

## Protocol-independent Content Negotiation Framework

### Status of this Memo

This memo provides information for the Internet community. It does not specify an Internet standard of any kind. Distribution of this memo is unlimited.

### Copyright Notice

Copyright (C) The Internet Society (1999). All Rights Reserved.

### Abstract

A number of Internet application protocols have a need to provide content negotiation for the resources with which they interact. MIME media types [1,2] provide a standard method for handling one major axis of variation, but resources also vary in ways which cannot be expressed using currently available MIME headers.

This memo sets out terminology, an abstract framework and goals for protocol-independent content negotiation, and identifies some technical issues which may need to be addressed.

The abstract framework does not attempt to specify the content negotiation process, but gives an indication of the anticipated scope and form of any such specification. The goals set out the desired properties of a content negotiation mechanism.

### Table of Contents

1. Introduction.....	2
1.1 Structure of this document .....	3
1.2 Discussion of this document .....	3
2. Terminology and definitions.....	3
3. Framework.....	7
3.1 Abstract framework for content negotiation .....	8
3.1.1 The negotiation process.....	9
3.2 Abstract model for negotiation metadata .....	10
3.3 Text representation for negotiation metadata .....	11
3.4 ASN.1 description of negotiation metadata .....	11
3.5 Protocol binding guidelines .....	11
4. Goals.....	12

4.1 Generic framework and metadata goals .....	12
4.2 Protocol-specific deployment goals .....	12
5. Technical issues.....	14
5.1 Non-message resource transfers .....	14
5.2 End-to-end vs hop-by-hop negotiations .....	14
5.3 Third-party negotiation .....	15
5.4 Use of generic directory and resolution services ....	15
5.5 Billing issues .....	15
5.6 Performance considerations .....	15
5.7 Confidence levels in negotiated options .....	16
6. Security Considerations.....	16
6.1 Privacy .....	16
6.2 Denial of service attacks .....	17
6.3 Mailing list interactions .....	17
6.4 Use of security services .....	17
6.5 Disclosure of security weaknesses .....	18
6.5.1 User agent identification.....	18
6.5.2 Macro viruses.....	18
6.5.3 Personal vulnerability.....	18
6.6 Problems of negotiating security .....	18
7. Acknowledgements.....	18
8. References.....	19
9. Author's Address.....	19
10. Full Copyright Statement.....	20

## 1. Introduction

A number of Internet application protocols have a need to provide content negotiation for the resources with which they interact. While MIME media types [1, 2] provide a standard method for handling one major axis of variation, resources also vary in ways which cannot be expressed using currently available MIME headers.

This memo sets out terminology, a framework and some goals for a protocol-independent content negotiation framework, and identifies some technical issues which may need to be addressed.

The framework does not attempt to specify the content negotiation process; rather it gives an indication of the anticipated scope and form of any such specifications.

The statement of goals is intended to set out the desired properties of a content negotiation framework, while trying to avoid any assumption of the form that framework may take.

## 1.1 Structure of this document

The main part of this memo addresses four main areas:

Section 2 defines some of the terms which are used with special meaning.

Section 3 outlines a proposed framework for describing protocol-independent content negotiation.

Section 4 describes various goals for content negotiation.

Section 5 discusses some of the technical issues which are raised by this document, with cross-references to other work where appropriate.

## 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](mailto:ietf-medfree@imc.org)

To subscribe to this list, send a message with the body 'subscribe' to "[ietf-medfree-request@imc.org](mailto: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/>

## 2. Terminology and definitions

This section introduces a number of terms which are used with specific meaning in the content negotiation documents. Many of these have been copied and adapted from [5].

The terms are listed in alphabetical order.

### Capability

An attribute of a sender or receiver (often the receiver) which indicates an ability to generate or process a particular type of message content.

**Characteristic**

Some description of a sender or receiver which indicates a possible capability or preference.

**Choice message**

A choice message returns a representation of some selected variant or variants, together with the variant list of the negotiable resource. It can be generated when the sender has sufficient information to select a variant for the receiver, and also requires to inform the receiver about the other variants available.

**Connected mode**

A mode of operation in which sender and receiver are directly connected, and hence are not prevented from definitively determining each other's capabilities. (See also: Session mode)

**Content feature**

(see Feature)

**Content negotiation**

An exchange of information (negotiation metadata) which leads to selection of the appropriate representation (variant) when transferring a data resource.

