

Table 1: Complete translation of FIPA performatives to μ ACP encodings. Each μ ACP expression is a finite sequence of messages of the form $\langle \text{hdr}, v, O, p \rangle$ where $v \in \{\text{PING}, \text{TELL}, \text{ASK}, \text{OBSERVE}\}$ and O encodes FIPA parameters via TLV fields. CID = correlation identifier; PROP = proposition; ACT = action.

FIPA Performatives	μ ACP Encoding $\tau(f)$
INFORM(s, r, φ)	[TELL($s, r, \text{TLV}(\text{PROP} = \varphi)$)]
INFORM-IF(s, r, φ)	[TELL($s, r, \text{TLV}(\text{PROP} = \varphi)$) iff $\text{Bel}_{s(\varphi)}$]
INFORM-REF(s, r, x)	[TELL($s, r, x; \text{TLV}(\text{ref} = 1)$)]
REQUEST(s, r, α)	[ASK($s, r, \text{TLV}(\text{ACT} = \alpha)$), TELL($r, s, \text{TLV}(\text{DONE} = \alpha)$)]
QUERY-IF(s, r, ψ)	[ASK($s, r, \text{TLV}(\text{QUERY} = \psi)$)]
QUERY-REF(s, r, ψ)	[ASK($s, r, \text{TLV}(\text{QUERY_REF} = \psi)$)]
CFP($s, G, task$)	[ASK($s, g, \text{TLV}(\text{CALL} = task)$) $g \in G$]
PROPOSE($p, q, prop$)	[TELL($p, q, \text{TLV}(\text{PROPOSAL} = prop)$)]
ACCEPT-PROPOSAL($q, p, prop$)	[TELL($q, p, \text{TLV}(\text{ACCEPT} = prop)$)]
REJECT-PROPOSAL($q, p, prop$)	[TELL($q, p, \text{TLV}(\text{REJECT} = prop)$)]
SUBSCRIBE(s, r, e)	[ASK($s, r, \text{TLV}(\text{SUBSCRIBE} = e)$), OBSERVE($r, s, \text{TLV}(\text{NOTIFY_ON} = e)$)]
AGREE(r, s, α)	[TELL($r, s, \text{TLV}(\text{AGREE} = \alpha)$)]
REFUSE(r, s, α)	[TELL($r, s, \text{TLV}(\text{REFUSE} = \alpha)$)]
CANCEL(s, r, cid)	[TELL($s, r, \text{TLV}(\text{CID} = cid, \text{CANCEL} = 1)$)]
FAILURE($r, s, reason$)	[TELL($r, s, \text{TLV}(\text{FAIL} = reason)$)]
CONFIRM(s, r, φ)	[TELL($s, r, \varphi; \text{TLV}(\text{CONFIRM} = 1)$)]
DISCONFIRM(s, r, φ)	[TELL($s, r, \neg\varphi; \text{TLV}(\text{DISCONFIRM} = 1)$)]
PROPAGATE(s, G, m)	[TELL($s, g, \text{TLV}(\text{PROP} = m)$) $g \in G$]
PROXY(s, r, f)	[ASK($s, r, \text{TLV}(\text{PROC} = f, \text{CID} = c)$)]
FORWARD(s, r, m)	[TELL($s, r, m; \text{TLV}(\text{FORWARD} = 1)$)]
NOT-UNDERSTOOD(s, r, m)	[PING($s, r, \text{TLV}(\text{ERR}, \text{orig} = m)$)]
UNSUBSCRIBE(s, r, cid)	[TELL($s, r, \text{TLV}(\text{CID} = cid, \text{UNSUBSCRIBE} = 1)$)]
BROKER($s, broker, m$)	[ASK($s, broker, \text{TLV}(\text{BROKER} = 1, \text{PAYLOAD} = m)$)]
INFORM-DONE(s, r, α)	[TELL($s, r, \text{TLV}(\text{DONE} = \alpha)$)]
INFORM-REF(s, r, x)	[TELL($s, r, \text{TLV}(\text{REF} = x)$)]
PROXY-IF(s, r, f)	[ASK($s, r, \text{TLV}(\text{PROC} = f, \text{IF} = 1)$)]

Table 1: Complete translation of FIPA performatives to μ ACP encodings. Each μ ACP expression is a finite sequence of messages of the form $\langle \text{hdr}, v, O, p \rangle$ where $v \in \{\text{PING}, \text{TELL}, \text{ASK}, \text{OBSERVE}\}$ and O encodes FIPA parameters via TLV fields. CID = correlation identifier; PROP = proposition; ACT = action.

