

Internet Engineering Task Force
INTERNET DRAFT
June 25, 1999

Expires December 25, 1999

<[draft-rosen-megaco-structures-and-encoding-00.txt](#)>

Brian Rosen
FORE Systems
Paul Sijben
Lucent Technologies
Joerg Ott
Universitaet Bremen

Unifying MEGACO/H.GCP Structures and Encoding

Status of this document

This document is an Internet-Draft and is in full conformance with all provisions of [Section 10 of RFC2026](#)

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

The list of current Internet-Drafts can be accessed at
<http://www.ietf.org/ietf/lid-abstracts.txt>

The list of Internet-Draft Shadow Directories can be accessed at
<http://www.ietf.org/shadow.html>.

To view the entire list of current Internet-Drafts, please check the "lid-abstracts.txt" listing contained in the Internet-Drafts Shadow Directories on ftp.is.co.za (Africa), ftp.nordu.net (Northern Europe), ftp.nis.garr.it (Southern Europe), munnari.oz.au (Pacific Rim), ftp.ietf.org (US East Coast), or ftp.isi.edu (US West Coast).

Abstract

The MEGACO/H.GCP draft protocol has several structures that are not completely defined, viz, how properties are described and grouped, the template concept, the package concept, etc. There is also an extended debate on the encoding of the protocol, and particularly, the encoding of the command parameters. This contribution proposes a unified approach to all of these issues. This Internet Draft was rushed to meet the Oslo deadline. As such, it was not thoroughly reviewed, especially to see that this ID conformed to its referenced fundamental ideas on

capabilities.

Internet draft Unifying MEGACO Structures and Encoding April 16, 1999

1. Introduction

In MEGACO/H.GCP, Terminations have Properties. The fundamental operation that is required is to set Properties to specific values. Each Command (like Add) needs to be able to express Properties and values. In some cases, values are not fully specified. Wildcards (any or choose) have been proposed. One important requirement is that the list of possible Properties needs to be extended, as new kinds of gateways are designed.

To aid in making commands short, it has been suggested that we have Templates, which are named, and which specify a list of Properties and values for those Properties. The template name could be substituted for the individual properties and values. We observe that the representation of a Template is identical to the representation of the Properties to a Command, that is, a list of Properties and their values. We also observe that if a Template could substitute for a list of Properties and values, by allowing a Template name in the list, complex Templates could be constructed from simpler Templates.

The current MEGACO syntax groups Properties into Descriptors. Descriptors are a group of related Properties. Descriptors are primarily a convenience, but they allow some Properties to be used twice; once for the "local" side, and again for the "remote" side.

In addition to setting the value of a Property, MEGACO requires that the MGC be able to discover what Properties a Termination has, and the allowed values for those Properties (capabilities). Allowed values can be stated as a choice from a list of allowed values or as min/max values.

In practice, capability representation (that which is returned by Audit) is more complex, because real MGs have more complex resource limitations. The oft-cited example is a DSP that can do several things, but not necessarily all combinations. For example, it might be able to handle 24 lines of G.729a, but only 16 lines of G.726. In general, capability representation can be expressed as counted, nested property/allowed value sets with and/or operations, for example codec=

{24 x G729.a or 16 x G.726}.

It is clear that describing such capabilities may be arbitrarily complex. To reduce complexity, one might define variables representing subtrees of the complete capability sets, give each subtree a name and use the name in higher level descriptions. Such a mechanism looks like a Template.

In addition to Properties, Terminations may have Events and/or Signals applied to them. Events and Signals have names. It has been proposed

Rosen, Sijben, Ott

[Page 2]

Internet draft Unifying MEGACO Structures and Encoding April 16, 1999

that they have parameters, which could limit proliferation of Events or Signals that have similar function, but some variation. An example is the "ring" tone, which has many national variants. Signals and Events have a grouping (Packages), and are listed in Descriptors in Commands. Events and Signals must be auditable, like Properties. The Package concept, which groups events and signals into a single definition when they are related, is attractive to apply, in some way, to Properties.

Finally, expressing Properties and their values, Descriptors, Templates, Capabilities, Events and Signals, etc, have to be encoded. SDP and H.245-like syntax have been proposed, and it has been pointed out that both would have to be enhanced to represent the expressive power of the other, as well as to represent complex capabilities.

The encoding debate is fraught with "religion" stemming from the representation of similar capabilities in the MGC - MGC protocols. If you assume that you are using SIP, then you want the MEGACO/H.GCP protocol to use SDP, and if you are using H.323 for MGC-MGC, then you want to use H.245.

All of this is preface to a proposal to unify and completely specify how all of these various elements of MEGACO/H.GCP.

[2.](#) Proposal

[2.1.](#) Encoding using TLVs

We begin by proposing that a property, event, or signal be represented by a Type/Length/Value structure (TLV). This is its encoding. It is neither the text string representation of SDP, nor the ASN.1 BER syntax of H.245. TLVs are a very common way to define a compact, but flexible

encoding of arbitrary parameters.

We propose that for MEGACO/H.GCP, Type and Length are two bytes each. This means that a simple one byte Property encodes to 5 bytes, and a 30 character string encodes to 34 bytes. As is conventional, a zero in the Length field causes the first 4 bytes of the Value to be treated as an extended Length.

