cittadino specificando l'anno di competenza, il codice del cittadino e la tipologia di tributo
- emette un ordine di pagamento
- invia l'ordine di incasso alla Banca Tesoriere dell'ente
- aspetta il ritorno dalla Banca Tesoriere per aggiornare la posizione tributaria del cittadino

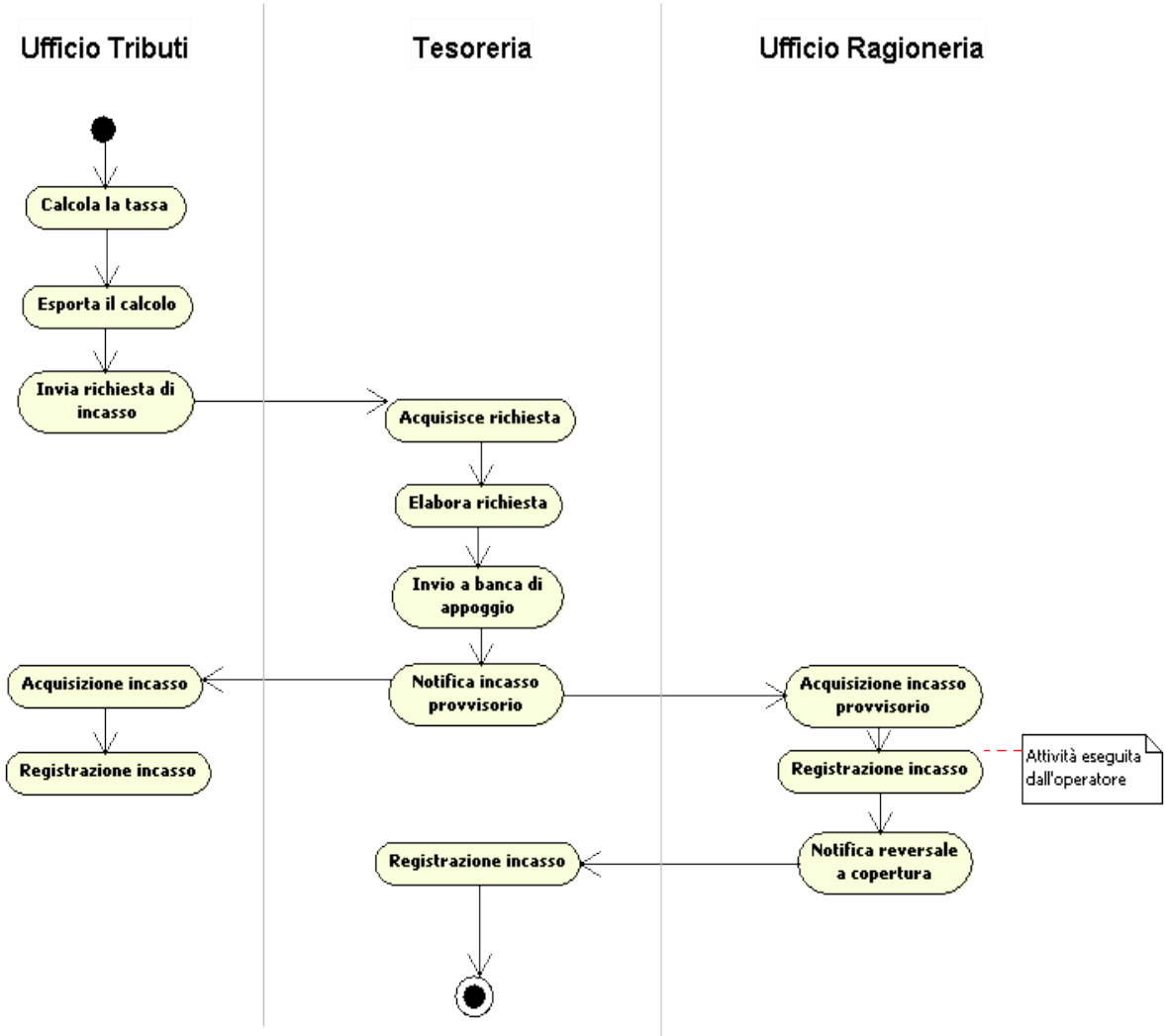
Il flusso lato **Banca Tesoriere**:

- acquisisce l'ordine di pagamento
- elabora (trasforma in RID) e smista alla banca di appoggio (domiciliazione)
- invia la notifica dell'incasso provvisorio all'Ufficio Ragioneria dell'ente
- invia la notifica dell'incasso all'Ufficio Tributi dell'ente
- acquisisce la reversale a copertura dall'Ufficio Ragioneria dell'ente
- registra la reversale

Il flusso lato **Ufficio Ragioneria**:

- acquisisce la notifica di incasso provvisorio dalla Banca Tesoriere dell'ente
- elabora e contabilizza l'incasso
- invia alla Banca Tesoriere la reversale a copertura

In termini di activity diagram:



Il diagramma non fa riferimento a situazioni di errore ma descrive il caso positivo.

Ufficio Tributi

Il calcolo del tributo da versare viene eseguito tramite il sw Civilia di DeltaDator. Il sw è disponibile su due differenti architetture, la prima di tipo Client/Server basata su piattaforma Centura, la seconda di tipo Web/J2EE.

Il sw prevede che l'operatore dell'ente esegua la funzione di calcolo del tributo ed a seguire esegua la funzione di esportazione della richiesta nel formato proprietario (vedi Appendice A) e che dovrà essere inviato alla Banca Tesoriere per la successiva conversione in RID e l'invio alla banca sula quale il cittadino ha la domiciliazione.

Come funziona oggi

Come detto in precedenza, per dettagliare la business logic si è scelto l'esempio della TARSU (Tassa sui Rifiuti Solidi Urbani). La logica è comunque applicabile in generale anche ad altre tipologie di tasse (ICI, Acquedotto, Tosap, Cosap, ..).

Iniziamo con il dire che la TARSU viene calcolata due volte all'anno. In queste occasioni viene eseguito dall'operatore il calcolo su tutti i contribuenti. Abbiamo un primo calcolo generale applicato su tutti, un secondo (per differenza) applicato su chi ha avuto della variazioni (superficie dell'immobile, nucleo familiare, ..).

L'esito del calcolo produce delle transazioni sui dati del database dell'applicativo Civilia. Una volta completato il calcolo, l'operatore esegue l'estrazione in un file (vedi Appendice A). Ad oggi esistono due tipologie di file, uno per il pagamento tramite bollettini, l'altro per il pagamento tramite RID.

Il pagamento della somma dovuta può avvenire in modalità rateale (fino a 4 rate). Il flusso esportato prevede i necessari placeholder per le 4 rate, mentre per il RID saranno 4 record distinti.

Attualmente, il flusso esportato viene inviato alla Banca Tesoriere in varie modalità: spedizione dischetto, mail, ftp, http. Il ritorno dalla banca invece contempla solo i casi di morosità, questo significa che essendoci una scadenza per ogni pagamento, l'ente inferisce che il cittadino abbia pagato se non è nella lista dei morosi. La lista dei morosi viene inviata dalla banca all'ente in formato cartaceo.

Una volta ricevuta questa lista, l'operatore dell'ente regolarizza uno ad uno i contribuenti utilizzando manualmente le funzionalità offerte dal sw Civilia.

Banca Tesoriere

Per l'invio alla banca del flusso si può utilizzare il componente sw di DeltaDator per la gestione della tesoreria che ad oggi viene utilizzato dai clienti DeltaDator per l'invio di flussi (mandati, reversali, RID, ..) alla tesoreria enti (ad oggi il fondo comune delle casse rurali trentine).

Il sw offre una serie di funzionalità quali la scelta flusso, invio flusso, ricezione flusso, validazione flusso, conversione flusso, interrogazione, visualizzazione log; funziona in

ambiente AS400

Come funziona oggi

Nel dettaglio, il modulo sw è una infrastruttura che fornisce i servizi necessari per il trasferimento di flussi informativi da un sender ad un receiver, di seguito anche chiamati genericamente attori. Gli attori possono essere sia applicazioni informatiche che esseri umani. Ad esempio:

- un impiegato comunale invia alla applicazione di tesoreria il flusso dei mandati mediante upload da un apposito sito
- una applicazione bancaria invia ai propri clienti l'estratto conto per e-mail
- l'applicazione dei tributi invia all'applicazione bancaria il flusso dei rid per la riscossione

Il sw fornisce l'infrastruttura di comunicazione, di controllo degli accessi, di logging, di consultazione dei flussi trasferiti, di supporto all'eventuale controllo sintattico ed alla conversione dei flussi.

Gli attori abilitati ad almeno un trasferimento di flussi vengono censiti nella tabella *attori*, che contiene anche i dati necessari per la loro autenticazione (password, indirizzo e-mail, certificato digitale, …).

In questa tabella sarà anche possibile inserire attori di gruppo, per gestire le situazioni in cui uno stesso tipo di flusso può essere ricevuto od inviato da molti attori dello stesso tipo (es: i clienti della banca che ricevono l'estratto conto via e-mail). Gli elementi di un gruppo vengono sempre registrati nella tabella attori ma con l'indicazione del gruppo al quale appartengono.

Anche tutti i tipi di flussi trattati sono censiti in una apposita tabella, nella quale sono indicati gli attori del flusso, le modalità di input-output e gli eventuali programmi di validazione o di conversione da lanciare per il trasferimento del flusso (ovviamente tali programmi dovranno essere realizzati appositamente per ogni tipo di flusso).

Per ogni flusso trasferito (o comunque acquisito) viene scritto un record nella tabella log, con tutti i dati di interesse del trasferimento (data ora, sender, receiver, flusso, stato del flusso, …).

Ufficio Ragioneria

La gestione della ragioneria è realizzata dai sw CiviliaOpen e CiviliaWeb di DeltaDator. Il prodotto CiviliaOpen è realizzato con architettura Client/Server ed è basato sul sistema di sviluppo Centura. Il prodotto CiviliaWeb è invece realizzato con architettura J2EE.

La ragioneria si aspetta di ricevere una notifica di incasso provvisorio. L'incasso viene associato in automatico al capitolo di incasso previsto per questa tipologia di incasso, l'operatore della ragioneria esegue i necessari controlli e completa manualmente con gli eventuali dati mancanti, emette un ordine di incasso (reversale a copertura) che viene inviato alla banca per validare l'incasso stesso e permettere al tesoriere di associare l'incasso al capitolo di entrata.

Come funziona oggi

Importazione

- L'utente seleziona la fase di importazione dati.
- Il sistema chiede all'utente di selezionare il file di riferimento.
- Il sistema verifica che il file non contenga informazioni già importate precedentemente.
- In caso di anomalie avvisa l'utente con una segnalazione
- Il sistema importa in una tabella le informazioni contenute nel file

Verifica dei dati importati

- L'utente seleziona la fase di gestione dati
- L'utente selezione uno o tutti i servizi
- L'utente seleziona la funzione "controllo"
- Il sistema effettua, relativamente ai servizi selezionati, una serie di controlli atti alla validazione di ogni record come "registrabile"
- Il sistema evidenzia per ogni record gli eventuali errori riscontrati
- Il sistema contrassegna come registrabili i record "controllati" senza errori
- L'utente provvede alla sistemazione delle anomalie segnalate e riesegue la funzione di controllo.

Assegnazione

- Mentre l'importazione viene fatta da chi ha accesso al TLQ, le bollette hanno un proprio responsabile per la gestione/registrazione. In genere chi esegue lo scarico "smista" le bollette per competenza. Per fare ciò, seleziona e assegna un nome responsabile (una sigla scritta a mano). La selezione può essere fatta a monte sul file di testo prima di importarlo, ma deve poi essere possibile modificarla per eventuali errori.

Registrazione dei dati importati

- L'utente avvia il processo di registrazione delle informazioni per i record "registrabili" relativamente a uno o a tutti i servizi
- L'utente seleziona le proprie righe da registrare inserendo il responsabile
- Il sistema richiede l'imputazione di eventuali descrizioni accessorie (SI, una descrizione da mettere in testa nelle note/riversali PRIMA del testo automatico)
- Il sistema inserisce un accertamento per ogni raggruppamento costituito da: servizio / capitolo / programma / centro elementare/ (NB: atto autorizzatorio = "PA" (presa d'atto) con numero = prima bolletta)
- Il sistema inserisce una nota di entrata per ogni raggruppamento
- Il sistema associa ad ogni nota registrata l'accertamento di riferimento
- Il sistema inserisce una registrazione in contabilità generale per ogni nota inserita
- Il sistema inserisce una registrazione in contabilità analitica per ogni registrazione in contabilità generale
- Il sistema inserisce una reversale per ogni raggruppamento o per ogni bolletta in dipendenza di una spunta
- Reversale a copertura (cod. 99 su stampa1)
- Indicazione in descrizione della presenza di flusso allegato o del numero della bolletta in dipendenza della spunta di cui sopra
- Il sistema registra su ogni record processato le informazioni relative a numero accertamento – numero nota – numero reversale
- Il sistema definisce come "registrato" ogni record processato

Controlli post registrazione

- L'utente produce una stampa che ha come contenuto l'elenco dei documenti registrati. Selezione per data / servizio / numero bolletta

Invio delle reversali a copertura

- Dopo aver emesso gli ordinativi l'operatore procede alla stampa degli stessi per un ulteriore processo di controllo, e per sottoporre gli stessi alla firma del responsabile ovvero colui che autorizza le operazioni di pagamento/incasso. In questa fase gli ordinativi possono essere anche raggruppati in distinte che riepilogano, in genere giornalmente, tutti i pagamenti o gli incassi ordinati. Gli ordinativi firmati vengono portati poi in tesoreria affinché possano essere eseguiti.
- Nel caso in cui sia attivato un sistema di ordinativo informatico con

apposizione di firma digitale agli ordinativi di pagamento ed incasso, non è necessario l'invio dell'ordinativo cartaceo firmato dal responsabile. Valgono a tutti gli effetti le informazioni trasmesse in via telematica.

- Per attivare l'invio alla tesoreria, l'utente seleziona la fase di esportazione mandati / reversali.

