TRADE/CEFACT/1997/CRP.15
5 September 1997

Anti-Segment Collision Techniques

A proposal for the way forward

Source: UK Delegation
Status: Input paper
Action: For noting by GE.1 at the September 1997 session, agenda item 3, and review by the JRT

Anti-Segment Collision Techniques

0. Background

Segment Collision occurs if:

i) two or more identically tagged stand-alone segments appear in a message structure, with no mandatory segment with a different tag (or with no mandatory segment group triggered by a segment with a different tag) intervening between them;

ii) two or more identically tagged segments appear in a message structure at the same or at different levels within a segment group structure (or within dependent segment group structures), with no mandatory segment with a different tag (or with no mandatory segment group triggered by a segment with a different tag) intervening between them, both before and after the colliding segment.[1]

The problem arises because UN/EDIFACT messages make use of generic segments and it is often necessary to specify two segments of the same type at different places within the message. This is compounded by the fact that segment groups are not explicitly identified in transmitted messages.

One of the main ways that designers prevent segment collision is by specifying a 'dummy' segment (one that is not really required) into the message between the offending segments. This approach does not accord with the prevailing wisdom that messages should reflect business models more accurately and makes the task of mapping from an in-house application to a message unnecessarily complex.

The reason for preparing this consolidation paper is to take stock of all of the solutions that have been proposed for anti-segment collision techniques over the years in order to review the situation. This report is based therefore upon the research of other papers from a wide range of sources, all of which have been on the agendas of the SDG (Syntax Development Group); or UN/ECE WP.4/ GE.1; or EBES TAG (European Board for EDI Standards Technical Assessment Group), formerly known as the Western European TAG. However, some of the referenced papers may not have been circulated beyond the group to which they were input, as is customary.

1. Introduction

The development of an anti-segment collision technique has had a long and chequered history, and discussions have taken place within every UN/EDIFACT technical group to no avail. Several proposals have been forwarded but each met with the same fate - a split vote.

At a recent JRT the issue was raised again and an urgent solution was requested. From a timing perspective this request coincided with the fact that the new version of the syntax was undergoing ISO fast-track approval, so it did not seem possible to reach a solution without seriously impacting the publication time-frame of the new syntax.

On the assumption that, once published, it would be very difficult to update the new syntax with a major corrigenda, it is time to look again at the solutions that do *not* require syntax amendment. It should be noted that if it is possible to agree upon such an anti-collision technique then it follows that the technique may be used with

current translation software and the current syntax (version 3) also. This is a major advantage over any techniques that do require syntax change.

Out of the seven anti-segment techniques widely discussed by various UN/EDIFACT groups, there are only two solutions that fall into this category: the extended UNS solution and the Snn technique. The other five solutions are included in this paper for reference purposes in Appendix A.

2. The extended UNS solution

The proposal to extend the use of the UNS segment was made by the SDG[2] as their last attempt to provide an anti-segment collision technique. The idea is relatively straightforward: The UNS is already used to prevent segment collisions between segments in different sections of a sectionalised message, so if the restrictions on usage were removed, the UNS could be used elsewhere in messages to prevent segment collision. SDG recommended that the message design restriction of only two UNS segments per message be removed, and that the segment should not be restricted to specification at level 0 of a message.

2.1 Advantages

The main advantage of the extended UNS segment is familiarity of usage, since message designers already use the segment for sectionalised messages. Like all of the other solutions, it prevents segment collisions between segment groups.

2.2 Disadvantages

The disadvantages of the extended UNS solution are that it does not solve all types of collisions. It would also require detailed guidance to explain how and where to use the UNS within the message design guidelines. On a minor point, it would be necessary to modify slightly the definition of the UNS, but this could probably be accomplished without too much difficulty since a major corrigenda to the syntax would not be required.

3. The Snn technique

The Snn technique was proposed by a Joint Venture group of insurance associations[3] and was based upon a method that has been tested by them in practice in Europe.

The proposal was to use the Snn segment (see Appendix B for a re-specification of the segment) as the trigger segment of a segment group. It would be user defined at message design stage and used when necessary to avoid collision, or whenever the user prefers to use the facility systematically. The 'nn' characters in the segment tag would be an integer (such as '01' for S01 and '02' for S02) and would only be used once each within a message. The content of the segment, though not important for a syntax check, was to be used to provide the segment group number and description of the segment group for which the Snn segment was acting as a trigger.

