



FAKULTÄT  
FÜR INFORMATIK  
Faculty of Informatics

# **ebInterface AK Meeting**

**25. September 2013  
ebInterface 4.x Requirements**

Philipp Liegl

- Diskussion der Anforderungen für ebInterface 4.1
- 21 Anforderungen seit dem letzten AK am 4.9.2013
  - 14 erledigt bzw. Vorschläge erstellt
  - 7 Vorschläge müssen noch diskutiert werden
- Weitere Punkte
  - Reductions (Base Amount) – Hr. Axamit
  - Gutschriftsverfahren Hr. Bogad
  - ConsolidatorIDs
  - Potentielle neue Entität: Supplier und Regulierer

# Umgesetzte Anforderungen

Aktuelle Umsetzung:

```
<xs:complexType name="AddressIdentifierType" mixed="true">  
  <xs:attribute ref="AddressIdentifierType" use="optional"/>  
</xs:complexType>
```

"Sauberere Umsetzung":

```
<xs:complexType name="AddressIdentifierType">  
  <xs:simpleContent>  
    <xs:extension base="xs:string">  
      <xs:attribute ref="AddressIdentifierType" use="optional"/>  
    </xs:extension>  
  </xs:simpleContent>  
</xs:complexType>
```

- Wird wie im Vorschlag von Hr. Helger umgesetzt.

# Anforderung 5 – ebInterface Dokumentation in englischer Sprache

Hallo,

nicht wirklich eine technische Frage, aber ich wüsste kein passendes Forum dazu:

Gibt es eine englische Übersetzung der ebi 4.0 Spezifikation?

Auf der WKO-Seite ebInterface English ist zu lesen:

Zitat

The development of ebInterface continues, currently it is available in version 4.0. First steps of internationalisation on the technical level and on the organizational level (CEN/ISSS, NORMAPME) have already been taken.

Gibt es da bereits erste Ergebnisse?

Der Hintergrund meiner Frage: Ich benötige für ausländische Entwicklungspartner Spezifikationen in englischer Sprache. Ich gehe davon aus, dass die deutsche Version immer die normative Version sein wird - aber auch ein Entwurf in englischer Sprache wäre bereits hilfreich.

Mit freundlichen Grüßen,

Peter Gerstbach

- [First DRAFT ebInterface 4.0 – The Austrian e-Invoice standard](#)
- Wer Interesse an der englischen DRAFT Version hat -> Email an AustriaPRO
- Link wurde ins Forum gestellt
  - <http://www.ebinterface.org/forum/index.php?page=Thread&postID=564#post564>

Hallo,

wie wird in ebi-XML eine Zustellanweisung korrekt angegeben?

Beispielsweise "c/o" oder "zu Handen"?

Ist dafür AddressExtension geeignet (laut ebi: Zusätzlicher Name oder Zusatz zur Adresse als Freitext)?

Viele Grüße,

Peter Gerstbach



- Für c/o Information kann das Feld "Contact" in Address verwendet werden

## Anforderung 7: OrderID ohne regulären Ausdruck

Anbei möchte ich ihnen einen Änderungsvorschlag für die nächste ebInterface Version unterbreiten:

Das Feld „OrderID“ – also die Bestellreferenz – ist derzeit vom Typ „AlphaNumType“. Mein Vorschlag ist, diesen auf „xs:string“ zu ändern, um auch andere Zeichen wie z.B. „:“ oder „/“ unterbringen zu können!

Aufgefallen ist mir das deshalb weil wir eine Support-Anfrage bzgl. eines ungültigen Beispiels für die Auftragsreferenz des Bundes bekommen haben (auf [http://test.erb.gv.at/index.jsp?p=info\\_channel](http://test.erb.gv.at/index.jsp?p=info_channel)).

Dort wurde moniert, dass „/“ kein gültiges Zeichen ist.

Im Zuge der Recherche ist mir auch aufgefallen, dass unser Trennzeichen „:“ gar kein gültiger Teil der OrderID ist, unsere Implementierung sich aber nie darüber beschwert hat.

D.h. wir müssen unser Trennzeichen jetzt anpassen.

Der jetzige SimpleType für einen Alphanumerischen Wert:

```
<xs:simpleType name="AlphaNumType">
  <xs:restriction base="xs:string">
    <xs:pattern value="[0-9 | A-Z | a-z | -_äöüÄÖÜß]+"/>
  </xs:restriction>
</xs:simpleType>
<xs:simpleType name="AlphaNumIDType">
  <xs:restriction base="AlphaNumType">
    <xs:maxLength value="35"/>
  </xs:restriction>
</xs:simpleType>
```

ergibt leider wirklich keinen Sinn, da die Regular Expression nicht korrekt ist.

Den "Fehler" schleifen wir bereits seit Version 1.0 mit - aufgefallen ist er bisher noch nie, da eben "jede" OrderID gültig war.

```
<xs:simpleType name="AlphaNumType">  
  <xs:restriction base="xs:string">  
    <xs:pattern value="[0-9 | A-Z | a-z | -_äöüÄÖÜß]+"/>  
  </xs:restriction>  
</xs:simpleType>
```

AlphaNumType wird entfernt und durch xs:string ersetzt.

```
<xs:simpleType name="AlphaNumIDType">  
  <xs:restriction base="AlphaNumType">  
    <xs:maxLength value="35"/>  
  </xs:restriction>  
</xs:simpleType>
```

AlphaNumIDType wird in IDType umbenannt, basiert auf xs:string und  
maxLength=255

- Aktueller BIC Typ ist nicht korrekt (Regular Expression ist fehlerhaft)

```
<xs:simpleType name="BICType">  
  <xs:restriction base="xs:string">  
    <xs:pattern value="[0-9 | A-Z | a-z]{8}([0-9 | A-Z | a-z]{3})?"/>  
  </xs:restriction>  
</xs:simpleType>
```

```
<xs:simpleType name="BICType">  
  <xs:restriction base="xs:string">  
    <xs:whiteSpace value="collapse"/>  
    <xs:pattern value="[0-9A-Za-z]{8}([0-9A-Za-z]{3})?"/>  
  </xs:restriction>  
</xs:simpleType>
```

- Änderung wie vorgeschlagen umsetzen
- Info in die Doku was `<xs:whiteSpace value="collapse"/>` bedeutet

**-> `<xs:whiteSpace value="collapse"/>` ist nicht notwendig und kann weggelassen werden**

## Anforderung 10 – Sicherstellung der Berechnungsreihenfolge bei ReductionAndSurcharge

- Bei gemischter Angabe von Reductions and Surcharges – wie wird die Berechnungsreihenfolge sichergestellt?



- Aus der ebInterface 4.x Wunschliste:

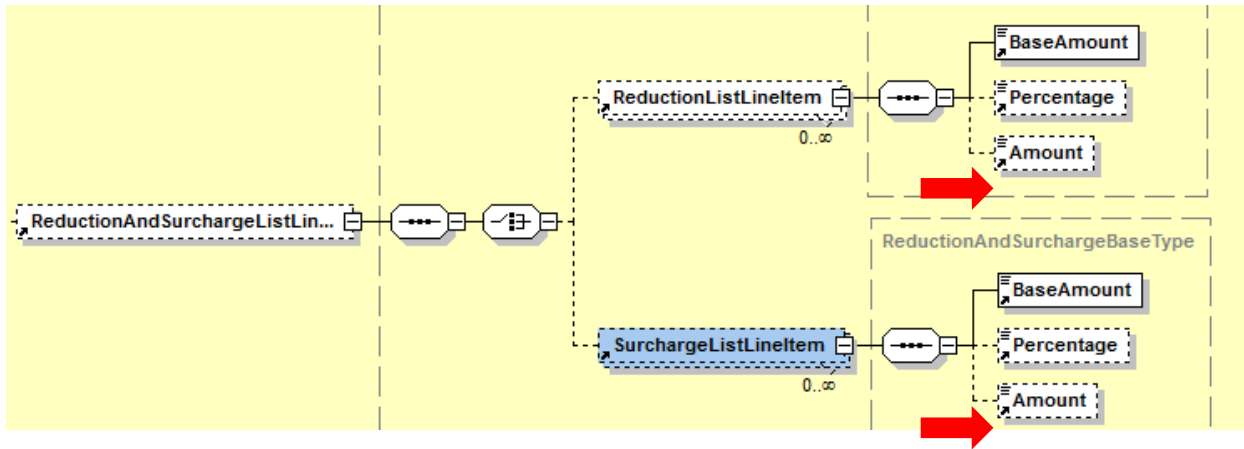
In der Dokumentation ist entsprechend zu vermerken, dass die Berechnung von Auf- und Abschlägen entsprechend der Reihenfolge zu erfolgen hat, in der Auf- und Abschläge im Element angegeben worden sind.

