FOUNDATION FOR INTELLIGENT PHYSICAL AGENTS

FIPA Quality of Service Ontology Specification

Document title	FIPA Quality of Service Ontology Specification		
Document number	SC00094A	**Document source**	FIPA TC Nomadic Application Support
Document status	Standard	**Date of this status**	2002/12/03
Supersedes	None		
Contact	fab@fipa.org		
Change history	See *Informative Annex A — ChangeLog*		

http://www.fipa.org/
Geneva, Switzerland

Notice

Use of the technologies described in this specification may infringe patents, copyrights or other intellectual property rights of FIPA Members and non-members. Nothing in this specification should be construed as granting permission to use any of the technologies described. Anyone planning to make use of technology covered by the intellectual property rights of others should first obtain permission from the holder(s) of the rights. FIPA strongly encourages anyone implementing any part of this specification to determine first whether part(s) sought to be implemented are covered by the intellectual property of others, and, if so, to obtain appropriate licenses or other permission from the holder(s) of such intellectual property prior to implementation. This specification is subject to change without notice. Neither FIPA nor any of its Members accept any responsibility whatsoever for damages or liability, direct or consequential, which may result from the use of this specification.

Foreword

The Foundation for Intelligent Physical Agents (FIPA) is an international organization that is dedicated to promoting the industry of intelligent agents by openly developing specifications supporting interoperability among agents and agent-based applications. This occurs through open collaboration among its member organizations, which are companies and universities that are active in the field of agents. FIPA makes the results of its activities available to all interested parties and intends to contribute its results to the appropriate formal standards bodies where appropriate.

The members of FIPA are individually and collectively committed to open competition in the development of agent-based applications, services and equipment. Membership in FIPA is open to any corporation and individual firm, partnership, governmental body or international organization without restriction. In particular, members are not bound to implement or use specific agent-based standards, recommendations and FIPA specifications by virtue of their participation in FIPA.

The FIPA specifications are developed through direct involvement of the FIPA membership. The status of a specification can be either Preliminary, Experimental, Standard, Deprecated or Obsolete. More detail about the process of specification may be found in the FIPA Document Policy [f-out-00000] and the FIPA Specifications Policy [f-out-00003]. A complete overview of the FIPA specifications and their current status may be found on the FIPA Web site.

FIPA is a non-profit association registered in Geneva, Switzerland. As of June 2002, the 56 members of FIPA represented many countries worldwide. Further information about FIPA as an organization, membership information, FIPA specifications and upcoming meetings may be found on the FIPA Web site at http://www.fipa.org/.

Contents

1 Scope

This document deals with a Quality of Service ontology. It contains specifications for:

- Defining an ontology for representing the Quality of Service of the FIPA Message Transport Service.

2 Overview

The ability to automatically adjust to changes in a transparent and integrated fashion is essential for *nomadicity*; nomadic end-users are usually professionals in areas other than computing. Furthermore, today's mobile computer systems are already very complex to use as productivity tools. Thus, nomadic end-users need all the support that a FIPA agent-based distributed system can deliver and adaptability to the changes in the environment of nomadic end-users is an important issue. To be able to adapt to the changes, an agent must be aware of the changes in the environment.

The `fipa-qos` ontology can be used by agents when communicating about the Quality of Service (QoS). The ontology provides basic vocabulary for QoS. Additionally, the `fipa-qos` ontology supports two methods to get QoS information: a single query and a subscription. For example, an agent may query current QoS values from another agent using, for example, the `fipa-query` interaction protocol [FIPA00027] or the agent may subscribe to notifications when something interesting happens in the QoS using the `fipa-subscribe` interaction protocol [FIPA00035]. These notifications may be dispatched at a predefined interval or when some changes in the QoS occur. The former mechanism (periodic notification) can be used if the agent wants to be informed about the QoS values on a regular basis, for example the value of the throughput every five seconds. The latter mechanism (on occurrence notification) is useful when the agent does not care about QoS values until something relevant to its task happens. For example, an agent that is sending real-time data may need to be informed, when the throughput value drops below the given threshold.

3 Quality of Service Ontology

3.1 Object Descriptions

This section describes a set of frames that represent the classes of objects in the domain of discourse within the framework of the `fipa-qos` ontology.

The following terms are used to describe the objects of the domain:

- **Frame**. This is the mandatory name of this entity that must be used to represent each instance of this class.

- **Ontology**. This is the name of the ontology, whose domain of discourse includes the parameters described in the table.

- **Parameter**. This is the mandatory name of a parameter of this frame.

- **Description**. This is a natural language description of the semantics of each parameter.

- **Presence**. This indicates whether each parameter is mandatory or optional.

- **Type**. This is the type of the values of the parameter: Integer, Word, String, URL, Term, Set or Sequence.