Example

In the following segment table, a message is shown with a collision between the 'ABC' segments in segment group 1 and segment group 2.

```
ABC   M 1
DEF   M 5
Gr 1 ------------ M 5 ----------+
ABC   M 1                       |
 JKL  C 5                       |
Gr 2 -------------M 5 --------+ |
ABC   M 1                     | |
MNO   M 5                     | |
PQR   M 5 ---------------------+
```

The segment collision shown above could be prevented by using the Snn segments as follows:

```
ABC   M 1
DEF   M 5
Gr 1 ------------ M 5 ----------+
S01   M 1                       |
ABC   M 1                       |
 JKL  C 5                       |
Gr 2 -------------M 5 --------+ |
S02   M 1                     | |
ABC   M 1                     | |
MNO   M 5                     | |
PQR   M 5 ---------------------+
```

The method gained widespread user support within Europe, since it required no change to translation software. Within two regions, however, the technique did not gain approval for reasons relating to multiple functionality. (The use of the segment tag, Snn, was to prevent segment collision whereas the content of the Snn segment was used to specify and describe a segment group.) Later, the same technique was proposed as a message design solution by the Healthcare message design group and met with the same fate.

3.1 Advantages

The main advantage of the Snn technique is that it has been proven to run on existing translation software without the need for modification. Like some of the explicit techniques proposed by other groups, it could help to provide the location of errors in anomaly reports and there is no need for the translation software to 'look into' the data of the segment. In addition, the technique has one advantage that is unique amongst all of the proposals made to UN/EDIFACT, yet seems to have been largely overlooked: *It would provide the possibility to release user segments from the role of trigger segment.* In many cases, for example when two closely related segments are grouped together, it is not always possible to specify universally which segment is the 'key' to the segment group because business practices differ within some user communities - in the case where a name and a reference are to be provided it is not always possible to identify which should be the trigger. Message developers get around this problem by specifying a different segment to act as a 'dummy' trigger segment - hardly a standardisation approach. Another important advantage is that the technique could be presented as a message design solution to be employed at the message design stage, thus requiring no change to syntax version 4.

3.2 Disadvantages

Technically speaking, a segment tag has never before been used to perform a stated function. In the case of the 'Snn' segment, the tag is undoubtedly used to prevent segment collision - a function that it performs well. In fact the segment tag performs the function so well that any data contained within the segment specification (i.e. the data elements) may be considered as extraneous. This led to the conclusion within two regions that the segment, since it *did* contain data elements, had to be rejected because of multiple functionality.

However, it may be possible to re-address the issue of multiple functionality by suggesting that the 'nn' in the Snn segment tag has to be independent of the segment group number to which it acts as trigger. The reason for this is ease of maintenance: If a new segment group is subsequently added to an existing message specification, resulting in an incremental change to other segment groups, it would not be necessary to amend the 'nn' of any groups headed by Snn. If this principle is accepted, then one could argue that an intrinsic requirement of a segment collision avoidance technique is to identify the location in the message where it is used so that any error reports can locate the problem immediately. Therefore it would be useful to design the Snn segment to contain a single data element which identified the segment group number for which the Snn acts as trigger. If necessary, the definition of the segment could stipulate that it is for use as a trigger segment - the fact that it also happens to prevent segment collision is a useful by-product. Fundamentally, this approach - a single stand-alone data element to identify the segment group number - is similar to that of the UNX and UNY segment proposal. However, as discussed above, the Snn technique, unlike that of UNX and UNY, does not require modifications to translation software.

Another apparent reason for the rejection of the Snn technique (though source documents quoting this as a reason have proved difficult to find) was that its specification within the data segments directory would be unorthodox. To overcome this, a note would be required to explain that the message developer, at design stage, had to allocate the integer 'nn' whenever the segment was to be specified in a message. (Snn would be shown in the segment directory, whereas S01, S02, S03 etc. would be shown in the message specification.)

4. The way forward

The fact of the matter is that all of the solutions presented within this paper (including those in Appendix A) will work to some degree or another. The only way of choosing between the solutions is to have some objective criteria by which to select the most appropriate. The technical intricacies of one anti-collision solution versus another have been discussed *ad infinitum* within the UN/EDIFACT process since at least 1989. It is now time to decide whether it is possible to agree on a solution that we can all live with, even if it is not the one that is considered by everyone to be the perfect choice.

It is not possible at this stage to incorporate an anti-segment collision technique into the syntax without significant change. Therefore, the way forward should be to choose a technique that is independent from the syntax and has the added value of being available for use with today's translation software packages.

Upon comparison of the two solutions that meet this criteria, it is the Snn technique that takes precedence over the extended UNS technique. The Snn technique has been proven to work in practice, it does not require any syntax changes, the translation software does not need to look at the data and, as an excellent bonus, it removes the *requirement* always to use traditional user segments (such as RFF,

LIN, NAD etc.) as trigger segments when often it is inconvenient to do so. The technique, like all of the others, is of course an optional feature: The Snn segment, like any other user segment in the UN/EDIFACT directory, is to be used at the discretion of the message developer, so no existing message needs to be changed unless it is modified by users for a business reason and no new message will include the Snn segment unless there is a user requirement. The only determinable disadvantage is that the Snn technique is unconventional, but this can surely be solved through well-worded guidelines.