In Dokumentation aufgenommen:

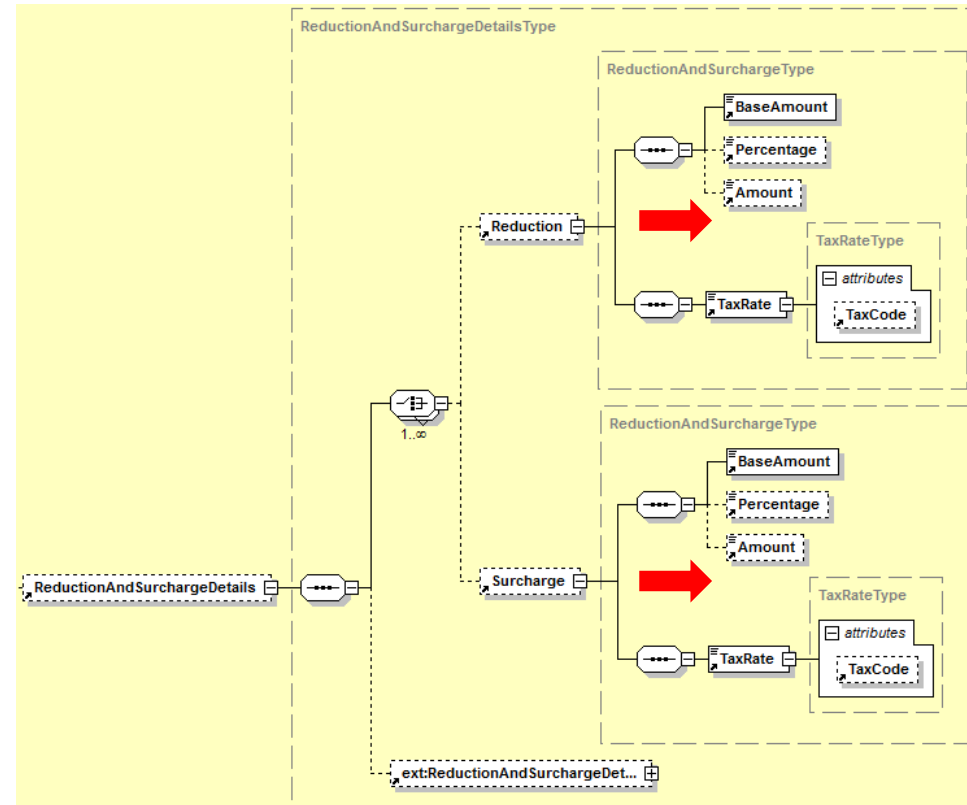
- $\text{LineItemAmount} = \text{Menge} \times \text{Preis} \pm \text{Zuschläge/Abschläge}$ .
- Die Berechnung innerhalb eines Surcharge/ReductionItems geht immer vom BaseAmount aus

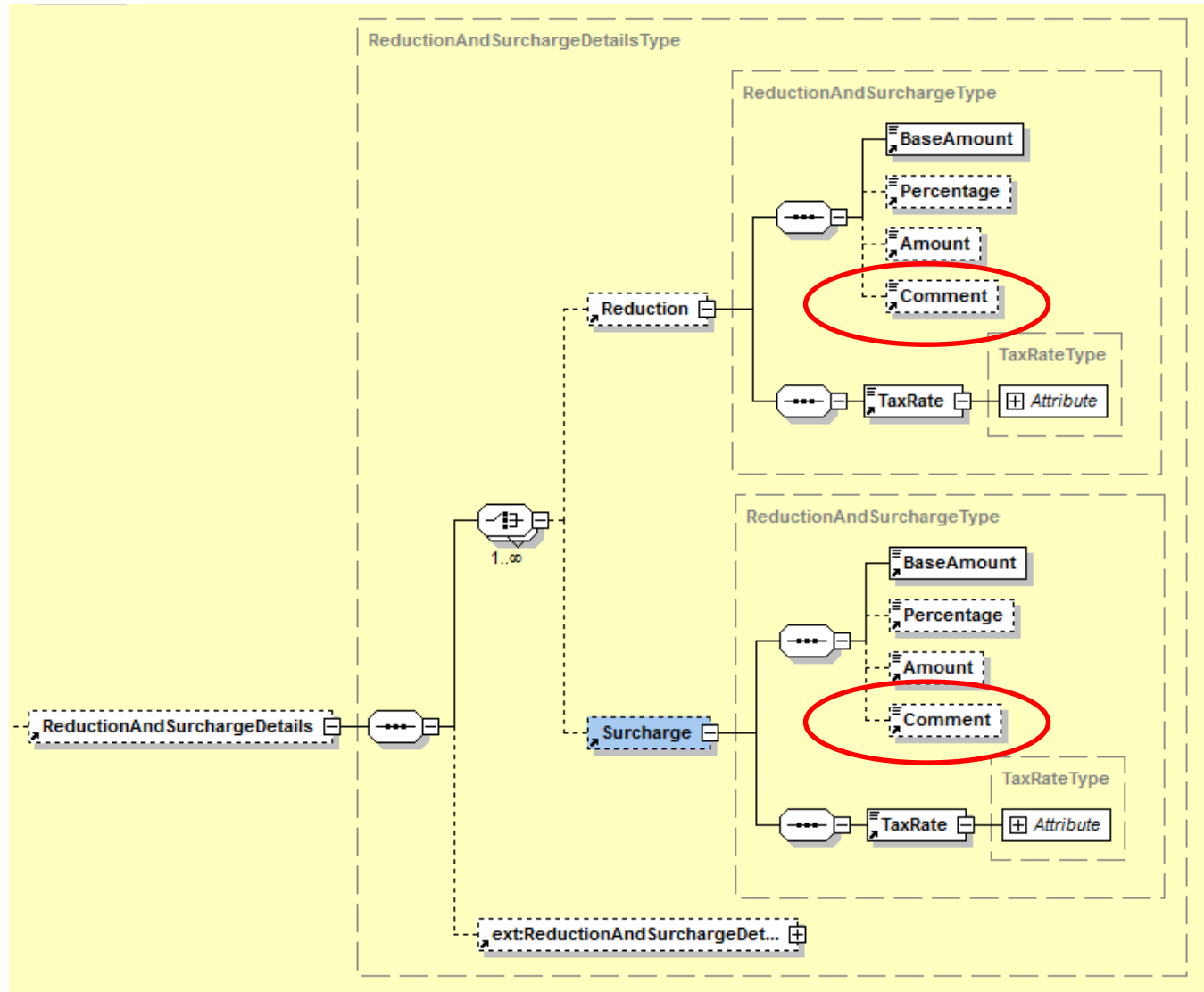
- Es wäre aus unserer Sicht wünschenswert, wenn es pro Reduction und Surcharge-Element (auf Lineltem und auf globaler Ebene) ein zusätzliches Optionales "Description" Feld vom Typ xs:string gibt, in dem die Gründe für Auf- oder Abschlag definiert werden können.