FIPA Performatives	μ ACP Encoding $\tau(f)$
PROXY- REF(s, r, x)	[ASK($s, r, \text{TLV}(\text{PROC} = x, \text{REF} = 1)$)]
PROPAGATE- IF(s, G, ψ)	[TELL($s, g, \text{TLV}(\text{PROP} = \psi, \text{IF} = 1)$) $g \in G$]
PROPAGATE- REF(s, G, x)	[TELL($s, g, \text{TLV}(\text{PROP} = x, \text{REF} = 1)$) $g \in G$]
REQUEST- WHEN(s, r, α, ψ)	[ASK($s, r, \text{TLV}(\text{ACT} = \alpha, \text{WHEN} = \psi)$)]
REQUEST- WHENEVER(s, r, α, ψ)	[ASK($s, r, \text{TLV}(\text{ACT} = \alpha, \text{WHENEVER} = \psi)$)]
QUERY- REF(s, r, x)	[ASK($s, r, \text{TLV}(\text{QUERY_REF} = x)$)]
STREAM- ALL(s, r, ψ)	[OBSERVE($s, r, \text{TLV}(\text{STREAM} = \psi, \text{ALL} = 1)$)]
STREAM- SOME(s, r, ψ)	[OBSERVE($s, r, \text{TLV}(\text{STREAM} = \psi, \text{SOME} = 1)$)]
STREAM- NONE(s, r, ψ)	[OBSERVE($s, r, \text{TLV}(\text{STREAM} = \psi, \text{NONE} = 1)$)]
STREAM- REF(s, r, x)	[OBSERVE($s, r, \text{TLV}(\text{STREAM} = x, \text{REF} = 1)$)]
STREAM-IF(s, r, ψ)	[OBSERVE($s, r, \text{TLV}(\text{STREAM} = \psi, \text{IF} = 1)$)]
STREAM-REF- IF(s, r, x, ψ) IF= ψ]	[OBSERVE($s, r, \text{TLV}(\text{STREAM} = x, \text{REF} = 1,$ WHEN= ψ)]
STREAM-REF- WHEN(s, r, x, ψ) WHEN= ψ]	[OBSERVE($s, r, \text{TLV}(\text{STREAM} = x, \text{REF} = 1,$ WHENEVER= ψ)]
STREAM-REF- ALL(s, r, x) ALL=1]	[OBSERVE($s, r, \text{TLV}(\text{STREAM} = x, \text{REF} = 1,$ ALL=1)]
STREAM-REF- SOME(s, r, x) SOME=1)]	[OBSERVE($s, r, \text{TLV}(\text{STREAM} = x, \text{REF} = 1,$ SOME=1)]

Table 1: Complete translation of FIPA performatives to μ ACP encodings. Each μ ACP expression is a finite sequence of messages of the form $\langle \text{hdr}, v, O, p \rangle$ where $v \in \{\text{PING}, \text{TELL}, \text{ASK}, \text{OBSERVE}\}$ and O encodes FIPA parameters via TLV fields. CID = correlation identifier; PROP = proposition; ACT = action.

FIPA Performatives	μ ACP Encoding $\tau(f)$
STREAM-REF- NONE(s, r, x) NONE=1]	[OBSERVE(s, r , TLV (STREAM = x , REF = 1, NONE=1)]
STREAM-IF- ALL(s, r, ψ) ALL=1]	[OBSERVE(s, r , TLV (STREAM = ψ , IF = 1, ALL=1)]
STREAM-IF- SOME(s, r, ψ) SOME=1)]	[OBSERVE(s, r , TLV (STREAM = ψ , IF = 1, SOME=1)]
STREAM-IF- NONE(s, r, ψ) NONE=1)]	[OBSERVE(s, r , TLV (STREAM = ψ , IF = 1, NONE=1)]
STREAM-WHEN- ALL(s, r, ψ) ALL=1)]	[OBSERVE(s, r , TLV (STREAM = ψ , WHEN = 1, ALL=1)]
STREAM-WHEN- SOME(s, r, ψ) SOME=1)]	[OBSERVE(s, r , TLV (STREAM = ψ , WHEN = 1, SOME=1)]
STREAM-WHEN- NONE(s, r, ψ) NONE=1)]	[OBSERVE(s, r , TLV (STREAM = ψ , WHEN = 1, NONE=1)]
STREAM- WHENEVER- ALL(s, r, ψ) ALL=1)]	[OBSERVE(s, r , TLV (STREAM = ψ , WHENEVER = 1, ALL=1)]
STREAM- WHENEVER- SOME(s, r, ψ) SOME=1)]	[OBSERVE(s, r , TLV (STREAM = ψ , WHENEVER = 1, SOME=1)]
STREAM- WHENEVER- NONE(s, r, ψ) NONE=1)]	[OBSERVE(s, r , TLV (STREAM = ψ , WHENEVER = 1, NONE=1)]
STREAM-REF-IF- ALL(s, r, x, ψ)	[OBSERVE(s, r , TLV (STREAM = x , REF = 1, ALL=1)]