**Data resource**

A network data object that can be transferred. Data resources may be available in multiple representations (e.g. multiple languages, data formats, size, resolutions) or vary in other ways. (See also: Message, Resource)

**Feature**

A piece of information about the media handling properties of a message passing system component or of a data resource.

**Feature tag**

A name that identifies a "feature".

**Feature set**

Information about a sender, recipient, data file or other participant in a message transfer which describes the set of features that it can handle.

Where a 'feature' describes a single identified attribute of a resource, a 'feature set' describes full set of possible attributes.

**List message**

A list message sends the variant list of a negotiable resource, but no variant data. It can be generated when the sender does not want to, or is not allowed to, send a particular variant.

**Media feature**

information that indicates facilities assumed to be available for the message content to be properly rendered or otherwise presented. Media features are not intended to include information that affects message transmission.

**Message**

Data which is transmitted from a sender to a receiver, together with any encapsulation which may be applied. Where a data resource is the original data which may be available in a number of representations, a message contains those representation(s) which are actually transmitted. Negotiation metadata is not generally considered to be part of a message.

Message data is distinguished from other transmitted data by the fact that its content is fully determined before the start of transmission.

**Negotiated content**

Message content which has been selected by content negotiation.

**Negotiation**

(See: content negotiation)

**Negotiable resource**

A data resource which has multiple representations (variants) associated with it. Selection of an appropriate variant for transmission in a message is accomplished by content negotiation between the sender and recipient.

**Negotiation metadata**

Information which is exchanged between the sender and receiver of a message by content negotiation in order to determine the variant which should be transferred.

**Neighbouring variant**

A particular representation (variant) of a variant resource which can safely be assumed to be subject to the same access controls as the variant resource itself. Not all variants of a given variant resource are necessarily neighbouring variants. The fact that a particular variant

is or is not a neighbouring variant has implications for security considerations when determining whether that variant can be sent to a receiver in place of the corresponding variant resource. It may also have implications when determining whether or not a sender is authorized to transmit a particular variant.

Preference

An attribute of a sender or receiver (often the receiver) which indicates an preference to generate or process one particular type of message content over another, even if both are possible.

Receiver A system component (device or program) which receives a message.

Receiver-initiated transmission

A message transmission which is requested by the eventual receiver of the message. Sometimes described as 'pull' messaging. E.g. an HTTP GET operation.

Resource A document, data file or facility which is accessed or transmitted across a network. (See also: Data resource)

Sender A system component (device or program) which transmits a message.

Sender-initiated transmission

A message transmission which is invoked by the sender of the message. Sometimes described as 'push' messaging. E.g. sending an e-mail.

Session mode

A mode of message transmission in which confirmation of message delivery is received by the sender in the same application session (usually the same transport connection) that is used to transmit the message. (See also: connected mode, store and forward mode)

Store and forward mode

A mode of message transmission in which the message is held in storage for an unknown period of time on message transfer agents before being delivered.

Syntax The form used to express some value; especially the format used to express a media feature value, or a feature set. (See also: feature value, feature set, type.)

**Transmission**

The process of transferring a message from a sender to a receiver. This may include content negotiation.

**Type**

The range of values that can be indicated by some identifier of variable; especially the range of values that can be indicated by a feature tag. (See also: feature, syntax.)

NOTE: this differs from usage employed by the LDAP/X.500 directory community, who use the terms "attribute type" to describe an identifier for a value in a directory entry, and "attribute syntax" to describe a range of allowed attribute values.

**User agent**

A system component which prepares and transmits a message, or receives a message and displays, prints or otherwise processes its contents.

**Variant**

One of several possible representations of a data resource.

**Variant list**

A list containing variant descriptions, which can be bound to a negotiable resource.

**Variant description**

A machine-readable description of a variant resource, usually found in a variant list. A variant description contains a variant resource identifier and various attributes which describe properties of the variant.

**Variant resource**

A data resource for which multiple representations (variants) are available.

