

IETF media feature registration WG
Internet draft

Graham Klyne
Content Technologies/5GM
15 December 1998
Expires: June 1999

W3C Composite Capability/Preference Profiles

[<draft-ietf-conneg-W3C-ccpp-01.txt>](#)

Status of this memo

This document is an Internet-Draft. 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''.

To view the entire list of current Internet-Drafts, please check the "1id-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).

Copyright Notice

Copyright (C) 1998, The Internet Society

Abstract

This document suggests some possible areas for extending the IETF 'conneg' working group's capability description framework, described in [2,3,4]. The suggested areas for extension have been motivated by WWW Consortium (W3C) work on Composite Capability/Preference Profiles (CCPP) [5] that parallels some aspects of IETF 'conneg' work.

It is presented as a discussion document, with a view to maybe integrating some of these ideas into ongoing 'conneg' work.

Table of contents

1. Introduction.....	2
1.1 Structure of this document	3
1.2 Discussion of this document	3
1.3 Amendment history	3
2. Tag-independent content negotiation.....	4
2.1 WWW resource transfer scenario	4
2.2 Conclusions	5
2.3 Consequences	5
3. Assumed feature values.....	5
3.1 Unifying "required" and "default" values	6
3.2 Message transmission model	7
3.3. Problem summary	8
3.4 Proposed extension to 'conneg' framework	10
3.4.1 Assumed values construct.....	10
3.4.2 End of chain indicator.....	10
3.4.3 Extended form of conjunction.....	10
3.5 Additional reduction rules	11
3.6 Extended canonicalization rules	12
3.7 Examples	12
3.7.1 The "font" problem.....	12
3.7.2 Resolution dependent font.....	14
3.7.3 Combining default values.....	15
4. XML representation of capability descriptions.....	17
5. Mapping between RDF and media features.....	17
6. Conclusions.....	21
7. Security considerations.....	21
8. Full copyright statement.....	22
9. Acknowledgements.....	22
10. References.....	22
11. Author's address.....	24

[1. Introduction](#)

This document suggests some possible areas for extending the IETF 'conneg' working group's capability description framework, described in [[2](#),[3](#),[4](#)]. The suggested areas for extension have been motivated by WWW Consortium (W3C) work on Composite Capability/Preference Profiles (CCPP) [[5](#)] that parallels some aspects of IETF 'conneg' work.

It is presented as a discussion document, with a view to maybe integrating some of these ideas into ongoing 'conneg' work.

Klyne

[Page 2]

1.1 Structure of this document

The main part of this draft addresses the following main areas:

[Section 2](#) discusses tag-independent negotiation procedures, one of the goals of the 'conneg' work, with particular reference to WWW operations

[Section 3](#) suggests a framework for describing "assumed" feature values.

[Section 4](#) suggests an approach to using XML to carry capability information structured according to the 'conneg' framework.

[Section 5](#) suggests an approach to mapping between RDF [[6](#)] and 'conneg' media feature registrations.

1.2 Discussion of this document

Discussion of this document should take place on the content negotiation and media feature registration mailing list hosted by the Internet Mail Consortium (IMC):

Please send comments regarding this document to:

`ietf-medfree@imc.org`

To subscribe to this list, send a message with the body 'subscribe' to "ietf-medfree-request@imc.org".

To see what has gone on before you subscribed, please see the mailing list archive at:

<http://www.imc.org/ietf-medfree/>

1.3 Amendment history

- | | |
|-----|---|
| 00a | 19-Oct-1998
Document initially created. |
| 00b | 20-Oct-1998
Added sections on tag-independent negotiation, XML syntax and RDF mapping. |
| 00c | 31-Oct-1998
Fixed up section numbers. |
| 01a | 15-Dec-1998 |

Minor editorial revisions for re-issue. Updated CC/PP
reference to published W3C Note.

