

Cronologia delle revisioni

La stesura originale di questo documento risale al 30 Aprile 2006

Il 27 Agosto 2008, a seguito di segnalazioni di errore o incongruenza sottoposte dalla ditta M.A.I.O.R. Srl di Lucca il documento è stato sottoposto a revisione; le parti modificate sono **evidenziate in verde**

Il 7 Settembre 2008, a seguito di altre segnalazioni di incongruenza sottoposte dalla ditta IVU il documento è stato sottoposto ad ulteriore revisione; le parti modificate sono **evidenziate in giallo**

Il 15 Settembre 2009, a seguito di segnalazioni di errore o incongruenza sottoposte dalla ditta M.A.I.O.R. Srl di Lucca il documento è stato sottoposto a revisione; le parti modificate sono **evidenziate in azzurro**

Osservatorio Trasporti Regione Toscana

Specifiche dei formati di Comunicazione Orari TPL

DbcXml

Indice

- **1 – Generalità**
 - 1.a – Scopo del documento
 - 1.b – Vantaggi dell’adozione di un formato XML
 - 1.c – Relazione tra i precedenti formati DbcTxt-DbcGis e DbcXml
 - 1.d – Formati DbcXml Level 1 e Level 2
 - 1.e – Risorse SW correlate
 - 1.f – XML in pillole
- **2 - formato DbcXml Level 1**
 - 2.a – Struttura generale del documento XML
 - 2.b – il tag <DbcXml>
 - 2.c – i tags <Fermate> e <Fmt>
 - 2.d – i tags <Percorsi> e <Itn>
 - 2.e – i tags <Geom> e <Pt>
 - 2.f – i tags <CorseStandard>, <CorsaStd> e <StdFmt>
 - 2.g – i tags <Cadenze> e <Cad>
 - 2.h – i tags <Calendario> e <Kal>
 - 2.i – i tags <PianoCorse>, <Lotto>, <Ente> e <Linea>
 - 2.j – i tags <AzLinea>, <Gestore>, <Subappalto> e <Corsa>
 - 2.k – i tags <Corsa> e <Periodo>
- **3 - formato DbcXml Level 2**
 - 3.a – Struttura generale del documento XML
 - 3.b – il tag <DbcXml>
 - 3.c – i tags <TmpNodi>, <TmpNod> e <RtArc>
 - 3.d – i tags <TmpArchi>, <TmpArc> e <Pt>
 - 3.e – i tags <KillArchi>, <KillArc> e <Pt>
 - 3.f – i tags <Fermate> e <Fmt>
 - 3.g – i tags <Rivendite> e <Riv>
 - 3.h – i tags <Tratte> e <Trt>
 - 3.i – i tags <SeqArchi> e <TrtArc>
 - 3.j – i tags <Percorsi> e <Itn>
 - 3.k – i tags <SeqTratte> e <ItnTrt>
 - 3.l – i tags <CorseStandard>, <CorsaStd> e <StdFmt>
 - 3.m – i tags <Cadenze> e <Cad>
 - 3.n – i tags <Calendario> e <Kal>
 - 3.o – i tags <PianoCorse>, <Lotto>, <Ente> e <Linea>
 - 3.p – i tags <AzLinea>, <Gestore>, <Subappalto> e <Corsa>
 - 3.q – i tags <Corsa> e <Periodo>

Appendice A – codifiche e formati DbcXml

Appendice B – corrispondenze elementi tra DbcTxt/DbcGis e DbcXml

Appendice C – trasmissione telematica delle comunicazioni DbcXml - logica interpretativa del Server OTRT - criteri di aggiornamento del DB centralizzato dell’OTRT

1 - Generalità

1.a - Scopo del documento:

Il presente documento descrive la struttura dei formati *DbcXml* adottati dall'Osservatorio Trasporti della Regione Toscana (*OTRT*) per la trasmissione neutrale dei dati che descrivono la struttura delle Reti TPL e dei relativi Orari di Servizio.

Si presuppone che il *lettore tipo* del documento sia un esperto di gestione del Trasporto Pubblico non necessariamente dotato di specifiche competenze informatiche, per cui vengono privilegiati gli aspetti descrittivi e funzionali in termini colloquiali.

1.b - Vantaggi dell'adozione di un formato XML:

XML (*eXtensible Markup Language*) è uno standard definito dal **W3C** (*World Wide Web Consortium*) per consentire agevolmente l'interscambio universale dei dati specie in contesti Web. XML presenta caratteristiche di estrema potenza e flessibilità ma è anche intrinsecamente semplice e non impone nessun vincolo particolare di ambiente e/o di piattaforma. Di fatto XML rappresenta uno standard universale ormai largamente diffuso in tutti i contesti.

La caratteristica principale di XML è quella di consentire la rappresentazione di qualsiasi categoria di dati tramite un elementare formato testuale e contemporaneamente di definire in modo rigoroso una struttura sintattica che ne renda univoca l'interpretazione.

Quindi, riassumendo, XML è stato adottato dall'OTRT essenzialmente per i seguenti motivi:

- E' pienamente conforme agli standards universali
- Semplifica i processi telematici di interscambio dati
- XML è *human-legible*, cioè può essere immediatamente letto da qualsiasi operatore utilizzando un banale text-editor. Utilizzando un *tool* più specifico (p.es. un browser Web, tipo Internet Explorer o FireFox, è direttamente possibile interpretare anche la struttura logica del documento XML)
- Generare un flusso XML è possibile (e semplice) utilizzando qualsiasi piattaforma e qualunque linguaggio di programmazione
- XML consente di definire strutture estremamente complesse all'interno di un unico documento. Quindi nello specifico del TPL consente di descrivere completamente una rete TPL (cartografia delle fermate e dei percorsi) e l'offerta di servizio ad essa relativa (orari corsa, etc) all'interno di un unico flusso.

1.c - Relazioni tra i precedenti formati DbcTxt-DbcGis e DbcXml:

La semantica dei dati (contenuto informativo) è esattamente identica tra il nuovo formato ed i precedenti; non cambia assolutamente nulla.

Semplicemente il nuovo formato DbcXml utilizza una diversa notazione formale per rappresentare i medesimi dati già richiesti per i formati precedenti.

Quindi è concettualmente possibile (e molto semplice) trasporre i dati dal formato DbcTxt al formato DbcXml [o viceversa].

Considerato che non sussistono variazioni di sorta rispetto ai contenuti informativi richiesti, l'operazione di generazione di un flusso DbcXml si configura quindi nei termini di una traduzione assolutamente automatica tra formati perfettamente equivalenti.

1.d - Formati DbcXml Level 1 e Level 2:

Attualmente l'OTRT richiede una descrizione delle reti TPL integralmente basata sul Grafo Stradale ufficiale RT [specifiche OTRT 2004-2005].

Tuttavia è prevista nell'arco del biennio 2006-2007 la gestione di un regime transitorio temporaneo che consente in via provvisoria la descrizione delle reti TPL anche in modalità non conforme rispetto al Grafo RT, così come definite dalle precedenti specifiche [OTRT 2001-2002].

Conseguentemente risultano definite due diverse versioni del formato DbcXml:

- DbcXml **Level 1** da utilizzarsi per l'inoltro delle comunicazioni secondo il vecchio standard [conformità approssimativa con il Grafo RT]
- DbcXml **Level 2** da utilizzarsi per l'inoltro delle comunicazioni secondo il nuovo standard [conformità rigorosa con il Grafo RT]

1.e - Risorse SW correlate:

Per quanto riguarda le applicazioni SW che l'OTRT mette a disposizione delle Aziende TPL al fine di agevolare la produzione di comunicazioni conformi agli standards, la situazione è la seguente:

- Le applicazioni **GaiaDbc + DbcPlus** già precedentemente offrivano un pieno supporto alla gestione delle comunicazioni DbcTxt + DbcGis secondo gli standards OTRT 2001-2002
- L'applicazione **GaiaTpl** già precedentemente offriva un pieno supporto alla gestione delle comunicazioni DbcTxt2 + DbcGis2 secondo lo standard OTRT 2004-2005
- Quindi **DbcPlus** ora supporta la produzione di comunicazioni **DbcXml Level 1**
- Mentre **GaiaTpl** ora supporta la produzione di comunicazioni **DbcXml Level 2**
- Quindi allo stato attuale risulta possibile utilizzare le applicazioni SW OTRT anche nel ruolo di filtro per la conversione automatica dai precedenti formati DbcTxt+DbcGis ai nuovi formati DbcXml

1.f - XML in pillole:

- Un documento XML deve essere strutturato utilizzando dei **tags** (marcatori) che si identificano in quanto sono racchiusi tra una coppia di parentesi angolari aperte e chiuse [**<tag_name>**].
- Di norma per ciascun **tag iniziale** deve essere esplicitamente definito un **tag finale** corrispondente; il **tag finale** si qualifica in quanto presenta il medesimo nome del **tag iniziale** ma presenta un prefisso / [**</tag_name>**]; tutto ciò che è contenuto tra un **tag iniziale** ed il corrispondente **tag finale** appartiene ad una medesima entità.
- Per gli elementi semplici è ammesso evitare la dichiarazione esplicita del **tag finale**; in questo caso però il marcatore / deve precedere la parentesi angolare di chiusura del **tag** [**<tag_name />**]
- A ciascun **tag** può eventualmente essere associata una lista di attributi qualificati; a ciascun attributo corrisponde un valore [**<tag_name attr1="value1" attr2="value2" />**]
- Un **tag** può racchiudere al suo interno ulteriori **tags**, e così via, in modo tale da consentire l'articolazione di gerarchie anche molto complesse di entità strutturate [**tags nidificati**].
- Ciascun documento XML deve identificarsi esplicitamente in quanto la prima riga del documento deve indicare: **<?xml version="1.0" encoding="ISO-8859-1" ?>** [giusto per i più curiosi: **ISO-8859-1** indica che il documento XML utilizza il normale alfabeto Latino Occidentale]
- Dato che i caratteri maggiore >, minore <, apice ' , virgolette ", ed e-commerciale & hanno un significato speciale per XML, non possono essere utilizzati all'interno dei valori associati ad un attributo; devono quindi essere necessariamente essere sostituiti utilizzando le corrispondenti notazioni speciali: **>**; **<**; **'**; **"e;**; **&**;

2 - formato DbcXml Level 1

2.a - Struttura generale del documento XML:

Una comunicazione DbcXml Level 1 descrive al suo interno una rete TPL [comprese le rappresentazioni cartografiche delle fermate e dei percorsi] e l'offerta di trasporto ad essa associata [in termini di orari corsa e calendari di servizio] relativamente ad un periodo di validità temporale specificato.

Di regola una comunicazione DbcXml deve descrivere complessivamente un'intera rete TPL integrata a livello di Lotto di Gara, e quindi deve essere prodotta dal soggetto gestore del Lotto [Scarl].

La struttura generale di un documento DbcXml Level 1 è la seguente:

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<!DOCTYPE DbcXml SYSTEM "http://www.osservatorio-trasporti.it/DbcXmlLevel1.dtd">
<DbcXml level="1.0" ..>
  <Fermate>
    ..
  </Fermate>
  <Percorsi>
    ..
  </Percorsi>
  <Cadenze>
    ..
  </Cadenze>
  <Calendario>
    ..
  </Calendario>
  <PianoCorse>
    ..
  </PianoCorse>
</DbcXml>
```

- La dichiarazione `<!DOCTYPE ...>` richiama la URL in cui si trova la **DTD** [*Document Type Definition*] che contiene le specifiche formali del documento XML
 - nel caso si tratti di una comunicazione in formato **Level 1** occorrerà fare riferimento alla URL: <http://www.osservatorio-trasporti.it/DbcXmlLevel1.dtd>
 - invece nel caso si tratti di una comunicazione in formato **Level 2** occorrerà fare riferimento alla URL: <http://www.osservatorio-trasporti.it/DbcXmlLevel2.dtd>
- L'intera comunicazione è incapsulata all'interno del tag `<DbcXml ...>...</DbcXml>`; **level="1.0"** indica appunto che si tratta di una comunicazione in formato **Level 1**
- All'interno della sezione corrispondente al tag `<Fermate>...</Fermate>` si trova la lista delle fermate presenti nella rete TPL [comprese le relative rappresentazioni cartografiche]
- All'interno della sezione corrispondente al tag `<Percorsi>...</Percorsi>` si trova la lista dei percorsi presenti nella rete TPL [comprese le relative rappresentazioni cartografiche]
- All'interno della sezione corrispondente al tag `<Cadenze>...</Cadenze>` si trova la lista delle cadenze-corsa utilizzate per la definizione dei calendari di servizio
- All'interno della sezione corrispondente al tag `<Calendario>...</Calendario>` si trova la lista delle cadenze-giorno che definiscono il calendario di servizio
- All'interno della sezione corrispondente al tag `<PianoCorse>...</PianoCorse>` si trova la definizione dell'offerta di servizio [in termini di orari corsa]
- **N.B.** l'ordine relativo delle sezioni sopra elencate deve essere tassativamente quello indicato, in modo tale da semplificare la verifica dei vincoli relazionali esistenti tra i diversi elementi.

2.b - Il tag <DbcXml ..>

Questo *tag* specifica i parametri generali di presentazione della comunicazione; la struttura completa è la seguente:

```
<DbcXml level="1.0" azienda=".." name=".."
      data=".." nro=".." inizio=".." fine=".."
      agente=".." tipo="..">..</DbcXml>
```

- **level** deve essere sempre impostato come **1.0** nel caso delle comunicazioni **Level 1**
- **azienda** deve contenere il codice numerico OTRT corrispondente all'Azienda che presenta la comunicazione. Di norma la comunicazione deve essere presentata da un'Azienda Scarl assegnataria di un Lotto di Gara.
- **name** deve contenere la denominazione dell'azienda. *Questo valore serve esclusivamente per migliorare la human-legibility del codice XML, quindi verrà ignorato durante le elaborazioni. Farà comunque fede il codice indicato da azienda*
- **data** deve indicare la data di presentazione della comunicazione. Le date devono sempre essere espresse come **GG/MM/AAAA**
- **nro** deve indicare il protocollo progressivo di invio delle comunicazioni. *Questo valore serve esclusivamente per migliorare la human-legibility del codice XML, quindi verrà ignorato durante le elaborazioni. Farà comunque fede l'ordine temporale di ricezione delle comunicazioni da parte del Server OTRT*
- **inizio** deve indicare la data di inizio del periodo di validità del piano di servizio. Per buona norma le comunicazioni devono essere preventive, quindi deve valere la relazione **inizio** >= **data** Le date devono sempre essere espresse come **GG/MM/AAAA**
- **fine** deve indicare la data di fine del periodo di validità del piano di servizio; ovviamente deve sempre valere la relazione **inizio** <= **fine**. Nel caso in cui il Server OTRT riceva una comunicazione successiva relativa ad un periodo temporale già precedentemente comunicato, si intende che l'ultima comunicazione sostituisce *in toto* quanto comunicato in precedenza. Le date devono sempre essere espresse come **GG/MM/AAAA**
- **agente** deve indicare il nominativo del funzionario Aziendale che ha curato la trasmissione della comunicazione. *Questo valore serve esclusivamente per migliorare la human-legibility del codice XML, quindi verrà ignorato durante le elaborazioni.*
- **tipo** specifica il ruolo della comunicazione. Può assumere uno dei seguenti valori letterali:
 - **BUDGET** – indica una comunicazione del Budget Contrattuale. Questa deve essere inviata solo ed esclusivamente in occasione della stipula iniziale di un Contratto di Servizio. *Quindi per norma verrà inviata un'unica comunicazione del Budget*
 - **PLANNING** – indica una comunicazione del Piano di Servizio. Questa deve essere sollecitamente inoltrata in ciascuna occasione in cui si applichi una qualsiasi variazione, anche minimale, dell'offerta di servizio tale da alterare gli orari corsa e/o i percorsi di linea. *Quindi di regola è atteso l'invio di frequenti comunicazioni del Piano di Servizio aggiornato. La ricezione di una comunicazione del Piano di Servizio causerà automaticamente l'aggiornamento del DB OTRT.*
 - **TEST** – indica una comunicazione priva di qualsiasi valore effettivo. Una comunicazione di Test verrà valutata ed analizzata dal Server OTRT, ma la sua ricezione non avrà alcuna conseguenza di sorta in termini di aggiornamento del DB OTRT. *Quindi le comunicazioni di Test servono solo ed esclusivamente come strumento diagnostico e di verifica del SW*
 - **PROJECT** – indica una comunicazione di tipo Ipotesi / Progetto. Una comunicazione di questo tipo verrà memorizzata permanentemente dal Server OTRT, ma non avrà conseguenze per quanto riguarda l'aggiornamento del Piano di Servizio. *Quindi una*

comunicazione Ipotesi / Progetto serve solo per formalizzare p.es. un'ipotesi di ristrutturazione del servizio a fini di studio e valutazione.

- **UNKNOWN** – se **tipo** non indica uno dei valori precedenti, la comunicazione verrà ritenuta completamente inaccettabile e sarà conseguentemente respinta, anche se formalmente ineccepibile.

2.c - I tags <Fermate> e <Fmt ..>

La sezione identificata dai tags <Fermate>..</Fermate> deve contenere la lista delle fermate che contraddistinguono la rete TPL.

Ciascuna singola fermata è descritta all'interno di un tag <Fmt .. /> che ne specifica gli attributi e la posizione cartografica georeferenziata; quindi la struttura della sezione <Fermate> è:

```
<Fermate>
  <Fmt .. />
  <Fmt .. />
  <Fmt .. />
  ..
</Fermate>
```

=====

La struttura completa del tag <Fmt .. /> è la seguente:

```
<Fmt code=".." name=".." ubic=".."
      x=".." y=".." />
```

- **code** deve esprimere il codice Aziendale univoco assegnato alla fermata
- **name** deve indicare la denominazione associata alla fermata
- **ubic** deve indicare l'ubicazione [Via, nro civico / Strada + km] della fermata. *Questo valore è opzionale e può essere omissso valorizzando p.es. a spazi*
- **x** deve indicare la coordinata cartografica corrispondente alla fermata [secondo Gauss-Boaga]
- **y** deve indicare la coordinata cartografica corrispondente alla fermata [secondo Gauss-Boaga]
N.B. le coordinate devono essere espresse in metri con almeno un decimale; come separatore della parte frazionale deve essere utilizzato il carattere punto.

2.d - I tags <Percorsi> e <Itn ..>

La sezione identificata dai tags <Percorsi>..</Percorsi> deve contenere la lista dei percorsi utilizzati dalla rete TPL.

Ciascun singolo percorso è descritto all'interno di un tag <Itn .. />..</Itn> che ne specifica gli attributi e la rappresentazione cartografica georeferenziata; quindi la struttura della sezione <Percorsi> è:

```
<Percorsi>
  <Itn ..>..</Itn>
  <Itn ..>..</Itn>
  <Itn ..>..</Itn>
  ..
</Percorsi>
```

A sua volta ciascun singolo tag <Itn ..></Itn> è strutturato come segue:

```
<Itn ..>
  <Geom ..>..</Geom>
  <CorseStandard>..</CorseStandard>
</Itn>
```

- l'elemento <Geom ..></Geom> specifica la descrizione cartografica georeferenziata relativa al percorso
- l'elemento <CorseStandard></CorseStandard> invece specifica le matrici degli orari di transito delle corse standard associate al percorso
- per ciascun percorso deve essere specificata un'unica geometria univoca che congiunga il capolinea di partenza con il capolinea di arrivo.
- analogamente per ciascun percorso può essere specificato un'unica lista di corse standard

=====

La struttura completa del tag <Itn .. /> è la seguente:

```
<Itn code=".." name=".." metri="..">
```

- **code** deve esprimere il codice Aziendale univoco assegnato al percorso
- **name** deve indicare la descrizione sommaria del percorso. *Questo valore è opzionale e può essere omissso valorizzando p.es. a spazi*
- **metri** deve indicare la lunghezza complessiva del percorso **N.B.** la lunghezza deve essere espressa in metri con almeno un decimale; come separatore della parte frazionale deve essere utilizzato il carattere punto.

2.e - I tags <Geom ..> e <Pt ..>

L'elemento identificato dai tags <Geom ..>.</Geom> descrive una geometria di tipo lineare [polilinea], e quindi deve contenere la lista ordinata dei singoli vertici georeferenziati che descrivono la geometria in oggetto. Quindi la struttura dell'elemento <Geom ..> è:

```
<Geom ..>
  <Pt .. />
  <Pt .. />
  <Pt .. />
  ..
</Geom>
```

=====

La struttura completa del tag <Geom ..> è la seguente:

```
<Geom pts="..">
```

- **pts** deve indicare il numero dei vertici che compongono la polilinea [i.e. deve coincidere con il numero di elementi <Pt ..> subordinati]

=====

La struttura completa del tag <Pt ..> è la seguente:

```
<Pt x=".." Y=".." />
```

- **x** deve indicare la coordinata cartografica corrispondente alla fermata [secondo Gauss-Boaga]
- **y** deve indicare la coordinata cartografica corrispondente alla fermata [secondo Gauss-Boaga] **N.B.** le coordinate devono essere espresse in metri con almeno un decimale; come separatore della parte frazionale deve essere utilizzato il carattere punto.
- **caveat**: dato che XML è un formato testuale, e quindi tende ad occupare molto spazio, per la codifica delle entità geometriche (già di per se onerosa) occorre applicare un opportuno **criterio di compressione spaziale** che opera come segue:
 - al primo elemento <Pt ..> associato alla geometria vengono sempre assegnate le coordinate cartografiche reali [secondo Gauss-Boaga]
 - a tutti gli elementi successivi al primo invece deve essere assegnato il valore della differenza rispetto al punto immediatamente precedente misurato sull'asse corrispondente. *Ovviamente la differenza deve essere gestita come un valore signed*

2.f - I tags <CorseStandard>, <CorsaStd ..> e <StdFmt ..>

L'elemento identificato dai *tags* <CorseStandard>..</CorseStandard> deve contenere la lista delle corse standard associate ad un singolo percorso TPL.

Per corsa standard si intende una corsa astratta caratterizzata da una sequenza di fermate e da una matrice oraria di marcia; tutte le corse concrete associate ad una corsa standard ereditano da quest'ultima l'orario di marcia reale opportunamente ricalcolato sulla base dell'orario di partenza assegnato alla corsa reale.

Ciascuna singola corsa standard quindi è descritta in termini di una successione ordinata di elementi <StdFmt ..> tali da rappresentare la successione delle fermate ed i relativi intertempi di transito; quindi la struttura complessiva dell'elemento <CorseStandard> è:

```

<CorseStandard>
  <CorsaStd ..>
    <StdFmt .. />
    <StdFmt .. />
    <StdFmt .. />
    ..
  </CorsaStd>
  <CorsaStd ..>
    <StdFmt .. />
    <StdFmt .. />
    <StdFmt .. />
    ..
  </CorsaStd>
  ..
</CorseStandard>

```

=====

La struttura completa del *tag* <CorsaStd ..> è la seguente:

```
<CorsaStd id=".." tempo="..">
```

- **id** deve contenere un opportuno valore numerico univoco che consenta successivamente di associare le corse reali alla corsa standard di riferimento
- **tempo** deve indicare la durata complessiva della corsa standard espressa in minuti. In altri termini deve misurare il numero di minuti intercorrenti tra la partenza dal capolinea iniziale e l'arrivo al capolinea finale.

=====

La struttura completa del tag `<StdFmt ..>` è la seguente:

```
<StdFmt sub=".." metri=".." code=".." arriva=".."
  parte=".." primaria="." facolt="."
  nonferma="." />
```

- **sub** deve contenere un opportuno valore numerico progressivo che consenta di ordinare la lista delle fermate associate a `<CorsaStd ..>` *E' fortemente gradito che gli elementi <StdFmt ..> vengano inseriti rispettando l'ordine naturale di percorrenza.*
- **metri** deve indicare la distanza progressiva misurata a partire dal capolinea di partenza. *Quindi al capolinea di partenza deve obbligatoriamente corrispondere una progressiva pari a 0 (zero).* Non devono essere indicate le cifre decimali.
- **code** deve esprimere il codice Aziendale che identifica la fermata corrispondente. **N.B.** se la fermata corrispondente non è stata dichiarata in un elemento `<Fmt ..>` all'interno della sezione `<Fermate>`, e quindi il riferimento relazionale non può essere soddisfatto, si verifica una condizione di **errore fatale**; conseguentemente l'intera comunicazione verrà respinta in quanto inutilizzabile.
- **arriva** indica l'intertempo di arrivo alla fermata espresso in minuti e misurato a partire dal capolinea iniziale. *Quindi i tempi associati al capolinea di partenza saranno sempre impostati a 0 (zero)*
- **parte** indica l'intertempo di ripartenza dalla fermata espresso in minuti e misurato a partire dal capolinea iniziale. *Quindi i tempi assegnati al capolinea di arrivo dovranno necessariamente coincidere con la durata complessiva della corsa*
- **primaria** indica se la fermata corrispondente ha un rilievo particolare. Può assumere i valori S o N
- **facolt** indica se la fermata corrispondente è facoltativa. Può assumere i valori S o N
- **nonferma** indica se si tratta di una fermata vera e propria [con servizio di salita-discesa dei passeggeri] oppure di un semplice punto di riferimento orario. Può assumere i valori S o N

2.g - I tags <Cadenze> e <Cad ..>

La sezione identificata dai tags <Cadenze>..</Cadenze> deve contenere la lista delle cadenza-corsa utilizzate per codificare i calendari di servizio delle corse.

Ciascuna singola cadenza è descritta all'interno di un tag <Cad .. /> che ne specifica gli attributi; quindi la struttura della sezione <Cadenze> è:

```
<Cadenze>
  <Cad .. />
  <Cad .. />
  <Cad .. />
  ..
</Cadenze>
```

=====

La struttura completa del tag <Cad .. /> è la seguente:

```
<Cad code=".." name=".." />
```

- **code** deve esprimere il codice Aziendale univoco assegnato alla cadenza
- **name** deve indicare la denominazione associata alla cadenza

2.h - I tags <Calendario> e <Kal ..>

La sezione identificata dai tags <Calendario>..</Calendario> deve contenere la lista delle cadenza-giorno utilizzate per codificare i calendari di servizio delle corse.

Ciascuna singola cadenza-giorno è descritta all'interno di un tag <Kal .. /> che ne specifica quali cadenze sono attive in un giorno dato; quindi la struttura della sezione <Calendario> è:

```
<Calendario>
  <Kal .. />
  <Kal .. />
  <Kal .. />
  ..
</Calendario>
```

=====

La struttura completa del tag <Kal .. /> è la seguente:

```
<Kal code=".." data=".." note=".." />
```

- **code** deve esprimere il codice Aziendale univoco assegnato alla cadenza. **N.B.** se la cadenza corrispondente non è stata dichiarata in un elemento <Cad ..> all'interno della sezione <Cadenze>, e quindi il riferimento relazionale non può essere soddisfatto, si verifica una condizione di **errore fatale**; conseguentemente l'intera comunicazione verrà respinta in quanto inutilizzabile.
- **data** deve indicare la data del giorno in cui la cadenza è attiva. Le date devono sempre essere espresse come **GG/MM/AAAA**
- **note** può caratterizzare opzionalmente le date particolari (Santo Patrono etc) *Questo valore serve esclusivamente per migliorare la human-legibility del codice XML, quindi verrà ignorato durante le elaborazioni.*

2.i - I tags <PianoCorse>, <Lotto ..>, <Ente ..> e <Linea ..>

La sezione identificata dai tags <PianoCorse>..</PianoCorse> deve contenere la lista delle corse che definiscono il Piano di Esercizio e quindi l'offerta di servizi TPL.

Il Piano di Servizio è strutturato in modo tale da riflettere strettamente l'articolazione dei Contratti di Servizio; quindi la struttura complessiva della sezione <PianoCorse> è:

```

<PianoCorse>
  <Lotto ..>
    <Ente .. >
      <Linea ..>
        ..
      </Linea>
      <Linea ..>
        ..
      </Linea>
    ..
  </Ente>
  <Ente .. >
    <Linea ..>
      ..
    </Linea>
    <Linea ..>
      ..
    </Linea>
  ..
</Ente>
..
</Lotto>
</PianoCorse>

```

=====

La struttura completa del tag <Lotto ..> è la seguente:

```
<Lotto code=".." name="..">
```

- **code** deve contenere il codice numerico OTRT corrispondente al Lotto di Gara cui si riferisce la comunicazione. Di norma una comunicazione deve essere presentata da un'Azienda Scarl assegnataria di un Lotto di Gara, e deve quindi riferirsi ad un unico Lotto.
- **name** deve contenere la denominazione del Lotto di Gara. *Questo valore serve esclusivamente per migliorare la human-legibility del codice XML, quindi verrà ignorato durante le elaborazioni. Farà comunque fede il codice indicato da code*

=====

La struttura completa del tag <Ente ..> è la seguente:

```
<Ente code=".." name="..">
```

- **code** deve contenere il codice numerico OTRT corrispondente all'Ente territoriale (Provincia, Comune etc) cui appartengono le linee subordinate. **N.B.** l'Ente territoriale proprietario di una linea non deve essere confuso con l'Ente intestatario del Contratto di Servizio. Può infatti legittimamente accadere che questi non coincidano affatto; il corretto criterio da seguire è il seguente:
 - se i percorsi afferenti ad una determinata linea si svolgono completamente, oppure per la parte prevalente, all'interno di un solo Comune, allora la linea è di proprietà di quel Comune
 - nel caso in cui connettano almeno due diversi Capoluoghi di Comune, allora la linea è di proprietà della Provincia
 - se infine connettono almeno due diversi Capoluoghi di Provincia, allora la linea è di proprietà della Regione
- **name** deve contenere la denominazione dell'Ente. *Questo valore serve esclusivamente per migliorare la human-legibility del codice XML, quindi verrà ignorato durante le elaborazioni. Farà comunque fede il codice indicato da code*

=====

La struttura completa del tag <Linea ..> è la seguente:

```
<Linea sigla=".." name=".." denom="..">
```

- **sigla** deve contenere la sigla standard della linea. *Quindi deve corrispondere esattamente al valore riportato sugli orari di palina, sui libretti orari, nei Contratti di Servizio etc*
- **name** deve contenere la denominazione convenzionale della linea. *Questo valore dovrebbe auspicabilmente essere concordato consensualmente tra Aziende ed Enti in modo tale da implementare un sistema rigorosamente coerente e consistente*
- **denom** deve contenere una sommaria descrizione del percorso prevalente della linea [lista delle principali località o poli di interesse serviti]

2.j - I tags <AzLinea ..>, <Gestore ..>, <Subappalto ..> e <Corsa ..>

L'elemento identificato dai tags <Linea ..>..</Linea> deve contenere la lista delle corse che definiscono il Piano di Esercizio relativo ad una specifica linea; quindi la struttura complessiva dell'elemento <Linea ..> è:

```

<Linea ..>
  <AzLinea ..>
    <Gestore .. >
      <Subappalto ..>
        <Corsa ..>..</Corsa>
        <Corsa ..>..</Corsa>
        ..
      </Subappalto>
      <Subappalto ..>
        ..
      </Subappalto>
    ..
  </Gestore>
</AzLinea>
<AzLinea ..>
  ..
</AzLinea>
..
</Linea>

```

=====

La struttura completa del tag <AzLinea ..> è la seguente:

```
<AzLinea sigla="..">
```

- **sigla** contiene il valore del codice Linea utilizzato internamente al DB Aziendale

=====

La struttura completa del tag **<Gestore ..>** è la seguente:

<Gestore code=".." name="..">

- **code** deve contenere il codice numerico OTRT corrispondente all'Azienda TPL responsabile della gestione del servizio. *Nel caso di Lotti di Gara assegnati ad un'Azienda Unica ovviamente questo valore indicherà sempre l'Azienda Assegnataria del Lotto. In tutti gli altri casi [Lotti di Gara assegnati ad una Scarl] andrà invece indicata quale tra le diverse Aziende associate nel consorzio gestisce effettivamente il servizio.*
- **name** deve contenere la denominazione dell'Azienda. *Questo valore serve esclusivamente per migliorare la human-legibility del codice XML, quindi verrà ignorato durante le elaborazioni. Farà comunque fede il codice indicato da **code***

=====

La struttura completa del tag **<Subappalto ..>** è la seguente:

<Subappalto code=".." name="..">

- **code** deve contenere il codice numerico OTRT corrispondente all'Azienda TPL che effettivamente eroga il servizio. *Nel caso in cui il servizio sia affidato in subappalto andrà indicato il codice dell'Azienda subappaltante. In tutti gli altri casi [servizi erogati direttamente dall'Azienda senza ricorrere a subappalto] andrà invece ripetuto il codice dell'Azienda indicato nell'elemento **<Gestore ..>** di appartenenza.*
- **name** deve contenere la denominazione dell'Azienda. *Questo valore serve esclusivamente per migliorare la human-legibility del codice XML, quindi verrà ignorato durante le elaborazioni. Farà comunque fede il codice indicato da **code***

=====

La struttura completa del tag <Corsa ..> è la seguente:

```
<Corsa id=".." v=".." code=".." IdStd=".." parte=".."
EnteCode=".." EnteName="..">
```

- **id** deve contenere un opportuno valore numerico univoco. *Preferibilmente verrà riportato l'ID univoco che identifica la corsa all'interno del DB Aziendale*
- **v** indica il verso di marcia della corsa. A indica il verso di andata, R indica il verso di ritorno. Nel caso di linee circolari può anche assumere convenzionalmente un valore '-'
- **code** deve indicare il codice identificativo della corsa assegnato dall'Azienda. *Se l'Azienda non utilizza nessun sistema interno per la codifica delle corse può essere valorizzato a spazi*
- **IdStd** deve indicare il codice univoco della corsa standard di riferimento. *Quindi ciascuna corsa erediterà la sequenza delle fermate e la matrice oraria di marcia definita per la corsa standard di riferimento. N.B.* se la corsa standard corrispondente non è stata dichiarata in un elemento <CorsaStd ..> all'interno della sezione <Percorsi>, e quindi il riferimento relazionale non può essere soddisfatto, si verifica una condizione di **errore fatale**; conseguentemente l'intera comunicazione verrà respinta in quanto inutilizzabile.
- **parte** deve indicare l'orario di partenza della corsa. I tempi devono essere espressi come **HH:MM** *Quindi i valori legali per gli orari vanno da 00:00 fino a 23:59*
- **EnteCode** deve contenere il codice numerico OTRT corrispondente all'Ente intestatario del Contratto di Servizio cui appartiene la corsa. **N.B.** le corse appartenenti ad una medesima linea possono anche appartenere a Contratti di Servizio differenti; ed è anche legittimamente possibile che l'Ente intestatario del Contratto di Servizio differisca dall'Ente proprietario della linea
 - *Vedi note precedenti relative al tag <Ente ...>*
- **EnteName** deve contenere la denominazione dell'Ente intestatario del Contratto di Servizio cui appartiene la corsa. *Questo valore serve esclusivamente per migliorare la human-legibility del codice XML, quindi verrà ignorato durante le elaborazioni. Farà comunque fede il codice indicato da EnteCode*

2.k - I tags <Corsa ..> e <Periodo ..>

Ciascun elemento identificato dai tags <Corsa ..>..</Corsa> deve specificare almeno un periodo di effettuazione [calendario di servizio].

Ciascun singolo periodo è descritto all'interno di un tag <Periodo .. /> che ne specifica sia l'estensione temporale che la cadenza associata; quindi la struttura dell'elemento <Corsa ..> è:

```
<Corsa ..>
  <Periodo .. />
  <Periodo .. />
  ..
</Corsa>
```

=====

La struttura completa del tag <Periodo .. /> è la seguente:

```
<Periodo inizio=".." fine=".." code=".."
      excl=".." />
```

- **inizio** deve indicare la data di inizio del periodo di validità della corsa. Le date devono sempre essere espresse come **GG/MM/AAAA**
- **fine** deve indicare la data di fine del periodo di validità della corsa. Le date devono sempre essere espresse come **GG/MM/AAAA** Per buona norma le comunicazioni devono essere consistenti, quindi devono valere le relazioni <DbcXml>inizio <= <Periodo>inizio e <DbcXml>fine >= <Periodo>fine
- **code** deve esprimere il codice Aziendale univoco assegnato alla cadenza. **N.B.** se la cadenza corrispondente non è stata dichiarata in un elemento <Cad ..> all'interno della sezione <Cadenze>, e quindi il riferimento relazionale non può essere soddisfatto, si verifica una condizione di **errore fatale**; conseguentemente l'intera comunicazione verrà respinta in quanto inutilizzabile.
- **excl** indica se si tratta di una fermata di esclusione contenuta all'interno di un periodo di effettuazione della corsa. Può assumere i valori S o N *Nel caso in cui risulti dichiarato un periodo di esclusione che non è compreso all'interno di un periodo di effettuazione, il periodo di esclusione verrà ignorato. Nel caso in cui invece l'estensione del periodo di esclusione sia solo parzialmente compresa all'interno di un periodo di effettuazione verranno esclusi solo i giorni realmente effettuati.* **N.B.** in ogni caso la presenza all'interno di una comunicazione di una corsa che non risulti mai effettuata neppure per un singolo giorno provocherà una condizione di **errore fatale**; conseguentemente l'intera comunicazione verrà respinta in quanto inutilizzabile.

3 - Il formato DbcXml Level 2

3.a - Struttura generale del documento XML:

Una comunicazione DbcXml Level 2 descrive al suo interno una rete TPL [comprese le rappresentazioni cartografiche delle fermate e dei percorsi] e l'offerta di trasporto ad essa associata [in termini di orari corsa e calendari di servizio] relativamente ad un periodo di validità temporale specificato.

Il formato Level 2 prevede la totale aderenza agli standards più evoluti stabiliti dall'OTRT, e quindi presuppone che tutti gli elementi cartografici siano rigorosamente coerenti rispetto al Grafo Stradale Ufficiale della Regione Toscana.

Di regola una comunicazione DbcXml deve descrivere complessivamente un'intera rete TPL integrata a livello di Lotto di Gara, e quindi deve essere prodotta dal soggetto gestore del Lotto [Scarl].

La struttura generale di un documento DbcXml Level 2 è la seguente:

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<DbcXml level="2.0" ..>
  <TmpNodi>
    ..
  </TmpNodi>
  <TmpArchi>
    ..
  </TmpArchi>
  <KillArchi>
    ..
  </KillArchi>
  <Fermate>
    ..
  </Fermate>
  <Rivendite>
    ..
  </Rivendite>
  <Tratte>
    ..
  </Tratte>
  <Percorsi>
    ..
  </Percorsi>
  <Cadenze>
    ..
  </Cadenze>
  <Calendario>
    ..
  </Calendario>
  <PianoCorse>
    ..
  </PianoCorse>
</DbcXml>
```

- L'intera comunicazione è incapsulata all'interno del tag **<DbcXml ...>...</DbcXml>**; **level="2.0"** indica appunto che si tratta di una comunicazione in formato **Level 2**
- All'interno della sezione corrispondente al tag **<TmpNodi>...</TmpNodi>** si trova la lista dei nodi del grafo stradale modificato aggiunti dall'azienda TPL [comprese le relative rappresentazioni cartografiche]

- All'interno della sezione corrispondente al tag `<TmpArchi>...</TmpArchi>` si trova la lista degli archi del grafo stradale modificato aggiunti dall'azienda TPL [comprese le relative rappresentazioni cartografiche]
- All'interno della sezione corrispondente al tag `<KillArchi>...</KillArchi>` si trova la lista degli archi [o porzioni di arco] del grafo stradale originale RT soppressi dall'azienda TPL [comprese le relative rappresentazioni cartografiche]
- All'interno della sezione corrispondente al tag `<Fermate>...</Fermate>` si trova la lista delle fermate presenti nella rete TPL [comprese le relative rappresentazioni cartografiche]
- All'interno della sezione corrispondente al tag `<Rivendite>...</Rivendite>` si trova la lista delle rivendite dei titoli di viaggio a supporto della rete TPL [comprese le relative rappresentazioni cartografiche]
- All'interno della sezione corrispondente al tag `<Tratte>...</Tratte>` si trova la lista delle tratte che congiungono due fermate consecutive descritte in termini di sequenze ordinate di archi del grafo stradale [comprese le relative rappresentazioni cartografiche]
- All'interno della sezione corrispondente al tag `<Percorsi>...</Percorsi>` si trova la lista dei percorsi presenti nella rete TPL [i percorsi sono descritti in termini di sequenze ordinate di tratte]
- All'interno della sezione corrispondente al tag `<Cadenze>...</Cadenze>` si trova la lista delle cadenze-corsa utilizzate per la definizione dei calendari di servizio
- All'interno della sezione corrispondente al tag `<Calendario>...</Calendario>` si trova la lista delle cadenze-giorno che definiscono il calendario di servizio
- All'interno della sezione corrispondente al tag `<PianoCorse>...</PianoCorse>` si trova la definizione dell'offerta di servizio [in termini di orari corsa]
- **N.B.** l'ordine relativo delle sezioni sopra elencate deve essere tassativamente quello indicato, in modo tale da semplificare la verifica dei vincoli relazionali esistenti tra i diversi elementi.