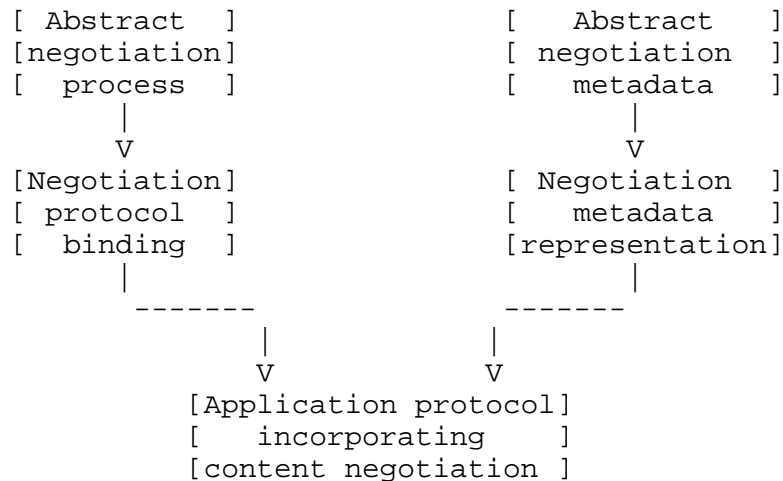
### 3. Framework

For the purposes of this document, message transmission protocol capabilities are explicitly disregarded: it is presumed that these will be dealt with separately by some orthogonal mechanism.

Content negotiation covers three elements:

1. expressing the capabilities of the sender and the data resource to be transmitted (as far as a particular message is concerned),
2. expressing the capabilities of a receiver (in advance of the transmission of the message), and
3. a protocol by which capabilities are exchanged.

These negotiation elements are addressed by a negotiation framework incorporating a number of design elements with dependencies shown:



Within this overall framework, expressing the capabilities of sender and receiver is covered by negotiation metadata. The protocol for exchanging capabilities is covered by the abstract negotiation framework and its binding to a specific application protocol.

Application protocol independence is addressed by separating the abstract negotiation process and metadata from concrete representations and protocol bindings.

### 3.1 Abstract framework for content negotiation

The negotiation framework provides for an exchange of negotiation metadata between the sender and receiver of a message which leads to determination of a data format which the sender can provide and the recipient can process. Thus, there are three main elements which are the subjects of the negotiation process and whose capabilities are described by the negotiation metadata: the sender, the transmitted data file format and the receiver.



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 a 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.

### 3.1.1 The negotiation process

Negotiation between the sender [S] and the receiver [R] consists of a series of negotiation metadata exchanges that proceeds until either party determines a specific data file (T) to be transmitted. If the sender makes the final determination, it can send the file directly. Otherwise the receiver must communicate its selection to the sender who sends the indicated file.

This process implies an open-ended exchange of information between sender and receiver. Not every implementation is expected to implement this scheme with the full generality thus implied. Rather, it is expected that every concrete negotiation can be viewed as a subset of this process.

For example, Transparent Content Negotiation (TCN) [5] uses a model in which one of the following happens:

- o The recipient requests a resource with no variants, in which case the sender simply sends what is available.

- o A variant resource is requested, in which case the server replies with a list of available variants, and the client chooses one variant from those offered.
- o The recipient requests a variant resource, and also provides negotiation metadata (in the form 'Accept' headers) which allows the server to make a choice on the client's behalf.

Another, simpler example is that of fax negotiation: in this case the intended recipient declares its capabilities, and the sender chooses a message variant to match.

Each of these can be viewed as a particular case of the general negotiation process described above. Similar observations can be made regarding the use of directory services or MIME 'Multipart/alternative' in conjunction with e-mail message transmission.

### 3.2 Abstract model for negotiation metadata

A simple but general negotiation framework has been described, which is based on the exchange of negotiation metadata between sender and recipient. The mechanism by which data is exchanged is not important to the abstract negotiation framework, but something does need to be said about the general form of the metadata.

The terminology and definitions section of this document places constraints on the form of negotiation metadata, and the descriptions that follow should be read in conjunction with the definitions to which they refer.

Negotiation metadata needs to encompass the following elements:

- o Media feature: a way to describe attributes of a data resource.
- o Feature set: a description of a range of possible media feature combinations which can be: offered by a sender; represented by a data file format; or processed by a receiver.
- o One or more naming schemes for labelling media features and feature sets. These should be backed up by some kind of registration process to ensure uniqueness of names and to encourage a common vocabulary for commonly used features.
- o A framework of data types for media features, indicating the range and properties of value types which can be represented.