2. Tag-independent content negotiation

One can imagine two extremes of content negotiation procedure:

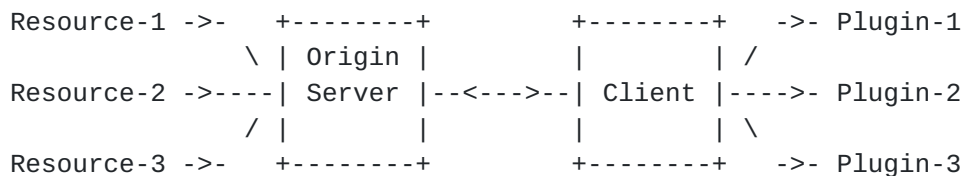
- o One in which the decisions about whether the features in a data resource match some set of capabilities are made in full knowledge of the exact meaning of every feature and capability.
- o The other, in which the matching decision is made without any knowledge of the particular features concerned.

A practical procedure is likely to lie somewhere between these extremes. The purpose of this section is to argue that a procedure that minimizes the required knowledge of particular feature tags is likely to be more useful than one which depends heavily upon such knowledge.

This approach is described here as "tag-independent negotiation": negotiation that can proceed without specific knowledge of the tags used to describe various media features of the participating entities.

2.1 WWW resource transfer scenario

Consider a WWW transaction scenario that involves content negotiation:



In this scenario, the active negotiation takes place between the origin server and the receiving client.

The resources available to the origin server are possibly passive data files, with no opportunity for interaction between data resource and server.

The plugins available to the browser may well be dispatched as separate programs, with limited opportunity for ongoing interaction between the plugin and the client. Also, there may be a high overhead associated with activating a plugin, so it is not desirable to activate one or more plugins simply to determine whether a data resource is acceptable to that plugin.

2.2 Conclusions

In an environment like WWW where new and changed features are deployed on a regular basis, it is clearly not desirable that detailed knowledge be hard-coded into either the origin server or the client software. This would rather defeat the idea of extending functionality through plugins and data resource abstractions.

Thus, to avoid early obsolescence of servers and clients, we seek a negotiation framework that permits a data resource to describe itself to the origin server, a plugin to describe itself to a browser, and allows the server and browser to conduct negotiations without any further knowledge.

There are doubtless some cases where the degree of negotiating flexibility indicated here is not essential: "Internet appliances" and other dedicated devices spring to mind. But even for such devices, it can be argued that the flexibility of tag-independent negotiation will simplify their configuration.

2.3 Consequences

Some consequences of the conclusion just outlined are:

- o The capability description should be able to describe dependencies between content features in such a way that they can be handled by the negotiation protocol.
- o The vocabulary used to describe content features should avoid having multiple ways to describe the same capabilities.

3. Assumed feature values

The W3C CC/PP [5] work embodies a concept that is not currently supported in the IETF 'conneg' framework [3]; namely the construction of a set of capability and preference values from a number of separate sources (e.g. hardware platform defaults, software platform defaults, user preferences). Also, there is a desire to have a compact way to represent a small number of differences from some baseline set of capabilities. What is not covered by this work, that the IETF 'conneg' work can supply, is a framework for actually combining the various information sources.

Also, there is an identified capability gap in the IETF 'conneg' work: the ability to indicate that some particular feature **MUST** be supplied by a communicating counterparty. This has been characterized as the "font problem": how can a sender require that some particular font is supported by a recipient? Currently the 'conneg' framework operates to effectively ignore a request for an unknown font, or other unrecognized feature.

3.1 Unifying "required" and "default" values

The two situations we wish to describe are:

- (A) "required" values: given two feature set descriptions, the feature set match is to fail if one of them does not define a value for a feature tag tested by the other. This is exemplified by the "fonts" problem: if a data resource uses some font indicated by the feature tag "Font-obscure", and the receiver does not explicitly declare an ability to deal with that font, then we should assume that the feature sets do not match. As currently defined, the 'conneg' algebraic framework [3] would allow a feature set match in the absence of an explicit denial of such support.
- (B) "override" and "default" values: given two feature set descriptions that contain different constraints involving some feature tag, the constraint from one of the feature sets is to override the constraint in the other. This is exemplified by the scenario outlined in the W3 CC/PP draft [5], in which a hardware manufacturer may supply a default profile for a device which can then be overridden by local user preferences.

A mechanism that can address both of these requirements is one which allows a feature set description to indicate "assumed" values for a given feature.