3.b - Il tag `<DbcXml ..>`

Questo tag specifica i parametri generali di presentazione della comunicazione; la struttura completa è la seguente:

```
<DbcXml level="2.0" azienda=".." name=".."
      data=".." nro=".." inizio=".." fine=".."
      agente=".." tipo="..">...</DbcXml>
```

- **level** deve essere sempre impostato come **2.0** nel caso delle comunicazioni **Level 2**
- **azienda** deve contenere il codice numerico OTRT corrispondente all'Azienda che presenta la comunicazione. Di norma la comunicazione deve essere presentata da un'Azienda Scarl assegnataria di un Lotto di Gara.
- **name** deve contenere la denominazione dell'azienda. *Questo valore serve esclusivamente per migliorare la human-legibility del codice XML, quindi verrà ignorato durante le elaborazioni. Farà comunque fede il codice indicato da **azienda***
- **data** deve indicare la data di presentazione della comunicazione. Le date devono sempre essere espresse come **GG/MM/AAAA**
- **nro** deve indicare il protocollo progressivo di invio delle comunicazioni. *Questo valore serve esclusivamente per migliorare la human-legibility del codice XML, quindi verrà ignorato durante le elaborazioni. Farà comunque fede l'ordine temporale di ricezione delle comunicazioni da parte del Server OTRT*
- **inizio** deve indicare la data di inizio del periodo di validità del piano di servizio. Per buona norma le comunicazioni devono essere preventive, quindi deve valere la relazione **inizio** >= **data** Le date devono sempre essere espresse come **GG/MM/AAAA**

- **fine** deve indicare la data di fine del periodo di validità del piano di servizio; ovviamente deve sempre valere la relazione **inizio** <= **fine**. Nel caso in cui il Server OTRT riceva una comunicazione successiva relativa ad un periodo temporale già precedentemente comunicato, si intende che l'ultima comunicazione sostituisce *in toto* quanto comunicato in precedenza. Le date devono sempre essere espresse come **GG/MM/AAAA**
- **agente** deve indicare il nominativo del funzionario Aziendale che ha curato la trasmissione della comunicazione. *Questo valore serve esclusivamente per migliorare la human-legibility del codice XML, quindi verrà ignorato durante le elaborazioni.*
- **tipo** specifica il ruolo della comunicazione. Può assumere uno dei seguenti valori letterali:
 - **BUDGET** – indica una comunicazione del Budget Contrattuale. Questa deve essere inviata solo ed esclusivamente in occasione della stipula iniziale di un Contratto di Servizio. *Quindi per norma verrà inviata un'unica comunicazione del Budget*
 - **PLANNING** – indica una comunicazione del Piano di Servizio. Questa deve essere sollecitamente inoltrata in ciascuna occasione in cui si applichi una qualsiasi variazione, anche minimale, dell'offerta di servizio tale da alterare gli orari corsa, i percorsi di linea. *Quindi di regola è atteso l'invio di frequenti comunicazioni del Piano di Servizio aggiornato.* La ricezione di una comunicazione del Piano di Servizio causerà automaticamente l'aggiornamento del DB OTRT.
 - **TEST** – indica una comunicazione priva di qualsiasi valore pratico. Una comunicazione di Test verrà valutata ed analizzata dal Server OTRT, ma la sua ricezione non avrà alcuna conseguenza di sorta in termini di aggiornamento del DB OTRT. *Quindi le comunicazioni di Test servono solo ed esclusivamente come strumento diagnostico e di verifica del SW*
 - **PROJECT** – indica una comunicazione di tipo Ipotesi / Progetto. Una comunicazione di questo tipo verrà memorizzata permanentemente dal Server OTRT, ma non avrà conseguenze per quanto riguarda l'aggiornamento del Piano di Servizio. *Quindi una comunicazione Ipotesi / Progetto serve solo per formalizzare p.es. un'ipotesi di ristrutturazione del servizio a fini di studio e valutazione.*
 - **UNKNOWN** – se **tipo** non indica uno dei valori precedenti, la comunicazione verrà ritenuta completamente inaccettabile e sarà conseguentemente respinta, anche se formalmente ineccepibile.

3.c - I tags <TmpNodi>, <TmpNod ..> e <RtArc>

La sezione identificata dai tags <TmpNodi>..</TmpNodi> deve contenere la lista dei nodi del grafo stradale modificato aggiunti dall'Azienda TPL.

Ciascun singolo nodo aggiunto è descritto all'interno di un tag <TmpNod ..></TmpNod> che ne specifica gli attributi e la posizione cartografica georeferenziata. A sua volta un elemento <TmpNod ..></TmpNod> può contenere nessuno, uno o più elementi <RtArc .. />, ciascuno dei quali identifica un arco del Grafo Stradale Ufficiale RT intercettato dal nodo aggiunto; quindi la struttura della sezione <TmpNodi> è:

```
<TmpNodi>
  <TmpNod ..>
  </TmpNod>
  <TmpNod ..>
    <RtArc .. />
  </TmpNod>
  <TmpNod ..>
    <RtArc .. />
    <RtArc .. />
    ..
  </TmpNod>
  ..
</TmpNodi>
```

Anche nel caso [improbabile, ma teoricamente ammissibile] in cui l'Azienda TPL abbia utilizzato esclusivamente gli archi ed i nodi presenti nel Grafo Stradale Ufficiale RT, senza quindi procedere a definire un proprio grafo stradale modificato, la sezione <TmpNodi></TmpNodi> deve essere dichiarata esplicitamente; in questo caso particolare naturalmente non andrà specificato nessun elemento <TmpNod ..></TmpNod>, quindi avremo:

```
<TmpNodi>
</TmpNodi>
```

=====

La struttura completa del tag <TmpNod ..> è la seguente:

```
<TmpNod id=".." x=".." y="..">
```

- **id** deve contenere un opportuno valore numerico univoco tale da identificare in maniera non ambigua il nodo aggiunto.
- **x** deve indicare la coordinata cartografica corrispondente al nodo aggiunto [secondo Gauss-Boaga]
- **y** deve indicare la coordinata cartografica corrispondente al nodo aggiunto [secondo Gauss-Boaga] **N.B.** le coordinate devono essere espresse in metri con almeno un decimale; come separatore della parte frazionale deve essere utilizzato il carattere punto.

=====

La struttura completa del tag **<RtArc .. />** è la seguente:

```
<RtArc code=".." />
```

- **code** deve indicare il codice univoco che identifica l'arco intercettato all'interno del Grafo Stradale Ufficiale RT di riferimento [**CODELE**]. *Naturalmente deve risultare soddisfatto un criterio di prossimità spaziale tra il nodo aggiunto e l'arco intercettato, per cui possono darsi i seguenti casi:*
 - *le coordinate spaziali del nodo aggiunto coincidono con un vertice appartenente alla polilinea che descrive l'arco*
 - *le coordinate spaziali del nodo aggiunto identificano un punto di intersezione con uno dei segmenti appartenenti alla polilinea che descrive l'arco*

3.d - I tags **<TmpArchi>**, **<TmpArc ..>** e **<Pt ..>**

La sezione identificata dai tags **<TmpArchi>..</TmpArchi>** deve contenere la lista degli archi del grafo stradale modificato aggiunti dall'Azienda TPL.

Ciascun singolo arco aggiunto è descritto all'interno di un tag **<TmpArc ..></TmpArc>** che ne specifica gli attributi. A sua volta ciascun elemento **<TmpArc ..></TmpArc>** deve contenere una sequenza ordinata di elementi **<Pt .. />**, tale da descrivere cartograficamente la geometria dell'arco aggiunto; quindi la struttura della sezione **<TmpArchi>** è:

```
<TmpArchi>
  <TmpArc ..>
    <Pt .. />
    <Pt .. />
    ..
  </TmpArc>
  <TmpArc ..>
    <Pt .. />
    <Pt .. />
    ..
  </TmpArc>
  ..
</TmpArchi>
```

Anche nel caso [improbabile, ma teoricamente ammissibile] in cui l'Azienda TPL abbia utilizzato esclusivamente gli archi ed i nodi presenti nel Grafo Stradale Ufficiale RT, senza quindi procedere a definire un proprio grafo stradale modificato, la sezione **<TmpArchi>** deve essere dichiarata esplicitamente; in questo caso particolare naturalmente non andrà specificato nessun elemento **<TmpArc ></TmpArc>** quindi avremo:

```
<TmpArchi>
</TmpArchi>
```

=====

La struttura completa del tag <TmpArc ..> è la seguente:

```
<TmpArc id=".." RtNodIni=".." TmpNodIni=".."
      RtNodFin=".." TmpNodFin=".."
      topon=".." pts="..">
```

- **id** deve contenere un opportuno valore numerico univoco tale da identificare in maniera non ambigua l'arco aggiunto.
- **RtNodIni** identifica il codice del nodo di partenza dell'arco aggiunto [nel caso in cui si tratti di un nodo già definito all'interno del Grafo Stradale Ufficiale RT]
- **TmpNodIni** identifica il codice del nodo di partenza dell'arco aggiunto [nel caso in cui si tratti di un nodo appartenente al grafo modificato e quindi aggiunto dall'Azienda TPL] *Quindi, dato che il nodo o appartiene al grafo ufficiale o appartiene al grafo modificato, la presenza del valore di **RtNodIni** e di **TmpNodIni** è mutuamente esclusiva; occorre definire entrambi gli attributi, ma uno solo di essi deve assumere un valore non nullo.*
- **RtNodFin** identifica il codice del nodo di arrivo dell'arco aggiunto [nel caso in cui si tratti di un nodo già definito all'interno del Grafo Stradale Ufficiale RT]
- **TmpNodFin** identifica il codice del nodo di arrivo dell'arco aggiunto [nel caso in cui si tratti di un nodo appartenente al grafo modificato e quindi aggiunto dall'Azienda TPL] *Quindi, dato che il nodo o appartiene al grafo ufficiale o appartiene al grafo modificato, la presenza dei valori di **RtNodFin** e di **TmpNodFin** è mutuamente esclusiva; occorre definire entrambi gli attributi, ma uno solo di essi deve assumere un valore non nullo.*
- **topon** deve indicare il toponimo associato all'arco stradale. *Considerato che le Aziende TPL non sono soggetti ufficialmente abilitati all'aggiornamento della toponomastica stradale, l'assegnazione del toponimo è fortemente gradita ma non strettamente obbligatoria. Quindi può anche legittimamente valorizzato a spazi.*
- **pts** deve indicare il numero dei vertici che compongono la polilinea che rappresenta cartograficamente l'arco [i.e. deve coincidere con il numero di elementi <Pt ..> subordinati]

=====

La struttura completa del tag <Pt ..> è la seguente:

```
<Pt x=".." Y=".." />
```

- **x** deve indicare la coordinata cartografica corrispondente al vertice [secondo Gauss-Boaga]
- **y** deve indicare la coordinata cartografica corrispondente al vertice [secondo Gauss-Boaga] **N.B.** le coordinate devono essere espresse in metri con almeno un decimale; come separatore della parte frazionale deve essere utilizzato il carattere punto .
- **caveat**: dato che XML è un formato testuale, e quindi tende ad occupare molto spazio, per la codifica delle entità geometriche (già di per se onerosa) occorre applicare un opportuno **criterio di compressione spaziale** che opera come segue:
 - al primo elemento <Pt ..> associato alla geometria vengono sempre assegnate le coordinate cartografiche reali [secondo Gauss-Boaga]
 - a tutti gli elementi successivi al primo invece deve essere assegnato il valore della differenza rispetto al punto immediatamente precedente misurata sull'asse corrispondente. *Ovviamente la differenza deve essere gestita come un valore signed*
- **N.B.** dato che sussistono precisi vincoli di coerenza spaziale e topologici tra gli elementi [archi e nodi] del grafo stradale, valgono i seguenti vincoli:
 - le coordinate spaziali assegnate al primo vertice della polilinea che rappresenta l'arco devono necessariamente coincidere con le coordinate associate al nodo iniziale.
 - le coordinate spaziali assegnate all'ultimo vertice della polilinea che rappresenta l'arco devono necessariamente coincidere con le coordinate associate al nodo finale.
- Qualora un vincolo di coerenza relazionale [identificazione dei nodi di inizio e fine associati all'arco] non possa essere soddisfatto, si verifica una condizione di **errore fatale**; conseguentemente l'intera comunicazione verrà respinta in quanto inutilizzabile. La medesima condizione si verificherà anche nel caso in cui non risulti soddisfatto un criterio di coerenza spaziale e topologica [difformità nelle coordinate dei vertici iniziali e finali rispetto a quelle dichiarate per i nodi corrispondenti]

3.e - I tags **<KillArchi>**, **<KillArc ..>** e **<Pt ..>**

La sezione identificata dai tags **<KillArchi>..</KillArchi>** deve contenere la lista degli archi [o porzioni di arco] presenti all'interno del Grafo Stradale Ufficiale RT per i quali l'Azienda TPL suggerisce l'opportunità dell'eliminazione in quanto obsoleti, non conformi, errati etc etc.