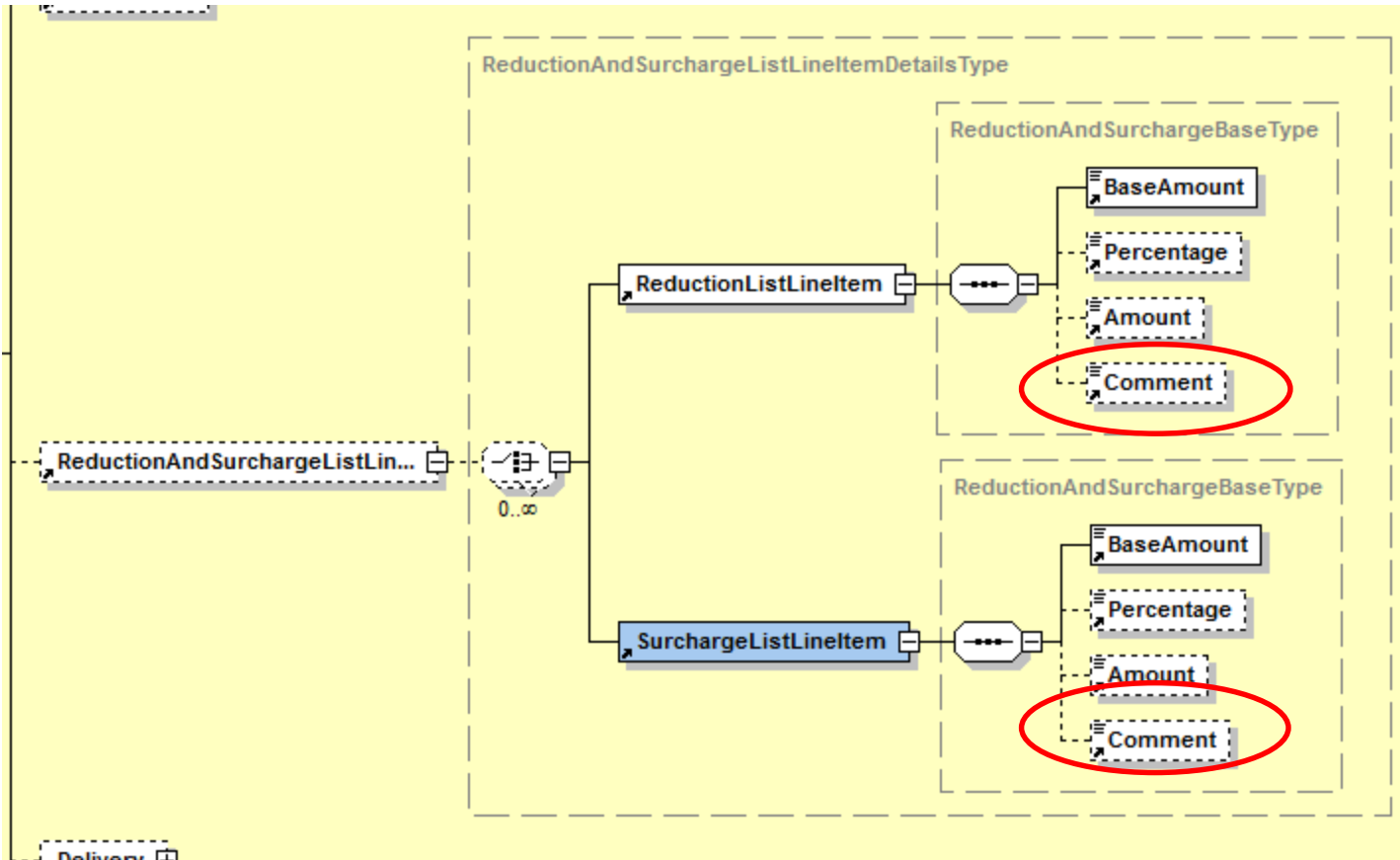
vg P. Helger



Optionales Feld "Comment" für Reductions und Surcharges auf globaler und ListLineItem Ebene aufnehmen.

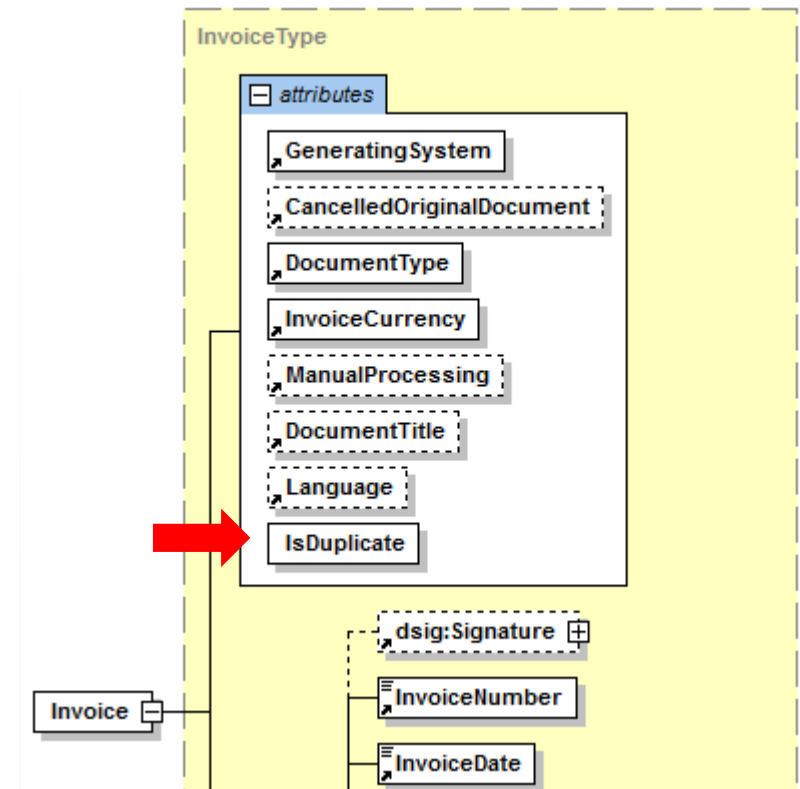






- Auszeichnung einer Rechnungs-Kopie
  - Möglichkeit zur Kennzeichnung einer ebInterface Rechnung als Kopie

- **Vorschlag:** Einführung eines Attributes auf ROOT Ebene
  - isDuplicate=true|false
- Vorschlag angenommen.



- Derzeit wird die Zahlungsart in ebInterface über das Element "PaymentMethod" abgewickelt. Um zu spezifizieren, um welchen Zahlungstyp es sich handelt (NoPayment, UniversalBankTransaction oder DirectDebit) muss ein relativ komplexes Konstrukt in der folgenden Form in einer XML-Datei angegeben werden:  
`<eb:PaymentMethod xsi:type="eb:UniversalBankTransactionType">...`

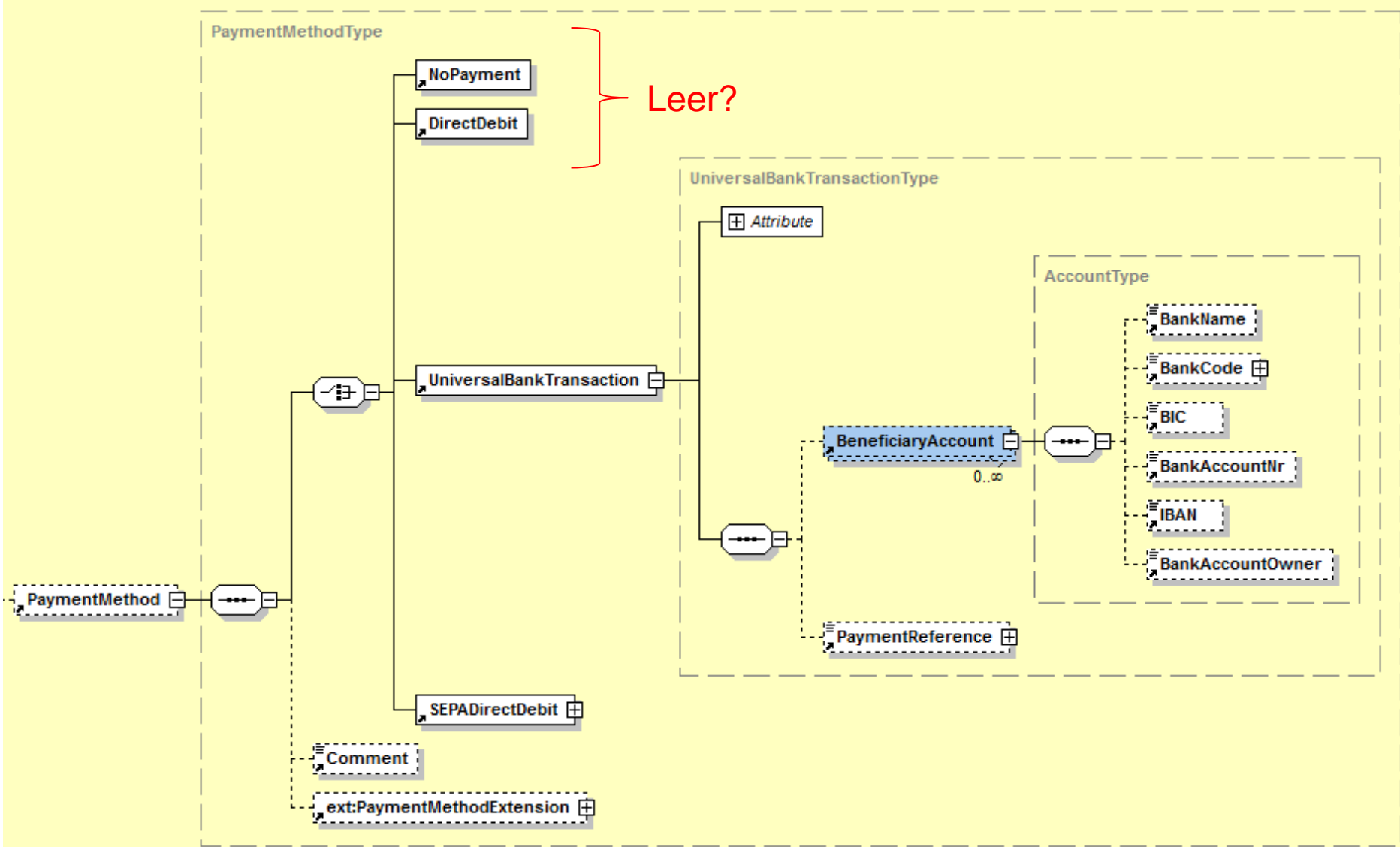
Meiner Ansicht nach wäre es einfacher (und schöner, und es gäbe besseren Tool-Support), wenn man die Spezifikation im XSD von  
`<xs:element ref="PaymentMethod" minOccurs="0"/>`