Case (A):

A data resource requires a given font to be available, and its feature set description provides "assumed" values for the font availability that are used only if the recipient does not itself reference that feature. Thus, if the data resource indicates:

(Font-obscure=TRUE)

Assume: (Font-obscure=FALSE)

and the recipient does not supply a constraint for 'Font-obscure', then the assumed value is used to force a feature set match failure.

Klyne

[Page 6]

Case (B):

A hardware platform capability description consists entirely of assumed feature values. When combined with user preferences, these provide values for those features that are not mentioned in the user preferences.

The ideas here can be extended to deal with combinations of more than two feature sets.

The assumed value idea breaks a number of assumptions of the current 'conneg' algebraic model [3], which is based pretty much on simplified predicate algebra. The sections that follow reconcile this idea with the existing framework.

3.2 Message transmission model

The following discussion is based on a message transmission model that assumes multiple feature set constraints. The following example is from [1], section 3.1:

The life of a data resource may be viewed as:

(C) (T) (F)
[A]-->--[S]-->--[R]-->--[U]

where:

[A] = author of document
(C) = original document content
[S] = message sending system
(T) = transmitted data file (representation of (C))
[R] = receiving system
(F) = formatted (rendered) document data (presentation of (C))
[U] = user or consumer of a document

Here, it is [S] and [R] who exchange negotiation metadata to decide the form of (T), so these elements are the focus of our attention.

Negotiation metadata provided by [S] would take account of available document content (C) (e.g. availability of resource variants) as well as its own possible ability to offer that content in variety of formats.

Negotiation metadata provided by [R] would similarly take account of the needs and preferences of its user [U] as well as its own capabilities to process and render received data.

This example suggests a generic framework within which negotiation is conducted: a chain of entities, each of which may impose some constraint(s) on the message features that can usefully be transferred:

[.]-->--[E1]-->--[E2]-->-- ... -->--[En]-->--[.]

'[.]' are used here indicate specific begin and end points of the chain, and '[Ei]' are entities that may impose feature constraints. If the feature constraint predicate associated with '[Ei]' is 'FCi', then the composite feature constraint for the entire chain is the conjunction of the individual constraints:

(& FC1 FC2 ... FCn)

using the notation of [\[3\]](#).

[3.3. Problem summary](#)

The 'conneg' algebraic framework depends on combining feature sets by set intersection (i.e. a logical AND of the predicates that describe feature sets).

For the purposes of this discussion, all feature set expressions are reduced to disjunctive normal form: thus, every feature set is described as a disjunction (union) of one or more subsets, each of which is described by a conjunction of atomic feature comparisons; e.g.

(| (& (size=s1) (resolution=r1) (color=c1))
(& (size=s2) (resolution=r2) (grey=g2)))

This is without loss of generality: the combining framework description [\[3\]](#) shows how more complex expressions can be reduced to this canonical form.

Now consider the following observations:

- (1) all atomic feature comparisons have the form '(ftag relop fvalue)', where 'ftag' is a feature tag that uniquely identifies a feature, 'relop' is a comparison operator and 'fvalue' is a particular value associated with the feature tag.
- (2) The conjunction operator is symmetric:

(& A B) == (& B A)

- (3) The conjunction operator is associative:

$$(& A (& B C)) == (& (& A B) C)$$

This property is important because it means that the feature sets are not required to be combined in any specific order. This point is raised in the CC/PP memo [5], and this is a property that should be preserved.

- (4) The assumed value mechanism outlined above is inherently anti-symmetric. One of the feature sets used takes precedence over the other:

$$(\text{Assume: } A B) \neq (\text{Assume: } B A)$$

- (5) The assumed value mechanism, by its very nature, is sensitive to the particular feature tags used.
- (6) Assumed values should be used only when an explicit constraint for a feature tag cannot be found. For example, if a data resource has the requirement:

$$(\text{Font-obscure}=\text{TRUE})$$

and also indicates an assumed value:

$$\text{Assume: } (\text{Font-obscure}=\text{FALSE})$$

The assumed value must be ignored if the recipient asserts:

$$(\text{Font-obscure}=\text{TRUE})$$

This requirement is captured without imposing an order of evaluation on the feature set matching process (because such imposition is considered to make the entire process more complex and error-prone).

- (7) A feature set constraint must be reduced to the predicate form currently defined by the 'conneg' framework [3] (i.e. all default values must be eliminated) before a feature collection (i.e. some specific feature values) can be tested with it.