Ciascun singolo arco [o porzione di arco] da sopprimere è descritto all'interno di un tag **<KillArc ..></KillArc>** che ne specifica gli attributi. A sua volta ciascun elemento **<KillArc ..>** deve contenere una sequenza ordinata di elementi **<Pt .. />**, tale da descrivere cartograficamente la geometria dell'arco da sopprimere; quindi la struttura della sezione **<KillArchi>** è:

```
<KillArchi>
  <KillArc ..>
    <Pt .. />
    <Pt .. />
    ..
  </KillArc>
  <KillArc ..>
    <Pt .. />
    <Pt .. />
    ..
  </KillArc>
  ..
</KillArchi>
```

Anche nel caso in cui l'Azienda TPL non abbia riscontrato nessun arco [o porzione di arco] per il quale si ritiene utile ed opportuno proporre l'eliminazione dal Grafo Stradale Ufficiale RT, la sezione **<KillArchi>** deve essere dichiarata esplicitamente; in questo caso particolare naturalmente non andrà specificato nessun elemento **<KillArc ></KillArc>** quindi avremo:

```
<KillArchi>
</KillArchi>
```

=====

La struttura completa del tag <KillArc ..> è la seguente:

```
<KillArc code=".." RtNodIni=".." TmpNodIni=".."
      RtNodFin=".." TmpNodFin=".."
      topon=".." pts="..">
```

- **code** deve indicare il codice univoco che identifica all'interno del Grafo Stradale Ufficiale RT l'arco da sopprimere in tutto o in parte [CODELE].
- **RtNodIni** identifica il codice del nodo di partenza dell'arco da sopprimere [nel caso in cui si tratti di un nodo già definito all'interno del Grafo Stradale Ufficiale RT]
- **TmpNodIni** identifica il codice del nodo di partenza dell'arco da sopprimere [nel caso in cui si tratti di un nodo appartenente al grafo modificato e quindi aggiunto dall'Azienda TPL] *Quindi, dato che il nodo o appartiene al grafo ufficiale o appartiene al grafo modificato, la presenza del valore in **RtNodIni** e di **TmpNodIni** è mutuamente esclusiva; occorre definire entrambi gli attributi, ma uno solo di essi deve assumere un valore non nullo.*
- **RtNodFin** identifica il codice del nodo di arrivo dell'arco da sopprimere [nel caso in cui si tratti di un nodo già definito all'interno del Grafo Stradale Ufficiale RT]
- **TmpNodFin** identifica il codice del nodo di arrivo dell'arco da sopprimere [nel caso in cui si tratti di un nodo appartenente al grafo modificato e quindi aggiunto dall'Azienda TPL] *Quindi, dato che il nodo o appartiene al grafo ufficiale o appartiene al grafo modificato, la presenza del valore in **RtNodFin** e di **TmpNodFin** è mutuamente esclusiva; occorre definire entrambi gli attributi, ma uno solo di essi deve assumere un valore non nullo.*
- **topon** deve indicare il toponimo associato all'arco stradale. *Considerato che le Aziende TPL non sono soggetti ufficialmente abilitati all'aggiornamento della toponomastica stradale, l'assegnazione del toponimo è fortemente gradita ma non strettamente obbligatoria. Quindi può anche legittimamente valorizzato a spazi.*
- **pts** deve indicare il numero dei vertici che compongono la polilinea che rappresenta cartograficamente l'arco o porzione di arco [i.e. deve coincidere con il numero di elementi <Pt ..> subordinati]
- **N.B.** nel caso in cui **RtNodIni** ed **RtNodFin** identifichino i nodi estremi dell'arco così come originariamente definito all'interno del Grafo Stradale Ufficiale RT, si intende che viene richiesta la soppressione dell'intero arco. In tutti gli altri casi [uno dei nodi estremi identifica un nodo aggiunto **TmpNodxx**] viene richiesta la soppressione di una porzione dell'arco originario, che evidentemente è stato spezzato in più parti dall'inserimento del nodo aggiuntivo. *Quindi l'identificazione della parte da sopprimere viene poi univocamente specificata tramite esplicita definizione della geometria associata*

=====

La struttura completa del tag <Pt ..> è la seguente:

```
<Pt x=".." y=".." />
```

- **x** deve indicare la coordinata cartografica corrispondente al vertice [secondo Gauss-Boaga]
- **y** deve indicare la coordinata cartografica corrispondente al vertice [secondo Gauss-Boaga] **N.B.** le coordinate devono essere espresse in metri con almeno un decimale; come separatore della parte frazionale deve essere utilizzato il carattere punto .
- **caveat**: dato che XML è un formato testuale, e quindi tende ad occupare molto spazio, per la codifica delle entità geometriche (già di per se onerosa) occorre applicare un opportuno **criterio di compressione spaziale** che opera come segue:
 - al primo elemento <Pt ..> associato alla geometria vengono sempre assegnate le coordinate cartografiche reali [secondo Gauss-Boaga]
 - a tutti gli elementi successivi al primo invece deve essere assegnato il valore della differenza rispetto al punto immediatamente precedente misurata sull'asse corrispondente. *Ovviamente la differenza deve essere gestita come un valore signed*
- **N.B.** dato che sussistono precisi vincoli di coerenza spaziale e topologici tra gli elementi [archi e nodi] del grafo stradale, valgono i seguenti vincoli:
 - le coordinate spaziali assegnate al primo vertice della polilinea che rappresenta l'arco devono necessariamente coincidere con le coordinate associate al nodo iniziale.
 - le coordinate spaziali assegnate all'ultimo vertice della polilinea che rappresenta l'arco devono necessariamente coincidere con le coordinate associate al nodo finale.
 - nel caso specifico degli archi [o porzioni di arco] per le quali si richiede l'eliminazione del Grafo Stradale Ufficiale RT sussiste l'ulteriore vincolo per cui tutti i vertici compresi tra il secondo ed il penultimo [esclusi quindi i nodi iniziale e finale] debbono obbligatoriamente coincidere con un vertice presente nella rappresentazione originale dell'arco Ufficiale RT
- Qualora un vincolo di coerenza relazionale [identificazione dei nodi di inizio e fine associati all'arco] non possa essere soddisfatto, si verifica una condizione di **errore fatale**; conseguentemente l'intera comunicazione verrà respinta in quanto inutilizzabile. La medesima condizione si verificherà anche nel caso in cui non risulti soddisfatto un criterio di coerenza spaziale e topologica [difformità nelle coordinate dei vertici iniziali e finali rispetto a quelle dichiarate per i nodi corrispondenti]

3.f - I tags <Fermate> e <Fmt ..>

La sezione identificata dai tags <Fermate>..</Fermate> deve contenere la lista delle fermate che contraddistinguono la rete TPL.

Ciascuna singola fermata è descritta all'interno di un tag <Fmt .. /> che ne specifica gli attributi e la posizione cartografica georeferenziata; quindi la struttura della sezione <Fermate> è:

```
<Fermate>
  <Fmt .. />
  <Fmt .. />
  <Fmt .. />
  ..
</Fermate>
```

=====

La struttura completa del tag <Fmt .. /> è la seguente:

```
<Fmt code=".." name=".." ubic=".." x=".." y=".."
      RtArc=".." TmpArc=".." RtNod=".." TmpNod=".." />
```

- **code** deve esprimere il codice Aziendale univoco assegnato alla fermata
- **name** deve indicare la denominazione associata alla fermata
- **ubic** deve indicare l'ubicazione [Via, nro civico / Strada + km] della fermata. *Questo valore è opzionale [in quanto desumibile dall'associazione della fermata con un elemento del grafo stradale] e può essere omesso valorizzando p.es. a spazi*
- **x** deve indicare la coordinata cartografica corrispondente alla fermata [secondo Gauss-Boaga]
- **y** deve indicare la coordinata cartografica corrispondente alla fermata [secondo Gauss-Boaga] **N.B.** le coordinate devono essere espresse in metri con almeno un decimale; come separatore della parte frazionale deve essere utilizzato il carattere punto.
- **RtArc** identifica il codice dell'arco del grafo stradale associato alla fermata [nel caso in cui si tratti di un arco già definito all'interno del Grafo Stradale Ufficiale RT]
- **TmpArc** identifica il codice dell'arco del grafo stradale associato alla fermata [nel caso in cui si tratti di un arco appartenente al grafo modificato e quindi aggiunto dall'Azienda TPL]
- **RtNod** identifica il codice del nodo del grafo stradale associato alla fermata [nel caso in cui si tratti di un nodo già definito all'interno del Grafo Stradale Ufficiale RT]
- **TmpNod** identifica il codice del nodo del grafo stradale associato alla fermata [nel caso in cui si tratti di un nodo appartenente al grafo modificato e quindi aggiunto dall'Azienda TPL] *Quindi, dato che la fermata deve essere associata ad un unico elemento del grafo stradale [Ufficiale o modificato], la presenza del valore in **RtArc**, **TmpArc**, **RtNod** e **TmpNod** è mutuamente esclusiva; occorre definire tutti e quattro gli attributi, ma uno solo di essi deve assumere un valore non nullo.*

3.g - I tags <Rivendite> e <Riv ..>

La sezione identificata dai tags <Rivendite>..</Rivendite> deve contenere la lista delle rivendite di titoli di viaggio associate alla rete TPL [biglietterie, bar, tabacchi, edicole etc].

Ciascuna singola rivendita è descritta all'interno di un tag <Riv .. /> che ne specifica gli attributi e la posizione cartografica georeferenziata; quindi la struttura della sezione <Rivendite> è:

```
<Rivendite>
  <Riv .. />
  <Riv .. />
  <Riv .. />
  ..
</Rivendite>
```

=====

La struttura completa del tag <Riv .. /> è la seguente:

```
<Riv code=".." name=".." ragsoc=".." ubic=".."
  chiuso=".." x=".." y=".." />
```

- **code** deve esprimere il codice Aziendale univoco assegnato alla rivendita. **N.B.** è tassativamente obbligatorio assegnare una codifica Aziendale univoca alle rivendite, anche considerando che in una fase successiva verrà attivata la Comunicazione periodica relativa alle statistiche di vendita dei titoli di viaggio.
- **name** deve indicare la denominazione associata alla rivendita
- **ragsoc** deve indicare la ragione sociale completa dell'esercizio. *Questo valore è opzionale e può essere omissso valorizzando p.es. a spazi*
- **ubic** deve indicare l'ubicazione [Via, nro civico / Strada + km] della rivendita. *Questo valore è obbligatorio e non può essere omissso, in quanto è un elemento essenziale per consentire una completa informazione al pubblico.*
- **chiuso** deve indicare il giorno [o periodo] di chiusura. . *Questo valore è opzionale e può essere omissso valorizzando p.es. a spazi*
- **x** deve indicare la coordinata cartografica corrispondente alla rivendita [secondo Gauss-Boaga]
- **y** deve indicare la coordinata cartografica corrispondente alla rivendita [secondo Gauss-Boaga] **N.B.** le coordinate devono essere espresse in metri con almeno un decimale; come separatore della parte frazionale deve essere utilizzato il carattere punto .

3.h - I tags <Tratte> e <Trt ..>

La sezione identificata dai tags <Tratte>..</Tratte> deve contenere la lista delle tratte fermata-fermata utilizzate dalla rete TPL.

Ciascuna singola tratta è descritta all'interno di un tag <Trt .. />..</Trt> che ne specifica gli attributi, la lista ordinata di archi del grafo stradale percorsi, e la rappresentazione cartografica georeferenziata; quindi la struttura della sezione <Tratte> è:

```

<Tratte>
  <Trt ..>..</Trt>
  <Trt ..>..</Trt>
  <Trt ..>..</Trt>
  ..
</Tratte>

```

A sua volta ciascun singolo tag <Trt ..></Trt> è strutturato come segue:

```

<Trt ..>
  <SeqArchi>..</SeqArchi>
  <Geom ..>..</Geom>
</Trt>

```

- l'elemento <SeqArchi></SeqArchi> specifica la lista ordinata degli archi stradali intercettati dalla tratta
- l'elemento <Geom ..></Geom> specifica la descrizione cartografica georeferenziata relativa alla tratta
- per ciascuna tratta deve essere naturalmente essere specificata un'unica geometria univoca che congiunga la fermata di partenza con la fermata di arrivo.
- ovviamente l'entità geometrica che descrive cartograficamente la tratta deve rispettare rigorosi vincoli di coerenza spaziale rispetto alla sequenza di archi del grafo associata.

=====

La struttura completa del tag `<Trt .. />` è la seguente:

```
<Trt id=".." fmIni=".." fmFin=".." metri="..">
```

- **id** deve contenere un opportuno valore numerico univoco tale da identificare in maniera non ambigua la tratta.
- **fmIni** deve indicare il codice che identifica la fermata di partenza della tratta.
- **fmFin** deve indicare il codice che identifica la fermata di arrivo della tratta.
- **metri** deve indicare la lunghezza complessiva della tratta **N.B.** la lunghezza deve essere espresse in metri con almeno un decimale; come separatore della parte frazionale deve essere utilizzato il carattere punto.
- **N.B.** se il codice che identifica fermata di inizio o fine tratta non è stata dichiarato in un elemento `<Fmt ..>` all'interno della sezione `<Fermate>`, e quindi il riferimento relazionale non può essere soddisfatto, si verifica una condizione di **errore fatale**; conseguentemente l'intera comunicazione verrà respinta in quanto inutilizzabile.

3.i - I tags <SeqArchi> e <TrtArc ..>

L'elemento identificato dai tags <SeqArchi>..</SeqArchi> deve contenere la lista ordinata degli archi stradali associata ad una tratta che congiunge due fermate consecutive.

Ciascun singolo arco intercettato dalla tratta è riferito all'interno di un tag <TrtArc .. />; quindi la struttura dell'elemento <SeqArchi> è:

```
<SeqArchi>
  <TrtArc .. />
  <TrtArc .. />
  <TrtArc .. />
  ..
</SeqArchi>
```

=====

La struttura completa del tag <TrtArc .. /> è la seguente:

```
<TrtArc sub=".." RtArc=".." TmpArc=".."
      RtNod=".." TmpNod="..">
```

- **sub** deve contenere un opportuno valore numerico progressivo che consenta di ordinare la lista degli archi stradali associati a <SeqArchi> *E' fortemente gradito che gli elementi <TrtArc> vengano inseriti rispettando l'ordine naturale di percorrenza.*
- **RtArc** identifica il codice dell'arco del grafo stradale intercettato dalla tratta [nel caso in cui si tratti di un arco già definito all'interno del Grafo Stradale Ufficiale RT]
- **TmpArc** identifica il codice dell'arco del grafo stradale intercettato dalla tratta [nel caso in cui si tratti di un arco appartenente al grafo modificato e quindi aggiunto dall'Azienda TPL] *Quindi dato che l'arco stradale o appartiene al grafo ufficiale o appartiene al grafo modificato, la presenza del valore in **RtArc** e di **TmpArc** è mutuamente esclusiva; occorre definire entrambi gli attributi, ma uno solo di essi deve assumere un valore non nullo.*
- **RtNod** identifica il codice del nodo estremo associato all'arco stradale verso il quale avviene il movimento [nel caso in cui si tratti di un nodo già definito all'interno del Grafo Stradale Ufficiale RT]
- **TmpNod** identifica il codice del nodo estremo associato all'arco stradale verso il quale avviene il movimento [nel caso in cui si tratti di un nodo appartenente al grafo modificato e quindi aggiunto dall'Azienda TPL] *Quindi dato che il nodo o appartiene al grafo ufficiale o appartiene al grafo modificato, la presenza del valore in **RtNod** e di **TmpNod** è mutuamente esclusiva; occorre definire entrambi gli attributi, ma uno solo di essi deve assumere un valore non nullo.*

3.j - I tags <Percorsi> e <Itn ..>

La sezione identificata dai tags <Percorsi>..</Percorsi> deve contenere la lista dei percorsi utilizzati dalla rete TPL.