- o A way to combine media features into feature sets, capable of expressing feature dependencies within a feature set (e.g. 640x480 pixel size and 256 colours, or 800x600 pixel size and 16 colours).
- o Some way to rank feature sets based upon sender and receiver preferences for different feature values.

### 3.3 Text representation for negotiation metadata

A concrete textual representation for media feature values and feature set descriptions would provide a common vocabulary for feature data in text-based protocols like HTTP and SMTP.

In defining a textual representation, the issue of allowable character sets needs to be addressed. Whether or not negotiation metadata needs to support a full gamut of international characters will depend upon the framework of data types adopted for media features. As negotiation metadata would be used as a protocol element (not directly visible to the user) rather than part of the message content, support for extended character sets may be not required.

A textual representation for negotiation metadata would imply a textual representation for media feature names, and also for expressions of the media feature combining algebra.

### 3.4 ASN.1 description of negotiation metadata

For use with non-text-based protocols, an ASN.1 description and encoding designation for negotiation metadata could be helpful for incorporating the common negotiation framework into ASN.1-derived protocols like X.400, X.500, LDAP and SNMP.

An ASN.1 description of negotiation metadata formats suggests that separate media feature naming scheme based on ISO object identifiers would be valuable.

### 3.5 Protocol binding guidelines

Specific protocol bindings will be needed to use the abstract framework for negotiation.

Details of protocol bindings would be beyond the scope of this work, but guidelines maybe not. (SASL might provide a useful model here.)

#### 4. Goals

These goals are presented in two categories:

1. Negotiation framework and metadata goals which address the broad goals of negotiation in a protocol-independent fashion.
2. Specific goals which relate to the deployment of negotiation in the context of a specific protocol (e.g. relation to HTTP protocol operations, cache interactions, security issues, existing HTTP negotiation mechanisms, application to variant selection, etc.). These would be addressed by a specific protocol binding for the negotiation framework.

##### 4.1 Generic framework and metadata goals

- o A common vocabulary for designating features and feature sets.
- o A stable reference for commonly used features.
- o An extensible framework, to allow rapid and easy adoption of new features.
- o Permit an indication of quality or preference.
- o Capture dependencies between feature values
- o A uniform framework mechanism for exchanging negotiation metadata should be defined that can encompass existing negotiable features and is extensible to future (unanticipated) features.
- o Efficient negotiation should be possible in both receiver initiated ('pull') and sender initiated ('push') message transfers.
- o The structure of the negotiation procedure framework should stand independently of any particular message transfer protocol.
- o Be capable of addressing the role of content negotiation in fulfilling the communication needs of less able computer users.

##### 4.2 Protocol-specific deployment goals

- o A negotiation should generally result in identification of a mutually acceptable form of message data to be transferred.

- o If capabilities are being sent at times other than the time of message transmission, then they should include sufficient information to allow them to be verified and authenticated.
- o A capability assertion should clearly identify the party to whom the capabilities apply, the party to whom they are being sent, and some indication of their date/time or range of validity. To be secure, capability assertions should be protected against interception and substitution of valid data by invalid data.
- o A request for capability information, if sent other than in response to delivery of a message, should clearly identify the requester, the party whose capabilities are being requested, and the time of the request. It should include sufficient information to allow the request to be authenticated.
- o In the context of a given application, content negotiation may use one or several methods for transmission, storage, or distribution of capabilities.
- o The negotiation mechanism should include a standardized method for associating features with resource variants.
- o Negotiation should provide a way to indicate provider and recipient preferences for specific features.
- o Negotiation should have the minimum possible impact on network resource consumption, particularly in terms of bandwidth and number of protocol round-trips required.
- o Systems should protect the privacy of users' profiles and providers' inventories of variants.
- o Protocol specifications should identify and permit mechanisms to verify the reasonable accuracy of any capability data provided.
- o Negotiation must not significantly jeopardize the overall operation or integrity of any system in the face of erroneous capability data, whether accidentally or maliciously provided.
- o Intelligent gateways, proxies, or caches should be allowed to participate in the negotiation.
- o Negotiation metadata should be regarded as cacheable, and explicit cache control mechanisms provided to forestall the introduction of ad-hoc cache-busting techniques.

- o Automatic negotiation should not pre-empt a user's ability to choose a document format from those available.