3.4 Proposed extension to 'conneg' framework

Two extensions to the 'conneg' algebraic framework are proposed:

- (a) a "assumed values" construct
- (b) an "end-of-chain" indicator

and revision of the conjunction '(&...)' and other processing rules to take account of these new constructs.

3.4.1 Assumed values construct

This is expressed as:

(+ FC)

where 'FC' is a list of one or more feature constraint expressions. For an expression in the canonical disjunctive normal form, each is an atomic feature value constraint (i.e. contains a single feature comparison).

3.4.2 End of chain indicator

This is expressed as:

(.)

3.4.3 Extended form of conjunction

The canonical form of a conjunction is extended to allow:

(& ... (+ FL) FC (+ FR) ...)

That is, it contains separate, optional assumed value constructs to the left- and right- of any expression in the conjunction.

Referring to the message transmission chain model, these correspond to assumed values that are applied "upstream" and "downstream" along the chain. For example, when feature sets along a chain are combined:

(& FC1 (& (+ FL2) FC2 (+FR2)) FC3)

the 'FL2' assumed values applied to 'FC1', and values 'FR2' are applied to 'FC3'.

Assumed value constructs can appear anywhere in a conjunction, and the additional rules below indicate how to process these into the

canonical form.

Klyne

[Page 10]

The '(.)' construct can appear at the left or right hand end of a conjunction:

`(& (.) FC (.) )`

It may not appear anywhere else in a conjunction. Additional processing rules (below) show how these can be used to eliminate assumed value constructs from a conjunction, thus reducing it to a simple predicate form.

3.5 Additional reduction rules

The aim of these rules is to allow assumed values to propagate along the message transfer chain until either (a) it is determined they are redundant, or (b) the assumed values are used as actual feature constraints which can be evaluated in the normal way.

The following rules assume that all feature constraints are atomic; i.e. they refer to a single feature tag. The next section contains rules for canonicalization of complex predicates to a form with just atomic feature constraints.

Note that if there are any assumed value constructs present in a conjunction, re-ordering of the component constraints is allowed only according to reordering rule below.

(a) Re-ordering of assumed value constraints:

`(& ... (+FC1) FC2 ... ) ---> (& ... FC2 (+FC1) ... )`
`(& ... FC1 (+FC2) ... ) ---> (& ... (+FC2) FC1 ... )`
`(& ... (+FC1) (+FC2) ... ) ---> (& ... (+FC2) (+FC1) ... )`
ONLY IF: FC1 and FC2 reference different feature tags.

(b) Assumed value elimination:

`(& ... FC1 (+FC2) FC3 ... ) ---> (& ... FC1 FC3 ... )`
ONLY IF: FC1, FC2 and FC3 reference the same feature tag,

Two extended forms of this rule allow multiple assumed values to be eliminated:

`(& ... FC1 ... (+FC2) FC3 ... ) ---> (& ... FC1 FC3 ... )`
`(& ... FC1 (+FC2) ... FC3 ... ) ---> (& ... FC1 FC3 ... )`
ONLY IF: FC1, FC2 and FC3 reference the same feature tag,

(c) Default resolution:

`(& (.) (+FC1) ... ) ---> (& FC1 ... )`
`(& ... (+FC1) (.) ) ---> (& ... FC1 )`

Klyne

[Page 11]

3.6 Extended canonicalization rules

The above reduction rules assume a canonical form of predicate expressions. The canonicalization rules described for the 'conneg' framework [3] must be extended to handle assumed value constructs.

The main goal of these additional rules is to push assumed values "down" in the expression so that they are contained within all disjunctions and conjunctions, and themselves contain only atomic feature constraints.

(a) Separate default constructs:

`(+ FC1 FC2 ... ) ---> (+FC1) (+FC2) ...`

Following repeated application of this rule, all default constructs contain just one feature predicate expression (conjunction, disjunction or atomic).

(b) Move assumed value into conjunction:

`(+ (& FC1 FC2 ... ) ) ---> (& (+FC1) (+FC2) ... )`

(c) Move assumed value into disjunction:

`(+ (| FC1 FC2 ... ) ) ---> (| (+FC1) (+FC2) ... )`

(d) Flatten assumed value nests:

`(+ ... (+FC1) ... ) ---> (+ ... FC1 ... )`

3.7 Examples

Expressions in the following examples assume left is "upstream" in the message flow.