Ciascun singolo percorso è descritto all'interno di un tag <Itn .. />..</Itn> che ne specifica gli attributi e la struttura spaziale in termini di sequenza ordinata di tratte fermata-fermata; quindi la struttura della sezione <Percorsi> è:

```
<Percorsi>
  <Itn ..>..</Itn>
  <Itn ..>..</Itn>
  <Itn ..>..</Itn>
  ..
</Percorsi>
```

A sua volta ciascun singolo tag <Itn ..></Itn> è strutturato come segue:

```
<Itn ..>
  <SeqTratte>..</SeqTratte>
  <CorseStandard>..</CorseStandard>
</Itn>
```

- l'elemento <SeqTratte></SeqTratte> specifica la sequenza ordinata di tratte fermata-fermata in cui si articola il percorso
- l'elemento <CorseStandard></CorseStandard> invece specifica le matrici degli orari di transito delle corse standard associate al percorso
- per ciascun percorso deve essere specificata un'unica sequenza di tratte che congiunga il capolinea di partenza con il capolinea di arrivo.
- analogamente per ciascun percorso può essere specificato un'unica lista di corse standard

=====

La struttura completa del tag <Itn .. /> è la seguente:

```
<Itn code=".." name=".." metri="..">
```

- **code** deve esprimere il codice Aziendale univoco assegnato al percorso
- **name** deve indicare la descrizione sommaria del percorso. *Questo valore è opzionale e può essere omissso valorizzando p.es. a spazi*
- **metri** deve indicare la lunghezza complessiva del percorso **N.B.** la lunghezza deve essere espresse in metri con almeno un decimale; come separatore della parte frazionale deve essere utilizzato il carattere punto.

3.k - I tags <SeqTratte> e <ItnTrt ..>

L'elemento identificato dai tags <SeqTratte>..</SeqTratte> deve contenere la lista ordinata delle tratte fermata-fermata che descrive un percorso.

Ciascuna singola tratta intercettata dal percorso è riferita all'interno di un tag <ItnTrt .. />; quindi la struttura dell'elemento <SeqTratte> è:

```
<SeqTratte>
  <ItnTrt .. />
  <ItnTrt .. />
  <ItnTrt .. />
  ..
</SeqTratte>
```

=====

La struttura completa del tag <ItnTrt .. /> è la seguente:

```
<ItnTrt sub=".." idTrt=".." metri="..">
```

- **sub** deve contenere un opportuno valore numerico progressivo che consenta di ordinare la lista delle tratte fermata-fermata associate a <SeqTratte> *E' fortemente gradito che gli elementi <ItnTrt> vengano inseriti rispettando l'ordine naturale di percorrenza.*
- **idTrt** identifica la tratta fermata-fermata **N.B.** se l'identificativo della tratta richiamata non è stato dichiarato in un elemento <Trt ..> all'interno della sezione <Tratte>, e quindi il riferimento relazionale non può essere soddisfatto, si verifica una condizione di **errore fatale**; conseguentemente l'intera comunicazione verrà respinta in quanto inutilizzabile.
- **metri** deve indicare la distanza progressiva [relativa alla fermata finale della tratta] misurata a partire dal capolinea iniziale del percorso **N.B.** la lunghezza deve essere espressa in metri con almeno un decimale; come separatore della parte frazionale deve essere utilizzato il carattere punto .

3.1 - I tags <CorseStandard>, <CorsaStd ..> e <StdFmt ..>

L'elemento identificato dai tags <CorseStandard>..</CorseStandard> deve contenere la lista delle corse standard associate ad un singolo percorso TPL.

Per corsa standard si intende una corsa astratta caratterizzata da una sequenza di fermate e da una matrice oraria di marcia; tutte le corse concrete associate ad una corsa standard ereditano da quest'ultima l'orario di marcia reale opportunamente ricalcolato sulla base dell'orario di partenza assegnato alla corsa reale.

Ciascuna singola corsa standard quindi è descritta in termini di una successione ordinata di elementi <StdFmt ..> tali da rappresentare la successione delle fermate ed i relativi intertempi di transito; quindi la struttura complessiva dell'elemento <CorseStandard> è:

```

<CorseStandard>
  <CorsaStd ..>
    <StdFmt .. />
    <StdFmt .. />
    <StdFmt .. />
    ..
  </CorsaStd>
  <CorsaStd ..>
    <StdFmt .. />
    <StdFmt .. />
    <StdFmt .. />
    ..
  </CorsaStd>
  ..
</CorseStandard>

```

=====

La struttura completa del tag <CorsaStd ..> è la seguente:

```
<CorsaStd id=".." tempo="..">
```

- **id** deve contenere un opportuno valore numerico univoco che consenta successivamente di associare le corse reali alla corsa standard di riferimento
- **tempo** deve indicare la durata complessiva della corsa standard espressa in minuti. In altri termini deve misurare il numero di minuti intercorrenti tra la partenza dal capolinea iniziale e l'arrivo al capolinea finale.

=====

La struttura completa del tag **<StdFmt ..>** è la seguente:

```
<StdFmt sub=".." metri=".." code=".." arriva=".."
  parte=".." primaria=".." facolt=".."
  nonferma=".." />
```

- **sub** deve contenere un opportuno valore numerico progressivo che consenta di ordinare la lista delle fermate associate a **<CorsaStd ..>** *E' fortemente gradito che gli elementi <StdFmt ..> vengano inseriti rispettando l'ordine naturale di percorrenza*
- **metri** deve indicare la distanza progressiva misurata a partire dal capolinea di partenza. *Quindi al capolinea di partenza deve obbligatoriamente corrispondere una progressiva pari a 0 (zero).* Non devono essere indicate le cifre decimali.
- **code** deve esprimere il codice Aziendale che identifica la fermata corrispondente. **N.B.** se la fermata corrispondente non è stata dichiarata in un elemento **<Fmt ..>** all'interno della sezione **<Fermate>**, e quindi il riferimento relazionale non può essere soddisfatto, si verifica una condizione di **errore fatale**; conseguentemente l'intera comunicazione verrà respinta in quanto inutilizzabile.
- **arriva** indica l'intertempo di arrivo alla fermata espresso in minuti e misurato a partire dal capolinea iniziale. *Quindi i tempi associati al capolinea di partenza saranno sempre impostati a 0 (zero)*
- **parte** indica l'intertempo di ripartenza dalla fermata espresso in minuti e misurato a partire dal capolinea iniziale. *Quindi i tempi assegnati al capolinea di arrivo dovranno necessariamente coincidere con la durata complessiva della corsa*
- **primaria** indica se la fermata corrispondente ha un rilievo particolare. Può assumere i valori S o N
- **facolt** indica se la fermata corrispondente è facoltativa. Può assumere i valori S o N
- **nonferma** indica se si tratta di una fermata vera e propria [con servizio di salita-discesa dei passeggeri] oppure di un semplice punto di riferimento orario. Può assumere i valori S o N

3.m - I tags <Cadenze> e <Cad ..>

La sezione identificata dai tags <Cadenze>..</Cadenze> deve contenere la lista delle cadenza-corsa utilizzate per codificare i calendari di servizio delle corse.

Ciascuna singola cadenza è descritta all'interno di un tag <Cad .. /> che ne specifica gli attributi; quindi la struttura della sezione <Cadenze> è:

```
<Cadenze>
  <Cad .. />
  <Cad .. />
  <Cad .. />
  ..
</Cadenze>
```

=====

La struttura completa del tag <Cad .. /> è la seguente:

```
<Cad code=".." name=".." />
```

- **code** deve esprimere il codice Aziendale univoco assegnato alla cadenza
- **name** deve indicare la denominazione associata alla cadenza

3.n - I tags <Calendario> e <Kal ..>

La sezione identificata dai tags <Calendario>..</Calendario> deve contenere la lista delle cadenza-giorno utilizzate per codificare i calendari di servizio delle corse.

Ciascuna singola cadenza-giorno è descritta all'interno di un tag <Kal .. /> che ne specifica quali cadenze sono attive in un giorno dato; quindi la struttura della sezione <Calendario> è:

```
<Calendario>
  <Kal .. />
  <Kal .. />
  <Kal .. />
  ..
</Calendario>
```

=====

La struttura completa del tag <Kal .. /> è la seguente:

```
<Kal code=".." data=".." note=".." />
```

- **code** deve esprimere il codice Aziendale univoco assegnato alla cadenza. **N.B.** se la cadenza corrispondente non è stata dichiarata in un elemento <Cad ..> all'interno della sezione <Cadenze>, e quindi il riferimento relazionale non può essere soddisfatto, si verifica una condizione di **errore fatale**; conseguentemente l'intera comunicazione verrà respinta in quanto inutilizzabile.
- **data** deve indicare la data del giorno in cui la cadenza è attiva. Le date devono sempre essere espresse come **GG/MM/AAAA**
- **note** può caratterizzare opzionalmente le date particolari (Santo Patrono etc) *Questo valore serve esclusivamente per migliorare la human-legibility del codice XML, quindi verrà ignorato durante le elaborazioni.*

3.o - I tags <PianoCorse>, <Lotto ..>, <Ente ..> e <Linea ..>

La sezione identificata dai tags <PianoCorse>..</PianoCorse> deve contenere la lista delle corse che definiscono il Piano di Esercizio e quindi l'offerta di servizi TPL.

Il Piano di Servizio è strutturato in modo tale da riflettere strettamente l'articolazione dei Contratti di Servizio; quindi la struttura complessiva della sezione <PianoCorse> è:

```

<PianoCorse>
  <Lotto ..>
    <Ente .. >
      <Linea ..>
        ..
      </Linea>
      <Linea ..>
        ..
      </Linea>
    ..
  </Ente>
  <Ente .. >
    <Linea ..>
      ..
    </Linea>
    <Linea ..>
      ..
    </Linea>
  ..
</Ente>
..
</Lotto>
</PianoCorse>

```

=====

La struttura completa del tag <Lotto ..> è la seguente:

```
<Lotto code=".." name="..">
```

- **code** deve contenere il codice numerico OTRT corrispondente al Lotto di Gara cui si riferisce la comunicazione. Di norma una comunicazione deve essere presentata da un'Azienda Scarl assegnataria di un Lotto di Gara, e deve quindi riferirsi ad un unico Lotto.
- **name** deve contenere la denominazione del Lotto di Gara. *Questo valore serve esclusivamente per migliorare la human-legibility del codice XML, quindi verrà ignorato durante le elaborazioni. Farà comunque fede il codice indicato da code*

=====

La struttura completa del tag <Ente ..> è la seguente:

```
<Ente code=".." name="..">
```

- **code** deve contenere il codice numerico OTRT corrispondente all'Ente territoriale (Provincia, Comune etc) cui appartengono le linee subordinate. **N.B.** l'Ente territoriale proprietario di una linea non deve essere confuso con l'Ente intestatario del Contratto di Servizio. Può infatti legittimamente accadere che questi non coincidano affatto; il corretto criterio da seguire è il seguente:
 - se i percorsi afferenti ad una determinata linea si svolgono completamente, oppure per la parte prevalente, all'interno di un solo Comune, allora la linea è di proprietà di quel Comune
 - nel caso in cui connettano almeno due diversi Capoluoghi di Comune, allora la linea è di proprietà della Provincia
 - se infine connettono almeno due diversi Capoluoghi di Provincia, allora la linea è di proprietà della Regione
-
- **name** deve contenere la denominazione dell'Ente. *Questo valore serve esclusivamente per migliorare la human-legibility del codice XML, quindi verrà ignorato durante le elaborazioni. Farà comunque fede il codice indicato da code*

=====

La struttura completa del tag <Linea ..> è la seguente:

```
<Linea sigla=".." name=".." denom="..">
```

- **sigla** deve contenere la sigla standard della linea. *Quindi deve corrispondere esattamente al valore riportato sugli orari di palina, sui libretti orari, nei Contratti di Servizio etc*
- **name** deve contenere la denominazione convenzionale della linea. *Questo valore dovrebbe auspicabilmente essere concordato consensualmente tra Aziende ed Enti in modo tale da implementare un sistema coerente*
- **denom** deve contenere una sommaria descrizione del percorso prevalente della linea [lista delle principali località o poli di interesse serviti]

3.p - I tags <AzLinea ..>, <Gestore ..>, <Subappalto ..> e <Corsa ..>

L'elemento identificato dai tags <Linea ..>..</Linea> deve contenere la lista delle corse che definiscono il Piano di Esercizio relativo ad una specifica linea; quindi la struttura complessiva dell'elemento <Linea ..> è:

```

<Linea ..>
  <AzLinea ..>
    <Gestore .. >
      <Subappalto ..>
        <Corsa ..>..</Corsa>
        <Corsa ..>..</Corsa>
        ..
      </Subappalto>
      <Subappalto ..>
        ..
      </Subappalto>
    ..
  </Gestore>
</AzLinea>
<AzLinea ..>
  ..
</AzLinea>
..
</Linea>

```

=====

La struttura completa del tag <AzLinea ..> è la seguente:

```
<AzLinea sigla="..">
```

- **sigla** contiene il valore del codice Linea utilizzato internamente al DB Aziendale

=====

La struttura completa del tag **<Gestore ..>** è la seguente:

<Gestore code=".." name="..">

- **code** deve contenere il codice numerico OTRT corrispondente all'Azienda TPL responsabile della gestione del servizio. *Nel caso di Lotti di Gara assegnati ad un'Azienda Unica ovviamente questo valore indicherà sempre l'Azienda Assegnataria del Lotto. In tutti gli altri casi [Lotti di Gara assegnati ad una Scarl] andrà invece indicata quale tra le diverse Aziende associate nel consorzio gestisce effettivamente il servizio.*
- **name** deve contenere la denominazione dell'Azienda. *Questo valore serve esclusivamente per migliorare la human-legibility del codice XML, quindi verrà ignorato durante le elaborazioni. Farà comunque fede il codice indicato da **code***

