TSG-RAN Meeting #12 ***RP-010415***
Stockholm, Sweden, 12 - 15 June 2001

Title: Backwards compatibility of Release 99 versions

Source: Motorola

Agenda item: 8

Document for: Discussion and decision

Introduction

“Backwards compatibility” is a term that 3GPP is starting to use more often. RAN WG 2 has recently had an attempt at setting a definition for this, R2-011242 in annex 2. Motorola has recently been considering this whole area and questions whether this is what other manufacturers and operators understand by the term.

Backwards compatibility

Motorola understands that the purpose of introducing this concept was to ensure that, where possible, corrections are made to specifications in a way that does not affect elements of the specifications that are seen as correct today.

For example if a location correction is made to a message that contains information elements relating to, say, location and call control, then call control part should remain exactly the same. This might entail adding an additional location IE at the end of the message and not using the old one.

The outcome is that location did not function correctly in the old version, but does in the new, hence the CR is “backwards compatible” for call control but the location feature is not.

Motorola thinks it would be better to try to get rid of this term in relation to version of the same release and focus on evaluating if CRs are really necessary, and when necessary to focus on solutions with minimum adverse effects.

The term should still be used in reference to releases where true backwards compatibility will continue to an essential requirement.

Discussion

Is the current definition working?

Motorola does not think it is, a scenario analysis is provided in annex 1. As a completely impartial example lets us look at a Motorola CR (one which we will see in this plenary) RAN 2- CR 074 to 25.304 on Emergency calls on barred cells (Motorola, Telia). Here there was a fault in the specification, the previously defined behaviour of the mobile was not the same as in GSM. The CR mended the problem in a “backwards compatible” manner. But the fact is that a March 01 and a June 01 mobile will function differently. Generally the problem is, again, one of definition.

Another area of great concern is that WG assumptions have been used in this debate, for example (CR 075r1 to 25.321 on Clarification on compressed mode) “This is in line with WG2 assumptions but needs to be considered in the implementation”. There is no requirement for any manufacturer or operator to attend these meetings, whether a CR meets the WG assumptions should be irrelevant, only the specification should be used.

Stability

This meeting will probably approve around 300 release 99 CRs. These changes need to be made because there are faults with the current version of the specifications. At the RAN #11 meeting 393 release 99 CR were approved.

As we move forward 3GPP will have to check that all CRs approved in the future are backwards compatible with all previous versions of a release. Since the changes from plenary to plenary are large this will become an increasingly difficult task.

Motorola considers that it will be very difficult for 3GPP to make a definitive statement that any version of a release is backwards compatible with any degree of confidence. Making the current definition only applicable to single CRs adds little value.

Completeness

To date only RAN 2 and 3 have addressed this issue. RAN 1 and 4 can also approve CRs that have an effect on version interworking. If RAN 4 where to approve a CR to reduce the receiver sensitivity then this is clearly not “backwards compatible”. Networks would be designed where the later release of mobile would drop calls.

Conclusions

Motorola does not consider the use of the term backwards compatible in relation to version of a release to be of value in the RAN working groups.

As it is used today it could be misinterpreted to mean that if only “backwards compatible” CRs are approved then different versions of a release are totally compatible. By definition CRs would not be approved unless faults had been found in an earlier release, correcting these faults leads to some aspect of the system behaving differently.

Motorola think it would be better to try to get rid of this term and focus on evaluating if CRs are really necessary, and when necessary to focus on solutions with minimum adverse effects.

True backwards compatibility between releases continues to an essential requirement.

Annex 1

Scenario

Let us take an example. Operator CoolOp decides to purchase a new 3G system, he buys a switch from Beemuns, RANs from Motoring, and Finkia. In the commercial agreement they all decide to use the March 01 version of release 99, a decision made because this is the backwardly compatible version. The network is built and deployed ready for operation in March 2002. Sonyson releases their multi-media mobile, which complies with the December 01 version of release 99.

Will it work?

All the elements of the infrastructure are specified with the same version, so this should remove any version incompatibility issues. As we are all aware (look at the volume of interface related CRs seen in this meeting) it is not a forgone conclusion that all the manufacturers will have interpreted the specifications in the same way. This is one of the most common causes for the category F CRs. In the spirit of the standard, future versions of rel-99 will improve the understanding and hence the stability. It is also likely that the volume of CRs will peak when the first sets of interoperability trials are finished.

There is another difficult issue that faces this whole discussion, the definition of working, or correctly function. When addressing this most organisations start with a minimum set of functionality, which must function correctly on the first day of commercial roll-out. To help 3GPP the GSM association provided a list of basic bearers, and RAN has contributed to the list of mandatory items in the handset. However in release 99 3GPP does not have a list of features and does not track changes against them, and Motorola does not suggest a change in this policy.