3.7.1 The "font" problem

Sender/data resource description:

`(& (Font-1=TRUE) (Font-2=TRUE)
 (+ (Font-1=FALSE) (Font-2=FALSE)))`

Recipient description:

`(& (Font-1=TRUE) (Font-3=TRUE) )`

Combine these to obtain the full transmission path description (recall that feature sets are matched by combining them with '(&...)', and left is upstream in the transmission path):

```
(& (.)
  (& (Font-1=TRUE) (Font-2=TRUE)           // Sender..
    (+ (Font-1=FALSE) (Font-2=FALSE) ) )
  (& (Font-1=TRUE) (Font-3=TRUE) )         // Receiver..
  (.) )
```

What follows is a sequence of rewriting rule applications that show how this expression is processed to eliminate the (+...) constructs and, in the process, the receiver's lack of support for 'Font-2' causes the feature set match to fail.

--> [expand (+...)]

```
(& (.)
  (& (Font-1=TRUE) (Font-2=TRUE)
    (+ (Font-1=FALSE) ) (+ (Font-2=FALSE) ) )
  (& (Font-1=TRUE) (Font-2=TRUE) (Font-4=TRUE) )
  (.) )
```

--> [flatten (&...(&...)...)]

```
(& (.)
  (Font-1=TRUE) (Font-2=TRUE)
  (+ (Font-1=FALSE) ) (+ (Font-2=FALSE) )
  (Font-1=TRUE) (Font-2=TRUE) (Font-4=TRUE) )
  (.) )
```

--> [collect features, being careful to preserve ordering of constraints with the same feature tag]:

```
(& (.)
  (Font-1=TRUE) (+ (Font-1=FALSE) ) (Font-1=TRUE)
  (Font-2=TRUE) (+ (Font-2=FALSE) )
  (Font-3=TRUE) )
  (& (.)
```

--> [eliminate un-needed default]

```
(& (.)
  (Font-1=TRUE) (Font-1=TRUE)
  (Font-2=TRUE) (+ (Font-2=FALSE) )
  (Font-3=TRUE) )
  (& (.)
```


--> [Resolve non-eliminated default: move to end, then use (+...)
(.) rewriting rule]

```
(& (.)  
  (Font-1=TRUE) (Font-1=TRUE)  
  (Font-2=TRUE) (Font-2=FALSE)  
  (Font-3=TRUE) )  
(& (.)
```

--> [Combine (&...) with same tag:

```
(& (.)  
  (Font-1=TRUE)  
  (FALSE)  
  (Font-3=TRUE)  
  (.) )
```

Which leaves a conjunction containing FALSE, indicating that there is no feature set matching all the capability constraints.

Note that the receiver's (Font-3=TRUE) was not eliminated because it did not use the (+...) mechanism to say, in effect, that the sender must use it.

3.7.2 Resolution dependent font

A slightly more complex example might be:

Sender/data resource description:

```
(| (& (dpi>=100) (Font-simple=TRUE) (+ (Font-simple=FALSE) ) )  
  (& (dpi>=300) (Font-fancy=TRUE) (+ (Font-fancy=FALSE) ) ) )
```

This would match the recipient capabilities:

```
(& (dpi=100) (Font-simple=TRUE) )
```

or

```
(& (dpi=300) (Font-simple=TRUE) (Font-fancy=TRUE) )
```

But it would not match the recipient capability:

```
(& (dpi=100) (Font-fancy=TRUE) )
```


3.7.3 Combining default values

Finally, note there is a possibility to use nested '(& ... (.))' constructs for defaults which are restricted to a particular entity in the transmission chain, for example (which is the type of scenario discussed in the CC/PP draft):

```
(& <sender>
  (& <system values> (+ <system values> )
    <local values> (.) ) ... )
```