auf folgendes Konstrukt ändert:

```
<xs:choice minOccurs="0">
  <xs:element ref="NoPayment" />
  <xs:element ref="DirectDebit"/>
  <xs:element ref="UniversalBankTransaction"/>
</xs:choice>
```



- Vorschlag wird angenommen.



- Abbildung einer SEPA Überweisung in ebInterface
- Input BRZ (Hr. Helger)

Ich darf Ihnen wieder mal einen Erweiterungsvorschlag für ebInterface 4.0 zukommen lassen. Es geht dabei um eine neue Zahlungsmethode „SEPA Direct Debit“ (SEPA Lastschriftverfahren).

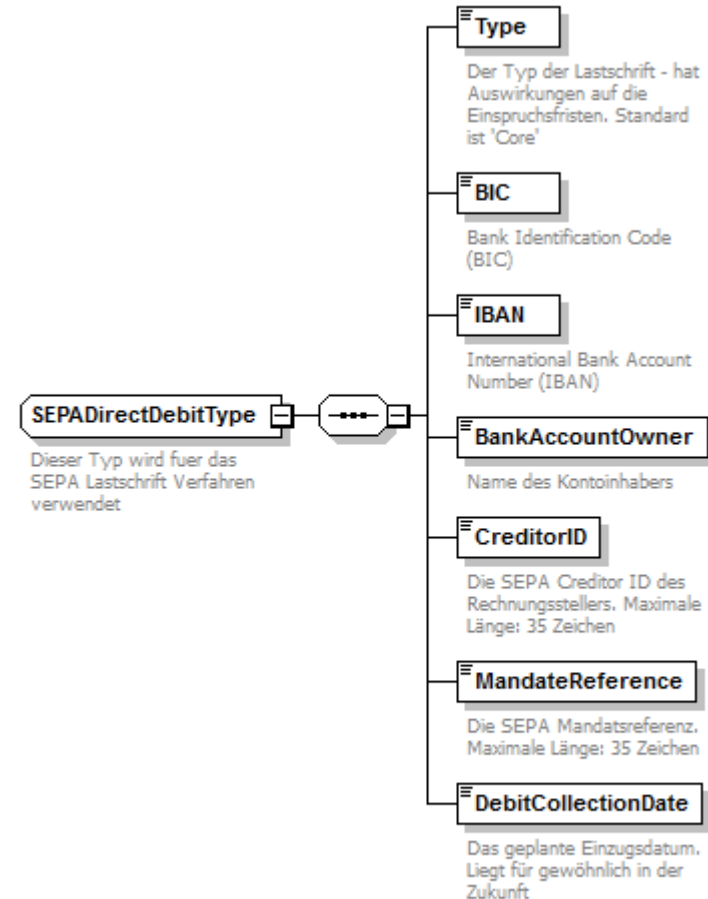
Davon gibt es 2 Typen: „Core“ und „B2B“ – diese unterscheiden sich hauptsächlich durch die Einspruchsfristen (Core: länger als bei B2B)

Die Felder sind: BIC, IBAN, Name des Kontoinhabers, SEPA Creditor ID, SEPA Mandatsreferenz und das geplante Einzugsdatum.

Ich kann Ihnen folgenden Pseudo-XML-Schema-Code für die Deklaration anbieten, wobei es für ebInterface leicht angepasst werden muss:

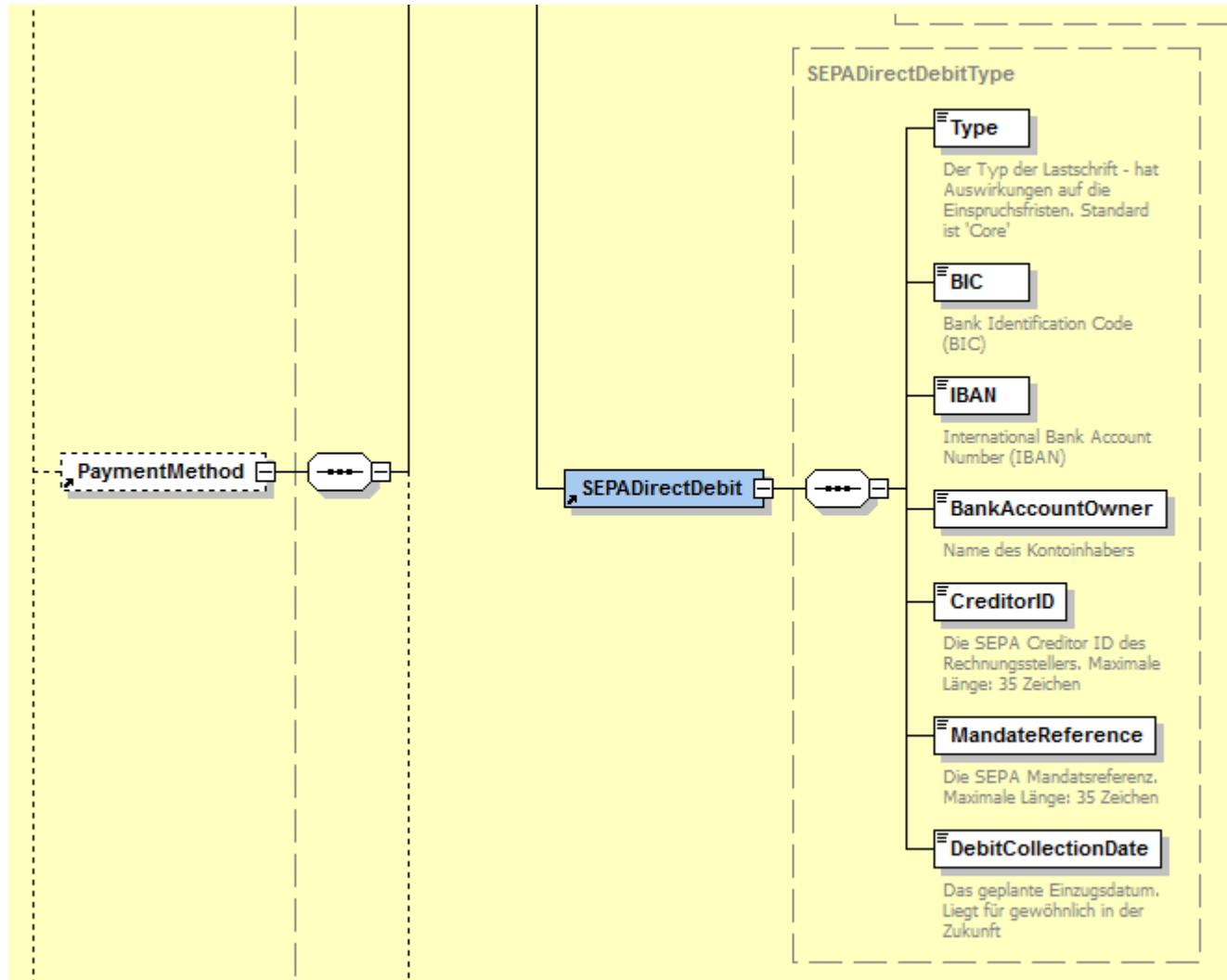
- Vorschlag BRZ:

In ebInterface Aufnahme als neues Element im Choice von PaymentMethod



Generated by XMLSpy

[www.altova.com](http://www.altova.com)



- BRZ klärt welche Felder verpflichtend sind.