  La selezione, fatta per entrata o spesa separatamente, può avvenire per range numerico, da numero ordinativo a numero ordinativo o da numero di distinta a numero distinta, o per utente (l'ordinativo è associato all'operatore che lo ha inserito - funzione attualmente attiva solo per l'ordinativo informatico).
- Il sistema produce un file, in genere un file di testo che corrisponde a specifiche fornite dalla tesoreria, che contiene gli ordinativi estratti secondo la selezione dell'operatore.
- Nel produrre il file il sistema riconosce, e contrassegna come tali, gli ordinativi emessi a copertura e, per il tracciati che lo prevedono, associa le informazioni relative ai provvisori di tesoreria che vanno a regolarizzare.
- Il file viene trasmesso, con sistemi diversi, alla tesoreria (dischetto consegnato a mano, Procedure Web messe a disposizione dalla tesoreria, sistemi di firma di terze parti…).
- La tesoreria importa e verifica le informazioni trasmesse dall'ente e, previa verifica della corrispondenza con l'ordinativo cartaceo firmato dal responsabile dell'ente o della validità della firma digitale apposta in caso di ordinativo informatico, provvede ad eseguire le operazioni richieste.
- La tesoreria trasmette all'ente il giornale di cassa che riporta le operazioni di pagamento o riscossione effettuate per conto dell'ente.
- L'ente registra le operazioni di carico e scarico di cassa.

Le informazioni che provengono dalla tesoreria e che sono necessarie alla contabilità sono le seguenti:

- Data flusso (GGMMAAAAHHMMSS)
- Numero bolletta
- Codice servizio
- Data bolletta
- Causale tesoreria
- Importo riscosso
- Riferimento di riscossione (es. versante)
- Data valuta
- Responsabile
- Nota

L'innovazione

E' evidente dalla descrizione del contesto attuale che l'integrazione dei processi è realizzata con meccanismi tra loro eterogenei e con attività di tipo manuale.

Esistono quindi spazi di evoluzione tecnologica e metodologica che ci permettono di uniformare l'esposizione di servizi da parte dei componenti sw coinvolti e che facilitino l'automazione del processo di pagamento tributi.

Dal punto di vista funzionale e di processo, il nuovo scenario è concettualmente analogo a quanto descritto nell'activity diagram precedente; l'innovazione funzionale consiste nell'automatizzare lo scambio di informazioni tra gli attori.

Ufficio Tributi

Le migliorie iniziano con l'invio automatico del flusso alla Banca Tesoriere. Anche la gestione del feedback dalla Banca Tesoriere può essere automatizzata.

E' possibile ad esempio chiedere alla Banca Tesoriere (con cadenza giornaliera) il giornale di cassa nel quale sono registrati i movimenti contabili della giornata. Dal giornale, con opportuni filtri è possibile estrarre i solo movimenti (incassi) relativi al pagamento del tributo.

Il giornale di cassa così filtrato può essere importato e “contabilizzato” automaticamente da Civilia superando quindi l'attività manuale attualmente in essere.

Affinché il nuovo processo possa funzionare, alla richiesta di pagamento emessa dai Tributi dovranno essere aggiunte le informazioni necessarie a:

- ✔ associare il futuro incasso al capitolo di entrata definito dall'ufficio ragioneria dell'ente; per poter automatizzare l'incasso è necessario che le informazioni inviate dall'ufficio tributi e veicolate dalla banca siano il codice del capitolo di incasso, la cifra e la causale.
- ✔ identificare il cittadino per permettere di gestire correttamente le informazioni di ritorno provenienti dalla Tesoreria. Per fare ciò potrebbe bastare il codice di pagamento unico o della rata i-esima già previsti nel tracciato in Appendice A.

Banca Tesoriere

Per questo attore è necessario prevedere la possibilità di ricevere richieste non più prevenienti da un operatore umano ma da un sw. Deve esporre i servizi necessari ai richiedenti Ufficio Tributi e Ufficio Ragioneria, deve essere in grado di comunicare all'Ufficio Ragioneria la disponibilità di un incasso relativo al capitolo Tributi.

La Banca Tesoriere riceve dall’Ufficio Tributi il flusso degli ordini di incasso relativi ai tributi. Il flusso non ha il tracciato dei RID, poiché contiene dati che servono alla tesoreria ma non alla banca d’appoggio

Effettua automaticamente il controllo sintattico del flusso; in caso positivo lo accetta, registra l’avvenuta ricezione (crea il record di log) e la notifica automaticamente all’Ufficio Tributi.

Carica il flusso nell’archivio incassi. Contestualmente, trattandosi di incassi tramite RID, prepara il flusso RID corrispondente e lo inoltra sulla rete interbancaria.

Trattandosi di incassi tramite RID, appena emessi i RID la Tesoreria considera gli incassi come effettuati, accredita il conto corrente dell’ente e passa gli incassi allo stato di “effettuati”.

A fine giornata la Tesoreria produce il giornale di cassa, che include tutti gli incassi ed i pagamenti ricevuti/effettuati in giornata da ogni ente.

L'Ufficio Ragioneria dell'ente acquisisce il giornale di cassa, emette ed invia alla Tesoreria un ordine di incasso (Reversale a copertura) per la imputazione degli incassi al corretto capitolo di bilancio.

La Tesoreria effettua la ricezione e la eventuale conversione della reversale e la inoltra alla Tesoreria.

Infine carica a bilancio la reversale e la incrocia con i corrispondenti incassi effettuati.

Ufficio Ragioneria

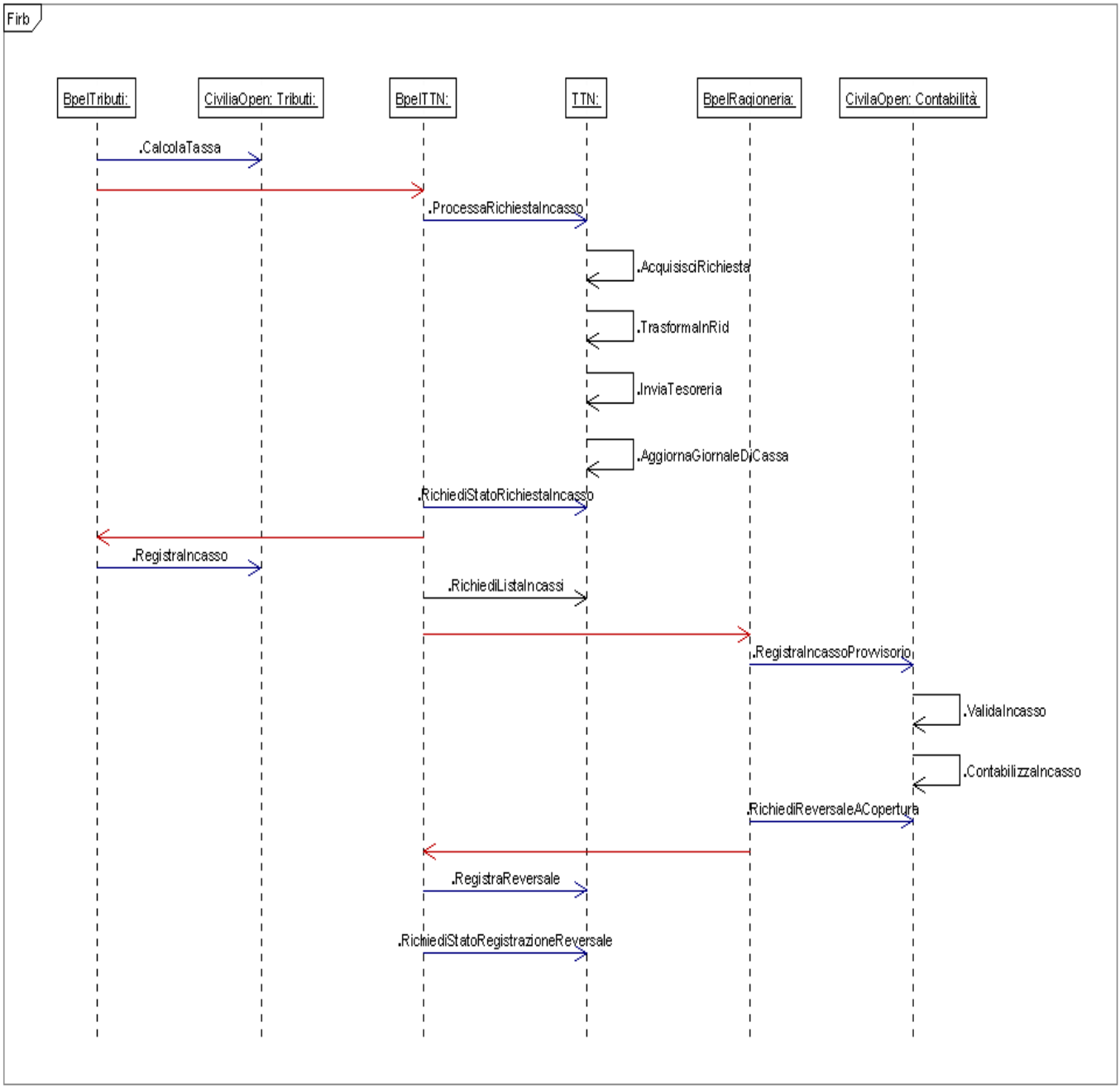
Anche in questo caso, dovrà essere esposto un servizio di acquisizione della disponibilità di un incasso e conferma dell'avvenuta contabilizzazione da notificare alla Banca Tesoriere.

L'Ufficio Ragioneria acquisisce automaticamente la richiesta di incasso provvisorio inviata dalla Tesoreria, l'operatore , notificato dell'acquisizione, esegue le opportune verifiche e completa la richiesta per la successiva contabilizzazione.

Viene quindi automaticamente inviata alla Tesoreria una richiesta di reversale a copertura per aggiornare gli aspetti contabili gestiti dalla Tesoreria.

Sequence diagram

La figura seguente rappresenta in termini di sequence diagram i servizi esposti dai tre attori e il processo che coordina i servizi stessi:



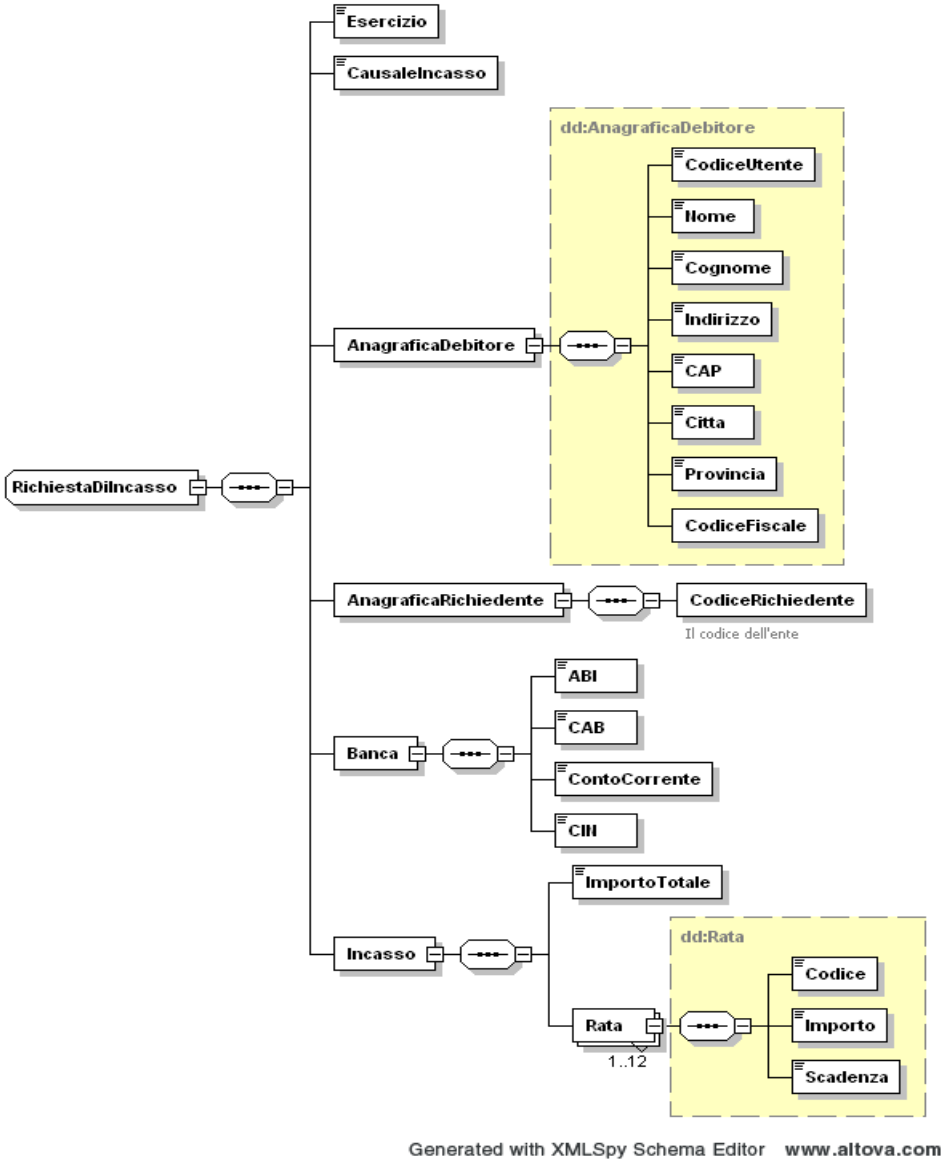
Riprendendo il sequence diagram, vengono di seguito dettagliati i servizi esposti:

Ufficio Tributi

I servizi individuati sono i seguenti:

- **CalcolaTassa** che esegue il calcolo del tributo specificato relativamente ad un

contribuente,ad un periodo di tempo ed a una tipologia di tassa. Ritorna il calcolo e una serie di informazioni aggiuntive necessarie la processamento automatico della richiesta di incasso. La figura seguente mostra l'oggetto ritornato:



- **RegistraIncassoTassa** che gestisce e registra nel sistema l'avvenuto pagamento, dato il codice incasso (vedi campo Codice definito in dd:Rata) e la data dell'incasso (vedi campo Scadenza definito in dd:Rata),

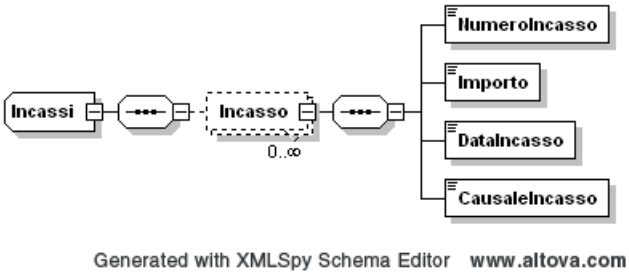
Banca Tesoriere

I servizi individuati sono i seguenti:

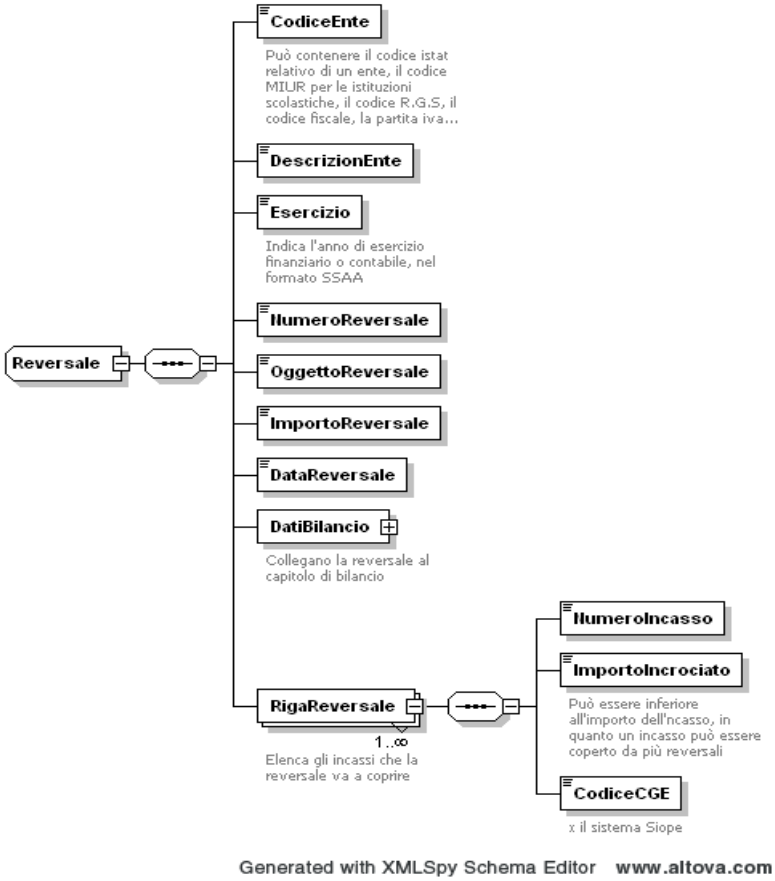
- **ProcessaRichiestaIncasso** che accetta in input una richiesta di incasso, esegue le normalizzazioni e trasformazioni necessarie (ad esempio in RID) e

immette la richiesta sul circuito interbancario. Ritorna un identificativo univoco della richiesta.

- **RichiediStatoRichiestaIncasso** che dato l'identificativo della richiesta, ritorna se l'incasso è effettivamente avvenuto o in che stato si trova
- **RichiediListaIncassi** che dato il codice ente, l'esercizio, la causale di incasso e una range di date,ritorna la lista degli incassi. La figura seguente mostra l'oggetto lista incassi:



- **RegistraReversale** che registra la reversale. La figura seguente mostra l'oggetto reversale:



- **RichiediStatoRegistrazioneReversale** che, dato l'identificativo della richiesta (vedi output dell'operazione RegistraReversale) ritorna se la richiesta

è andata a buon fine, è in lavorazione o è fallita.

Ufficio Ragioneria

Servizi esposti

I servizi individuati sono i seguenti:

- **RegistraIncassoProvvisorio** che accetta in input una richiesta di incasso, esegue le normalizzazioni e trasformazioni necessarie (ad esempio in RID) e immette la richiesta sul circuito interbancario. Ritorna un identificativo univoco della richiesta.
- **RichiediReversaleACopertura** che ritorna la reversale dato l'identificativo dell'incasso provvisorio

Tecnologie abilitanti

Fino ad ora non i è parlato di tecnologia volendo focalizzare l'approccio funzionale.

Coerentemente con l'approccio tecnologico e metodologico definito da IRST, i servizi esposti sono modellati e realizzati come web services, mentre per la parte di processo si utilizza BPEL che ha il compito di orchestrare i webservices.

Cenni sui WebServices

I WebServices sono descritti tramite il linguaggio WSDL. Esso è la formale descrizione dell'interfaccia e della locazione fisica (indirizzo) del servizio. Il WSDL può essere diviso in due categorie: descrizione astratta e descrizione concreta del servizio.

La descrizione astratta stabilisce le caratteristiche dell'interfaccia senza fare riferimento alle tecnologie abilitanti. I concetti principali della descrizione astratta sono i seguenti:

- portType
- operation
- message

Il *portType* fornisce una vista di alto livello dell'interfaccia del servizio e raggruppa le funzioni chiamate *operations.* Ogni operazione rappresenta una specifica azione che può essere eseguita dal servizio. Possiamo dire che un'operazione sia comparabile ad un metodo pubblico di componente sw. Il *messagge* è invece assimilabile al concetto di parametri di input e output di un metodo.

La descrizione concreta concerne gli aspetti tecnologici e di comunicazione (ad esempio il protocollo di comunicazione). Anche in questo caso abbiamo 3 concetti:

- binding
- port
- service

Il *binding* rappresenta una possibile tecnologia di trasporto che il servizio può usare per comunicare. SOAP è la più comune forma di binding. Il binding può essere applicato all'intera interfaccia oppure alla singola operazione. La *port* rappresenta l'indirizzo fisico al quale il servizio può essere acceduto con uno specifico protocollo. Il termine *service* è invece usato per identificare il gruppo di endpoints.

Un WSDL spesso fa riferimento ad uno *schema XSD* per formalizzare la struttura dei *message* in ingresso e uscita. Fa anche riferimento ad un ulteriore documento di *policy* che ha lo scopo di definire le regole di fruizione dei servizi (ad esempio lo SLA).

Questi documenti (WSDL, XSD, Policy) rappresentano nel loro insieme il *service contract* cioè quel set di informazioni e regole che permettono ad richiedente del

servizio di poter fruire con successo del servizio stesso.

Cenni su BPEL

BPEL é l'acronimo di Business Process Execution Language; è un linguaggio basato su XML ed è costruito per descrivere formalmente i processi commerciali ed industriali in modo da permettere una suddivisione dei compiti tra attori diversi.

Un'applicazione BPEL viene invocata come Web Service ed interagisce con il mondo esterno esclusivamente invocando altri Web Services. L'ambiente runtime all'interno del quale viene eseguito il generico processo è detto *motore BPEL*.

Lo standard che definisce l'uso di BPEL nelle interazioni tra Web services è chiamato BPEL4WS o WS-BPEL.

Il linguaggio BPEL permette di descrivere un processo di business mediante un insieme di attività, che possono essere *semplici* o *strutturate*. Le attività semplici esprimono una generica *azione* (ad es. invoca servizio, ricevi risposta, assegna valore ad una variabile, termina processo, etc...), mentre quelle strutturate sono normalmente utilizzate per raggruppare attività semplici allo scopo di esprimere loop, operazioni condizionali, esecuzione sequenziale, esecuzione concorrente, etc... L'intero processo è descritto mediante un'unica attività strutturata (*top-level activity*), generalmente di tipo sequenziale.

BPEL mette altresì a disposizione dei costrutti per esprimere le cosiddette transazioni di lungo periodo (*long running transactions*, LRT), che rappresentano un'estensione delle transazioni ACID al caso di processi di lunga durata mediante la nozione di compensazione delle attività eseguite. Ancora, il meccanismo della correlazione è utilizzato per tener traccia di una certa conversazione a livello business, identificando così una sorta di sessione tra più partecipanti ad una stessa istanza di processo.

La Web-Servizzazione

I componenti sw di **Civilia Open** sono realizzati come già menzionato con la tecnologia Centura. Questa offre nativamente nel proprio Toolkit di sviluppo un modulo per lo sviluppo nativo di webservice.

Purtroppo questo modulo non è sufficientemente sofisticato per supportare i webservices definiti in questo progetto, per cui è necessario realizzato un wrapper in dotnet (trattadosi di piattaforma Windows) che implementa l'interfaccia webservices, gestisce la comunicazione e il binding, chiama i moduli nativi Centura tramite DCOM

Per le prove di integrazione è stato allestito un ambiente di test raggiungibile da internet.

I componenti sw di **Civilia Web** sono realizzati, come già menzionato, con la tecnologia Java/J2EE. Questa offre nativamente dei potenti framework per l'implementazione dei webservice. Nel nostro caso viene utilizzato il framework Axis.

Per le prove di integrazione è stato allestito un ambiente di test raggiungibile da

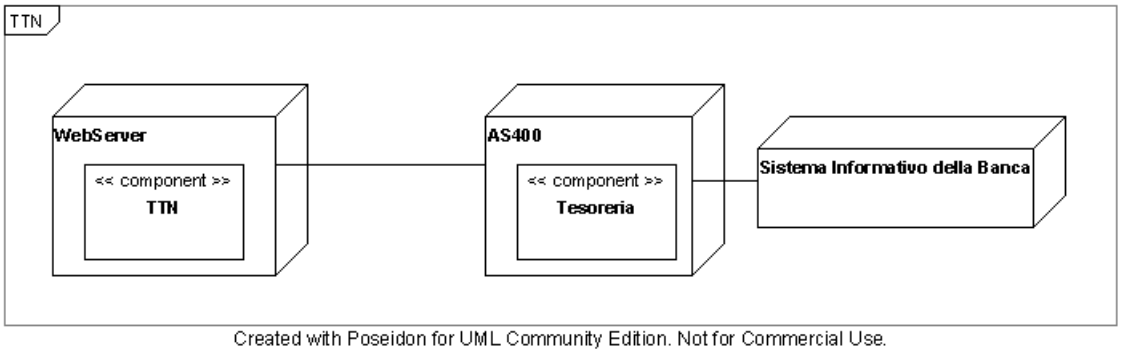
internet.

Per quanto riguarda la **Tesoreria**, i webservices sono stati sviluppati sulle specifiche J2EE in particolare sulla tecnologia Jsp (Java Server Pages) e Servlet, il Core di supporto è sviluppato in Java 1.5, la connettività con la base dati è effettuata tramite i driver JDBC. L'application server è Tomcat 5.

Per lo sviluppo dei WebServices è stato utilizzato il framework Axis. Per le prove di integrazione è stato allestito un ambiente di test raggiungibile da internet.

Per l'implementazione dei webservice e in particolare la loro integrazione con il mondo As400, è stato realizzato un componente Java chiamato TTN che si occupa di integrare i servizi nativi AS400.

Nella figura seguente viene mostrata l'architettura fisica.



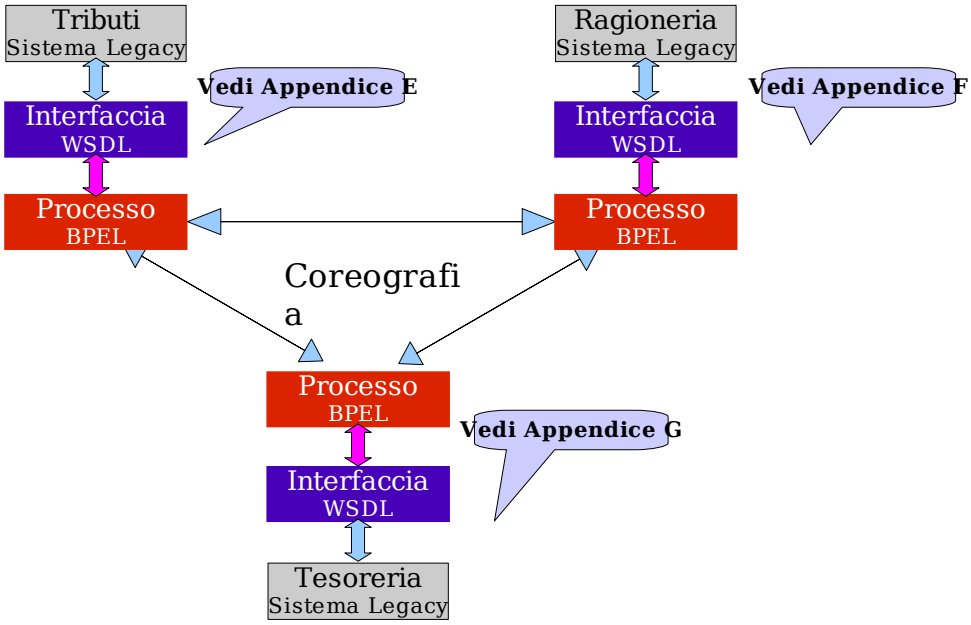
I processi BPEL

L'utilizzo di BPEL ci offre due possibili approcci alla gestione dei processi: coreografia e orchestrazione. La differenza tra coreografia e orchestrazione può essere spiegata con la metafora della rotonda stradale e di un sistema di semafori.

I semafori sono controllati un sistema centrale, la rotonda stradale definisce delle regole di circolazione che gli attori (automobilisti) vanno ad interpretare. Questo significa che nell'approccio coreografico i processi conoscono il contesto globale e sono parte attiva nella realizzazione del processo principale. Con l'orchestrazione i processi non conoscono il contesto generale ma rispondono puntualmente ad un processo orchestratore.

Nel primo passo per la realizzazione del prototipo si è affrontato il problema dal punto di vista dei processi BPEL inseriti in un contesto “coreografico” cioè senza un'entità esterna di coordinamento (orchestratore) ma con la logica di business nota agli attori coinvolti.

