3GPP TSG SA, Maui
15 - 18 December 2003

TDoc SA-030742

Title : **Improvements to 3GPP Release Structure**

Source: Nortel Networks
(Contact: Iain Sharp)

Agenda: 9.2 - Working Methods

1. Scope

This document discusses ways of changing the 3GPP release structure in order to improve:

- the time taken to add new features to the standard and have them implemented ("velocity")
- The efficiency of the process, particularly in terms of the amount of documents that have to be generated and managed

If the "velocity" of implementation can be improved for specific critical features then this will discourage the practice of pushing changes related to new features back in to very old releases in the hope this will improve the implementation speed. This has become common practice recently and carries considerable problems in terms of destabilising old specifications.

When considering changes to the 3GPP release structure it is also important to maintain:

- the quality of the specifications
- good version management/change management
- a simple conceptual framework so that the system can be easily understood

The existing 3GPP release process has proved effective over a long time period, and changes to the process should be approached with caution. Therefore this discussion only looks at incremental changes that can be added to the existing process. Any changes to the existing process also have to be manageable within 3GPP in terms of the effort required to introduce them. Significant restructuring of the existing specification base is not realistic.

2. Early Implementation of Features

"Features may be implemented when standardisation of that feature is completed whether or not the corresponding release is approved"
- frequently stated principle at 3GPP SA plenary

Most vendor companies do not implement all aspects of a release as a block. Instead they pick individual features based on the priorities of their customers. Therefore features that are urgently required by operators may be implemented quickly even if

they are only included in a late release of the standard. However the standard must be written so that this approach is technically possible.

“Early implementation of features” means to implement an individual feature before the 3GPP release which contains that feature is approved. This topic is important because arguments often occur over whether it is necessary to wait for the completion of a 3GPP release before implementing individual features contained in that release.

The position quoted above suggests that it is not in fact necessary to wait for the whole release. In principle individual features can be implemented early. Despite this companies are reluctant to embrace early implementations and therefore 3GPP release dates are still perceived as a gating factor on the availability of new features.

Why are releases considered to be important? A number of reasons can be identified:

- Dependencies and interactions may exist with other features in the release. These may prevent early implementations.
- Protocol version control mechanisms may link features to a new version of the protocol which can only be implemented once the whole release is completed (eg Application-contexts in MAP)
- The processes to stabilise and “freeze” individual features in the standards are unclear. Therefore people making early implementations risk churn in the feature’s specifications until the whole release is frozen.

Of the reasons listed above, the protocol version control problem is the most serious. If multiple features are tied together in a new protocol version then it can be technically impossible to make an early implementation of one feature before the other are also ready. The other points are problems but these can be addressed by analysis and risk management in implementing companies.

2.1 Dealing with the Protocol Version Problem

Annex A contains an analysis of some of the key GSM/UMTS protocols and their suitability to allow individual features to be introduced without introducing dependencies on other features in the release. The key protocols in GSM/UMTS do allow new features to be introduced in a way that is suitable for early implementation. However several important protocols do include a protocol version control mechanism which can cause problems.

2.1.1 Protocol Version Control Problem

Protocols such as MAP sometimes require that extensions are performed by introducing a new version of the protocol. When a new version of a protocol is created this causes particular problems for features that are planned for early implementation and impact the same protocol. This is regardless of whether or not the feature for early implementation is the feature that triggered the decision to generate a new protocol version. The reason for this is explained below.

Most systems that use protocol version control require that a device supporting a particular protocol version is at least able to deal with the full syntax of the new

protocol version. Therefore the new protocol version cannot be implemented until its full syntax is stable in the standards.

When a feature is added to 3GPP specifications it is targeted towards a particular release. Therefore the standards documents only specify the signalling for that feature in terms of the protocol versions used in that release. If there are no changes in the protocol versions for any interface impacted by a feature then from a protocol syntax point of view early implementation of the feature should be possible. The feature can be added independently to any node supporting the existing protocol version. However, when a new protocol version is created then any feature for early implementation that uses the impacted interfaces is constrained by the need to wait for the full syntax of the new protocol version before it can be implemented. The full syntax is not normally certain until all features in the new release are finalised.