## 5. Technical issues

### 5.1 Non-message resource transfers

The ideas for generic content negotiation have been conceived and developed in the context of message-oriented data transmissions.

Message data is defined elsewhere as a data whose entire content is decided before the start of data transmission. The following are examples of non-message data transfers.

- o streamed data,
- o interactive computations,
- o real-time data acquisition,

Does a proposed approach to negotiation based on message data reasonably extend to streamed data (e.g. data whose content is not fully determined by the time the first data items are transmitted)?

It may be that the metadata will be applicable, but the abstract negotiation process framework may be insufficient to these more demanding circumstances.

### 5.2 End-to-end vs hop-by-hop negotiations

Could this distinction place any special demands or constraints on a generic negotiation framework, or is this simply a protocol issue?

- o End-to-end negotiation gives greatest confidence in the outcome.
- o Hop-by-hop may have advantages in a network of occasionally-connected systems, but will place additional demands on intervening message transmission agents.

Hop-by-hop negotiation implies that negotiation responses are not necessarily a definitive indication of an endpoint system's capabilities. This in turn implies a possible need for time-to-live and re-verification mechanisms to flush out stale negotiation data.

Note that one of the stated goals is to allow proxies and caches to participate in the negotiation process, as appropriate.

### 5.3 Third-party negotiation

An extension of the hop-by-hop vs. end-to-end negotiation theme is to consider the implications of allowing any system other than an endpoint participant in the message transmission to supply negotiation metadata.

Any use of a third party in the negotiation process inevitably increases the possibilities for introducing errors into the negotiation metadata.

One particular example of a third party participant in a negotiation process that is frequently suggested is the use of a directory service using LDAP or similar protocols. What additional steps need to be taken to ensure reasonable reliability of negotiation metadata supplied by this means?

### 5.4 Use of generic directory and resolution services

It is clearly helpful to use existing protocols such as LDAP to exchange content negotiation metadata.

To achieve this, it be necessary to define directory or other schema elements which are specific to content negotiation. For example, an LDAP attribute type for a media feature set.

### 5.5 Billing issues

Negotiation may raise some billing-related issues in some contexts because it potentially incurs a two-way exchange of data not necessarily completed during a single connection. There is an issue of who pays for return messages, etc., in a non-connected environment like e-mail or fax.

### 5.6 Performance considerations

Negotiation can impact performance in both positive and negative ways.

The obvious negative impact arises from the exchange of additional data which necessarily consumes some additional bandwidth. There is also an issue of round-trip or third-party query delays while negotiation metadata is being exchanged before transmission of the message itself is commenced.

Over the Internet, there are some bandwidth/latency trade-offs which can be made. For example, in Internet e-mail the MIME type 'multipart/alternative' can be used to send multiple versions of a

resource: this preserves latency by using additional bandwidth to send a greater volume of data. On the other hand, HTTP [7] suggests a negotiation mechanism which preserves bandwidth at the cost of introducing a round-trip delay (section 12.2, Agent-driven negotiation).

To set against the negative performance impact of content negotiation, it is to be hoped that overall network efficiency is to be improved if it results in the most useful data format being delivered to its intended recipient, first time, almost every time.

### 5.7 Confidence levels in negotiated options

In some cases (e.g. when there has been a direct exchange of information with the remote system) the communicating parties will have a high degree of confidence in the outcome of a negotiation. Here, a data exchange can be performed without need for subsequent confirmation that the options used were acceptable.

In other cases, the options will be a best-guess, and it may be necessary to make provision for parties to reject the options actually used in preference for some other set.

This consideration is likely to interact with performance considerations.

A useful pattern, adopted by TCN [5], is to define a negotiation procedure which guarantees a correct outcome. This forms the foundation for a procedure which attempts to use easily-obtained but less reliable information in an attempt to optimize the negotiation process but that contains checks to guarantee the final result will be the same as would have been obtained by the full negotiation procedure. Such procedures sometimes have to resort to the original "full cycle" negotiation procedure, but in a majority of cases are expected to reach their conclusion by an optimized route.

## 6. Security Considerations

The purposes of this section is to identify and catalogue some security issues that feature negotiation protocols should consider.

### 6.1 Privacy

Privacy may be adversely affected by:

- o Unintended disclosure of personal information.