Appendix A

Anti-segment Collision Techniques requiring significant modifications to Syntax version 4.

There have been a number of solutions for preventing segment collision proposed in the past, as shown in the following list. Each of the solutions is explained in more detail below and have the common factor that they would require significant change to the syntax and would not therefore be available for use with current translators.

I. The UNX/UNY envelope solution
II. Segment number solution
III. Explicit techniques
IV. Segment number and tag solution (for colliding segments only)
V. The right parenthesis solution

Note that although one or two other solutions were also suggested, they never really appeared as detailed proposals but rather as ideas resulting from 'brainstorm' discussions. One such idea suggested combining the essence of the Snn technique with that of UNX and UNY to provide an Snn envelope to surround segment groups. Since these suggestions never really made it further than the white-board, they are not covered within this paper.

A.1 The UNX/UNY envelope solution

The UNX/UNY envelope solution originates from the PAEB region[4] and is based on a similar technique within the X12 syntax (i.e. the LS and LE segments). It can therefore be argued that it has been tested in practice.

The UNX and UNY segment was to be used in pairs to surround segment groups, with the presence of UNX indicating segment group start and the UNY indicating segment group end. The single data element contained in each of the two segments would be used to specify the segment group number being enveloped. It was proposed that a special segment designation - S (for Syntax) - be assigned to the UNX and UNY segments, allowing them to be used if and only if the offending conditional segment group is transmitted.

Example

The following example shows that in order to prevent segment collision between the two ABC segments, the UNX and UNY segments are placed around segment group 10.

```
...

ABC  C 1
DEF   C 5
UNX  S 1
------   Grp 10   -----   C 5 -----+
ABC  M 1                           |
JKL    C 5                         |
MNO  C 5 ---------------------- +
UNY   S 1
PQR    C 5
...
```

A.1.1 Advantages

The main advantage of the UNX/UNY solution is that, being a derivative of an X12 solution, it has been proven to work in practice.

A.1.2 Disadvantages

The technique is quite heavy in transmission and cannot be used to prevent segment collisions between stand-alone segments. The introduction of a new segment status of 'syntactic' was also seen as disadvantageous.

A.2 Segment number solution

This solution was proposed by the French delegation[5] in response to the UNX/UNY solution. Instead of transmitting the segment tag, the position (rank) of the segment in the message definition was to be transmitted instead. As an exception to the mechanism, it was suggested to keep the segment tags for service segments, thereby easing syntactical checks and the interface with other standards like X.435 (Pedi).

Example

For the message:
Header section
AAA C 10
BBB C 5
Detail section
Segment Group 1 C 999
...
Summary section
AAA C 5

Instead of sending:
AAA+...'AAA+...'BBB+...'AAA+...'

one would send:
1+...'1+...'2+...'4+...'

A.2.1 Advantages

The main advantages of this proposal are that it prevents all types of segment collision; it is a syntactical solution (translators do not need to 'look' at data); and, it is possible for anomaly reports to identify segments in error without ambiguity.

A.2.2 Disadvantages

The main disadvantage is that significant changes would be required of translation software. Furthermore, it was pointed out that: 'There is no clear connection between message definitions and transmissions, so it seems impossible to determine whether the segment being checked is the correct segment (i.e. that segment '42' is a 'LIN' segment, as it should be, and not an 'MOA' segment).[6]' It was also noted that the solution may cause compatibility problems[7]. In the case where a segment is inserted in a message, from one version to another, the number of a

segment may change. This means that, from a user point of view, the message will change even if the new segment is not used.

A.3 Explicit techniques

This solution combines solutions from PAEB and from France[8] and was put forward in a proposal by the SDG as an alternative to the UNX/UNY proposal.

The following text is an extract from the SDG meeting report:

'The proposed technique is to define, in the syntax, the segment tag as being a composite data element, in which the first component is mandatory (the segment tag), and the second component is conditional (to which a unique identification is assigned by message designers for each colliding segment). The composite structure would be as follows:

Sxxx	SEGMENT IDENTIFICATION			
	0013	Segment tag	an3	M
	nnnn	Segment tag extension	an..3	C

Note: A composite already exists in the current syntax for explicit nesting, which will be modified for the purpose.

For any segment not the subject of a collision, the normal syntax rules for exclusion of a component in a composite will apply, in which case, only the segment tag will be transferred for this composite data element.

For two or more identically tagged segments being the subject of collision, the designers of the message will be responsible for assigning the unique identification to each of these segments in the message specification.