Using the "indirect references" example from the CC/PP memo [5], we could construct a framework like this:

```
(& (& (Hardware-defaults) (+ (Hardware-defaults) )
  (Hardware-platform) (.) )
  (& (Software-defaults) (+ (Software-defaults) )
    (Software-platform) (.) )
  (EpocEmail)
  (EpocCalendar)
  (UserPreferences)
  (.) )
```

where

```
(Hardware-defaults) :-
  (& (Vendor="Nokia")
    (Model="2160")
    (Type="PDA")
    (ScreenSize="800x600x24")
    (CPU="PPC")
    (Keyboard="Yes")
    (Speaker="Yes")
    (Memory="16Mb" )

(Hardware-platform) :-
  (Memory="32Mb")

(Software-defaults) :-
  (& (OS="EPOC1.0")
    (HTMLVersion="4.0")
    (JavaScriptVersion="4.0")
    (WAPVersion="1.0")
    (WMLScriptVersion="1.0" )

(Software-platform) :-
  (& (Sound="Off")
    (Images="Off" )
```



```
(EpocEmail) :-  
  (& (Version="EpocEmail1.0")  
    (HTMLVersion="4.0") )  
  
(EpocCalendar) :-  
  (& (Version="EpocCalendar1.0")  
    (HTMLVersion="4.0") )
```

end

The repetition of '(Hardware-defaults)' and '(Software-defaults)' is needed to force application of the default value through reduction rule 2.5(b), "Assumed value elimination". This is, however, a little clumsy and suggests introduction of a new construct and reduction rule.

The new construct, a default end of chain indicator, is expressed as:

```
(*)
```

and its purpose is to force elimination of any unused default value, using the following rule

Default value elimination:

```
(& (*) ... (+FC1) FC2 ... ) ---> (& (*) ... FC2 ... )  
ONLY IF: FC1 and FC2 reference the same feature tag.
```

Thus, the above feature set expression could be re-written as:

```
(& (& (*) (+ (Hardware-defaults) ) (Hardware-platform) (.) )  
  (& (*) (+ (Software-defaults) ) (Software-platform) (.) )  
  (EpocEmail)  
  (EpocCalendar)  
  (UserPreferences)  
  (.) )
```

where

(etc.)

4. XML representation of capability descriptions

The 'conneg' feature set description framework is well-suited to being expressed in XML. The previously given example:

```
( | (& (dpi>=100) (Font-simple=TRUE) (+ (Font-simple=FALSE) ) )  
  (& (dpi>=300) (Font-fancy=TRUE) (+ (Font-fancy=FALSE) ) ) )
```

might be represented (in conjunction with an appropriate XML tag definitions) using something like the following XML:

```
<or>  
  <and>  
    <GE dpi 100 />  
    <EQ Font-simple TRUE />  
    <ASSUME>  
      <EQ Font-simple FALSE />  
    </ASSUME>  
  </and>  
  <and>  
    <GE dpi 300 />  
    <EQ Font-fancy TRUE />  
    <ASSUME>  
      <EQ Font-fancy FALSE />  
    </ASSUME>  
  </and>  
</or>
```

[[[I am not expert in XML, and the above is almost certainly flawed. But I do believe that it is possible to use XML broadly in this way, and would welcome further comment from any XML cognoscenti.]]]

5. Mapping between RDF and media features

The following extract is from the RDF Model and Syntax specification:

"This document introduces a model for representing RDF metadata as well as a syntax for encoding and transporting this metadata in a manner that maximizes the interoperability of independently developed web servers and clients. The syntax presented here uses the Extensible Markup Language [XML]: one of the goals of RDF is to make it possible to specify semantics for data based on XML in a standardized, interoperable manner. [...] It is also important to understand that this XML syntax is only one possible syntax for RDF, and that alternate ways to represent the same RDF

data model may emerge."

Klyne

[Page 17]

It is interesting to note that the 'conneg' feature expression framework shares with the RDF framework this clear intent to separate the data model from its representative syntax.

In this section, the RDF data model is explored and related to the 'conneg' feature registration namespace. The RDF data model is very rich compared with that used by 'conneg', so should easily be capable of representing the concepts used.