- o Spoofed requests for negotiation data simply for the purposes of gathering information, and not as part of a bona fide message transmission.

## 6.2 Denial of service attacks

Service denial may be caused by:

- o Injection of false negotiation data.
- o Excessive requests for negotiation data

## 6.3 Mailing list interactions

Content negotiation with final recipients is somewhat at odds with normal practice for maintaining lists for redistribution of Internet mail.

It may be appropriate for a sender to negotiate data formats with a list manager, and for a list manager to negotiate with message recipients. But the common practice of keeping confidential the identities and addresses of mailing list subscribers suggests that end-to-end negotiation through a mailing list is not consistent with good security practice.

## 6.4 Use of security services

Protocols that employ security services for message transfer should also apply those services to content negotiation:

- o Authenticated requests for negotiation metadata provide a means for a potential recipient to moderate the distribution of media capability information.
- o Authentication of negotiation metadata provides a means for potential message senders to avoid using incorrect information injected by some other party.
- o Encryption of negotiation data may help to prevent disclosure of sensitive capability-related information to snoopers.
- o Conducting a negotiation exchange over an authenticated or encrypted protocol session (e.g. SASL), transport connection or network path (e.g. TLS, IPSEC) can provide for mutual authentication of both parties in an exchange of negotiation data.

## 6.5 Disclosure of security weaknesses

### 6.5.1 User agent identification

Disclosure of capability information may allow a potential attacker to deduce what message handling agent is used, and hence may lead to the exploitation of known security weaknesses in that agent.

### 6.5.2 Macro viruses

Macro viruses are a widespread problem among applications such as word processors and spreadsheets. Knowing which applications a recipient employs (e.g. by file format) may assist in a malicious attack. However, such viruses can be spread easily without such knowledge by sending multiple messages, where each message infects a specific application version.

### 6.5.3 Personal vulnerability

One application of content negotiation is to enable the delivery of message content that meets specific requirements of less able people. Disclosure of this information may make such people potential targets for attacks that play on their personal vulnerabilities.

## 6.6 Problems of negotiating security

If feature negotiation is used to decide upon security-related features to be used, some special problems may be created if the negotiation procedure can be subverted to prevent the selection of effective security procedures.

The security considerations section of GSS-API negotiation [8] discusses the use of integrity protecting mechanisms with security negotiation.

## 7. Acknowledgements

Some material in this memo has been derived from earlier memos by Koen Holtman, Andrew Mutz, Ted Hardie, Larry Masinter, Dan Wing, Neil Joffe. Matters relating to the importance and relevance of content negotiation to less-able users were raised by Al Gilman.

This memo has also been informed by the debates of the IETF "conneg" working group.

## 8. References

- [1] Freed, N. and N. Borenstein, "Multipurpose Internet Mail Extensions (MIME) Part 1: Format of Internet message bodies", RFC 2045, November 1996.
- [2] Freed, N. and N. Borenstein, "Multipurpose Internet Mail Extensions (MIME) Part 2: Media Types", RFC 2046, November 1996.
- [3] Holtman, K., et al., "The Alternates Header Field", Work in Progress.
- [4] Hardie, T., "Scenarios for the Delivery of Negotiated Content", Work in Progress.
- [5] Holtman, K. and A. Mutz, "Transparent Content Negotiation in HTTP", RFC 2295, March 1998.
- [6] Wing, D., "Indicating Supported Media Features Using Extensions to DSN and MDN", RFC 2530, March 1999.
- [7] Fielding, R., Gettys, J., Mogul, J., Frytyk, H. and T. Berners-Lee, "Hypertext Transfer Protocol -- HTTP/1.1", RFC 2068, January 1997.
- [8] Blaize, E. and D. Pinkas, "The Simple and Protected GSS-API Negotiation Mechanism", RFC 2478, December 1998.

## 9. Author's Address

Graham Klyne

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

Content Technologies Ltd.

1220 Parkview, Arlington Business Park  
Theale  
Reading, RG7 4SA  
United Kingdom.

Phone: +44 1491 641 641

+44 118 930 1300

Fax: +44 1491 641 611

+44 118 930 1301

EMail: GK@ACM.ORG

## 10. Full Copyright Statement

Copyright (C) The Internet Society (1999). 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.

## Acknowledgement

Funding for the RFC Editor function is currently provided by the Internet Society.