It is recommended that these unique identifications be assigned independently of the segments' position within the message structure.'

A.3.1 Advantages

The advantages of this technique over the UNX/UNY is that it is not necessary for translation software to 'look into' the data of the segment. Also, in the case of a error reporting, the exact positioning of the segments is given.

A.3.2 Disadvantages

The disadvantages are that, firstly, translators would need to be modified in a non upwardly compatible way; and, secondly, the overheads are large if the technique is used in conjunction with segment group trigger segments that can repeat a large number of times.

A.4 Segment number and tag solution (for colliding segments only)

This proposal, submitted by Norway (see reference 6), is similar to the explicit techniques above, but for one difference. In the message definition, add the segment position to all segments where the potential for collision exists (but only then) as follows:

"Segment tag": "Segment position" (i.e. AAA:42)

It was suggested that ":" can be used as a separator, as explicit looping is to be removed from the future syntax.

One other distinction was suggested for this technique: It was recommended that segment positions should not be changed if new segments are added to a message, thus the segments need not always be sequentially numbered.

The following example has segments 'BBB' in collision. The segment positions are 5 and 11. Segment 11 is in subgroup 'CCC'. Both 'BBB's are conditional but only group 5 is repeatable.

AAA+...'BBB:5+...'CCC+...'BBB:11+...'BBB:5+...'

A.4.1 Advantages

No requirement for conversion software to look at the data. It is only used in cases where there are segments in collision, therefore it is an economical solution in terms of data transmission.

A.4.2 Disadvantages

The solution would require modifications to the way in which translation software operates, so it is not upwardly compatible from a syntax perspective.

A.5 The right parenthesis solution

The 'right parenthesis' solution was proposed by Sweden[9] and gained favour within several message design groups for its cost-effectiveness and elegance. The proposal suggests that a special service character, such as ')' be used as a 'group terminator' replacing the apostrophe in the very last segment interchanged in each segment group. For nested segment groups, the technique would allow for more than one right parenthesis to appear together following the last segment in the final segment group.

Example

```
010     Group 1 ---------- C 10 ------------+
020     ABC   M 1                           |
030     Group 2 ---------- C 10 ---------+  |
040     DEF   M 1                        |  |
050     Group 3 -----------C 10 -----+   |  |
060     GHI    M 1                   |   |  |
070     JKL    C 9 -------------------+   |  |
080     JKL    C 3 ----------------------+--+
090     DEF    C 3
100     GHI    C 3
110     JKL    C 3
120     XXX    M 1
```

For one occurrence each of group 1, group 2 (with no JKL position 080) and group 3 (with 2 occurrences of JKL position 070), the message would be transmitted as: ABC+...'DEF+...'GHI+...)JKL+...))XXX+...'....

A.5.1 Advantages

The right parenthesis technique has the advantage of low data transmission and hence cost effectiveness. Since there is a character to end segment data (the apostrophe) and to separate one data element from another, the right parenthesis solution also seems to follow the logical order of EDIFACT. In addition, the generation of the character can be left entirely to the syntax software.

A.5.2 Disadvantages

The disadvantage of the right parenthesis technique is that a change would be required of the syntax and of translation software.

Appendix B - Re-specification of the Snn Segment

The Snn segment could be presented as a relatively simple specification, using only one data element to identify the number of the segment group to which the Snn segment is acting as trigger. The important consideration is the note that should accompany the segment in the directory.

Snn SEGMENT GROUP TRIGGER

Function: To identify uniquely a segment group, thereby preventing segment collision.

Note: The 'nn' in the segment tag shall be replaced with a number, incremented by one each time the segment is specified within a single message e.g. S01, S02. For maintenance reasons, the number shall not represent the number of the segment group to which the segment acts as trigger.

9999	Segment Group Number	M	n..3
Desc:	To identify the number of the Segment Group.		

Appendix C - References

[1] Definition extracted from SDG/49 Com.2 Rev.1. Author: Don Trafford, SDG Chair
[2] SDG Executive Summary, Meeting 14 - London 9-13 January 1995. Reference: Trade/WP.4/R.1111
[3] BRMA/RAA - LIMNET - RINET proposal for segment collision problem. Reference: TAG47.224
[4] SDG/49 Add.1 Rev.2. Author: K-D Naujok
[5] Another syntactical solution to the collision problem. Trade/WP.4/R.1024
[6] Comments submitted by Norsk EDIPRO. Reference: TAG46.202
[7] Comments submitted by AFNOR and EDIFRANCE on behalf of France. Reference: TAG46.197
[8] Syntax Development Group - Executive Summary, Meeting 12, Ann Arbor, 18-22 April 1994.
[9] Revision of EDIFACT syntax, submitted by SWEPRO on 21 February 1994. Reference: TAG46.203