In some cases it may not be possible to know in advance whether a new protocol version will be introduced in a particular release. Therefore during the release development what was originally specified on an existing version of a protocol may finally only be standardised for a new protocol version.

2.1.2 Consequences for Standardisation

Considering the above discussion we can see that the technical ability to allow early implementation of individual features is not guaranteed. Working groups will be able to take the right design choices if they know in advance that it may be required to make an early implementation of a particular feature. This information should be provided at an early stage of the feature development.

Information from working groups on features' dependencies and linkage to specific protocol versions would be useful to companies assessing the possibility for early implementation. This could be obtained by extending the work item template to include a section for completion by working groups after the feature is completed.

2.1.3 Conclusion on Actions Relating to Protocol Versions

Firstly this discussion has shown that not all features are technically suitable for early implementation. To make sure that features are suitable for early implementation this requirement needs to be identified to working groups.

It is proposed that:

1) Requirements specifications or the work item template should identify features where technical support for early implementation is required or desirable.

2) That the work item template be extended to include a section that can be filled-in after the work item is completed to allow working groups to report on the suitability of the feature for early implementation and its dependencies and interactions.

2.2 Freezing of Individual Features

The normal process in 3GPP to stabilise specifications is to agree the "freezing" of a 3GPP release. Changes to frozen releases are only allowed under progressively more restrictive criteria which limit changes to the correction of essential errors.

The applicability of the same idea to individual features prior to the completion of a release is not clear. Though in principle the same process could be followed it is not often done in standards and the tools and processes used do not make specific provision for this event.

Without a good formal process to freeze individual features the early implementation of features will contain an element of risk which must be managed by the implementer and their customers. This risk could be reduced by clarifying and strengthening the formal process for freezing individual work items. One way to do this could be to add a work item status box to the work item form (eg “Not approved”, “Approved and working”, “Approved and technically complete”, “Approved and Frozen”, “Deleted or stopped”.

2.3 Conclusion - Early Implementation of Features

As things stand, the early implementation of features involves risks and problems that implementing features from a completed release does not involve. From that point of view it cannot be said that individually completed features outside a release are a sound basis for implementation. As such people will continue to push for complete releases as a basis for implementation. As long as this continues the 3GPP release cycle will be a bottle-neck for new services.

In order to reduce this problem a number of proposals have been made. If these proposals were adopted the problems with early implementations would be reduced. This would alleviate many of the conflicts and pressures that currently apply to feature and release scheduling.

The proposals are to:

- Explicitly plan which features may require an early implementation.
- Take account of protocol properties when designing features for early implementation.
- Capture the possibility for early implementation when individual features are completed.
- Document the “frozen” status of individual features and clarify the processes for the individual freezing of features.

3. Creating Multiple Release Streams

Currently 3GPP specifications are released as a single stream. All parts of the system are bundled in to a single release structure. It has been suggested that splitting the system in to several modules (eg Access Stratum, Non-Access Stratum) and then creating a separate release stream for each module would improve the process.

This section discusses this idea and compares it to the existing arrangement.

3.1 Definition and Motivation for Different Release Streams

The motivation for using different release streams is the observation that not all parts of the 3GPP system necessarily develop at the same rate. Trying to synchronise releases across the system pushes some areas faster than they want and holds back

others. By creating multiple release streams each module can adjust the release dates to suit their own needs.

An obvious way to structure the different steams would be to separate “RAN” and “Core Network” parts of the system. The IMS which is a kind of overlay on the Packet Domain of the Core Network could also be made in to a separate module. To be strict it must be remembered that 3GPP specifies mobile as well as network aspects. Therefore the “RAN” module is really the “Access Stratum” module and so on. The diagram below illustrates the separation of the system in to three streams.