IETF 'conneg' deals with the following value-sets:

```
Feature-tag      ::= Token
Feature-value    ::= Integer
                  | Rational
                  | Boolean
                  | Token
                  | String
Feature-collection ::= FINITE MAP Feature-tag TO Feature-value
Feature-set      ::= SET OF Feature-collection
```

Further, a "Feature-set" is described by a collection of feature value assertions:

```
Feature-assertion ::= ( "&", SEQUENCE OF Feature-assertion )
                  | ( "|", SEQUENCE OF Feature-assertion )
                  | ( "!", Feature-assertion )
                  | Atomic-assertion
Atomic-assertion  ::= ( Relation, Feature-tag, Feature-value )
Relation          ::= "EQ" | "NE" | "GE" | "LE"
```

A "Feature-collection" can be represented directly as an RDF resource, where each feature tag/value pair is an RDF property name and string value.

A "Feature-set" does not have such an obvious RDF representation. However, the characterization of RDF as having "predicates" that associate some "object value" with a "resource" suggests a representation. Atomic feature value assertions are also predicates that relate a feature tag with a value.

Thus, to fit the RDF model, we can model a feature as an RDF "resource", with a tag and one or more value assertions. The following example models a simple description of a VGA display, which using 'conneg' notation would be:

```
(& (pix-x<=640) (pix-y<=480) (color<=256) )
```


This display might be described by the following RDF:

```
<rdf:RDF>
  <rdf:Description ID="VGA display">
    <conneg:Feature-assertion>
      <conneg:Feature>
        <conneg:Feature-tag="pix-x"/>
        <conneg:LE="640"/>
      </conneg:Feature>
      <conneg:Feature>
        <conneg:Feature-tag="pix-y"/>
        <conneg:LE="480"/>
      </conneg:Feature>
      <conneg:Feature>
        <conneg:Feature-tag="color"/>
        <conneg:LE="256"/>
      </conneg:Feature>
    </conneg:Feature-assertion>
  </rdf:Description>
</rdf:RDF>
```

In this example, the multiple feature values within a "Feature-assertion" are taken to be implicitly ANDed together. Other representations are possible. All 'conneg'-related tags are assumed to be defined in a 'conneg' namespace.

To take a slightly more complex 'conneg' example:

```
(| (& (pix-x<=800) (pix-y<=600) (color<=256) )  
  (& (pix-x<=640) (pix-y<=480) (color<=65536) ) )
```

This might be represented as:

```
<rdf:RDF>  
  <rdf:Description ID="VGA display">  
    <conneg:Feature-assertion>  
      <conneg:OR>  
        <rdf:Description ID="Mode-1">  
          <conneg:Feature-assertion>  
            <conneg:Feature>  
              <conneg:Feature-tag="pix-x"/>  
              <conneg:LE="800"/>  
            </conneg:Feature>  
            <conneg:Feature>  
              <conneg:Feature-tag="pix-y"/>  
              <conneg:LE="600"/>  
            </conneg:Feature>  
            <conneg:Feature>  
              <conneg:Feature-tag="color"/>  
              <conneg:LE="256"/>  
            </conneg:Feature>  
          </conneg:Feature-assertion>  
        </rdf:Description>  
        <rdf:Description ID="Mode-1">  
          <conneg:Feature-assertion>  
            <conneg:Feature>  
              <conneg:Feature-tag="pix-x"/>  
              <conneg:LE="640"/>  
            </conneg:Feature>  
            <conneg:Feature>  
              <conneg:Feature-tag="pix-y"/>  
              <conneg:LE="480"/>  
            </conneg:Feature>  
            <conneg:Feature>  
              <conneg:Feature-tag="color"/>  
              <conneg:LE="65536"/>  
            </conneg:Feature>  
          </conneg:Feature-assertion>  
        </rdf:Description>  
      </conneg:OR>  
    </conneg:Feature-assertion>  
  </rdf:Description ID="VGA display">  
</rdf:RDF>
```


[[[I am not expert in RDF, but already I can see ways in which the above might to improved. At this stage, the key ideas I wish to test are: (a) that feature collections can be modelled directly as RDL resource descriptions, (b) that feature sets must be modelled as resources in their own right, and (c) that atomic value assertions should use the RDF "predicate" to indicate the kind of constraint that is being applied.]]]

6. Conclusions

One of the goals of these proposals is to allow the 'conneg' work to integrate seamlessly with ongoing work that is being undertaken by W3C.