Hallo,

ich hätte einen Änderungsvorschlag bezüglich des Elements <Address>

In ebi 4.0 ist das Element <Street> im Element <Address> verpflichtend, sollte auf minOccurs=0 geändert werden:  
Falls eine Adresse über ein Postfach angegeben wird, ersetzt dies die Angabe der Straße (zumindest bei der Österr. Post).

Viele Grüße,  
Peter Gerstbach

Aktuelle Implementierung:

```
<xs:complexType name="AddressType">
  <xs:sequence>
    <xs:element ref="AddressIdentifier" minOccurs="0" maxOccurs="unbounded"/>
    <xs:element ref="Salutation" minOccurs="0"/>
    <xs:element ref="Name"/>
    <xs:element ref="Street"/>
    <xs:element ref="POBox" minOccurs="0"/>
    <xs:element ref="Town"/>
    <xs:element ref="ZIP"/>
    <xs:element ref="Country"/>
    <xs:element ref="Phone" minOccurs="0"/>
    <xs:element ref="Email" minOccurs="0"/>
    <xs:element ref="Contact" minOccurs="0"/>
    <xs:element ref="AddressExtension" minOccurs="0" maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>
```

- **No action.** Der Leistungsempfänger und einer seiner Sitze muss ortsgenau spezifiziert sein (Postfach reicht nicht).



# Anforderung 16 – Längenbeschränkung bei SEPA-konformer Zahlungsreferenz

## PaymentMethod UniversalBankTransaction: SEPA-konforme Zahlungsreferenz

Das Element <PaymentReference> ist derzeit limitiert auf 12 Zeichen. Eine SEPA-Überweisung kann jedoch als Zahlungsreferenz bis zu 35 alphanumerische Zeichen enthalten!  
Die Prüfziffer kann aus bis zu 4 Ziffern bestehen.

Eine Beschreibung findet sich u.a. in folgender Datei, Seite 35:

[http://www.oenb.at/de/img/sepa\\_im\\_public\\_sector\\_tcm14-249338.pdf](http://www.oenb.at/de/img/sepa_im_public_sector_tcm14-249338.pdf)

Ist es überhaupt notwendig bei diesen Feldern die Länge zu spezifizieren?

Viele Grüße,  
Peter Gerstbach

```
<xs:complexType name="PaymentReferenceType">  
  <xs:simpleContent>  
    <xs:extension base="Digit12Type">  
      <xs:attribute ref="Checksum" use="optional"/>  
    </xs:extension>  
  </xs:simpleContent>  
</xs:complexType>
```

- ToDo TU Wien: Kompatibilität der UniversalBankTransaction zu SEPA herstellen.
- RegEx in CheckSum prüfen

- **Feldinhalte für Österreich**

- Empfänger (Name/Firma): 70 Zeichen
- IBAN (Empfänger): 20 Zeichen
- BIC (der Empfängerbank) : 11 Zeichen
- Betrag: 11 Zeichen (9 Vor-, 2 Nachkommastellen)
- Zahlungsreferenz: 35 Zeichen (alphanumerisch)
- Prüzfiffer: 4 Ziffern
- Verwendungszweck: 140 Zeichen (4 x 35 auf Zahlungsanweisung)
- IBAN (Kontoinhaber/Auftraggeber): 20 Zeichen
- Kontoinhaber/Auftraggeber (Name/Firma): 70 Zeichen

```

<xs:simpleType name="ChecksumType">
  <xs:restriction base="xs:string">
    <xs:pattern value="[0-9]{0,4}"/>
  </xs:restriction>
</xs:simpleType>

```

```

<xs:simpleType name="ID35Type">
  <xs:restriction base="xs:string">
    <xs:maxLength value="35"/>
  </xs:restriction>
</xs:simpleType>

```

```

<xs:complexType name="PaymentReferenceType">
  <xs:simpleContent>
    <xs:extension base="ID35Type">
      <xs:attribute ref="Checksum" use="optional"/>
    </xs:extension>
  </xs:simpleContent>
</xs:complexType>

```

## ToDo

- CheckSum anpassen, sodass auch alte CheckSums abbildbar sind.
- ID35Type auch bei SEPA nachziehen (wo notwendig)

- Lt. Diskussion zwischen Post und Bund
  - Zu Berechnungszwecken wäre ein Feld „LineItemGrossAmount“ hilfreich, welches den Gesamtbruttobetrag einer Rechnungszeile enthält

- **No action.** Wird nicht aufgenommen, da Abweichungen zwischen `LineItemGrossAmount` und den berechneten Werten ( $\text{Menge} \times \text{Preis} \pm \text{Zuschlag/Reduktion}$ ) zu erwarten sind.

## Rechnungs-Kommentar als eigenes Element

Ich fände es gut, wenn es ein optionales "Comment"-Feld auf oberster Ebene gäbe.

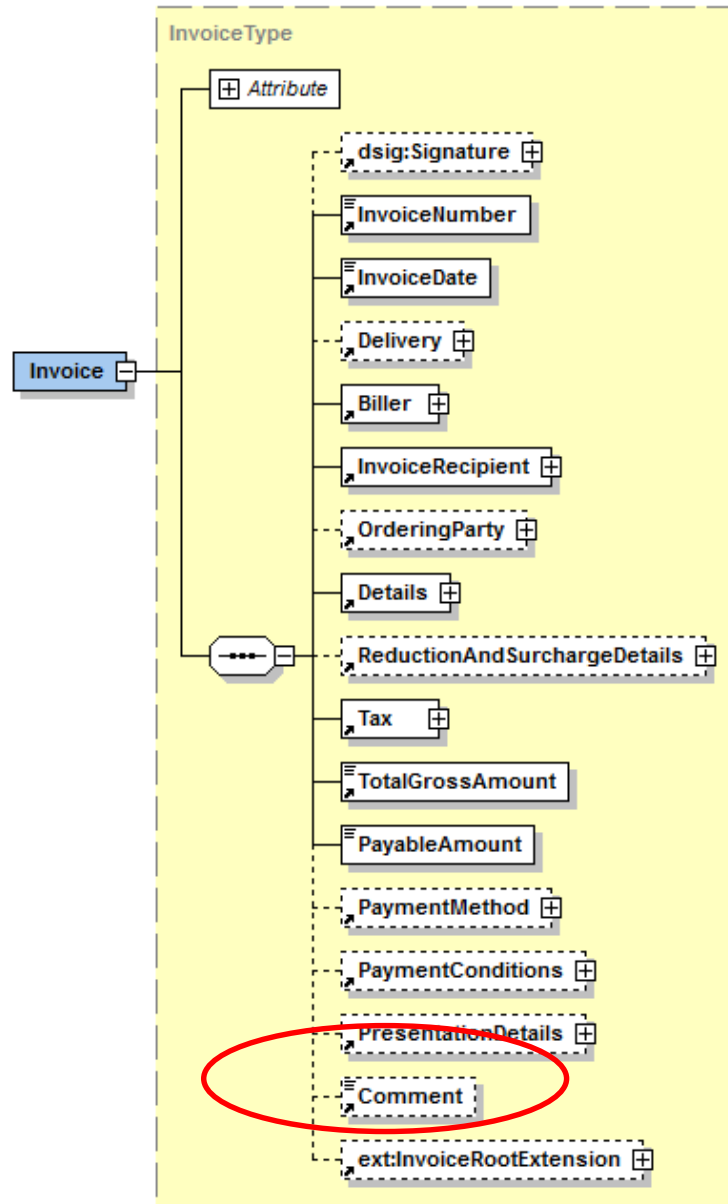
Dort könnten diverse (unkritische) Informationen zur Rechnung hinterlegt werden (z.B. die Beschreibung von Beilagen etc.)