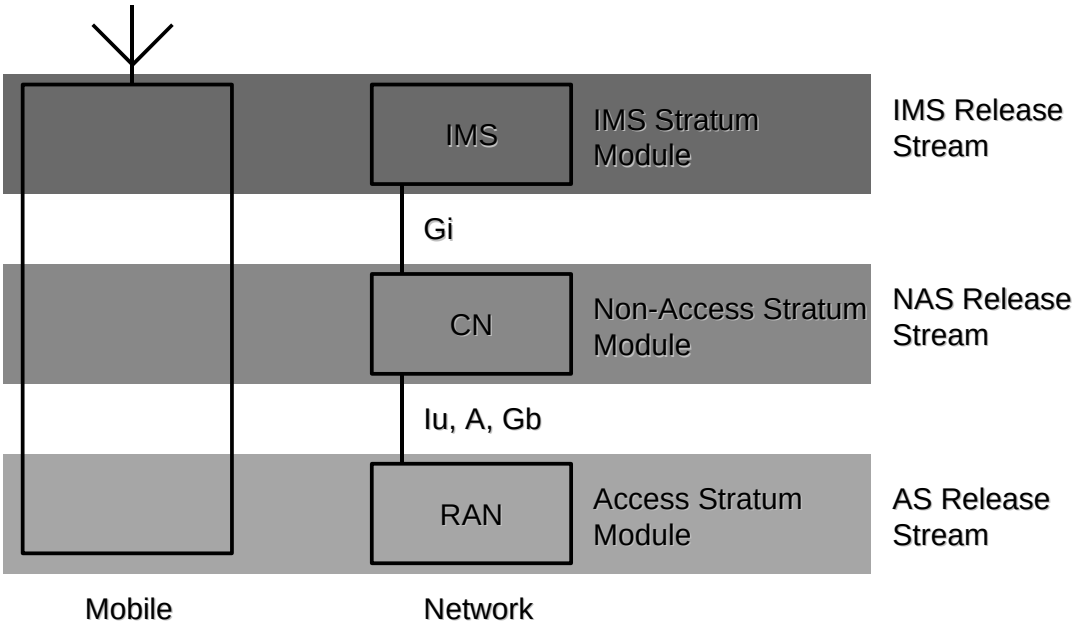

Figure 1 - Separation of Modules and Resulting Release Streams

A release structure based on this split might look as shown below:

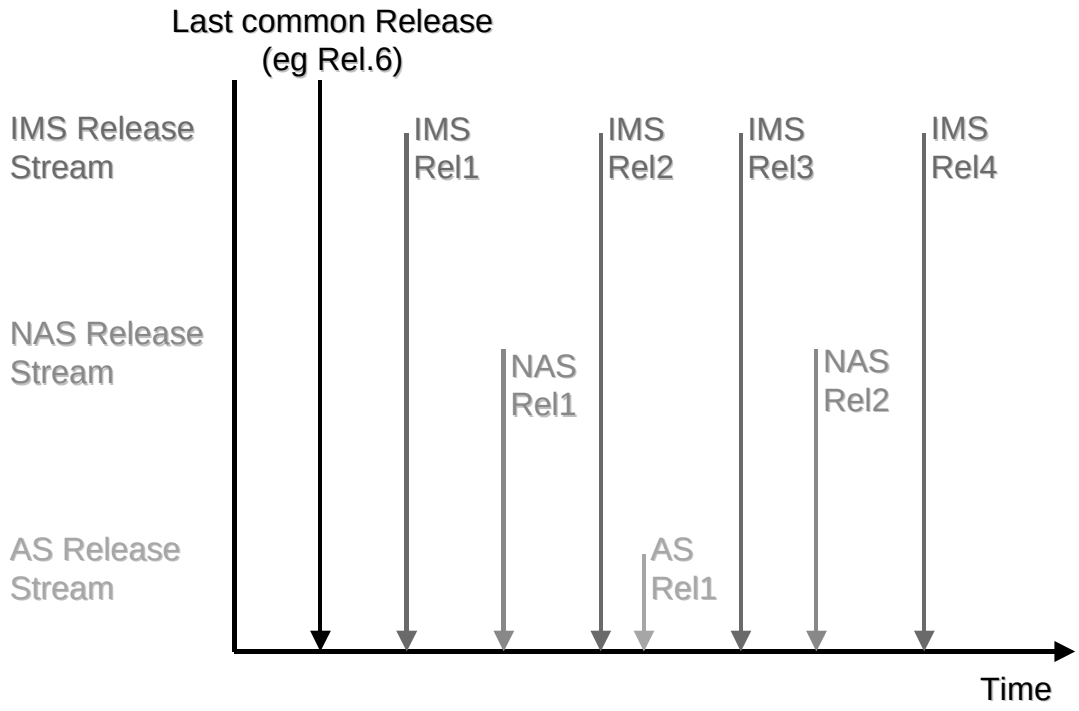

Figure 2 - Example Release Structure with Multiple Streams

These examples greatly simplify the real situation, but a number of conclusions can still be drawn.

3.1.1 Cross-Release Compatibility

Releases will need cross-compatibility between different streams.

If you consider figure 2 it can be seen that IMS Rel3 will need to at least be compatible with NAS Rel1 and NAS Rel2. In addition operational issues may require broader compatibility - eg IMS Rel2 is compatible with NAS Rel2.

Obviously cross-phase compatibility is not a new requirement for GSM/UMTS. However this scheme would extend the existing requirements. Currently cross-phase compatibility is primarily a concern on the radio interface, the SIM-ME interface and inter-PLMN interfaces. Within a PLMN it is normally acceptable to consider that the RAN and the CN are of equivalent functionality. In the multiple stream structure the need to support RANs and CNs of different releases becomes explicit.

3.1.2 Inter-Module Interfaces

In order to support the compatibility requirements the interfaces between the modules need to be well defined and the dependencies and requirements well documented. On the infrastructure the inter-module interfaces will probably be chosen to correspond to existing standardised interfaces. However in the mobile station the inter-module interfaces will probably correspond to internal interfaces for which no explicit documentation exists. Both the ME and the SIM may contain several different modules.

Supporting these inter-module interfaces will require careful work in the standards body. Cooperation between the groups responsible for each module will be needed.

3.1.3 Total Number of Specifications Generated

One possible advantage of the multi stream approach is that fewer specifications may be generated if more slowly evolving parts of the system create fewer new releases in their stream.

On average the GSM/UMTS system has had a new release approximately every 18 months in recent history. If we consider a multi stream approach what might be a release rate for each stream? New releases are unlikely to be less than 12 months apart as this is the minimum time needed for significant technical work, and to release more often then this would create a great management burden for the stream. Also the releases are unlikely to be more that 24 months apart as this would imply technical work on the module had almost stopped. Therefore it would be expected that even in a multistream model each stream would probably generate a new release at around the same rate as the current system releases. The total number of specifications in use would remain about the same.

3.1.4 Complexity

Tough the multi-stream release model is reasonably simple it still introduces significant extra complexity when compared to the current single release stream model. Special handing would be required for the specifications that deal with interfaces between the different modules. The cross-release compatibility of different protocols would have to be documented and managed. Overall the movement to a multi release stream model will add considerably to the management complexity of the system.

3.2 Conclusions - Multiple Release Streams

In this analysis the use of multiple release streams would not make any significant reduction in the total number of new specifications being released by 3GPP. Though some simplification would be made in terms of having to coordinate releases over a smaller part of the system this would be replaced by increased complexity in terms of having to deal with inter-module cross-phase issues.

On the whole the transition to multiple release streams does not seem to offer compelling advantages over the current method. Before confirming this decision input from all parts of the 3GPP project should be obtained. It is therefore recommended that the current system-wide release structure remains.