The Type is an IANA assigned value. Types have two subfields: Package and Item. IANA maintains a single list of Package assignments, and a list of Item assignments for each Package. Thus the name space of Item is unique within a Package. Length and Value are what you expect, Length in bytes and a byte string of the Value. A description of each Item in the Package specifies the representation of the Value (number, enumerated list, string, etc). For enumerated lists, IANA maintains a list of enumerations for each Item in each Package that requires such. We reserve one Package index for special Items. Besides the IANA

Rosen, Sijben, Ott

[Page 3]

Internet draft Unifying MEGACO Structures and Encoding April 16, 1999

registered standard packages, vendors will have their own packages which will allow vendor specific extensions, and we will reserve a Package index for "Experimental".

The convention used when displaying a Type is "Package:Item". For example, we might have an Type called "Audio:Codec" which is an Item called "Codec" defined in the Package called "Audio". Packages are described either in appendices to the MECAGO/H.GCP document, or in separate documents.

For each Item, two reserved values are usually defined for "all" and "choose". Some item definitions may need to use different conventions.

For a list of TLVs, we encode the list as a TLV, with a special Item, and a Value which is the concatenation of all the TLVs in the list. The Length is the total length of all of the concatenated TLVs.

[2.2.](#) Events and Signals

Events and Signals are Items in Packages. The Value is the parameter to the Event/Signal. If there is more than one parameter, then the Value of the Item is a concatenated list of TLVs, each of which represents one parameter.

[2.3.](#) Capabilities

To encode capabilities, we propose to have a TLV of Type "AND" and another of Type "OR". [Ott] [Kimchi] The value is a concatenated list of TLVs, Such a list could itself contain other AND/OR TLVs, of course. To encode mix/max we have another Item ("MinMax"), and if anyone thinks we need one, a "MinMaxIncrement". The first TLV in the Value of MinMax is the Min, the second is the Max. To encode quantities, we propose a TLV of Type QuantityOf. We will reserve a special item value to indicate the capability of the entire package.

[2.4.](#) Example

The general encoding for the Codec Item for example would have an index from the IANA assigned list of choices (representing G.711, G.726, ...). To indicate a capability of G.711 or G.726, we would encode this as:

```
Type: Megaco:OR
Length: 10 (If Type and Length are 2 bytes each)
Value:
    Type: Audio:Codec
    Length: 1
    Value: 1 (if IANA assigned index 1 to G.711)

    Type Audio:Codec
    Length: 1
    Value: 2 (if IANA assigned index 2 to G.726)
```

[2.5.](#) Descriptors

A Descriptor is a concatenated list of TLVs. The use of these Descriptors should be obvious. Underspecified properties may be encoded by use

of the AND, OR and MinMax TLVs as well as the "choose" and "all" Values.

[2.6.](#) Templates

For Templates, we propose to reserve a Package for use as a dynamic name store (a Template). A unique Item would be allocated dynamically, and it's value would be a concatenated list of TLVs (including nested AND/OR TLVs, and other Template TLVs). When a TLV referencing the template was encountered in a Descriptor, the current definition of the template would be substituted for the Template TLV.

[2.7.](#) Summary

A Property, Event or Signal is encoded as a TLV
Properties, Events and Signals are defined in Packages,
The Type of a TLV consists of a Package and an Item, which
are IANA assigned indexes
A descriptor is a TLV whose Type is an Item in the Megaco
Package and whose Value is a concatenated list
of TLVs.
Complex descriptors may be created by use of AND

and Megaco:OR TLVs, the value of which is a concatenated list of TLVs, any of which could be an AND/OR TLV

Descriptors may express ranges by use of MinMax and QuantityOf TLVs.

A Template is a prestored descriptor. A Template TLV may be used wherever any other TLV may be used, and may contain TLVs of Type Template, AND, OR, MinMax, etc.

A Capability is a Descriptor (Megaco:Capability) which may have QuantityOf, AND, OR and MinMax TLVs.

Capabilities are normally what Audit returns.

3. References

- * Cuervo, Huitema, Kelly, Rosen, Sijben, Zimmerer, "MEGACO Protocol", [draft-ietf-megaco-protocol-01.txt](#), April 1999.
- * ITU-T, Draft Recommendation H.GCP (05/99), "GATEWAY CONTROL PROTOCOL."
- * Kimchi, "Issues with using SDP or SDP sequences for MEGACO Protocol", [draft-kimchi-megaco-discuss-caps-00.txt](#), June 1999.
- * Ott, Kutscher, "Capability description for group cooperation", [draft-ott-mmusic-cap-00.txt](#), June 1999.

4. Authors' Addresses

Brian Rosen
FORE Systems
1000 FORE Drive
Warrendale, PA 15086
Phone: +1 724 742 6826
EMail: brosen@eng.fore.com

Paul Sijben
Lucent Technologies

Rosen, Sijben, Ott

[Page 6]

Internet draft Unifying MEGACO Structures and Encoding April 16, 1999

PO box 18
1270 AA Huizen
the Netherlands

Phone: +31 35 687 4774
Email: sijben@lucent.com

Joerg Ott
Universitaet Bremen,
TZI, MZH 5180 Bibliothekstr. 1 D-28359
Bremen Germany
Phone: +49 421 201-7028 0
Email: jo@tzi.org