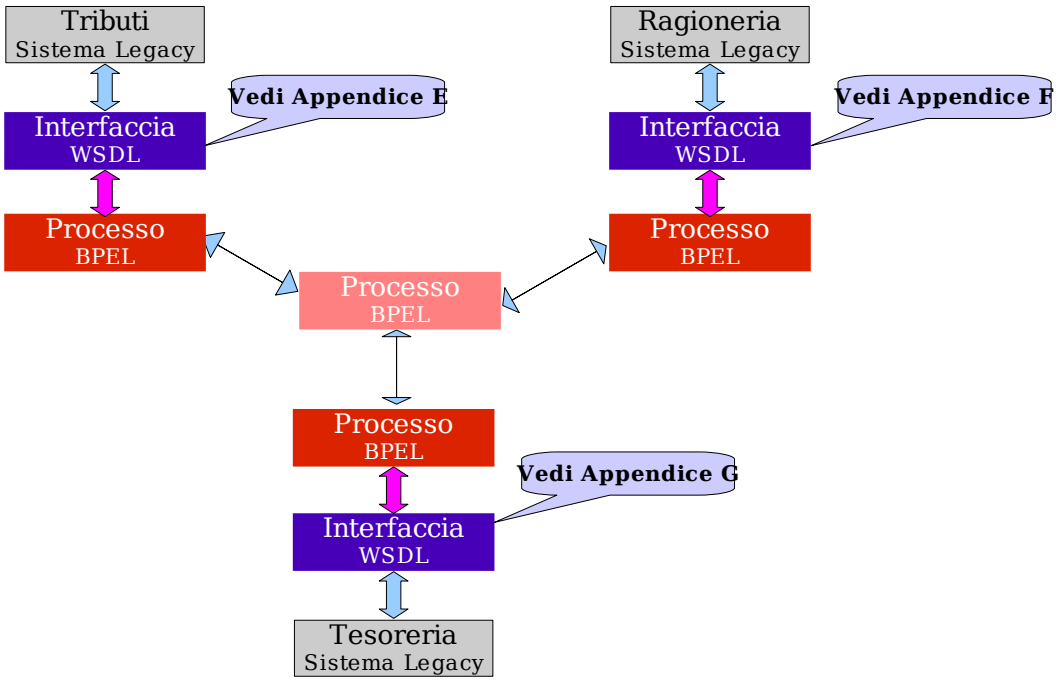
In questa ipotesi possiamo rappresentare il sistema come composto da:



Le interfacce rappresentano lo strato che esporta in modalità web services le funzionalità legacy che saranno invocate dai processi sovrastanti.

DeltaDator si occupa della definizione ed implementazione delle interfacce, IRST della parte processi. Si noti la separazione logica tra sistemi legacy, interfacce e processi. Questa stratificazione permette di cambiare gli stati con altri aventi implementazioni differenti, ad esempio sarà possibile sostituire la parte di processo realizzato con modalità coreografica e una realizzata con modalità orchestrata.

Nel caso di un approccio ad orchestrazione, avremmo la seguente situazione:



Casi di errore/eccezioni

I diagrammi fino ad ora rappresentati mostrano il caso nominale cioè quello che prevede il completamento con successo del processo.

Ovviamente nel mondo reale esiste una serie di casistiche che possono portare all'insuccesso dell'operazione. La tabella seguente riepiloga i casi di errore che possono verificarsi durante l'esecuzione del processo.

Test Case ID	***Errore***	***Azione correttiva***
CalcolaTassa_1	Errore di comunicazione: non si riesce ad invocare il webservice	Errore sw o di configurazione. Non ammesso in ambiente di produzione. Log del problema e fine del processo.
CalcolaTassa_2	Parametri non validi	Reinserisci nuovi parametri e riprova.
ProcessaRichiestaIncasso_1	Errore di comunicazione: non si riesce ad invocare il webservice	Retry. Finiti i retries, log del problema e fine/sospensione del processo. Sistemazione manuale del problema e riesecuzione/riattivazione del processo (in caso di riesecuzione, a partire dal calcolo della tassa)
ProcessaRichiestaIncasso_2	Parametri non validi. Si intende la validazione formale e sincrona.	Errore sw o di configurazione. Non ammesso in ambiente di produzione. Log del problema e fine/sospensione del processo. Sistemazione manuale del problema e riesecuzione/riattivazione del processo (in caso di riesecuzione, a partire dal calcolo della tassa)
RichiediStatoRichiestaIncasso_1	Errore di comunicazione: non si riesce ad invocare il webservice	Retry. Finiti i retries, log del problema e fine/sospensione del processo. Sistemazione manuale del problema e riesecuzione/riattivazione del processo (in caso di riesecuzione, a partire dal calcolo della tassa)
RichiediStatoRichiestaIncasso_2	Parametri non validi. Si intende la validazione formale e sincrona.	Errore sw o di configurazione. Non ammesso in ambiente di produzione. Log del problema e fine/sospensione del processo. Sistemazione manuale del problema e riesecuzione/riattivazione del processo (in caso di riesecuzione, a partire dal calcolo della tassa)
RichiediStatoRichiestaIncasso_3	Richiesta fallita. Si intende la validazione di dominio dei dati passati nella richiesta di incasso.	Errore sw o di configurazione. Non ammesso in ambiente di produzione. Log del problema e fine/sospensione del processo. Sistemazione manuale del problema e riesecuzione/riattivazione del processo (in caso di riesecuzione, a partire dal calcolo della tassa)

Test Case ID	*Errore*	*Azione correttiva*
RichiediStatoRichiestaIncasso_4	Il contribuente non ha pagato (insoluto).	Chiama *RegistraIncasso* specificando il problema nel parametro *esitoPagamento*. Fine del processo
RichiediListaIncassi_1	Errore di comunicazione: non si riesce ad invocare il webservice	Retry. Finiti i retries, log del problema e fine/sospensione del processo. Sistemazione manuale del problema e riesecuzione/riattivazione del processo (in caso di riesecuzione, a partire dalla richiesta lista incassi).
RichiediListaIncassi_2	Parametri non validi	Errore sw o di configurazione. Non ammesso in ambiente di produzione. Log del problema e fine/sospensione del processo. Sistemazione manuale del problema e riesecuzione/riattivazione del processo (in caso di riesecuzione, a partire dalla richiesta lista incassi).
RegistraIncassoProvvisorio_1	Errore di comunicazione: non si riesce ad invocare il webservice	Retry. Finiti i retries, log del problema e fine/sospensione del processo. Sistemazione manuale del problema e riesecuzione/riattivazione del processo (in caso di riesecuzione, a partire dalla richiesta lista incassi).
RegistraIncassoProvvisorio_2	Parametri non validi. Si intende la validazione formale e sincrona.	Errore sw o di configurazione. Non ammesso in ambiente di produzione. Log del problema e fine/sospensione del processo. Sistemazione manuale del problema e riesecuzione/riattivazione del processo (in caso di riesecuzione, a partire dalla richiesta lista incassi).
RichiediReversaleACopertura_1	Errore di comunicazione: non si riesce ad invocare il webservice	Retry. Finiti i retries, log del problema e fine/sospensione del processo. Sistemazione manuale del problema e riesecuzione/riattivazione del processo (in caso di riesecuzione, a partire dal ???).
RichiediReversaleACopertura_2	Parametri non validi. Si intende la validazione formale e sincrona.	Errore sw o di configurazione. Non ammesso in ambiente di produzione. Log del problema e fine/sospensione del processo. Sistemazione manuale del problema e riesecuzione/riattivazione del processo (in caso di riesecuzione, a partire da ???).
RichiediReversaleACopertura_3	Richiesta fallita. Si intende la validazione di dominio dei dati passati nella richiesta di registrazione dell'incasso provvisorio.	Errore sw o di configurazione. Non ammesso in ambiente di produzione. Log del problema e fine/sospensione del processo. Sistemazione manuale del problema e riesecuzione/riattivazione del processo (in caso di riesecuzione, a partire da ???).

Test Case ID	*Errore*	*Azione correttiva*
RegistraReversale_1	Errore di comunicazione	Retry. Finiti i retries, log del problema e fine/sospensione del processo. Sistemazione manuale del problema e riesecuzione/riattivazione del processo (in caso di riesecuzione, a partire da ???).
RegistraReversale_2	Parametri non validi. Si intende la validazione formale e sincrona.	Errore sw o di configurazione. Non ammesso in ambiente di produzione. Log del problema e fine/sospensione del processo. Sistemazione manuale del problema e riesecuzione/riattivazione del processo (in caso di riesecuzione, a partire da ???).
RichiediStatoRegistrazioneReversale_1	Errore di comunicazione: non si riesce ad invocare il webservice	Retry. Finiti i retries, log del problema e fine/sospensione del processo. Sistemazione manuale del problema e riesecuzione/riattivazione del processo (in caso di riesecuzione, a partire dal ???).
RichiediStatoRegistrazioneReversale_2	Parametri non validi. Si intende la validazione formale e sincrona.	Errore sw o di configurazione. Non ammesso in ambiente di produzione. Log del problema e fine/sospensione del processo. Sistemazione manuale del problema e riesecuzione/riattivazione del processo (in caso di riesecuzione, a partire da ???).
RichiediStatoRegistrazioneReversale_3	Richiesta fallita. Si intende la validazione di dominio dei dati passati nella richiesta di registrazione dell'incasso provvisorio.	Errore sw o di configurazione. Non ammesso in ambiente di produzione. Log del problema e fine/sospensione del processo. Sistemazione manuale del problema e riesecuzione/riattivazione del processo (in caso di riesecuzione, a partire da ???).

3. L'approccio al problema

Uno degli obiettivi del progetto è la definizione di una metodologia per approcciare il problema dell'inserimento in un contesto legacy delle tecnologie oggetto della sperimentazione.

La cosa ha un grande valore per DeltaDator come strumento di lavoro necessario alla replica della sperimentazione in ambiti di produzione più disparati. E' evidente che la separazione tra logica di business e esposizione dei servizi può essere considerata una problematica neutra rispetto al dominio di applicazione. Poter disporre di una lista di best practice per approcciare il problema garantisce il controllo delle attività e da visibilità al cliente su cosa si sta andando a fare.

La metodologia

L'approccio è di tipo “meet in the middle” e consta delle seguenti macrofasi:

- analisi del processo/definizione della coreografia della composizione
- web-servizzazione delle componenti
- implementazione della coreografia in BPEL

Analisi del processo - Definizione della coreografia

Rappresenta l'attività di analisi del processo del cliente e la sua rappresentazione coreografica intesa come definizione di alto livello dei servizi esposti e dei processi di business che li governano (si parla anche di abstract BPEL). Un esempio di questa analisi è l'activity diagram presentato in questo documento.

In questa fase è possibile realizzare dei moke-up che simulando il comportamento dei componenti permettono di validare il processo di coreografico o di orchestrazione.

Web-Servizzazione delle componenti

Rappresenta la fase di definizione dei WSDL cioè le specifiche di interfaccia dei web services che i componenti dovranno esporre.

Successivamente si passa alla fase di implementazione che normalmente si scontra con le problematiche tecnologiche di interfacciamento con i componenti legacy. E' una della fasi più delicate e potenzialmente meno sotto controllo in quanto i problemi di integrazione non sono sempre predicibili. Non da ultimo, il problema dell'integrazione con il componente qual'ora non sia disponibili sufficienti informazioni o il componente sia troppo blindato (tecniche di screen scraping).

Implementazione della coreografia in BPEL

Ultimo step, la definizione del processo o processi BPEL e la loro istanziazione in un ambiente di test per validare la soluzione.

4. Log delle attività svolte

In conformità con la metodologia descritta in precedenza, vengono di seguito elencate le attività concrete realizzate al fine di meglio contestualizzare quanto fatto:

- attività di self-training sulle tecnologie WebServices (WSDL, SOAP, AXIS) e BPEL
- esecuzione di un'attività di analisi dei domini (attuale stato dell'arte) con i referenti funzionali e tecnici dei prodotti DeltaDator Civilia e Tesoreria
- produzione di una prima versione dei 3 processi (abstract BPEL)
- validazione dei processi BPEL modellati assieme agli esperti di dominio
- definizione dei 3 file wsdl per le specifiche dei webservices e un file xsd per la definizione dei tipi comuni ai tre domini
- validazione dei WSDL assieme agli esperti di dominio
- a partire dalle specifiche WSDL sono state realizzate le attività di binding a Java e Centura per l'implementazione degli stubs e skeleton. Durante queste attività sono emerse svariati problemi di binding dei tipi non sempre supportati dal framework Axis e Centura che hanno innescato un processo iterativo di modifica ai WSDL.

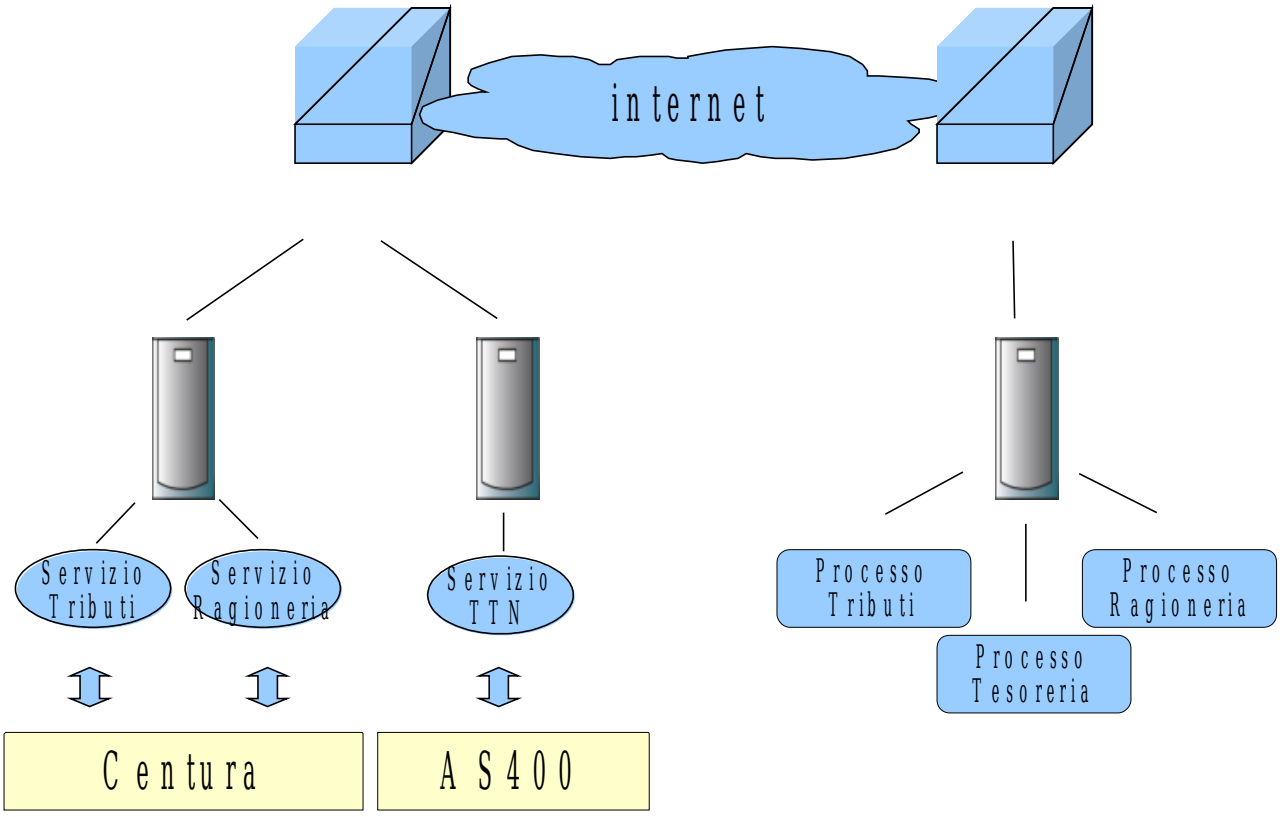
  In aggiunta, la piattaforma Centura si è rivelata ancora immatura per la realizzazione dei WebServices per cui è stato necessario realizzare un “Gateway” in DotNet che si fa carico di gestire l'invocazione del WebService (gestione richiesta, gestione protocolli Soap/Xml, binding) e di invocare il backend Centura via DCOM.