=====

La struttura completa del tag **<Subappalto ..>** è la seguente:

<Subappalto code=".." name="..">

- **code** deve contenere il codice numerico OTRT corrispondente all'Azienda TPL che effettivamente eroga il servizio. *Nel caso in cui il servizio sia affidato in subappalto andrà indicato il codice dell'Azienda subappaltante. In tutti gli altri casi [servizi erogati direttamente dall'Azienda senza ricorrere a subappalto] andrà invece ripetuto il codice dell'Azienda indicato nell'elemento **<Gestore ..>** di appartenenza.*
- **name** deve contenere la denominazione dell'Azienda. *Questo valore serve esclusivamente per migliorare la human-legibility del codice XML, quindi verrà ignorato durante le elaborazioni. Farà comunque fede il codice indicato da **code***

=====

La struttura completa del tag <Corsa ..> è la seguente:

```
<Corsa id=".." v=".." code=".." IdStd=".." parte=".."=".."
EnteCode=".." EnteName="..">
```

- **id** deve contenere un opportuno valore numerico univoco. *Preferibilmente verrà riportato l'ID univoco che identifica la corsa all'interno del DB Aziendale*
- **v** indica il verso di marcia della corsa. A indica il verso di andata, R indica il verso di ritorno. Nel caso di linee circolari può anche assumere convenzionalmente un valore '-'.
- **code** deve indicare il codice identificativo della corsa assegnato dall'Azienda. *Se l'Azienda non utilizza nessun sistema interno per la codifica delle corse può essere valorizzato a spazi*
- **IdStd** deve indicare il codice univoco della corsa standard di riferimento. *Quindi ciascuna corsa erediterà la sequenza delle fermate e la matrice oraria di marcia definite per la corsa standard di riferimento. N.B.* se la corsa standard corrispondente non è stata dichiarata in un elemento <CorsaStd ..> all'interno della sezione <Percorsi>, e quindi il riferimento relazionale non può essere soddisfatto, si verifica una condizione di **errore fatale**; conseguentemente l'intera comunicazione verrà respinta in quanto inutilizzabile.
- **parte** deve indicare l'orario di partenza della corsa. I tempi devono essere espressi come **HH:MM** *Quindi i valori legali per gli orari vanno da 00:00 fino a 23:59*
- **EnteCode** deve contenere il codice numerico OTRT corrispondente all'Ente intestatario del Contratto di Servizio cui appartiene la corsa. **N.B.** le corse appartenenti ad una medesima linea possono anche appartenere a Contratti di Servizio differenti; ed è anche legittimamente possibile che l'Ente intestatario del Contratto di Servizio differisca dall'Ente proprietario della linea
 - *Vedi note precedenti relative al tag <Ente ...>*
- **EnteName** deve contenere la denominazione dell'Ente intestatario del Contratto di Servizio cui appartiene la corsa. *Questo valore serve esclusivamente per migliorare la human-legibility del codice XML, quindi verrà ignorato durante le elaborazioni. Farà comunque fede il codice indicato da EnteCode*

3.q - I tags <Corsa ..> e <Periodo ..>

Ciascun elemento identificato dai tags <Corsa ..>..</Corsa> deve specificare almeno un periodo di effettuazione [calendario di servizio].

Ciascun singolo periodo è descritto all'interno di un tag <Periodo .. /> che ne specifica l'estensione temporale e la cadenza associata; quindi la struttura dell'elemento <Corsa ..> è:

```
<Corsa ..>
  <Periodo .. />
  <Periodo .. />
  ..
</Corsa>
```

=====

La struttura completa del tag <Periodo .. /> è la seguente:

```
<Periodo inizio=".." fine=".." code=".."
      excl="." />
```

- **inizio** deve indicare la data di inizio del periodo di validità della corsa. Le date devono sempre essere espresse come **GG/MM/AAAA**
- **fine** deve indicare la data di fine del periodo di validità della corsa. Le date devono sempre essere espresse come **GG/MM/AAAA** Per buona norma le comunicazioni devono essere consistenti, quindi devono valere le relazioni <DbcXml>inizio <= <Periodo>inizio e <DbcXml>fine >= <Periodo>fine
- **code** deve esprimere il codice Aziendale univoco assegnato alla cadenza. **N.B.** se la cadenza corrispondente non è stata dichiarata in un elemento <Cad ..> all'interno della sezione <Cadenze>, e quindi il riferimento relazionale non può essere soddisfatto, si verifica una condizione di **errore fatale**; conseguentemente l'intera comunicazione verrà respinta in quanto inutilizzabile.
- **excl** indica se si tratta di una fermata di esclusione contenuta all'interno di un periodo di effettuazione della corsa. Può assumere i valori S o N *Nel caso in cui risulti dichiarato un periodo di esclusione che non è compreso all'interno di un periodo di effettuazione, il periodo di esclusione verrà ignorato. Nel caso in cui invece l'estensione del periodo di esclusione sia solo parzialmente compresa all'interno di un periodo di effettuazione verranno esclusi solo i giorni realmente effettuati.* **N.B.** in ogni caso la presenza all'interno di una comunicazione di una corsa che non risulti mai effettuata neppure per un singolo giorno provocherà una condizione di **errore fatale**; conseguentemente l'intera comunicazione verrà respinta in quanto inutilizzabile.

Appendice A – codifiche e formati DbcXml

Le Comunicazioni nel formato DbcXml [sia di tipo Level 1 che di tipo Level 2] devono necessariamente essere prodotte utilizzando una codifica conforme al *character set* **ISO-8859-1** Latino Occidentale. Qualsiasi altra codifica verrà ritenuta non conforme [illegale] e quindi inaccettabile.

Per quanto concerne invece i formati di rappresentazione delle diverse categorie di dati, valgono le seguenti considerazioni generali:

- dati di categoria **alfanumerica**: devono essere rappresentati come stringhe letterali composte da una qualsiasi sequenza di caratteri appartenenti al *character set* **ISO-8859-1**. La lunghezza massima ammissibile è in ogni caso quella associata al dato corrispondente nei precedenti formati DbcTxt o DbcGis. Rappresentano esempi di dati alfanumerici:
 - o i codici Aziendali delle fermate, corse, rivendite, percorsi etc
 - o le descrizioni associate alle fermate, rivendite, linee, percorsi etc
- dati di categoria **numerica intera**: devono essere rappresentati come stringhe letterali composte da una qualsiasi sequenza di cifre [0-9]. La lunghezza massima ammissibile è in ogni caso quella associata al dato corrispondente nei precedenti formati DbcTxt o DbcGis. Rappresentano esempi di dati numerici interi:
 - o gli identificativi univoci assegnati alle corse, tratte etc
 - o il numero di vertici che compone una polilinea, il numero di minuti che indica una durata corsa etc
- dati di categoria **numerica con decimali**: devono essere rappresentati come stringhe letterali composte da una parte intera e da una parte frazionaria, separate da un carattere punto . Non è mai ammessa la presenza del segno positivo o negativo. Rappresentano esempi di dati numerici con decimali:
 - o le lunghezze in metri di tratte, percorsi etc
 - o le coordinate spaziale Gauss-Boaga che georeferenziano le fermate, nodi etc
- dati di categoria **numerica signed con decimali**: devono essere rappresentati come stringhe letterali composte da un segno + o -, una parte intera e da una parte frazionaria, separate da un carattere punto . Il segno positivo o negativo deve sempre precedere la parte intera. Rappresentano esempi di dati numerici signed con decimali:
 - o le coordinate differenziali assegnate ai vertici di polilinea successivo al primo
- dati di tipo **data**: devono sempre essere rappresentati nel formato GG/MM/AAAA
- dati di tipo **orario**: devono sempre essere rappresentati nel formato HH:MM

Appendice B – corrispondenze elementi tra DbcTxt / DbcGis e DbcXml

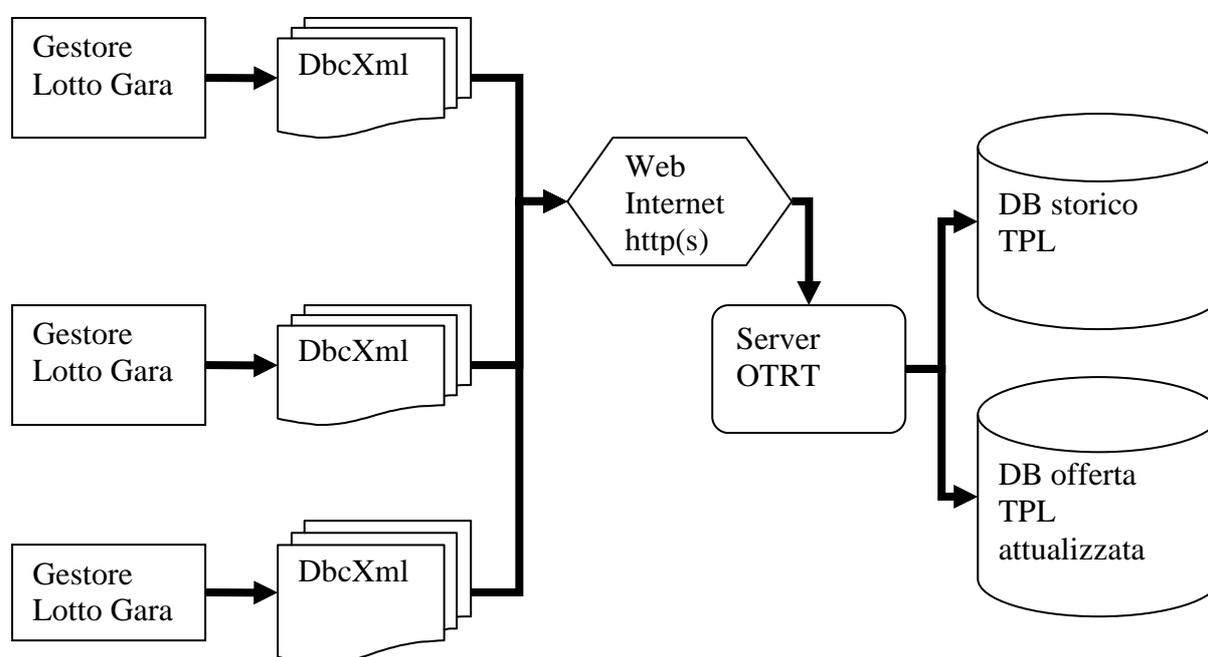
Flusso DbcTxt o DbcGis	Elemento	Elemento DbcXml	Annotazioni
RT_PROTO.txt	AZIENDA	<DbcXml azienda="..">	Gli attributi presenti in <DbcXml> corrispondono 1:1 con i campi di RT_PROTO
	DT_INVIO	<DbcXml data="..">	
	PROTOCOLLO	<DbcXml nro="..">	
	INIZIO	<DbcXml inizio="..">	
	FINE	<DbcXml fine="..">	
	RESP_LE	<DbcXml agente="..">	
RT_CADEN.txt	CADENZA	<Cad code="..">	Gli attributi di <Cad> corrispondono 1:1 ad RT_CADEN
	DENOM	<Cad name="..">	
RT_CALEN.txt	GIORNO	<Kal data="..">	Gli attributi di <Kal> corrispondono 1:1 ad RT_CALEN
	NOTE	<Kal note="..">	
	CADENZA	<Kal data="..">	
RT_HDORA.txt	PROG_CORSA	<Corsa id="..">	Mentre DbcTxt adottava un criterio ridondante ed assolutamente non normalizzato. DbcXml implementa una normalizzazione rigorosa, anche tramite l'identificazione delle corse standard implicite. Quindi la struttura è completamente rivoluzionata, anche se le informazioni sono sempre le medesime
	COD_CORSA	<Corsa code="..">	
	COD_ENTE	<Ente code="..">	
	LUNGHEZZA	<Itm metri="..">	
	TEMPO	<CorsaStd tempo="..">	
	LINEA	<AzLinea sigla="..">	
	VERSO	<Corsa v="..">	
	COD_PERC	<Itm code="..">	
DENOM	<Itm name="..">		
RT_EXTCOD.txt	LOTTO	<Lotto code="..">	Vedi punto precedente. DbcXml applica la normalizzazione estesa
	AZI_GES	<Gestore code="..">	
	AZI_SUB	<Subappalto code="..">	
RT_DTORA.txt	DETT_CORSA	<StdFmt sub="..">	Vedi punto precedente. DbcXml applica la normalizzazione estesa, ed utilizza l'identificazione implicita delle corse standard
	COD_FERMA	<Fmt code=".."> <StdFmt code="..">	
	DENOM	<Fmt name="..">	
	UBICAZ	<Fmt ubicaz="..">	
	DIST_PROG	<StdFmt metri="..">	
	ARRIVA	<StdFmt arriva="..">	
	PARTE	<StdFmt parte="..">	
	PRIMARIA	<StdFmt primaria="..">	
	FACOLT	<StdFmt facolt="..">	
NON_FERMA	<StdFmt nonferma="..">		
RT_PERIOD.txt	CADENZA	<Periodo code="..">	Gli attributi di <Periodo> corrispondono 1:1 ad RT_PERIOD
	INIZIO	<Periodo inizio="..">	
	FINE	<Periodo fine="..">	
	ESCLUSA	<Periodo excl="..">	
RT_NODI.shp	COD_FERMA	<Fmt code="..">	Gli shapefiles non servono più, dato che le geometrie sono descritte in XML in forma testuale, applicando anche la compressione geometrica
	Coordinate [x,y]	<Fmt x=".." y="..">	
RT_ITIN.shp	COD_PERC	<Itm code="..">	
	Coordinate polilinea	<Itm><Geom>...</Geom></Itm>	

Appendice C – trasmissione telematica delle comunicazioni DbcXml - logica interpretativa del server OTRT – criteri di aggiornamento del DB centralizzato dell'OTRT

L'adozione da parte dell'OTRT del formato DbcXml trova la sua principale motivazione nel fatto che rende possibile l'inoltro delle comunicazioni per via telematica e la loro successiva elaborazione in modalità automatica o semi-automatica.