Meiner Meinung nach könnte das existierende "Comment"-Element mit `minOccurs="0"` einfach zur Rechnung hinzugefügt werden.

vg P. Helger



- Auf Anforderung von BRZ und BBG: Aufnahme eines Comment-Felders auf ROOT Ebene vom typ xs:string.



# Offene Anforderungen

- [Codelistenvorschlag Philip Helger](#)
- [Codeliste ZUGFeRD](#)
  - Integration der ZUGFeRD Codes?

- Einführung einer neuen Spalte "Abkürzung wie in Österreich geläufig" – zB "kg" für Kilogramm, "mm" für Millimeter usw.
- ZUGFeRD Codes werden mit aufgenommen
- Codes sollen auch als Basis für die weiteren Dokumenttypen dienen
- Philip Helger veröffentlicht einer neue konsolidierte Version im Forum
- **Referenz auf die Codeliste in die Dokumentation aufnehmen. AustriaPRO übernimmt die weitere Wartung.**

- Welches Vorzeichen haben nach ebi-Standard die Beträge im Element "LineItemAmount", wenn der DocumentType "CreditMemo" ist?

S.g. Hr. Gerstbach!

Der Bund akzeptiert sowohl für Invoice als auch für CreditMemo nur positive Beträge in der Gesamtsumme.

Leider habe ich bei den Beispielrechnungen auf die Schnelle auch keine Beispiele für Gutschriften gefunden.

Wie bereits per E-Mail erklärt, wird seitens des Bundes an einer Lösung gearbeitet die zulässt, dass bei Verwendung von Einkäufergruppe bzw. Einkäufergruppe:SonstigeReferenz auch negative Beträge in einzelnen Teilen unterstützt werden, solange die Gesamtsumme positiv ist (sowohl für Invoice als auch für CreditMemo).

vg P. Helger

- Bei einer Gutschrift sind die Beträge auf Lineltems und der Endbetrag der Rechnung negativ
  - Auf Lineltem Ebene ist entweder der Preis oder die Menge negativ
- Die Aussage des Attributs DocumentType ist nur ein "zusätzliches Mascherl"
- Auch nachträgliche Präzisierung für ebInterface 3.x und 4.0
  - Entsprechende Updates in den Dokus

- CreditMemo -> positive Beträge (dann ist es eine Gutschrift) oder negative Beträge (dann ist es eine Rechnung) haben
- Invoice -> kann positive (dann ist es eine Rechnung) oder negative (dann ist es eine Gutschrift) Beträge haben
- SelfBilling -> positive Beträge (dann ist es eine Gutschrift) oder negative Beträge (dann ist es eine Rechnung) haben
- (Genauere Erklärung der Unterschiede zwischen den verschiedenen Typen in der Doku – SelfBilling stellt der Leistungsempfänger aus)



# Anforderung 4 – Optionale Elemente "Below the line"

Hallo,

In ebi 4.0 fehlen Elemente um Beträge "below-the-line" darzustellen.

Ein paar Beispiele:

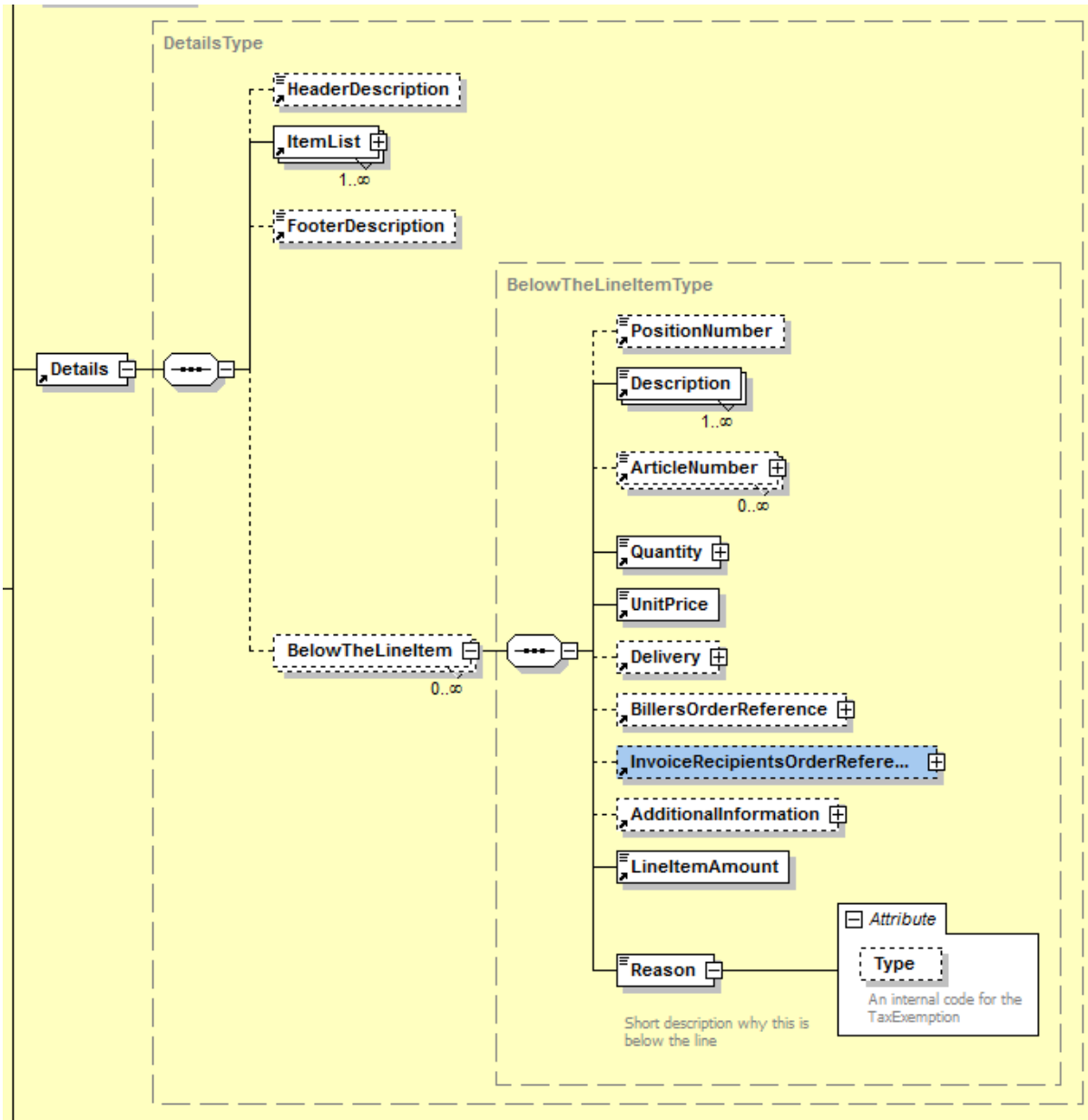
Leistungen Dritter: Diese dürften nicht als "normale" Rechnungspositionen mit 0% USt oder "nicht steuerbar angegeben werden, sondern nur below-the-line, also unterhalb des Rechnungsbruttobetrags.

Ein weiteres Beispiel sind Verzugszinsen, diese enthalten keine Steuern und dürfen ebenfalls nicht zu einem erhöhten Rechnungsbrutto führen, weil diese sonst bei erneuter Verzinsung zu Zinseszinsen führen würde.

Ein drittes Beispiel sind Guthaben, beispielsweise aus vorigen Rechnungen, die mit der aktuellen Rechnung gegengerechnet werden. Diese dürfen ebenfalls nicht zu einem (verringerten) Rechnungsbrutto führen, weil die Leistung ja bereits in Rechnung gestellt wurde. Dieser Fall ist also eher eine Saldo-Information und führt beim Rechnungsempfänger nicht zu einer Buchung aber müsste mit den internen durchzuführenden Adjustments übereinstimmen.

Die Summe aus dem Brutto-Rechnungsbetrag und diesen einzelnen Positionen sollte dann zu einem "zu zahlenden Betrag" führen.