  Quest'ultima attività ha richiesto delle giornate di formazione sulle tecnologia DotNet e sull'integrazione con Centura presso Sipac.
- sono stati esposti in internet da parte di DeltaDator i servizi definiti per completare i test di integrazione e interoperabilità con IRST.
- Sono stati definiti i casi non nominali e realizzati i relativi aggiornamenti ai processi WSDL e ai BPEL
- Realizzazione della demo del prototipo

5. La demo del prototipo

Una delle attività più gratificanti e significative del progetto è la realizzazione di una demo che permette di capire e far capire i risultati del lavoro svolto.

La demo prevede la distribuzione geografica dei servizi e dei processi tra le sedi DeltaDator e IRST. In aggiunta i sistemi sui quali rispondono i servizi sono a loro volta distinti e tecnologicamente eterogenei. La figura seguente illustra la topologia:



La demo prevede due processi, il primo che descrive un caso nominale, il secondo un caso non nominale.

Caso nominale

Di fatto è quanto descritto nel sequence diagram sopra rappresentato. In dettaglio:

1. ***Tributi::CalcolaTassa***
   Il metodo viene invocato dal processo BPEL con i seguenti parametri:
   - ✔ CodiceContribuente = 12345
   - ✔ AnnoInizio = 01/01/2006
   - ✔ AnnoFine = 01/01/2006
   - ✔ CodiceRiscossione = 0434

- ✔ CausaleIncasso = TRIBUTI_TARSU

Il metodo ritorna l'oggetto RichiestaDiIncasso così avvalorato:

- ✔ Esercizio = 2006
- ✔ CausaleIncasso = TRIBUTI_TARSU
- ✔ AnagraficaDebitore.CodiceUtente = 12345
- ✔ AnagraficaDebitore.Nome = Mario
- ✔ AnagraficaDebitore.Cognome = Rossi
- ✔ AnagraficaDebitore.Indirizzo = “Via Giuseppe Verdi 3”
- ✔ AnagraficaDebitore.CAP = 38100
- ✔ AnagraficaDebitore.Comune = Trento
- ✔ AnagraficaDebitore.Provincia = TN
- ✔ AnagraficaDebitore.CodiceFiscale = RSSMRA62M10L123D
- ✔ AnagraficaRichiedente.CodiceRichiedente = 022205
- ✔ Banca.ABI = 03135
- ✔ Banca.CAB = 01699
- ✔ Banca.ContoCorrente = 000020021777
- ✔ Banca.CIN = A
- ✔ Incasso.ImportoTotale = 526,10
- ✔ Incasso.Rata.Codice = 11111
- ✔ Incasso.Rata.Importo = 526,10
- ✔ Incasso.Rata.Scadenza = 30/10/2006

2. ***TTN:ProcessaRichiestaIncasso***

Il metodo viene invocato dal processo BPEL con i seguenti parametri:

- ✔ RichiestaDiIncasso = vedi valore di ritorno di Tributi:CalcolaTassa

Il metodo ritorna:

- ✔ IdRichiesta = 123

3. ***TTN:RichiediStatoRichiestaIncasso*** *(primo tentativo)*

Il metodo viene invocato dal processo BPEL con i seguenti parametri:

- ✔ IdRichiesta = 123

Il metodo ritorna:

- ✔ EsitoRichiestaIncasso.StatoIncasso = NonEsitato

4. ***TTN:RichiediStatoRichiestaIncasso*** *(secondo tentativo)*
   Il metodo viene invocato dal processo BPEL con i seguenti parametri:
   - ✔ IdRichiesta = 123

   Il metodo ritorna:
   - ✔ EsitoRichiestaIncasso.StatoIncasso = Pagato
   - ✔ EsitoRichiestaIncasso.DataIncasso = 30/10/2006
   - ✔ EsitoRichiestaIncasso.NumeroIncasso = 11111

5. ***Tributi:RegistraIncassoTassa***
   Il metodo viene invocato dal processo BPEL con i seguenti parametri:
   - ✔ CodicePagamento = 11111
   - ✔ DataIncasso = 30/10/2006
   - ✔ EsitoPagamento = Pagato

6. ***TTN:RichiediListaIncassi***
   Il metodo viene invocato dal processo BPEL con i seguenti parametri:
   - ✔ CodiceEnte = 022205
   - ✔ Esercizio = 2006
   - ✔ Causale = TRIBUTI_TARSU
   - ✔ DataInizio = 01/10/2006
   - ✔ DataFine = 30/10/2006

   Il metodo ritorna:
   - ✔ Incassi.Incasso.NumeroIncasso = 11111
   - ✔ Incassi.Incasso.Importo = 526,10
   - ✔ Incassi.Incasso.DataIncasso = 30/10/2006
   - ✔ Incassi.Incasso.CausaleIncasso = TRIBUTI_TARSU

7. ***Ragioneria:RegistraIncassoProvvisorio***
   Il metodo viene invocato dal processo BPEL con i seguenti parametri:
   - ✔ Importo = 526,10
   - ✔ DataIncasso = 30/10/2006
   - ✔ OrigineIncasso = TRIBUTI_TARSU
   - ✔ NumeroProvvisorio = 66666
   - ✔ Storno = false

- ✔ Esercizio = 01/01/2006
- ✔ AnagraficaVersante.CodiceUtente = 12345
- ✔ AnagraficaVersante.Nome = Mario
- ✔ AnagraficaVersante.Cognome = Rossi
- ✔ AnagraficaVersante.Indirizzo = “Via Giuseppe Verdi 3”
- ✔ AnagraficaVersante.CAP = 38100
- ✔ AnagraficaVersante.Comune = Trento
- ✔ AnagraficaVersante.Provincia = TN
- ✔ AnagraficaVersante.CodiceFiscale = RSSMRA62M10L123D

8. ***Ragioneria:RichiediReversaleACopertura*** *(primo tentativo)*
   Il metodo viene invocato dal processo BPEL con i seguenti parametri:
   - ✔ NumeroProvvisorio = 666666

   Il metodo ritorna:
   - ✔ Disponibile = 66666

9. ***Ragioneria:RichiediReversaleACopertura*** *(secondo tentativo)*
   Il metodo viene invocato dal processo BPEL con i seguenti parametri:
   - ✔ NumeroProvvisorio = 666666

   Il metodo ritorna:
   - ✔ Disponibile = true
   - ✔ Reversale.CodiceEnte = 022205
   - ✔ Reversale.DescrizioneEnte = “Comune di Trento”
   - ✔ Reversale.Esercizio = 2006
   - ✔ Reversale.NumeroReversale = 77777
   - ✔ Reversale.OggettoReversale = “Pagamento Rata Tributi Tarsu”
   - ✔ Reversale.Importo = 526,10
   - ✔ Reversale.DataReversale = 30/10/2006
   - ✔ Reversale.DatiBilancio.Gestione = C
   - ✔ Reversale.DatiBilancio.CodiceCapitolo = 4.18.07
   - ✔ Reversale.DatiBilancio.CodiceArticolo = 444
   - ✔ Reversale.DatiBilancio.Anno = 2006
   - ✔ Reversale.RigaReversale.NumeroIncasso = 88888

- ✔ Reversale.RigaReversale.ImportoIncrociato = 526,10
- ✔ Reversale.RigaReversale.CodiceCGE = EC4337

10. ***TTN:RegistraReversale***
    Il metodo viene invocato dal processo BPEL con i seguenti parametri:
    - ✔ Reversale = vedi valore di ritorno di Ragioneria:RichiediReversaleACopertura

    Il metodo ritorna:
    - ✗ IdRichiesta = 22222

11. ***TTN:RichiediStatoRegistrazioneReversale*** *(primo tentativo)*
    Il metodo viene invocato dal processo BPEL con i seguenti parametri:
    - ✔ dRichiesta = 22222

    Il metodo ritorna:
    - ✔ EsitoRegistrazioneReversale = InRegistrazione
    - ✔ Descrizione = “”

12. ***TTN:RichiediStatoRegistrazioneReversale*** *(secondo tentativo)*
    Il metodo viene invocato dal processo BPEL con i seguenti parametri:
    - ✔ dRichiesta = 22222

    Il metodo ritorna:
    - ✔ EsitoRegistrazioneReversale = Registrato
    - ✔ Descrizione = “Registrato con successo”

***Caso non nominale* (Test Case Id: RichiediStatoRichiestaIncasso_4)**

In questo caso il debitore di imposta non paga. In dettaglio:

1. ***Tributi::CalcolaTassa***
   Il metodo viene invocato dal processo BPEL con i seguenti parametri:
   - ✔ CodiceContribuente = 67890
   - ✔ AnnoInizio = 01/01/2006
   - ✔ AnnoFine = 01/01/2006
   - ✔ CodiceRiscossione = 0434
   - ✔ CausaleIncasso = TRIBUTI_TARSU

Il metodo ritorna l'oggetto RichiestaDiIncasso così avvalorato:

- ✔ Esercizio = 2006
- ✔ CausaleIncasso = TRIBUTI_TARSU
- ✔ AnagraficaDebitore.CodiceUtente = 67890
- ✔ AnagraficaDebitore.Nome = Paolo
- ✔ AnagraficaDebitore.Cognome = Incanna
- ✔ AnagraficaDebitore.Indirizzo = “Via sotto il ponte”
- ✔ AnagraficaDebitore.CAP = 38100
- ✔ AnagraficaDebitore.Comune = Trento
- ✔ AnagraficaDebitore.Provincia = TN
- ✔ AnagraficaDebitore.CodiceFiscale = NCNPLA80A01M345P
- ✔ AnagraficaRichiedente.CodiceRichiedente = 022205
- ✔ Banca.ABI = 03135
- ✔ Banca.CAB = 01699
- ✔ Banca.ContoCorrente = 000020021333
- ✔ Banca.CIN = A
- ✔ Incasso.ImportoTotale = 100,99
- ✔ Incasso.Rata.Codice = 11112
- ✔ Incasso.Rata.Importo = 100,99
- ✔ Incasso.Rata.Scadenza = 30/10/2006

2. ***TTN:ProcessaRichiestaIncasso***

Il metodo viene invocato dal processo BPEL con i seguenti parametri:

- ✔ RichiestaDiIncasso = vedi valore di ritorno di Tributi:CalcolaTassa

Il metodo ritorna:

- ✔ IdRichiesta = 124

3. ***TTN:RichiediStatoRichiestaIncasso***

Il metodo viene invocato dal processo BPEL con i seguenti parametri:

- ✔ IdRichiesta = 124

Il metodo ritorna:

- ✔ EsitoRichiestaIncasso.StatoIncasso = Insoluto
- ✔ EsitoRichiestaIncasso.Messaggio419 = P

4. ***Tributi:RegistraIncassoTassa***

Il metodo viene invocato dal processo BPEL con i seguenti parametri:

- ✔ CodicePagamento = 11112
- ✔ DataIncasso = 30/10/2006
- ✔ EsitoPagamento = Insoluto

Appendice A – Tracciato di esportazione Tributi

Il tracciato è composto da una serie di record, uno per riga. Il record è strutturato in da una sequenza di campi divisi dal separatore # (cancelletto).

La tabella seguente descrive i campi di un record con a fianco alcuni esempi.