When CoolOp tries to get the network up and running they discover that two suppliers have misinterpreted a handover procedure. The fault is tracked down to hole in a RAN specification. At the working group meetings this is corrected by a CR that is backwards compatible, using the RAN 2 definition. This CR is approved for the June version of the specifications. Motoring is lucky, they chose the approach used in the CR. Beemuns and Finkia have to go off and correct the software.

Point 1 – Clarifications to procedures can be seen as backwards compatible, it is feasible that all manufacturers have the same understanding.

Back at CoolOp the test engineers have hit another problem. A parameter which is supposed to be passed from the RNC to the MSCu is missing. A corrective CR is written, the ASN 1 backwards compatible procedure is used. The problem is small, the flashing LED feature does not work, but CoolOp really wants this, so all manufacturers are sent off to write some more code. Their competitor, DullOp, does not want this feature, so their vendors do not have to implement it, they know which features are affected due to the precise description in the “consequence if not approved” field.

Point 2 – Message extension for features that could be considered optional might be seen as backwards compatible. Not all manufacturers have to implement them.

Back again to our friends at CoolOp. Serious problems, the paging message is missing a key parameter. A corrective CR is written, the ASN 1 backwards compatible procedure is used. But now all manufacturers have to implement this change or a mobile cannot be paged, probably an essential feature. Under the current rules this is backwards compatible.

Point 3 – Any essential correction should NOT be seen as backwards compatible. Placing artificial constraints on the solution is simply glossing over the facts. It will lead to longer and longer messages without any need for them

CoolOp are up and running on schedule. Sonyson's new mobile is released and sent to the shops. CoolOps complete network is now running to the June 01 version, and the mobile to December 01. This will function correctly assuming the only backwards compatible CRs compiling to point 1 have been approved. The operator can control what features he deploys, so any CRs similar to point 2 will be OK if those features are not used.

CoolOps real challenge comes when ForeignOps Startup.com's March 01 mobile roams into his network. This mobile will only work if truly backward compatible CRs have been approved. The other issue facing Startup.com is that individually the CRs may be backwards compatible but when combined together the overall effect could be devastating.

Point 4 – 3GPP should only declare a version of a release backwards compatible if the combined set of CRs has been carefully analysed.

Annex 2

TSG-RAN Working Group 2 (Radio L2 and Radio L3) *R2-011242*
Busan, Korea, 21 - 25 May 2001

Title:	**Backwards compatibility**
Source:	**Chairman**
Document for:	**discussion and approval**

The following captures the current status of the discussions on methodology regarding backwards compatibility of Change Requests.

<u>Definition of a backwards compatible change</u>

A Change implemented in version N of a 3GPP release is backwards compatible when the following conditions are all met:

- Any functionality that was working in versions prior to version N still works with a UE that implements version N and a network implementing version N-1
- Any functionality that was working in versions prior to version N still works with a network that implements version N and a UE implementing version N-1
- Any functionality that was working in versions prior to version N still works with a UE that implements version N and a network implementing version N

Backwards compatibility with version N-1 should also mean BC with any prior version of the specifications in which the functionality was working, although exceptions may exist.

A backwards compatible change needs to be implemented by networks and UEs if they support the corrected functionality so that the standard (and the functionality that it intends to correct) works.

Possible actions when a functionality is found erroneous in release 99

- Make a backwards compatible change that corrects or deletes the function
- Make a non backwards compatible change that corrects the function
- State that the erroneous function is not supported in release 99, and make the correction in the next release

Action when a functionality is found ambiguous in release 99, or some text needed to clarify a common understanding

- Provide necessary clarifications
- State
- « Correction to a function where the specification was :
  - ambiguous or not sufficiently explicit.
- Would not affect implementations behaving like indicated in the CR, would affect implementations supporting the corrected functionality otherwise. »

Action when there are conflicting descriptions of a functionality in release 99

- Resolve conflict
- State
- « Correction to a function where the specification was :
  - Containing some contradictions.
- Would not affect implementations behaving like indicated in the CR, would affect implementations supporting the corrected functionality otherwise. »

Action when procedural text or rules missing for a functionality in release 99

- Add new description text
- State
- « Correction to a function where the specification was :
  - Procedural text or rules were missing.
- Would not affect implementations behaving like indicated in the CR, would affect implementations supporting the corrected functionality otherwise. »

Note : a combination of the 3 cases above may be used depending on the CR.

Backwards compatibility analysis

A backwards compatibility analysis should provide the following :

- Define clearly the functionality which does not work
- Describe the correction which is being brought
- When the change is not a backwards compatible change, state the consequence in the following cases:
    - Network implements the change, but not the UE
    - UE implements the change, but not the network