- **Reserved Values**. This is a list of FIPA-defined constants that can assume values for this parameter.

3.1.1 Quality of Service Description

This type of object represents the quality of service of the transport protocol or communication channel.

Frame	`qos`
Ontology	`fipa-qos`

Parameter	Description	Presence[1]	Type	Reserved Values
`line-rate`	The bandwidth in one direction over the link.	Optional	`rate-value`	
`throughput`	The number of user data bits successfully transferred in one direction across the link[2]. Successful transfer means that no user data bits are lost, added or inverted in transfer.	Optional	`rate-value`	
`throughput-std-dev`	The current standard deviation of the throughput within a time unit.	Optional	`rate-value`	
`rtt`	The round trip time which is the time required for a data segment to be transmitted to a peer entity and a corresponding acknowledgement sent back to the originating entity.	Optional	`time-value`	
`rtt-std-dev`	The standard deviation of the round-trip time within a time unit.	Optional	`time-value`	
`delay`	The (nominal) time required for a data segment to be transmitted to a peer entity.	Optional	`time-value`	

[1] While all of the parameters for this object are optional, a valid `qos` object will contain at least one parameter.

[2] See [ITUX135].

delay-std-dev	The standard deviation of the delay time within a time unit.	Optional	time-value	
mean-up-time	The expected uptime of an established link.	Optional	time-value	
omission-rate	The probability that a data segment is not transmitted correctly over a link.	Optional	probability-value	
ber	The ratio of the number of bit errors to the total number of bits transmitted in a given time interval[3].	Optional	probability-value	
frame-error-rate	The probability that a data segment is not transmitted correctly over a link.	Optional	probability-value	
conn-setup-delay	The (sampled) delay to establish a connection between communicating entities.	Optional	time-value	
conn-setup-failure-prob	The ratio of total call attempts that result in call setup failure to the total call attempts in a population of interest.	Optional	probability-value	
status	The connectivity status of the link. connected means that there (at least) logical connection between communicating entities. disconnected means that there is no connection between communicating entities, and the communicating entities are not establishing a connection at the moment. connecting means that there is no connection between communicating entities, but they are currently establishing a connection between them.	Optional	word	connected disconnected connecting

3.1.2 Rate Value

This type of object represents a data transfer value.

Frame **Ontology**	rate-value fipa-qos

Parameter	**Description**	**Presence**	**Type**	**Reserved Values**
direction	The direction in which this value is measured. inbound means the data transmission where the actor receives the data, and outbound means the data transmission where the actor transmits the data.	Mandatory	word	inbound outbound
unit	The unit in which the value is represented. bits/s means bits per seconds. kbits/s means kilobits per seconds. One kilobit is 2^10 bits. mbits/s means megabits per	Mandatory	word	gbits/s mbits/s kbits/s bits/s

[3] See [ITUE800].

	second. One megabit is 2^20 bits. `gbits/s` means gigabits per second. One gigabit is 2^30 bits.			
`value`	The rate value.	Mandatory	`number`	

3.1.3 Time Value

This type of object represents a time value.

Frame	`time-value`
Ontology	`fipa-qos`

Parameter	Description	Presence	Type	Reserved Values
`direction`	The direction in which this value is measured. `inbound` means the data transmission where the actor receives the data, and `outbound` means the data transmission where the actor transmits the data.	Optional[4]	`word`	`inbound` `outbound`

`unit`	The unit in which the value is represented. `h` means hours, `m` means minutes, `s` means seconds, and `ms` means milliseconds.	Mandatory	`word`	`h`
				`m`
				`s`
				`ms`

`value`	The time value.	Mandatory	`number`	

3.1.4 Probability Value

This type of object represents a probability value.

Frame	`probability-value`
Ontology	`fipa-qos`

Parameter	Description	Presence	Type	Reserved Values
`direction`	The direction in which this value is measured. `inbound` means the data transmission where the actor receives the data, and `outbound` means the data transmission where the actor transmits the data.	Optional	`word`	`inbound` `outbound`
`value`	The probability value which obeys the following axiom: `0  value  1`	Mandatory	`number`	

3.1.5 Time Type

This type of object represents the time type of a time value.

Frame	`time-type`
Ontology	`fipa-qos`

Parameter	Description	Presence	Type	Reserved Values
`value`	The value of the time-type.	Mandatory	`word`	`every` `after`

[4] This parameter is mandatory for those QoS values that have a different value depending upon the direction.

3.1.6 Communication Channel Description

This type of object represents a communication channel.