Nome del Campo	**Esempio di valore possibile**	**Nota**
CU	22	
NOME1	CONSORZIO	
INDIRIZZO	Via G. Roma N°180	
CAP	30081	
DEST	Rovereto	
PROV	TN	
CODFISC	00505210423	
ESTERO		
TOTALE	1326,66	
NOME2		
INDIRIZZO2		
CAP2		
DEST2		
PROV2		
UBIC1	Strada del vino 69 N.2/A	
CAT1	Magazzini e depositi	
MQ1	1680	
TARIFFA1	0,250000	
RIDUZIONE1	0,00	
MAGGIORAZIONE1		
CAUSALE1		
IMPORTO1	420,00	
TRIBUTO1	0434	
UBIC2	Strada del vino 69 N.2/A	
CAT2	Magazzini e depositi	
MQ2	300	
TARIFFA2	0,250000	
RIDUZIONE2	0,00	
MAGGIORAZIONE2		
CAUSALE2		
IMPORTO2	75,00	
TRIBUTO2	0434	
UBIC3	Strada del vino 69 N.2/A	
CAT3	Negozi beni non deperibili	
MQ3	126	
TARIFFA3	1,060000	
RIDUZIONE3	0,00	

Nome del Campo	Esempio di valore possibile	Nota
MAGGIORAZIONE3		
CAUSALE3		
IMPORTO3	133,56	
TRIBUTO3	0434	
UBIC4	Via G. Roma N.180	
CAT4	Uffici	
MQ4	770	
TARIFFA4	0,750000	
RIDUZIONE4	0,00	
MAGGIORAZIONE4		
CAUSALE4		
IMPORTO4	577,50	
TRIBUTO4	0434	
UBIC5		
CAT5		
MQ5		
TARIFFA5		
RIDUZIONE5		
MAGGIORAZIONE5		
CAUSALE5		
IMPORTO5		
TRIBUTO5		
UBIC6		
CAT6		
MQ6		
TARIFFA6		
RIDUZIONE6		
MAGGIORAZIONE6		
CAUSALE6		
IMPORTO6		
TRIBUTO6		
UBIC7		
CAT7		
MQ7		
TARIFFA7		
RIDUZIONE7		
MAGGIORAZIONE7		
CAUSALE7		
IMPORTO7		
TRIBUTO7		
UBIC8		
CAT8		
MQ8		
TARIFFA8		

Nome del Campo	Esempio di valore possibile	Nota
RIDUZIONE8		
MAGGIORAZIONE8		
CAUSALE8		
IMPORTO8		
TRIBUTO8		
UBIC9		
CAT9		
MQ9		
TARIFFA9		
RIDUZIONE9		
MAGGIORAZIONE9		
CAUSALE9		
IMPORTO9		
TRIBUTO9		
UBIC10		
CAT10		
MQ10		
TARIFFA10		
RIDUZIONE10		
MAGGIORAZIONE10		
CAUSALE10		
IMPORTO10		
TRIBUTO10		
UBIC11		
CAT11		
MQ11		
TARIFFA11		
RIDUZIONE11		
MAGGIORAZIONE11		
CAUSALE11		
IMPORTO11		
TRIBUTO11		
UBIC12		
CAT12		
MQ12		
TARIFFA12		
RIDUZIONE12		
MAGGIORAZIONE12		
CAUSALE12		
IMPORTO12		
TRIBUTO12		
UBIC13		
CAT13		
MQ13		

Nome del Campo	Esempio di valore possibile	Nota
TARIFFA13		
RIDUZIONE13		
MAGGIORAZIONE13		
CAUSALE13		
IMPORTO13		
TRIBUTO13		
UBIC14		
CAT14		
MQ14		
TARIFFA14		
RIDUZIONE14		
MAGGIORAZIONE14		
CAUSALE14		
IMPORTO14		
TRIBUTO14		
UBIC15		
CAT15		
MQ15		
TARIFFA15		
RIDUZIONE15		
MAGGIORAZIONE15		
CAUSALE15		
IMPORTO15		
TRIBUTO15		
UBIC16		
CAT16		
MQ16		
TARIFFA16		
RIDUZIONE16		
MAGGIORAZIONE16		
CAUSALE16		
IMPORTO16		
TRIBUTO16		
UBIC17		
CAT17		
MQ17		
TARIFFA17		
RIDUZIONE17		
MAGGIORAZIONE17		
CAUSALE17		
IMPORTO17		
TRIBUTO17		
UBIC18		
CAT18		

Nome del Campo	Esempio di valore possibile	Nota
MQ18		
TARIFFA18		
RIDUZIONE18		
MAGGIORAZIONE18		
CAUSALE18		
IMPORTO18		
TRIBUTO18		
UBIC19		
CAT19		
MQ19		
TARIFFA19		
RIDUZIONE19		
MAGGIORAZIONE19		
CAUSALE19		
IMPORTO19		
TRIBUTO19		
UBIC20		
CAT20		
MQ20		
TARIFFA20		
RIDUZIONE20		
MAGGIORAZIONE20		
CAUSALE20		
IMPORTO20		
TRIBUTO20		
PERC_ECA	5	
ADD_ECA	60,31	
PERC_MAGG_ECA	5	
MAGG_ECA	60,31	
PERC_TRIB_PROV	0,00	
TRIB_PROV	0,00	
TOT_NETTO	1206,06	
VCAMPO1	200600000050	Codice del pagamento unico
VCAMPO2	200600000051	Codice del pagamento prima rata
VCAMPO3	200600000052	Codice del pagamento seconda rata
VCAMPO4	200600000053	Codice del pagamento terza rata
VCAMPO5	200600000054	Codice del pagamento quarta rata
IMPO1	331,67	
IMPO2	331,67	
IMPO3	331,67	
IMPO4	331,65	
TOTLETT	Milletrecentoventisei/66	
TOTLETT1	Trecentotrentuno/67	
TOTLETT2	Trecentotrentuno/67	

Nome del Campo	Esempio di valore possibile	Nota
TOTLETT3	Trecentotrentuno/67	
TOTLETT4	Trecentotrentuno/65	
DATA1	28/02/2006	Data di scadenza pagamento unico o prima rata
DATA2	30/04/2006	Data di scadenza seconda rata
DATA3	30/06/2006	Data di scadenza terza rata
DATA4	30/08/2006	Data di scadenza quarta rata
SANZIONE		
INTERESSI		

Appendice B – Tracciato incasso Tesoreria

Codice ente (10)
con: obbligatorio
deve esistere nel file ENTI

Esercizio (4,0)
con: obbligatorio; deve essere un esercizio aperto

Tipo codice debitore (valori limitati) (1)
des: indica di che tipo è il CODICE DEBITORE
val: _=Codice di tesorererìa
1=Codice anagrafe generale
2=Codice attribuita all'Ente

Codice debitore (11)
con: non obbligatorio; se presente deve esistere nell'archivio PERSONE

Descrizione debitore (50)

Indirizzo debitore (30)

CAP debitore (5,0)

Località debitore (30)
dft: se indicato CODICE DEBITORE vengono acquisiti dal file PERSONE
con: obbligatori se non indicato CODICE DEBITORE

Descrizione causale emissione (280)
con: obbligatorio

Codice sottoconto (5)
con: non obbligatorio; se presente deve esistere sulla tabella SOTTOCONTI

Codice causale di incasso (3)
dft: viene acquisito dal file PERSONE ENTI se è stato specificato il CODICE DEBITORE e se per quel debitore sono state definite le condizioni per l'ente in esame
con: obbligatorio se non indicato CODICE DEBITORE; deve esistere sulla tabella CAUSALI DI INCASSO

Divisa dell'incasso (3)
des: indica la divisa con cui è stato effettuato l'incasso.
val: EUR=euro

Importo netto (14,2+1)
con: obbligatorio

Flag importo spese (1)
des: indica se l'importo delle spese deve essere valorizzato automaticamente dalla CAUSALE INCASSO
val: _=l'IMPORTO SPESE viene acquisito dalla causale
1=l'IMPORTO SPESE non viene acquisito dalla causale

Importo Spese (7,2)
dft: se FLAG IMPORTO SPESE=_ viene acquisito dalla CAUSALE INCASSO

Attribuzione Spese (1)
dft: assume significato solo se l'IMPORTO SPESE è diverso da 0
val: E=ENTE
V=VERSANTE
T=TESORIERE

Codice esenzione bollo (3)
con: se impostato deve esistere sulla tabella ESENZIONI

Attribuzione bollo (1)
des: assume significato solo se l'IMPORTO LORDO > dell'importo minimo previsto per l'assegnazione di bolli e se il CODICE ESENZIONE non è stato valorizzato
val: E=ENTE
V=VERSANTE
T=TESORIERE
dft: se non valorizzato viene assunto dalla CAUSALE INCASSO.

Valuta ente (data) (8,0)
con: Non obbligatorio.

Valuta versante (data) (8,0)
dft: se il tipo di incasso è in conto corrente viene calcolato in base alla tabella GRUPPI BANCHE

Conto corrente del versante (12)
dft: viene acquisito dal file PERSONE ENTI se è stato specificato il CODICE DEBITORE se per quel debitore sono state definite le condizioni per l'ente in esame
con: obbligatorio se il TIPO INCASSO è in conto corrente e non è stato indicato il CODICE DEBITORE; se valorizzato e se il versante è correntista della banca tesoriere deve essere presente sull'archivio CONTI CORRENTI

Incasso Fruttifero od Infruttifero (1)
val: F=Fruttifero
I=Infruttifero
con: obbligatorio se ente in tesoreria unica

Parte Corrente - Capitale (1)
val: C=Corrente
P=Capitale
con: obbligatorio se ente USL

Codice fiscale debitore (16)

P.IVA debitore (11)

Appendice C – Tracciato reversale (a copertura) Tesoreria

Testata

Tipo Record (1)
val: valore fisso ad 'T'.

Codice ente (10)
con: obbligatorio; deve esistere nel file ENTI

Esercizio (4,0)
con: obbligatorio; deve essere un esercizio aperto

Nr. reversale (5,0)
con: obbligatorio e univoco nell'ambito dell'ente e dell'esercizio

Oggetto della reversale (50)

Data della reversale (Data) (8,0)

Gestione (valori limitati) (1)
val: C=Competenza
R=Residuo
con: obbligatorio

Codice capitolo (7)

Codice articolo (3)

Anno (4,0)
con: identificano assieme un capitolo che deve essere presente sul file dei CAPITOLI obbligatori

Codice voce economica (2)
con: obbligatorio se l'ente utilizza il nuovo bilancio; in questo caso viene controllato.

Importo reversale (14,2+1)
con: obbligatorio

Reversale a copertura (S/N) (1)

Codice CGE (4)
con: obbligatorio per gli enti che utilizzano il sistema SIOPE
se valorizzato, comporta che tutti i codici gestionali sulle righe siano valorizzati allo stesso modo e coincidano.

Righe

Tipo Record (1)
val: valore fisso ad 'R'.

Tipo codice debitore (1)
des: indica di che tipo è il CODICE DEBITORE
val: _=Codice di tesorereria
1=Codice anagrafe generale
2=Codice attribuita all'Ente

Codice debitore (11)
con: non obbligatorio; se presente deve esistere nell'archivio PERSONE

Descrizione debitore (50)

Indirizzo debitore (30)

CAP debitore (5,0)

Località debitore (30)

Provincia (2)
dft: se indicato CODICE DEBITORE vengono acquisiti dal file PERSONE
con: obbligatori se non indicato CODICE DEBITORE

Descrizione causale emissione (280)
con: obbligatorio

Codice causale incasso (3)
dft: viene acquisito dal file PERSONE ENTI se è stato specificato il CODICE DEBITORE e se per quel debitore sono state definite le condizioni per l'ente in esame
con: obbligatorio se non indicato CODICE DEBITORE; deve esistere sulla tabella CAUSALI DI INCASSO

Importo lordo (14,2+1)
con: obbligatorio

Importo netto (14,2+1)

Sottoconto per somme vincolate (5)
con: il codice sottoconto, se indicato, deve essere presente sul file SOTTOCONTI

Divisa dell'incasso (3)
EUR=euro

Codice esenzione bollo (3)
con: se impostato deve esistere sulla tabella ESENZIONI

Attribuzione bollo (1)
des: assume significato solo se l'IMPORTO LORDO > dell'importo minimo previsto per l'assegnazione di bolli e se il CODICE ESENZIONE non è stato valorizzato
val: E=ENTE
V=VERSANTE

T=TESORIERE
dft: se non valorizzato viene assunto dalla CAUSALE INCASSO.

Valuta ente (data) (8,0)

Valuta debitore (data) (8,0)

Conto corrente del debitore(12)
dft: viene acquisito dal file PERSONE ENTI se è stato specificato il CODICE DEBITORE e se per quel debitore sono state definite le condizioni per l'ente in esame
con: obbligatorio se il TIPO INCASSO è addebito in conto corrente e non è stato indicato il CODICE DEBITORE; se valorizzato il DEBITORE deve essere correntista della banca tesoriere.

Incasso Fruttifero od Infruttifero (1)
val: F=Fruttifero
I=Infruttifero

Codice fiscale debitore (16)

P.IVA debitore (11)

Riferimento sospeso da regolarizzare (5,0)
Con: se valorizzato controlla l'esistenza del sospeso.

Codice CGE (4)
con: obbligatorio per gli enti che utilizzano il sistema SIOPE
deve coincidere con il codice della testata se presente sulla testata; altrimenti può essere un codice gestionale valido qualsiasi.

Appendice D – Tracciato giornale di cassa Tesoreria

Data giornale di cassa (8,0)
Data di effettuazione dei movimenti

Esercizio (4,0)
Esercizio di riferimento

Codice ente (10)
Codice attribuito dal tesoriere all'ente

Flag tipo movimento (2)
"**PE**"=provvisorio di entrata
"**PU**"=provvisorio di uscita
"**RV**"=movimento di entrata collegato a reversale
"**MN**"=movimento di uscita collegato a mandato

Progressivo movimento (7,0)
Progressivo attribuito dalla procedura tesoriere ai movimenti di entrata / uscita

Numero documento di riferimento (5,0)
Numero della reversale o del mandato di riferimento;
99999 in caso di riferimenti multipli

Importo del movimento (14,2) +1
Importo del movimento contabile

Importo regolarizzato del movimento (14,2) +1
Importo del movimento regolarizzato con documenti di bilancio (mandati / reversali)

Importo residuo da regolarizzare del movimento (14,2) +1
Importo del movimento non regolarizzato da documenti di bilancio

Anagrafica debitore/creditore (50)

Causale descrittiva del movimento (100)

Importo bollo (7,2) +1

Importo spese (7,2) +1

Sottoconto (5)
Codice sottoconto di imputazione:
- valore a spazi nel caso di imputazione conto libero
- codice sottoconto codifica tesoriere nel caso di imputazione conto vincolato

Data valuta del movimento (8,0)

Data di estrazione (8,0)
Data creazione file

Note:

File “giornale di cassa”

Il file prodotto è un file ASCII con struttura a campi di lunghezza fissa con formato record unico, un record per ogni movimento.

Le date sono esposte nel formato **aaaammgg.**

La grandezza di ogni campo è indicata a fianco del nome del campo.

Il tipo del campo ed è da intendersi:

- **alfanumerico** se non viene specificata la lunghezza della parte decimale
- **numerico** altrimenti.

La grandezza dei campi numerici è indicata con il formalismo **(i,d) +1** dove:

i indica il numero complessivo di cifre disponibili per il campo

d quante posizioni di queste sono riservate alla parte decimale

+1 indica la presenza di un carattere ulteriore per il segno dopo la parte riservata al campo con valori ammissibili **“+”** o **“-“**.

Tutti i campi numerici sono esposti e devono essere esposti con zeri non significativi.

Esempio:

importo a (14,2) +1

indica che il numero è di 14 cifre di cui due sono riservate alla parte decimale +1 carattere per il segno (in tutto 15 posizioni).