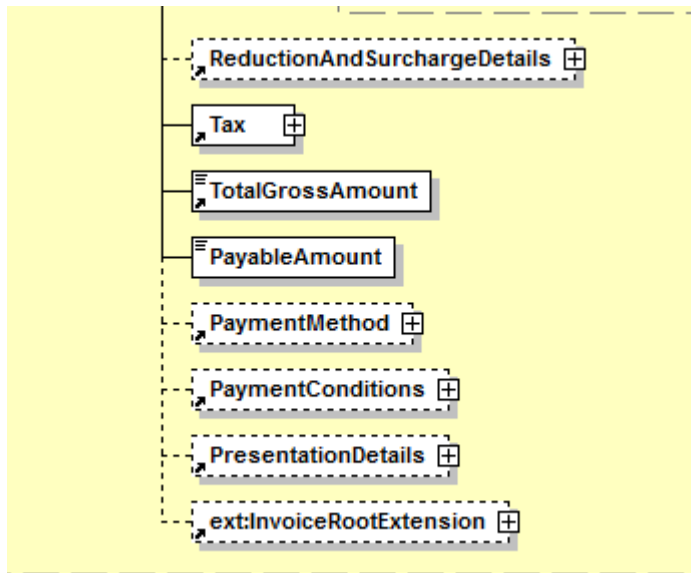
- Vorschlag für eine neue Struktur "Below the line" für die Abbildung von Nicht-rechnungsrelevanten Beträgen wie: Gutschriften, Verzugszinsen, Drittleistungen, usw.
- Alle Einträge in "Below the line" sind relevant für den Payable Amount
- Neues ROOT-Element (mandatory): PayableAmount
- Wenn keine Below the line Elemente vorhanden sind, dann ist der TotalGrossAmount = PayableAmount
- $\text{PayableAmount} = \text{TotalGrossAmount} + \text{SUM}(\text{Below the line items})$
- Neuer ComplexType auf Basis des jetzigen LineItemTypes (ev. ein gemeinsamer Supertyp)



- Felder für Below the line:
- Description (m)
  - Amount (m)
  - Reason[@Date(o)] (o)

# Anforderung 4 Vorschlag BRZ

- Payable Amount =  
TotalGrossAmount +  
SUM(BelowtheLineItems)



# Anforderung 9 – Limitierung von Auf- und Abschlägen

- Lt. Diskussion zwischen Post und Bund bezüglich Reduction und Surcharge (global und auf Item-Ebene):
  - Die Limitierung von nur Aufschlägen bzw. nur Abschlägen ist auf Dauer ein Hindernis. Es führt dazu, dass im Falle von einem Aufschlag und einem Abschlag stattdessen 2 Aufschläge (oder Abschläge) verwendet werden, wobei einer einen positiven Betrag hat, und einer einen negativen Betrag. Daher sollten diese Einträge auch mischbar sein. Dazu braucht es dann aber auch genaue Regeln, in welcher Reihenfolge die Einträge anzuwenden sind, da der BaseAmount meiner Ansicht nach hier nicht ausreicht!

- Bereits als Anforderung in der ebInterface 4.x Wunschliste aufgenommen

## < ReductionAndSurchargeListLinItemDetailsType >

- Erlaubt ist eine Liste von ReductionListLinItem oder SurchargeListLinItem  
- jedoch nicht beides gemischt

Aktueller Stand:

```
<xs:complexType name="ReductionAndSurchargeListLinItemDetailsType">
  <xs:sequence>
    <xs:choice>
      <xs:element ref="ReductionListLinItem" minOccurs="0" maxOccurs="unbounded"/>
      <xs:element ref="SurchargeListLinItem" minOccurs="0" maxOccurs="unbounded"/>
    </xs:choice>
  </xs:sequence>
</xs:complexType>
```

```
<ReductionAndSurchargeListLinItemDetails>
  <ReductionListLinItem>
    <BaseAmount>12</BaseAmount>
  </ReductionListLinItem>
  <ReductionListLinItem>
    <BaseAmount>12</BaseAmount>
  </ReductionListLinItem>
</ReductionAndSurchargeListLinItemDetails>
```

```
<ReductionAndSurchargeListLinItemDetails>
  <SurchargeListLinItem>
    <BaseAmount>12</BaseAmount>
  </SurchargeListLinItem>
  <SurchargeListLinItem>
    <BaseAmount>12</BaseAmount>
  </SurchargeListLinItem>
</ReductionAndSurchargeListLinItemDetails>
```

Eintrag im Schema  
nachziehen

## Vorschlag Anpassung von <ReductionAndSurchargeListLineItemDetailsType>

```
<xs:complexType name="ReductionAndSurchargeListLineItemDetailsType">
  <xs:choice maxOccurs="unbounded" minOccurs="0">
    <xs:element ref="ReductionListLineItem" />
    <xs:element ref="SurchargeListLineItem" />
  </xs:choice>
</xs:complexType>
```

Dadurch gemischte Angabe von Reductions und Surcharges auf ListLineItem Ebene möglich.

In die Dokumentation aufnehmen und klarer beschreiben:

Reduction = Subtraktion  
Surcharge = Addition

Beispiel für eine Invoice und ein CreditMemo aufnehmen und ins Forum stellen.

Hallo,

es gibt verschiedenste Sachverhalte, die in einer Rechnung in verschiedenen Positionen ausgewiesen werden und unterschiedlichste umsatzsteuerliche Konsequenzen haben (unterschiedliche Steuersätze, steuerbefreit, nicht steuerbar) und manche dieser Sachverhalte erfordern auch einen Hinweistext.

Laut ebi 4.0 sind Hinweistexte nur im Rahmen des Elements <TaxExemption> möglich, welches wiederum nur alternativ zu den <Item> Elementen in dem <VAT> Element steht.

Wenn jedoch auf einer Rechnung sowohl Lineltems mit sowie ohne Steuern (0%) vorkommen, ist es gesetzlich notwendig den Grund der Steuerbefreiung anzugeben. Auf Papier geschieht dies üblicherweise durch Fußnoten. In ebi sollte es diese Möglichkeit auf der Ebene des ListLineltem geben:

Beispielsweise wie folgt:

Zitat

```
<eb:ListLineltem>  
<eb:Description>Eine steuerfreie Leistung</eb:Description>  
<eb:Quantity eb:Unit="Stk.">1</eb:Quantity>  
<eb:UnitPrice>10.00</eb:UnitPrice>  
<eb:TaxRate>0.00</eb:TaxRate>  
<a1:TaxExemption>Steuerbefreiung gem § 6 UStG</a1:TaxExemption>  
<eb:LineltemAmount>10.00</eb:LineltemAmount>  
</eb:ListLineltem>
```

Je nach Fall sind auch unterschiedliche Hinweistexte notwendig, wodurch es nicht möglich ist mit einem einheitlichen Text alle Positionen zu beschreiben.



- Vorschlag:
  - Aus ListLineItemEbene ein xs:choice zwischen TaxRate und TaxExemption
  
  - Zu klären:
    - Brauchen wir bei TaxExemption auch einen Code in einem Attribut dazu (= Begründung der TaxExemption)
    - Wie ist die Darstellung der Steuersummenelement auf ROOT Ebene
  
    - GS1 stellte eine Codeliste für Steuerbefreiungs- und Steuerermässigungsgründe zur Verfügung
    - Darstellung des choices auf ROOT Ebene auch notwendig

- Break 25.09.2013

- Lt. Diskussion zwischen Post und Bund
  - Ein optionales Feld “BaseAmount” mit dem Standardwert “1” wäre hilfreich. Dieses kann verwendet werden, wenn Elemente in bestimmten Verpackungseinheiten (z.B: 1000) verkauft werden. Dadurch kann verhindert werden, dass sich der Einzelpreis nicht mehr mit 4 Dezimalstellen ausdrücken lässt.

- Menge: 5 NEP: 1,8€ => 9€
- Menge: 1000 NEP 0,0005 => 0,5€
- Menge: 10000 NEP **0,00005** => 0,5€
  
- BaseAmount: 1000 Menge: 10000; NEP => 0,05€ => 0,5€
- $0,05 / 1000 * 10000$

- Vorschlag für BaseAmount aufnehmen.
- **Wo aufnehmen?**

# Anforderung 21 – FurtherIdentification für InvoiceRecipient

## FurtherIdentification für InvoiceRecipient

Es gäbe meiner Meinung nach Bedarf für zusätzliche Identifizierungen des Rechnungsempfängers.