I have suggested some constructs and handling rules that I believe capture the required concepts embodied in the W3C work, and provide a way to manipulate them consistently within the 'conneg' framework already proposed.

[[[Ideally, the extensions to the conneg algebraic framework should be subjected to some mathematical analysis to show that application of the rules can never yield contradictory results, and that the behaviour is consistent with the concept of assumed values that it tries to capture.]]]

7. Security considerations

In general, these proposals are not believed to raise any security considerations not already inherrent in the 'conneg', 'RDF' or 'CC/PP' work.

One area can be seen where further consideration of security concerns might be required. The use of URLs, per CC/PP, to locate 'conneg' feature set descriptions might open up some exposure to privacy or denial of service attacks on the referenced description.

8. Full copyright statement

Copyright (C) The Internet Society 1998. All Rights Reserved.

This document and translations of it may be copied and furnished to others, and derivative works that comment on or otherwise explain it or assist in its implementation may be prepared, copied, published and distributed, in whole or in part, without restriction of any kind, provided that the above copyright notice and this paragraph are included on all such copies and derivative works. However, this document itself may not be modified in any way, such as by removing the copyright notice or references to the Internet Society or other Internet organizations, except as needed for the purpose of developing Internet standards in which case the procedures for copyrights defined in the Internet Standards process must be followed, or as required to translate it into languages other than English.

The limited permissions granted above are perpetual and will not be revoked by the Internet Society or its successors or assigns.

This document and the information contained herein is provided on an "AS IS" basis and THE INTERNET SOCIETY AND THE INTERNET ENGINEERING TASK FORCE DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

9. Acknowledgements

xxxx

10. References

- [1] "Requirements for protocol-independent content negotiation"
G. Klyne, Integralis Ltd.
Internet draft: <[draft-ietf-conneg-requirements-00.txt](#)>
Work in progress, March 1998.
- [2] "Media Feature Tag Registration Procedure"
Koen Holtman, TUE
Andrew Mutz, Hewlett-Packard
Ted Hardie, NASA
Internet draft: <[draft-ietf-conneg-feature-reg-03.txt](#)>
Work in progress, July 1998.

Klyne

[Page 22]

- [3] "A syntax for describing media feature sets"
Graham Klyne, 5GM/Content Technologies
Internet draft: <[draft-ietf-conneg-feature-syntax-00.txt](#)>
Work in progress, September 1998.

- [4] "Media Features for Display, Print, and Fax"
Larry Masinter, Xerox PARC
Koen Holtman, TUE
Andrew Mutz, Hewlett-Packard
Dan Wing, Cisco Systems
Internet draft: <[draft-ietf-conneg-media-features-02.txt](#)>
Work in progress, September 1998.

- [5] "Composite Capability/Preference Profiles (CC/PP):
A user side framework for content negotiation"
Franklin Reynolds, Nokia Research
W3C Note <<http://www.w3.org/TR/NOTE-CCPP/>>
November 1998

- [6] "Resource Description Framework (RDF) Model and Syntax
Specification"
Ora Lassila, Nokia Research Centre
Ralph R Swick, World Wide Web Consortium
W3C working draft: <<http://www.w3.org/TR/WD-rdf-syntax>>
October 1998

- [7] "Resource Description Framework (RDF) Schema Specification"
Dan Brickley, University of Bristol
R. V. Guha, Netscape
Andrew Layman, Microsoft
W3C working draft: <<http://www.w3.org/TR/WD-rdf-syntax>>
August 1998.

- [8] "Extensible Markup Language (XML) 1.0"
Tim Bray, Textuality and Netscape
Jean Paoli, Microsoft
C. M. Sperberg-McQueen, University of Illinois at Chicago
W3C Recommendation: <<http://www.w3.org/TR/REC-xml>>
February 1998.

11. Author's address

Graham Klyne
Content Technologies Ltd.
Forum 1
Station Road
Theale
Reading, RG7 4RA
United Kingdom

5th Generation Messaging Ltd.
5 Watlington Street
Nettlebed
Henley-on-Thames
RG9 5AB
United Kingdom.

Telephone: +44 118 930 1300

+44 1491 641 641

Facsimile: +44 118 930 1301

+44 1491 641 611

E-mail: GK@ACM.ORG