Nel caso di importo positivo di +111,22 (centoundici euro e 22 centesimi) avrò sul file: (00000000011122+)

Appendice E – Specifiche WSDL Tributi

WSDL location: F:\Firb\Tributi.wsdl
targetnamespace: tributi.firb.deltadator.com

services	bindings	porttypes	messages
TributiService	TributiBinding	TributiPortType	CalcolaTassaInput
			CalcolaTassaOutput
			OperazioneFallita
			OutputMessage
			ParametroNonValido
			RegistraIncassoTassaInput

service TributiService

ports	TributiPort
	binding y:TributiBinding

porttype TributiPortType

operations	CalcolaTassa	
	input	y:CalcolaTassaInput
	output	y:CalcolaTassaOutput
	fault	y:ParametroNonValido
	fault	y:OperazioneFallita
	RegistraIncassoTassa	
	input	y:RegistraIncassoTassaInput
	output	y:OutputMessage
	fault	y:ParametroNonValido
	fault	y:OperazioneFallita

message CalcolaTassaInput

parts	codiceContribuente
	type xs:string
	annoInizio
	type xs:date
	annoFine
	type xs:date
	codiceRiscossione
	type xs:string
	tipoTassa
	type dd:CausaleIncasso

message CalcolaTassaOutput

parts	infoTassa
	type dd:RichiestaDiIncasso

message OutputMessage

parts	esito
	type xs:string

message RegistraIncassoTassaInput

parts	codicePagamento
	type xs:string
	dataPagamento
	type xs:date
	esitoPagamento
	type xs:string

message ParametroNonValido

parts	errore
	type dd:E_ListaParametriNonValidi

message OperazioneFallita

parts	errore
	type dd:E_OperazioneFallita

Appendice F – Specifiche WSDL Tesoreria

WSDL location: F:\Firb\TTN.wsdl
targetnamespace: ttn.firb.deltadator.com

services	bindings	porttypes	messages
TTNService	TTNBinding	TTNPortType	OperazioneFallita
			ParametroNonValido
			ProcessaRichiestaIncassoInput
			ProcessaRichiestaIncassoOutput
			RegistraReversaleInput
			RegistraReversaleOutput
			RichiediListaIncassiInput
			RichiediListaIncassiOutput
			RichiediStatoRegistrazioneReversaleInput
			RichiediStatoRegistrazioneReversaleOutput
			RichiediStatoRichiestaIncassoInput
			RichiediStatoRichiestaIncassoOutput

service TTNService

ports	TTNPort
	binding y:TTNBinding

porttype TTNPortType

operations	ProcessaRichiestaIncasso
	input y:ProcessaRichiestaIncassoInput
	output y:ProcessaRichiestaIncassoOutput
	fault y:ParametroNonValido
	fault y:OperazioneFallita

	RichiediStatoRichiestaIncasso
	input y:RichiediStatoRichiestaIncassoInput
	output y:RichiediStatoRichiestaIncassoOutput
	fault y:ParametroNonValido
	fault y:OperazioneFallita
	RichiediListaIncassi
	input y:RichiediListaIncassiInput
	output y:RichiediListaIncassiOutput
	fault y:ParametroNonValido
	fault y:OperazioneFallita
	RegistraReversale
	input y:RegistraReversaleInput
	output y:RegistraReversaleOutput
	fault y:ParametroNonValido
	fault y:OperazioneFallita
	RichiediStatoRegistrazioneReversale
	input y:RichiediStatoRegistrazioneReversaleInput
	output y:RichiediStatoRegistrazioneReversaleOutput
	fault y:ParametroNonValido
	fault y:OperazioneFallita

message ProcessaRichiestaIncassoInput

parts	richiestaIncasso
	type dd:RichiestaDiIncasso

message ProcessaRichiestaIncassoOutput

parts	idRichiesta
	type xs:string

message RichiediStatoRichiestaIncassoInput

parts	idRichiesta
	type xs:string

message RichiediStatoRichiestaIncassoOutput

parts	esito
	type dd:EsitoRichiestaIncasso

message RichiediListaIncassiInput

parts	codiceEnte
	type xs:string
	esercizio
	type xs:integer
	causale
	type dd:CausaleIncasso
	dataInizio
	type xs:date

	dataFine type xs:date

message RichiediListaIncassiOutput

parts	incassi type dd:Incassi

message RegistraReversaleInput

parts	reversale type dd:Reversale

message RegistraReversaleOutput

parts	idRichiesta type xs:string

message RichiediStatoRegistrazioneReversaleInput

parts	idRichiesta type xs:string

message RichiediStatoRegistrazioneReversaleOutput

parts	esito type dd:EsitoRegistrazioneReversale

message ParametroNonValido

parts	errore
	type dd:E_ListaParametriNonValidi

message OperazioneFallita

parts	errore
	type dd:E_OperazioneFallita

Appendice G – Specifiche WSDL Ragioneria

WSDL location: F:\Firb\Ragioneria.wsdl
targetnamespace: ragioneria.firb.deltadator.com

services	bindings	porttypes	messages
RagioneriaService	RagioneriaBinding	RagioneriaPortType	OperazioneFallita
			OutputMessage
			ParametroNonValido
			RegistraIncassoProvvisorioInput
			RegistraIncassoProvvisorioOutput
			RichiediReversaleACoperturaInput
			RichiediReversaleACoperturaOutput

service RagioneriaService

ports	RagioneriaPort
	binding y:RagioneriaBinding

porttype RagioneriaPortType

operations	RegistraIncassoProvvisorio
	input y:RegistraIncassoProvvisorioInput
	output y:OutputMessage
	fault y:ParametroNonValido
	fault y:OperazioneFallita
	RichiediReversaleACopertura
	input y:RichiediReversaleACoperturaInput

	output y:RichiediReversaleACoperturaOutput fault y:ParametroNonValido fault y:OperazioneFallita

message RegistraIncassoProvvisorioInput

parts	importo type xs:decimal dataIncasso type xs:date origineIncasso type dd:CausaleIncasso numeroProvvisorio type xs:string storno type xs:boolean esercizio type xs:date anagraficaVersante type dd:AnagraficaDebitore

message RegistraIncassoProvvisorioOutput

parts	idIncasso
	type xs:string

message RichiediReversaleACoperturaInput

parts	numeroProvvisorio
	type xs:string

message RichiediReversaleACoperturaOutput

parts	reversale
	type dd:Reversale
	disponibile
	type xs:boolean

message ParametroNonValido

parts	errore
	type dd:E_ListaParametriNonValidi

message OperazioneFallita

parts	errore
	type dd:E_OperazioneFallita

message OutputMessage

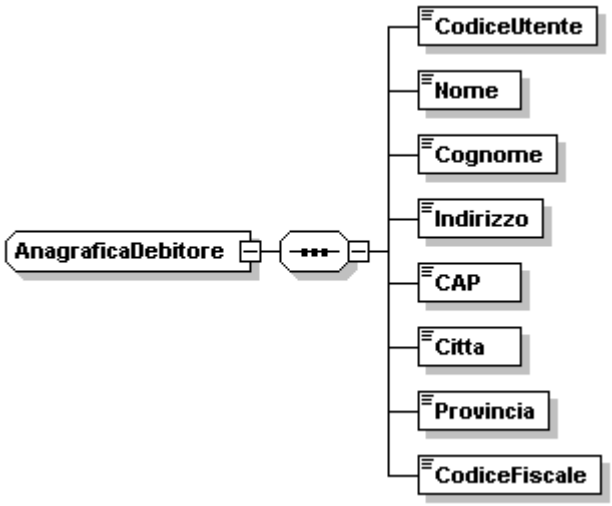
parts	esito
	type xs:string

Appendice H – Specifiche XSD Types

schema location: **Types.xsd**
attribute form default:
element form default: **unqualified**
targetNamespace: **types.firb.deltadator.com**

Complex types	Simple types
AnagraficaDebitore	**CausaleIncasso**
DatiParametroErrato	**GestioneDatiBilancio**
E_ListaParametriNonValidi	**Messaggio419**
E_OperazioneFallita	**StatoIncasso**
EsitoRegistrazioneReversale	**StatoRegistrazione**
EsitoRichiestaIncasso	
Incassi	
Incasso	
Rata	
Reversale	
RichiestaDiIncasso	

complexType **AnagraficaDebitore**

diagram	
namespace	types.firb.deltadator.com

children	**CodiceUtente Nome Cognome Indirizzo CAP Citta Provincia CodiceFiscale**

element **AnagraficaDebitore/CodiceUtente**

diagram	
type	**xs:string**

element **AnagraficaDebitore/Nome**

diagram	
type	**xs:string**

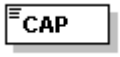
element **AnagraficaDebitore/Cognome**

diagram	
type	**xs:string**

element **AnagraficaDebitore/Indirizzo**

diagram	
type	**xs:string**

element **AnagraficaDebitore/CAP**

diagram	
type	**xs:string**

element **AnagraficaDebitore/Citta**

diagram	

type	**xs:string**

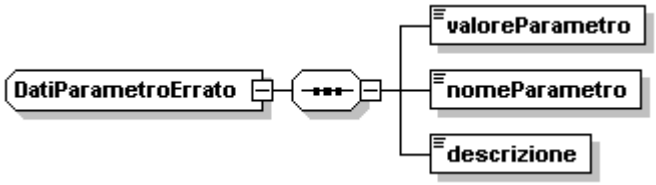
element **AnagraficaDebitore/Provincia**

diagram	
type	**xs:string**

element **AnagraficaDebitore/CodiceFiscale**

diagram	
type	**xs:string**

complexType **DatiParametroErrato**

diagram	
namespace	types.firb.deltadator.com
children	**valoreParametro** **nomeParametro** **descrizione**

element **DatiParametroErrato/valoreParametro**

diagram	
type	**xs:string**

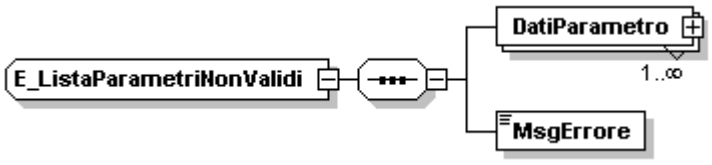
element **DatiParametroErrato/nomeParametro**

diagram	
type	**xs:string**

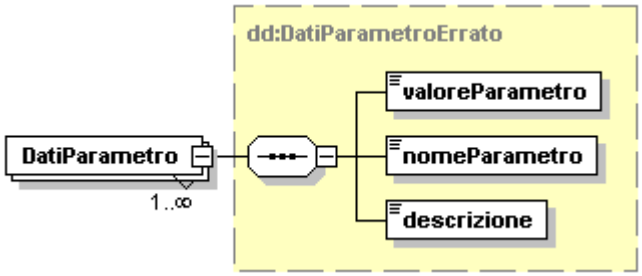
element **DatiParametroErrato/descrizione**

diagram	
type	**xs:string**

complexType **E_ListaParametriNonValidi**

diagram	
namespace	types.firb.deltadator.com
children	**DatiParametro** **MsgErrore**

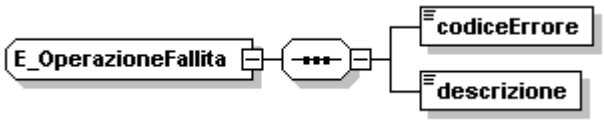
element **E_ListaParametriNonValidi/DatiParametro**

diagram	
type	**dd:DatiParametroErrato**
children	**valoreParametro** **nomeParametro** **descrizione**

element **E_ListaParametriNonValidi/MsgErrore**

diagram	
type	**xs:string**

complexType **E_OperazioneFallita**

diagram	
namespace	types.firb.deltadator.com
children	**codiceErrore** **descrizione**

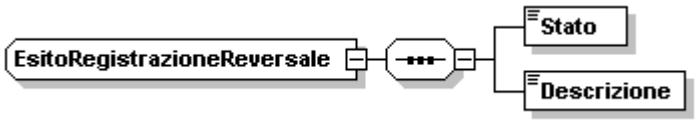
element **E_OperazioneFallita/codiceErrore**

diagram	
type	**xs:string**

element **E_OperazioneFallita/descrizione**

diagram	
type	**xs:string**

complexType **EsitoRegistrazioneReversale**

diagram	
namespace	types.firb.deltadator.com
children	**Stato** **Descrizione**

element **EsitoRegistrazioneReversale/Stato**

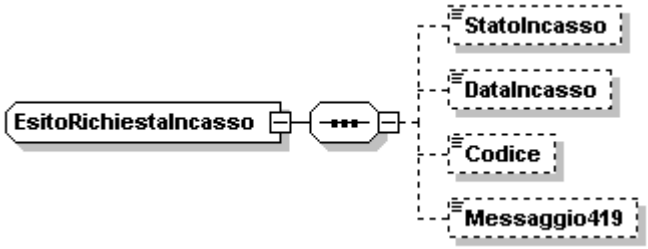
diagram	
type	**dd:StatoRegistrazione**

facets	enumeration InRegistrazione enumeration Registrato enumeration Fallito

element **EsitoRegistrazioneReversale/Descrizione**

diagram	
type	**xs:string**

complexType **EsitoRichiestaIncasso**

diagram	
namespace	types.firb.deltadator.com
children	**StatoIncasso DataIncasso Codice Messaggio419**

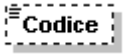
element **EsitoRichiestaIncasso/StatoIncasso**

diagram	
type	**dd:StatoIncasso**
facets	enumeration NonEsitato enumeration Pagato enumeration Insoluto enumeration Errato

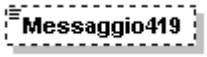
element **EsitoRichiestaIncasso/DataIncasso**

diagram	
type	**xs:date**

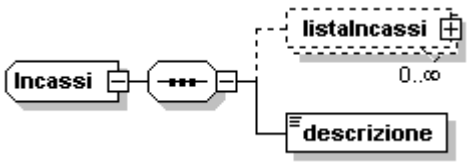
element **EsitoRichiestaIncasso/Codice**

diagram	
type	**xs:string**

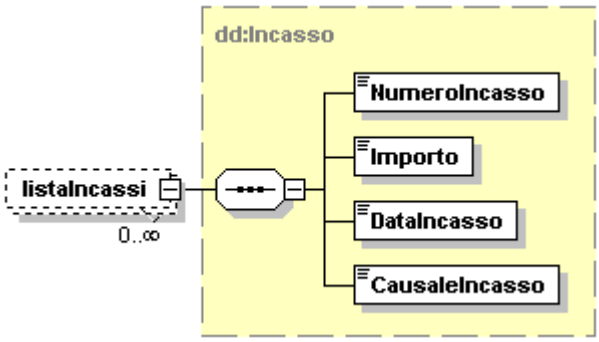
element **EsitoRichiestaIncasso/Messaggio419**

diagram	
type	**dd:Messaggio419**
facets	enumeration TRANSAZIONE_ESEGUITA enumeration PROBLEMA enumeration ERRORE

complexType **Incassi**

diagram	
namespace	types.firb.deltadator.com
children	**listaIncassi** **descrizione**