Derzeit gibt es neben der UID nur die BillersInvoiceRecipientID die laut Spezifikation für die Kundennummer (vom Rechnungsersteller vergeben) vorgesehen ist.

In der Praxis werden jedoch mehrere IDs dem Kunden zugeordnet, beispielsweise zusätzlich:

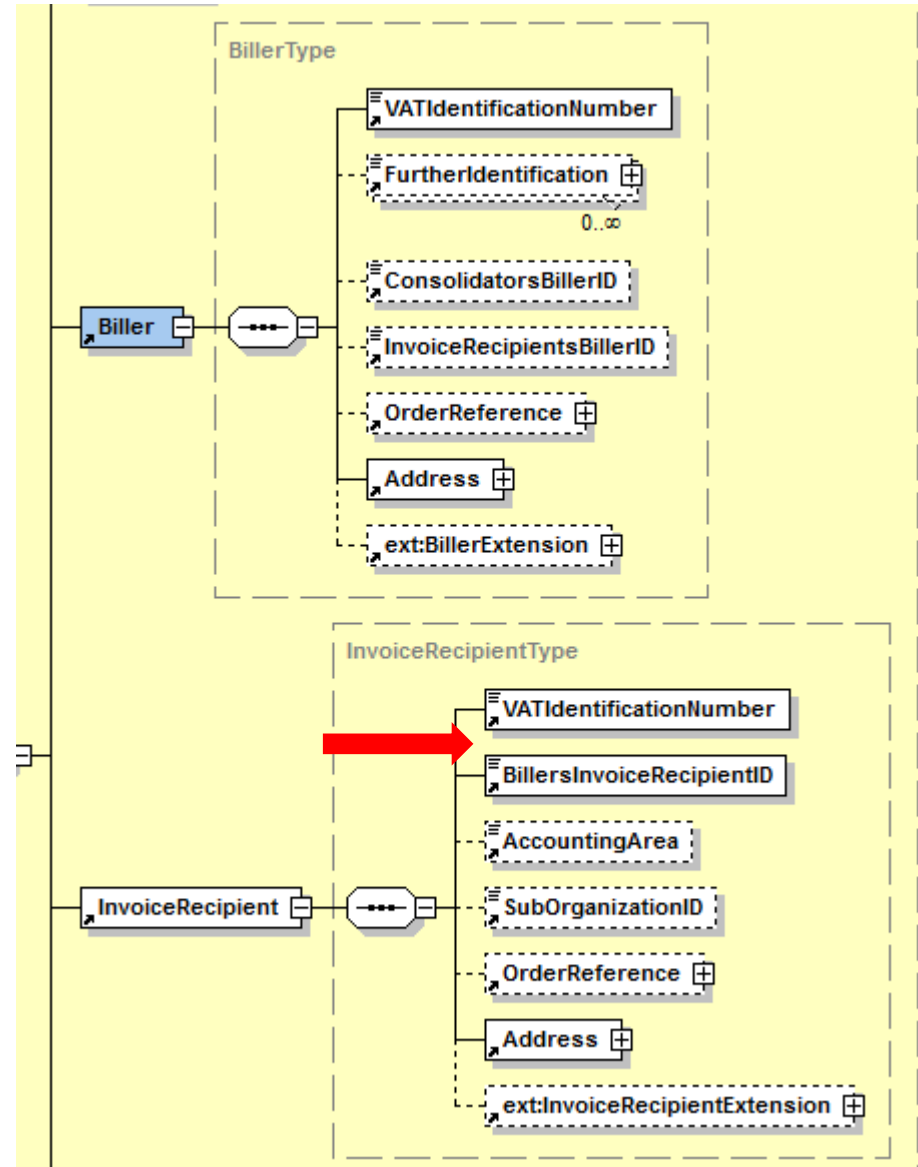
- Vertragsnummern
- Nummer, die ein Dienstleisters vergibt (z.B. Einkaufsgemeinschaft)

Der Fall 2. ist beispielsweise relevant für die BBG, wo ja jeder Kunde eine BBG Partnernummer hat. Mangels eines Feldes, schlägt die BBG vor, das Feld BillersInvoiceRecipientID für diese Information zu nutzen - wodurch aber wiederum die Kundennummer keinen Platz mehr hätte.

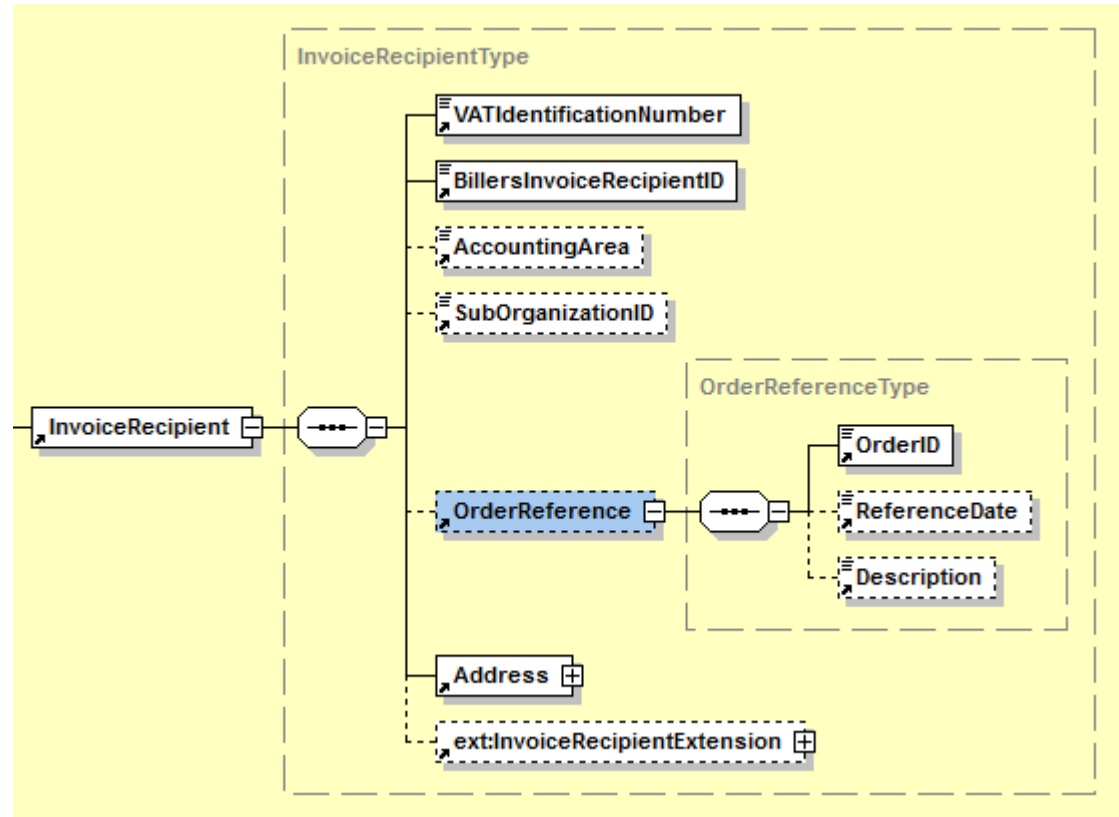
Fazit: Ich glaube wir brauchen mehr Elemente für IDs. Vorschlag: Nutzung einer flexiblen Lösung, analog zum Biller, mit "FurtherIdentification".

Viele Grüße,  
Peter Gerstbach

**Vorschlag:** Aufnahme von FurtherIdentification bei InvoiceRecipient



**Vorschlag:** Aufnahme eines eigenen ContractReference Elements in InvoiceRecipient, analog zu OrderReference





- FurtherIdentification wurden zu InvoiceRecipient aufgenommen
- Bezüglich ContractReference bei InvoiceRecipient bzw. auf ListLineItem-Ebene: Abwarten auf Feedback von Hr. Gerstbach (A1)

## Neue Anforderung 25.9.2013

- In die Doku aufnehmen wie man Reductions und Surcharges, Lineltem-Amounts usw. berrechnet