Frame	comm-channel
Ontology	fipa-qos

Parameter	**Description**	**Presence**[5]	**Type**	**Reserved Values**
name	The logical name of the communication channel.	Optional	word	
target-addr	The target transport address of the communication channel. This may also be the address of a gateway ACC.	Optional	url	
options	A list of optional parameters for the communication channel.	Optional	Set of property[6]	

3.1.7 Transport Protocol Description

This type of object represents a transport protocol.

Frame	transport-protocol
Ontology	fipa-qos

Parameter	**Description**	**Presence**	**Type**	**Reserved Values**
name	The logical name of the transport protocol.	Mandatory	word	
gw-addr	The transport address of the gateway ACC.	Optional	url	
dest-addr	The transport address of the ultimate destination. If this address is present, but gw-addr is not, then the Control Agent may select the most appropriate gateway transport address to use.	Optional	url	
options	A list of optional parameters for the transport protocol.	Optional	Set of property	

3.1.8 Property Template

This is a special object that is useful for specifying parameter/value pairs.

Frame	property
Ontology	fipa-qos

Parameter	**Description**	**Presence**	**Type**	**Reserved Values**
name	The name of the property.	Mandatory	string	
value	The value of the property.	Mandatory	term	

[5] Either the name parameter or the target-addr parameter must be present in this object.

[6] See [FIPA00023]

3.2 Predicate Descriptions

The following tables define usage and semantics of the predicates that are part of the `fipa-qos` ontology.

The following terms are used to describe the predicates of the `fipa-qos` domain:

- **Predicate**. This is the symbol that identifies the predicate in the ontology.

- **Ontology**. This is the name of the ontology, whose domain of discourse includes the function or the predicate described in the table.

- **Supported by**. This is the type of agent that supports this function or predicate.

- **Description**. This is a natural language description of the semantics of the function or the predicate.

- **Domain**. This indicates the domain over which the function predicate is defined. The arguments passed to the function or predicate must belong to the set identified by the domain.

- **Arity**. This indicates the number of arguments that a function or a predicate takes. If a function or a predicate can take an arbitrary number of arguments, then its arity is undefined.

3.2.1 Monitoring Information

Predicate	`qos-information`
Ontology	`fipa-qos`
Supported by	MA
Description	The predicate is true when the values of the QoS parameters defined in the `qos` object are true for a given communication channel or transport protocol. That is, the QoS of a communication channel or transport protocol is what is stated in the QoS object. Otherwise the predicate is false.
Domain	`comm-channel` /[7] `transport-protocol`[8] × `qos`
Arity	2

3.2.2 Time Constraint

Predicate	`time-constraint`
Ontology	`fipa-qos`
Supported by	MA
Description	If the `time-type` parameter has the value `every`, then the predicate is true every time interval defined in the `time-value` parameter. If the `time-type` parameter has the value `after`, the predicate is true after the time specified the `time-value` parameter. Otherwise the predicate is false.
Domain	`time-type` × `time-value`
Arity	2

[7] Where '/' is "exclusive or".

[8] Where '×' is Cartesian product.

3.2.3 Match Quality of Service Information

Predicate	`qos-match`
Ontology	`fipa-qos`
Supported by	MA
Description	An agent may subscribe to notifications about changes to the quality of service from an MA.
Domain	`qos-information` × `qos-information`
Arity	2

3.2.3.1 Matching Criterion

The `qos-match` predicate defined in this ontology mandates the implementation of the following matching criterion in order to determine the set of objects that satisfy the search criteria.

The first thing to note about the matching operation is that the `qos-match` predicate receives, as its first argument, an object description that evaluates to a structured object that will be used as an object template during the execution of the `qos-match` action. In the following explanation, the expressions *parameter template* and *value template* are used to denote a parameter of the object template, and the value of the parameter of the object template, respectively.

A registered object matches an object template if:

1. The class name of the object (that is, the object type) is the same as the class name of the object description template, and,

2. Each parameter of the object template is matched by a parameter of the object description.

A parameter matches a parameter template if the parameter name is the same as the template parameter name, and its value matches the value template.

Since the value of a parameter is a term, the rules for a term to match another term template must be given. Before, it must be acknowledged that the values of the parameters of descriptions kept by the MA can only be either `SLConstants`, `SLSets`, `SLSequences` or other object descriptions (for example, a `service-description`).

The `qos-match` action evaluates functional expressions before the object template is matched against the descriptions kept by the MA. This means that if the value of a parameter of an object description is a functional term (for example, `(plus 2 3)`), then what is seen by the matching process is the result of evaluating the functional term within the context of the receiving agent. A constant matches a constant template if they are equal.