The use of multiple release streams would force the issue of dependencies between different modules within GSM/UMTS to be addressed. Even within the single release stream structure the introduction of tighter management of the dependencies between different modules should be considered to ease deployment issues for new features.

4. Conclusions

This document has discussed aspects of the 3GPP release systems and ways it could be improved. This analysis suggests that there isn't a strong case for moving away from the current model of having one system wide release produced on a periodic basis. However, improvements in a number of areas would help make this process more effective:

- Process improvements should be applied to make the "early implementation" of features prior to the freezing of a whole release a more practical possibility.
- The work item template is updated as shown in the annex to help facilitate these improvements.
- The coupling between different modules of the system should be more carefully managed so that new features can be deployed while impacting as few modules as possible.

Annex A - Analysis of GSM/UMTS Protocols

This annex analysis some key GSM/UMTS protocols in terms of their version control properties and their suitability for supporting the introduction of individual features without the need to create new protocol versions that cannot be implemented until the whole release has been completed.

In this annex the term “message” is used to refer to the basic signalling event in the protocol regardless of the specific name used in the individual protocol (eg PDU, Operation). Similarly “parameter” is used to refer to a unit of information in the message regardless of the individual name (eg Information Element).

Protocol	Mechanisms for addition of new features	Mechanism to add a new message without creating a new protocol version which may tie several features together	Mechanism to add a new parameter to an existing message without creating a new protocol version which may tie several features together	Comments/Conclusion
DTAP 24.008	"Revision Level" which takes values "Phase 1", "Phase 2" or "Rel99 and beyond" Error handling for unknown messages and parameters.	New messages may be ignored (unacknowledged mode) or trigger a STATUS message (acknowledged mode). The sender may be able to determine if a new message is supported by a receiver (and thus avoid sending unsupported messages) if a flag is included in a previous message in the reverse direction.	Error handling allows new parameters to be added to existing messages in a clean way	The transition from "Phase 2" to "Rel 99" caused problems because of the change in revision level. Beyond Rel99 support for features is signalled individually. DTAP shouldn't be a barrier to the early implementation of features.

Protocol	Mechanisms for addition of new features	Mechanism to add a new message without creating a new protocol version which may tie several features together	Mechanism to add a new parameter to an existing message without creating a new protocol version which may tie several features together	Comments/Conclusion
MAP 29.002	1) Application context version 2) Ellipses extensions (“…”) 3) Extension container for proprietary extensions	**New messages can only be added in a new application context**	New parameters can be added using ellipses to most messages - only option is to have unknown parameters ignored	**Features that are suitable for early implementation cannot be in a release which contains a new application context for an impacted protocol package.** **MAP may be a problem for early features.**

Protocol	Mechanisms for addition of new features	Mechanism to add a new message without creating a new protocol version which may tie several features together	Mechanism to add a new parameter to an existing message without creating a new protocol version which may tie several features together	Comments/Conclusion
GTP 29.060	1) GTP protocol versions 2) Extension headers 3) Unknown messages ignored	New messages can be added to the existing protocol version and will be ignored by unsupporting nodes	New parameters can be added using extension headers	**Features that are suitable for early implementation and which have GTP impacts cannot be in a release which contains a new GTP protocol version.** **In releases without new protocol versions GTP should not be a problem for early features.**

Protocol	Mechanisms for addition of new features	Mechanism to add a new message without creating a new protocol version which may tie several features together	Mechanism to add a new parameter to an existing message without creating a new protocol version which may tie several features together	Comments/Conclusion
BSSMAP 48.008 and BSSGP 48.016	1) Add new octets to existing parameters 2) Unknown messages ignored but may generate an OA&M log at the receiving node 3) Implicitly, but via OA&M nodes may be configured to use compatible versions of protocols at either end of interface	New messages may be added but may generate stream of OA&M logs from unsupporting nodes.	Parameters may be extended	**Care is required to avoid problems with early features** **This is an intra-PLMN interface and therefore work arounds can be introduced without involving third-parties.**
RANAP 25.413	"The forwards and backwards compatibility of the protocol is assured by mechanism where all current and future messages, and IEs or groups of related IEs, include ID and criticality fields that are coded in a standard format that will not be changed in the future. These parts can always be decoded regardless of the standard version."			RANAP shouldn't be a barrier to the early implementation of features.