In questo modo sarà finalmente possibile realizzare un contesto informatico tale da consentire il tempestivo e costante aggiornamento dei dati relativi all'offerta TPL; ovviamente il principale beneficiario di questa innovazione sarà il Portale Regionale della Mobilità, che finalmente potrà trovarsi in condizione di erogare un'informazione al pubblico affidabile, verificata e non obsoleta.

Quindi il ciclo complessivo di processo delle Comunicazioni tempestivamente fornite dai diversi gestori dei Lotti di Gara può essere rappresentato come segue:



La trasmissione telematica delle Comunicazioni:

Il processo inizia nel momento in cui l'Azienda TPL responsabile della gestione complessiva di un Lotto di Gara [quindi normalmente una Scarl] ritiene di dovere comunicare un nuovo piano di servizio all'OTRT.

Si dà per scontato che l'Azienda TPL disponga quantomeno di opportuna connettività Web/Internet [presumibilmente di tipo ADSL o superiore] e di un normale PC connesso al Web [postazione *client*]

L'Azienda TPL deve mettersi in condizione di generare un documento conforme alle specifiche del formato DbcXml tale da rappresentare l'intera rete gestita e l'offerta di servizio pianificata complessiva. Questo obiettivo può essere raggiunto indifferentemente tramite i seguenti percorsi:

- L'Azienda ritiene opportuno utilizzare una delle applicazioni SW che l'OTRT mette gratuitamente a disposizione [*DbcPlus* o *GaiaTpl*]
- L'Azienda utilizza per la gestione un SW gestionale prodotto internamente o acquistato da fornitori esterni ma che comunque è in grado di generare un flusso DbcXml conforme.

- L'Azienda utilizza internamente un SW gestionale che non è in grado di generare un flusso DbcXml, ma che è capace di generare i precedenti formati DbcTxt+DbcGis. In questo caso è possibile utilizzare ***DbcPlus*** o ***GaiaTpl*** per integrare i dati e quindi generare un flusso DbcXml.
- *In ogni caso non ha alcuna rilevanza il tipo di strumenti SW che l'Azienda ritiene opportuno utilizzare. L'unico fattore rilevante è che in un modo o nell'altro alla fine sia possibile generare un flusso DbcXml pienamente conforme alle specifiche del formato.*

A questo punto occorre trasferire tramite Web/Internet il file DbcXml che contiene la Comunicazione in modo tale che giunga a destinazione sul *server* dell'OTRT.

Per ottenere questo risultato basta semplicemente che la postazione *client* stabilisca una connessione con il servizio di ricezione presente sul *server* OTRT e quindi effettui l'*upload* del file DbcXml; per standardizzare al massimo la trasmissione il protocollo di trasmissione da utilizzare è HTTP [non sicuro] o preferibilmente HTTPS [sicuro, non intercettabile da terze parti in quanto criptato].

Per evitare interferenze indebite ciascun utente Aziendale autorizzato sarà dotato di una specifica autenticazione di accesso personale [username + password].

Quindi operativamente si procederà come segue:

- Se l'Azienda utilizza ***DbcPlus*** o ***GaiaTpl***, entrambe le applicazioni incorporano direttamente le funzionalità necessarie per procedere all'invio telematico della comunicazione DbcXml
- Se invece l'Azienda non intende utilizzare ***DbcPlus*** o ***GaiaTpl*** è comunque possibile effettuare l'*upload* della comunicazione DbcXml utilizzando un normale *browser* Web [Microsoft **Internet Explorer**, Mozilla **Firefox** etc etc]

Avvertenza: i formati basati su XML tendono a generare files estremamente pesanti e di rilevanti dimensioni. P.es. una *tipica* Comunicazione DbcXml Level 1 relativa ad un intero Lotto di Gara di dimensioni provinciali occupa tra i 50 ed i 100 MB.

Quindi i tempi di trasmissione [anche utilizzando una connessione a banda larga] risulterebbero insopportabilmente lunghi.

Fortunatamente XML risulta estremamente comprimibile, specie utilizzando algoritmi di compressione potenti ed evoluti; quindi lo standard adottato dall'OTRT prevede di utilizzare l'algoritmo di compressione LZMA per i flussi DbcXml trasferiti in *upload* sul *server* OTRT.

Si tratta di un algoritmo estremamente efficace [la *tipica* comunicazione di cui sopra si riduce ad 1-4 MB] e che ha l'enorme vantaggio di essere supportato dalla disponibilità di SW di tipo *open-source*, p.es. l'ormai largamente diffuso **7-Zip** ed SDK correlato.

Quindi per ottenere la compressione LZMA dei flussi DbcXml è possibile ricorrere ad una delle seguenti opzioni:

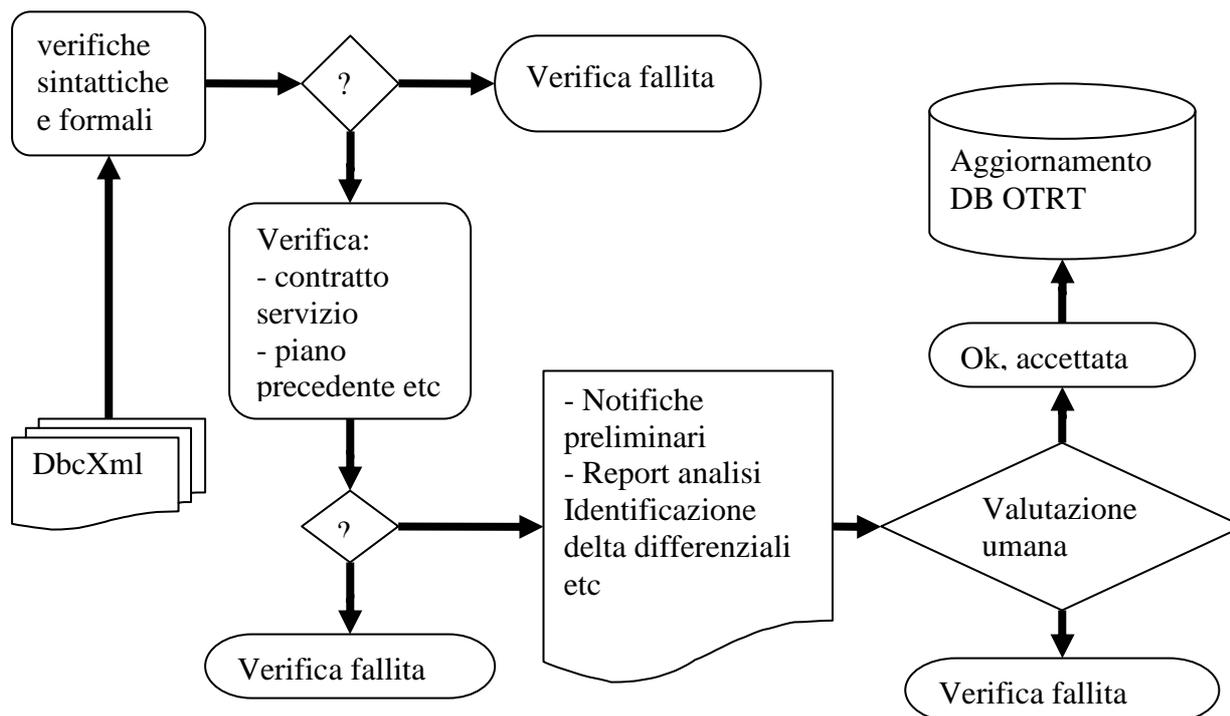
- Se l'Azienda utilizza ***DbcPlus*** o ***GaiaTpl***, entrambe le applicazioni incorporano direttamente le funzionalità necessarie per procedere all'invio telematico della comunicazione DbcXml già compressa secondo LZMA
- Se invece l'Azienda non intende utilizzare ***DbcPlus*** o ***GaiaTpl*** può comunque realizzare [tramite risorse interne o esterne] un compressore LZMA basato sulle librerie *open-source* dell'SDK.
- In ogni caso l'OTRT mette a disposizione [come risorsa gratuita] un compressore LZMA a riga di comando utilizzabile su piattaforme *Windows* nell'ambiente *Prompt dei comandi* / *Finestra MS-DOS*

La logica interpretativa del Server OTRT:

Non appena una Comunicazione DbcXml viene ricevuta dal *server* OTRT, si innesca automaticamente una concatenazione di eventi finalizzata a valutare l'accettabilità del materiale ricevuto.

Il processo non avrà caratteristiche di totale e cieca automaticità, dato che allo stato dell'arte pare decisamente prudente riservare uno spazio decisionale critico alla valutazione complessiva e di buon senso di un operatore umano dell'OTRT.

Comunque il processo di validazione semi-automatica effettuato dal server OTRT si svilupperà come segue:



Gli aspetti rilevanti sono i seguenti:

- In via preliminare verrà verificata la totale conformità del flusso ricevuto rispetto agli standards del formato DbcXml; con tutta evidenza si tratta essenzialmente di una verifica di tipo sintattico e quindi meramente formale.
- Quindi verrà verificata la consistenza interna dei dati sottoposti a vaglio; verrà verificata la consistenza dei riferimenti relazionali, dei vincoli spaziali e topologici etc. Anche in questo caso si tratta di una verifica esclusivamente formale.
- *Se la Comunicazione DbcXml non supera le fasi di verifica formale si intende direttamente respinta. In questo caso verrà automaticamente informato [via eMail] il mittente circa i motivi del rifiuto di accettazione. Contestualmente verranno informati in copia anche gli operatori dell'OTRT.*
- Altrimenti si procede ad una fase di verifiche di merito. Essenzialmente verrà valutata la consistenza quantitativa del nuovo Piano di Servizio rispetto al precedente, verrà valutata la consistenza rispetto agli standards previsti dal Contratto di Servizio etc.
- *Anche in questo caso se la verifica di merito fallisce la Comunicazione DbcXml si intende direttamente respinta. Quindi verranno informati sia il mittente che gli operatori OTRT.*
- Se invece le verifiche automatiche si concludono verificando l'apparente accettabilità della Comunicazione DbcXml, il processo si arresta temporaneamente in attesa di un consenso esplicito da parte di un operatore qualificato dell'OTRT.

- In ogni caso le verifiche preliminari generano automaticamente i seguenti elaborati:
 - o verrà informato il mittente circa l'apparente successo della verifica
 - o parallelamente verranno informati gli operatori OTRT
 - o verrà generato un report di valutazione essenzialmente incentrato sull'identificazione dei delta differenziali rispetto all'ultimo Piano di Servizio noto al sistema [in sostanza verranno evidenziate le modifiche apportate al Piano]

I criteri di aggiornamento del DB centralizzato dell'OTRT:

Il DB centralizzato gestito dall'OTRT sostanzialmente risulta articolato nei seguenti macro-blocchi funzionali:

- sottosistema **Contratti di Servizio**
 - o memorizza i parametri gestionali relativi ai diversi Contratti in essere. L'elemento qualificante consiste nel **Budget Contrattuale** che descrive analiticamente la struttura della rete TPL [in termini di cartografia delle fermate e dei percorsi] nonché la consistenza dell'offerta di trasporto [in termini di orari corsa e calendari di servizio]. In termini formali il *Budget Contrattuale* presenta esattamente la medesima struttura dati di un *Piano di Servizio*, però è profondamente diversa l'interpretazione funzionale. Infatti i *Piani di Servizio* riflettono la successiva dinamica evoluzione della rete TPL, così come si verrà configurando nel tempo, mentre invece il *Budget Contrattuale* è intrinsecamente un oggetto statico che rispecchia la situazione teorica consensualmente definita al momento della stipula.
- sottosistema **Storico**
 - o serve per l'archiviazione statica di tutti i successivi singoli *Piani di Servizio* che verranno via via comunicati. Non riveste alcun ruolo operativo diretto, in quanto serve piuttosto a garantire la rintracciabilità delle informazioni ed a ricostruire cronologicamente l'evoluzione dinamica delle reti TPL
- sottosistema **Piano di Servizio** attualizzato
 - o descrive l'intera offerta TPL attuale, valida per l'intera scala del territorio Regionale, ottenuta tramite integrazione delle varie Comunicazioni Aziendali al miglior livello di aggiornamento temporale correntemente disponibile. Di fatto costituisce il materiale primario per l'alimentazione del Portale Regionale della Mobilità.

Le modalità di interazione tra le Comunicazioni DbcXml fornite dalle Aziende TPL ed il DB centralizzato dell'OTRT dipende in larga misura dal tipo della Comunicazione, così come indicato dall'attributo `<DbcXml tipo = “..”>`

- se `<DbcXml tipo=”BUDGET”>` la comunicazione interagisce con il DB solo dopo esplicita autorizzazione e convalida da parte di un operatore OTRT. In questo caso andrà ad aggiornare il *Budget Contrattuale* e costituirà un vincolo di coerenza e consistenza per i *Piani di Servizio* successivamente comunicati. *Dato che un Budget Contrattuale è intrinsecamente un oggetto statico può essere comunicato una ed una sola volta.*
- se `<DbcXml tipo=”PLANNING”>` la comunicazione interagisce immediatamente con il DB, in quanto verrà direttamente registrata nella sezione *Storica* [anche se assumerà uno *stato sospeso*]. Solo dopo esplicita autorizzazione e convalida da parte di un operatore OTRT andrà ad aggiornare il *Piano di Servizio* attualizzato, e contestualmente la registrazione già avvenuta nella sezione *Storica* assumerà uno *stato esecutivo*.
- se `<DbcXml tipo=”PROJECT”>` la comunicazione interagisce immediatamente con il DB, in quanto verrà direttamente registrata nella sezione *Storica* ed assumerà uno *stato progettuale*. In nessun caso una comunicazione di progetto andrà ad aggiornare il *Piano di Servizio* attualizzato.

- Infine se **<DbcXml tipo="TEST">** oppure se la comunicazione viene respinta, non si avrà alcuna interazione con il DB. Resta comunque nella piena facoltà dall'OTRT conservare i flussi DbcXml comunque ricevuti, anche se di tipo non valido o irrilevante.