Informally, a sequence matches a sequence template if the elements of the sequence template are matched by elements of the sequence appearing in the same order. Formally, the following recursive rules apply:

1. An empty sequence matches an empty sequence, and,

2. The sequence `(cons` *`x sequence1`*`)` matches the sequence template `(cons` *`y sequence2`*`)` if:

   - *`x`* matches *`y`* and *`sequence1`* matches *`sequence2`*, or,

   - *`sequence1`* matches `(cons` *y* *`sequence2`*`)`.

Finally, a set matches a set template if each element of the set template is matched by an element of the set template. Notice that it is possible that the same element of the set matches more than one element of the set template.

3.2.3.2 Matching Examples

The following example matches the `qos-information` of communication channel named `gsm` every 10 seconds:

```
(iota ?x
  (and
    (time-constraint (time-type :value every) (time-value :value 10 :unit seconds))
    (qos-matches ?x
      (qos-information (comm-channel :name gsm)))))
```

The following example matches the `qos-information` of communication channel named `gsm` whenever the `rtt` value is 500 milliseconds:

```
(iota ?x
   (qos-matches ?x
     (qos-information
      (comm-channel :name gsm)
      (qos :rtt (rate-value :unit ms :value 500))))))
```

The following example matches the `qos-information` of communication channel named `gsm` whenever the `rtt` value is between 300 and 400 milliseconds:

```
(iota ?x
  (exists ?y
    (and
      (qos-matches ?x
         (qos-information
           (comm-channel :name gsm)
           (qos :rtt (rate-value :unit ms :value ?y)))))
    (> ?y 30) (< ?y 40))))
```

3.3 Exceptions

The exceptions for the `fipa-qos` ontology follow the same form and rules as specified in [FIPA00023].

3.3.1 Not Understood Exception Propositions

Communicative Act	`not-understood`
Ontology	`fipa-qos`

Predicate Symbol	**Arguments**	**Description**
`unsupported-act`	`string`	The receiving agent does not support the specific communicative act; the string identifies the unsupported communicative act.
`unexpected-act`	`string`	The receiving agent supports the specified communicative act, but it is out of context; the string identifies the unexpected communicative act.
`unsupported-value`	`string`	The receiving agent does not support the value of a message parameter; the string identifies the message parameter name.
`unrecognised-value`	`string`	The receiving agent cannot recognise the value of a message parameter; the string identifies the message parameter name.

3.3.2 Refusal Exception Proposition

Communicative Act	`refuse`
Ontology	`fipa-qos`

Predicate symbol	**Arguments**	**Description**
`unauthorised`		The sending agent is not authorised to perform the function.
`unsupported-function`	`string`	The receiving agent does not support the function; the string identifies the unsupported function name.
`missing-argument`	`string`	A mandatory function argument is missing; the string identifies the missing function argument name.

unexpected-argument	string	A mandatory function argument is present which is not required; the string identifies the required function argument that is not expected.
unexpected-argument-count		The number of function arguments is incorrect.
missing-parameter	string string	A mandatory parameter is missing; the first string represents the object name and the second string represents the missing parameter name.
unexpected-parameter	string string	The receiving agent does not support the parameter; the first string represents the function name and the second string represents the unsupported parameter name.
unrecognised-parameter-value	string string	The receiving agent cannot recognise the value of a parameter; the first string represents the object name and the second string represents the parameter name of the unrecognised parameter value.
unrecognised-comm-channel	comm-channel	The specified communication channel is not recognised; the string identifies the communication channel.
unsupported-protocol	transport-protocol	The specified transport protocol is not supported; the string identifies the transport protocol.

3.3.3 Failure Exception Propositions

Communicative Act	failure
Ontology	fipa-qos

Predicate symbol	Arguments	Description
internal-error	string	An internal error occurred; the string identifies the internal error.

4 References

[FIPA00023] FIPA Agent Management Specification. Foundation for Intelligent Physical Agents, 2000. `http://www.fipa.org/specs/fipa00023/`

[FIPA00027] FIPA Query Interaction Protocol Specification. Foundation for Intelligent Physical Agents, 2000. `http://www.fipa.org/specs/fipa00027/`

[FIPA00035] FIPA Subscribe Interaction Protocol Specification. Foundation for Intelligent Physical Agents, 2000. `http://www.fipa.org/specs/fipa00023/`

[ITUE800] Recommendation E.800 – Telephone Network and ISDN, Quality of Service, Network Management and Traffic Engineering, Terms and Definitions Related to Quality of Service and Network Performance Including Dependability. International Telecommunication Union, International Telecommunication Union, 1995.

[ITUX135] Recommendation X.135 – Speed of Service (delay and throughput), Performance Values for Public Data Networks when Providing Packet-Switched Services. International Telegraph and Telephone Consultative Committee, 1993.

5 Informative Annex A — ChangeLog

5.1 2002/12/03 - version A by FIPA Architecture Board

Entire document: Promoted to Standard status