Protocol	Mechanisms for addition of new features	Mechanism to add a new message without creating a new protocol version which may tie several features together	Mechanism to add a new parameter to an existing message without creating a new protocol version which may tie several features together	Comments/Conclusion
Iu-UP 25.415	1) Some ability to add new octets and add new parameters 2) Some ability to add new message types 3) Iu protocol version			Changes to this protocol need to be done with care. Its efficient encoding does not provide the same level of extensibility within a particular version is present on some other protocols. **Features that are suitable for early implementation and have Iu-UP impacts cannot be in a release which contains a new Iu-UP protocol version.** **In releases without new protocol versions Iu-UP should not be a problem for early features.**

Protocol	Mechanisms for addition of new features	Mechanism to add a new message without creating a new protocol version which may tie several features together	Mechanism to add a new parameter to an existing message without creating a new protocol version which may tie several features together	Comments/Conclusion
24.229 SIP	1) SIP protocol version 2) "Reasonable" error checking	In general a new message can be added without a new version (unknown messages are ignored)	In general a new parameter can be added without a new version (unknown parameters are ignored)	**Features that are suitable for early implementation and have SIP impacts cannot be in a release which contains a new SIP protocol version.** **In releases without new protocol versions SIP should not be a problem for early features. SIP is normally extensible without a new protocol version.**
29.229 Cx	Version control is still under development			

Other protocols that may be considered:

- (GE)RAN protocols
- CAMEL
- Dx

- H.248 based interfaces

Annex B - Proposed revisions to Work Item Template

Work Item Description

Title

(few words)

0. Work Item Status

	Not Approved
	Approved and work in progress
	Approved and technically complete
	Approved and frozen
	Deleted or stopped

1 3GPP Work Area

	Radio Access
	Core Network
	Services

2 Linked work items

(list of linked WIs)

3 Justification

Text (one to few paragraphs)

4 Objective

Text (one to few paragraphs)

5 Service Aspects

None/Text

6 MMI-Aspects

None/Text

7 Charging Aspects

None/Text

8 Security Aspects

None/Text

9 Impacts

Affects :	UICC apps	ME	AN	CN	Others
Yes					
No					
Don't know					

10 Expected Output and Time scale (to be updated at each plenary)

New specifications						
Spec No.	Title	Prime rsp. WG	2ndary rsp. WG(s)	Presented for information at plenary#	Approved at plenary#	Comments

Affected existing specifications				
Spec No.	CR	Subject	Approved at plenary#	Comments

10.1 Early Implementation

(Comment whether or not this work item may require “early implementation” before the release it gets assigned to is completed.)

11 Work item raporteurs

(name of physical person)

12 Work item leadership

(one WG)

13 Supporting Companies

(at least 4 companies)

14 Classification of the WI (if known)

	Feature (go to 14a)
	Building Block (go to 14b)
	Work Task (go to 14c)

14a The WI is a Feature: List of building blocks under this feature

(list of Work Items identified as building blocks)

14b The WI is a Building Block: parent Feature

(one Work Item identified as a feature)

14c The WI is a Work Task: parent Building Block

(one Work Item identified as a building block)

15 Work Item Completion Report

This information is only added once the work item is completed. It should be blank for incomplete work items.

15.1 Actual Impacts

(indicate which parts of the system are actually impacted and dependencies)

Affects:	UICC apps	ME	AN	CN	Others
Yes					
No					

15.2 Early Implementation

(Indicate any technical barriers which would prevent "early implementation" of this work item prior to the freezing of its parent release).

form change history:
2002-07-04: "USIM" box changed to "UICC apps"