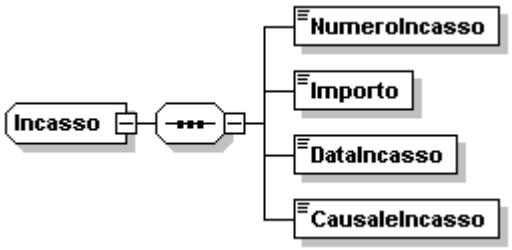
element **Incassi/listaIncassi**

diagram	
type	**dd:Incasso**
children	**NumeroIncasso** **Importo** **DataIncasso** **CausaleIncasso**

element **Incassi/descrizione**

diagram	
type	**xs:string**

complexType **Incasso**

diagram	
namespace	types.firb.deltadator.com
children	**NumeroIncasso** **Importo** **DataIncasso** **CausaleIncasso**

element **Incasso/NumeroIncasso**

diagram	
type	**xs:int**

element **Incasso/Importo**

diagram	
type	**xs:decimal**

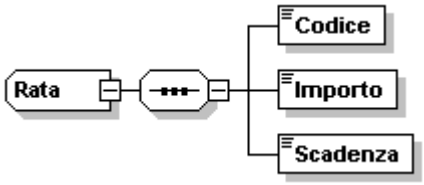
element **Incasso/DataIncasso**

diagram	
type	**xs:date**

element **Incasso/CausaleIncasso**

diagram	
type	**dd:CausaleIncasso**
facets	enumeration TRIBUTI_TAR SU enumeration TRIBUTI_ICI enumeration TRIBUTI_TOS AP enumeration TRIBUTI_COS AP enumeration TRIBUTI_ACQ UEDOTTO

complexType **Rata**

diagram	
namespace	types.firb.deltadator.com
children	**Codice Importo Scadenza**

element **Rata/Codice**

diagram	
type	**xs:string**

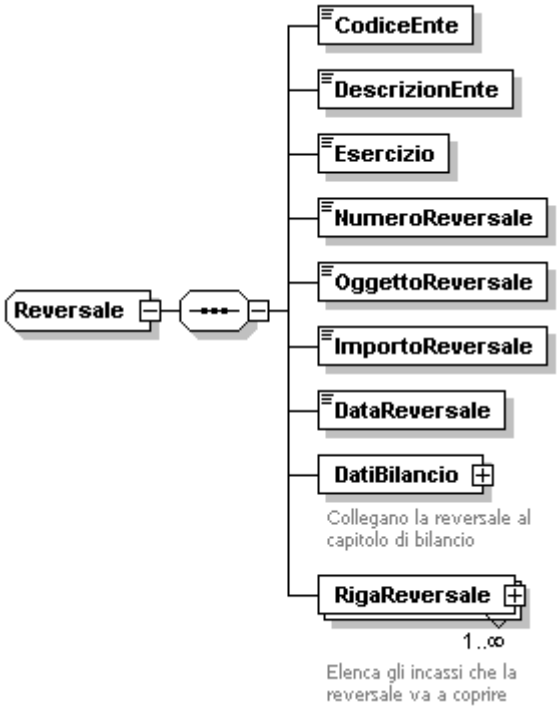
element **Rata/Importo**

diagram	
type	**xs:decimal**

element **Rata/Scadenza**

diagram	
type	**xs:date**

complexType **Reversale**

diagram	
namespa	types.firb.deltadator.com

ce	
children	**CodiceEnte DescrizionEnte Esercizio NumeroReversale OggettoReversale ImportoReversale DataReversale DatiBilancio RigaReversale**

element **Reversale/CodiceEnte**

diagram	
type	**xs:string**

element **Reversale/DescrizionEnte**

diagram	
type	**xs:string**

element **Reversale/Esercizio**

diagram	
type	**xs:integer**

element **Reversale/NumeroReversale**

diagram	
type	**xs:integer**

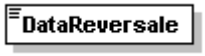
element **Reversale/OggettoReversale**

diagram	
type	**xs:string**

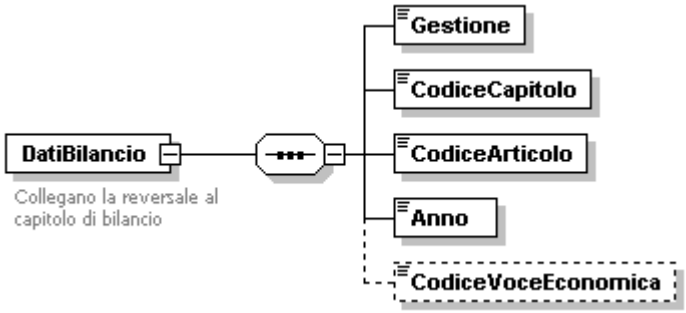
element **Reversale/ImportoReversale**

diagram	
type	**xs:decimal**

element **Reversale/DataReversale**

diagram	
type	**xs:date**

element **Reversale/DatiBilancio**

diagram	
children	**Gestione CodiceCapitolo CodiceArticolo Anno CodiceVoceEconomica**

element **Reversale/DatiBilancio/Gestione**

diagram	
type	**dd:GestioneDatiBilancio**
facets	enumeration C enumeration R

element **Reversale/DatiBilancio/CodiceCapitolo**

diagram	

type	**xs:string**

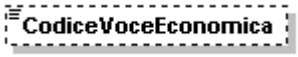
element **Reversale/DatiBilancio/CodiceArticolo**

diagram	
type	**xs:string**

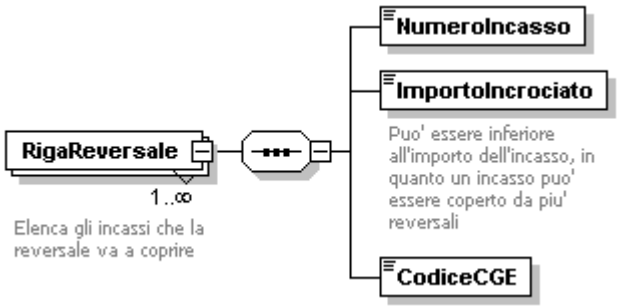
element **Reversale/DatiBilancio/Anno**

diagram	
type	**xs:integer**

element **Reversale/DatiBilancio/CodiceVoceEconomica**

diagram	
type	**xs:string**

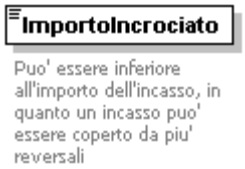
element **Reversale/RigaReversale**

diagram	
children	**NumeroIncasso** **ImportoIncrociato** **CodiceCGE**

element **Reversale/RigaReversale/NumeroIncasso**

diagram	
type	**xs:integer**

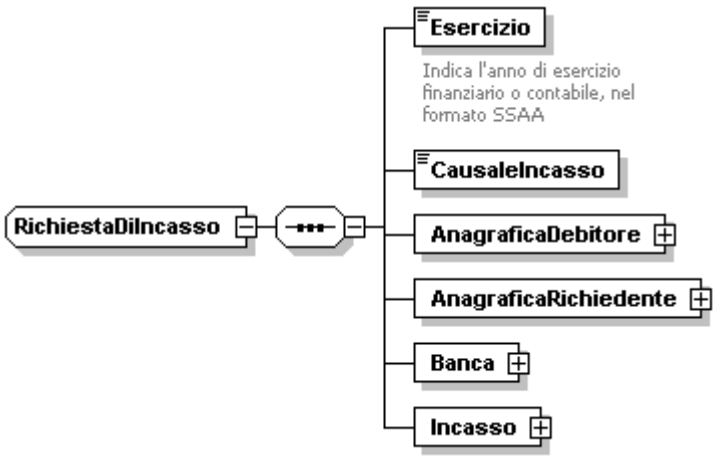
element **Reversale/RigaReversale/ImportoIncrociato**

diagram	
type	**xs:decimal**

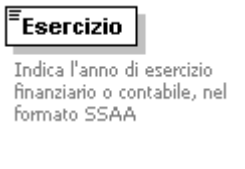
element **Reversale/RigaReversale/CodiceCGE**

diagram	
type	**xs:string**

complexType **RichiestaDiIncasso**

diagram	
namespace	types.firb.deltadator.com
children	**Esercizio** **CausaleIncasso** **AnagraficaDebitore** **AnagraficaRichiedente** **Banca** **Incasso**

element **RichiestaDiIncasso/Esercizio**

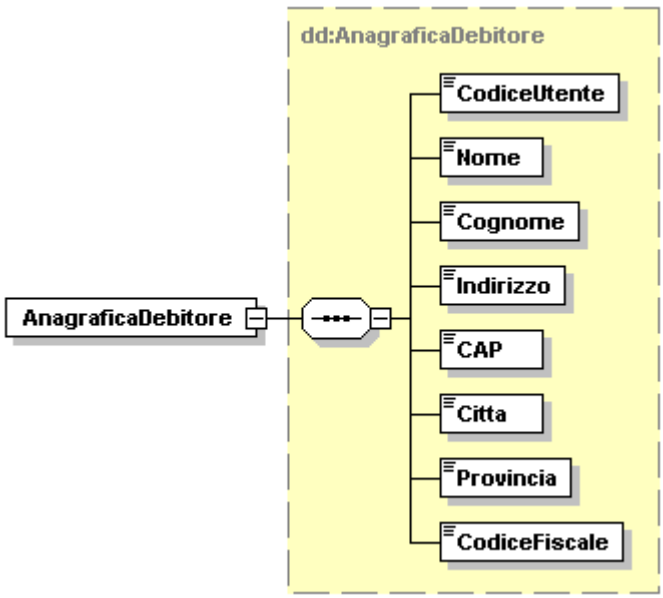
diagram	

type	**xs:integer**

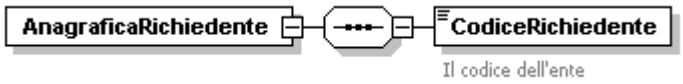
element **RichiestaDiIncasso/CausaleIncasso**

diagram	
type	**dd:CausaleIncasso**
facets	enumeration TRIBUTI_TAR SU enumeration TRIBUTI_ICI enumeration TRIBUTI_TOS AP enumeration TRIBUTI_COS AP enumeration TRIBUTI_ACQ UEDOTTO

element **RichiestaDiIncasso/AnagraficaDebitore**

diagram	
type	**dd:AnagraficaDebitore**
children	**CodiceUtente Nome Cognome Indirizzo CAP Citta Provincia CodiceFiscale**

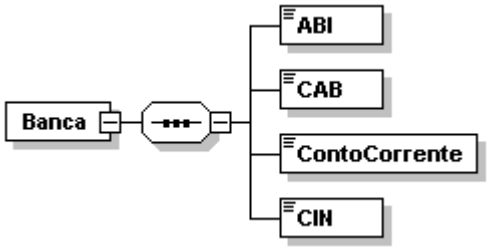
element **RichiestaDiIncasso/AnagraficaRichiedente**

diagram	
children	**CodiceRichiedente**

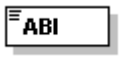
element **RichiestaDiIncasso/AnagraficaRichiedente/CodiceRichiedente**

diagram	
type	**xs:string**

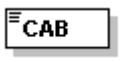
element **RichiestaDiIncasso/Banca**

diagram	
children	**ABI** **CAB** **ContoCorrente** **CIN**

element **RichiestaDiIncasso/Banca/ABI**

diagram	
type	**xs:string**

element **RichiestaDiIncasso/Banca/CAB**

diagram	
type	**xs:string**

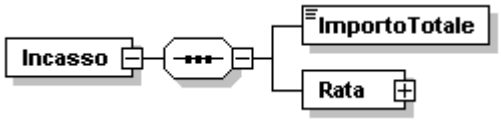
element **RichiestaDiIncasso/Banca/ContoCorrente**

diagram	
type	**xs:string**

element **RichiestaDiIncasso/Banca/CIN**

diagram	
type	**xs:string**

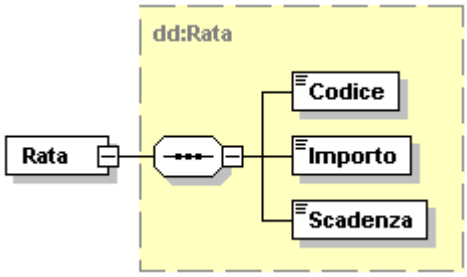
element **RichiestaDiIncasso/Incasso**

diagram	
children	**ImportoTotale** **Rata**

element **RichiestaDiIncasso/Incasso/ImportoTotale**

diagram	
type	**xs:decimal**

element **RichiestaDiIncasso/Incasso/Rata**

diagram	
type	**dd:Rata**
children	**Codice** **Importo** **Scadenza**

simpleType **CausaleIncasso**

namespace	types.firb.deltadator.com
type	restriction of **xs:string**
facets	enumeration TRIBUTI_TARSU enumeration TRIBUTI_ICI enumeration TRIBUTI_TOSAP enumeration TRIBUTI_COSAP enumeration TRIBUTI_ACQUEDOTTO

simpleType **GestioneDatiBilancio**

namespace	types.firb.deltadator.com
type	restriction of **xs:string**
facets	enumeration C enumeration R

simpleType **Messaggio419**

namespace	types.firb.deltadator.com
type	restriction of **xs:string**
facets	enumeration TRANSAZIONE_ESEGUITA enumeration PROBLEMA enumeration ERRORE

simpleType **StatoIncasso**

namespace	types.firb.deltadator.com
type	restriction of **xs:string**
facets	enumeration NonEsitato enumeration Pagato enumeration Insoluto enumeration Errato

simpleType **StatoRegistrazione**

namespace	types.firb.deltadator.com
type	restriction of **xs:string**
facets	enumeration InRegistrazione enumeration Registrato enumeration Fallito