Table 1: Complete translation of FIPA performatives to μ ACP encodings. Each μ ACP expression is a finite sequence of messages of the form $\langle \text{hdr}, v, O, p \rangle$ where $v \in \{\text{PING}, \text{TELL}, \text{ASK}, \text{OBSERVE}\}$ and O encodes FIPA parameters via TLV fields. CID = correlation identifier; PROP = proposition; ACT = action.

FIPA Performatives	μ ACP Encoding $\tau(f)$
IF= ψ , ALL = 1)]	
STREAM-REF-IF- SOME(s, r, x, ψ)	[OBSERVE(s, r , TLV (STREAM = x , REF = 1, SOME(s, r, x, ψ))]
IF= ψ , SOME = 1)]	
STREAM-REF-IF- NONE(s, r, x, ψ)	[OBSERVE(s, r , TLV (STREAM = x , REF = 1, NONE(s, r, x, ψ))]
IF= ψ , NONE = 1)]	
STREAM-REF- WHEN- ALL(s, r, x, ψ)	[OBSERVE(s, r , TLV (STREAM = x , REF = 1, WHEN- ALL(s, r, x, ψ))]
WHEN= ψ , ALL = 1)]	
STREAM-REF- WHEN- SOME(s, r, x, ψ)	[OBSERVE(s, r , TLV (STREAM = x , REF = 1, WHEN- SOME(s, r, x, ψ))]
WHEN= ψ , SOME = 1)]	
STREAM-REF- WHEN- NONE(s, r, x, ψ)	[OBSERVE(s, r , TLV (STREAM = x , REF = 1, WHEN- NONE(s, r, x, ψ))]
WHEN= ψ , NONE = 1)]	
STREAM-REF- WHENEVER- ALL(s, r, x, ψ)	[OBSERVE(s, r , TLV (STREAM = x , REF = 1, WHENEVER- ALL(s, r, x, ψ))]
WHENEVER= ψ , ALL = 1)]	
STREAM-REF- WHENEVER- SOME(s, r, x, ψ)	[OBSERVE(s, r , TLV (STREAM = x , REF = 1, WHENEVER- SOME(s, r, x, ψ))]
WHENEVER= ψ , SOME = 1)]	

Table 1: Complete translation of FIPA performatives to μ ACP encodings. Each μ ACP expression is a finite sequence of messages of the form $\langle \text{hdr}, v, O, p \rangle$ where $v \in \{\text{PING}, \text{TELL}, \text{ASK}, \text{OBSERVE}\}$ and O encodes FIPA parameters via TLV fields. CID = correlation identifier; PROP = proposition; ACT = action.

FIPA Performatives	μ ACP Encoding $\tau(f)$
STREAM-REF- WHENEVER- NONE(s, r, x, ψ)	[OBSERVE(s, r , TLV (STREAM = x , REF = 1, WHENEVER= ψ , NONE = 1)]
WHENEVER= ψ , NONE = 1] STREAM-IF- WHEN- ALL(s, r, ψ_1, ψ_2)	[OBSERVE(s, r , TLV (STREAM = ψ_1 , IF = 1, WHEN= ψ_2 , ALL = 1)] [OBSERVE(s, r , TLV (STREAM = ψ_1 , IF = 1, WHEN- SOME(s, r, ψ_1, ψ_2))]
WHEN= ψ_2 , SOME = 1] STREAM-IF- WHEN- NONE(s, r, ψ_1, ψ_2)	[OBSERVE(s, r , TLV (STREAM = ψ_1 , IF = 1, WHEN= ψ_2 , SOME = 1)] [OBSERVE(s, r , TLV (STREAM = ψ_1 , IF = 1, WHEN- NONE(s, r, ψ_1, ψ_2))]
WHEN= ψ_2 , NONE = 1] STREAM-IF- WHENEVER- ALL(s, r, ψ_1, ψ_2)	[OBSERVE(s, r , TLV (STREAM = ψ_1 , IF = 1, WHEN= ψ_2 , NONE = 1)] [OBSERVE(s, r , TLV (STREAM = ψ_1 , IF = 1, WHENEVER- ALL(s, r, ψ_1, ψ_2))]
WHENEVER= ψ_2 , ALL = 1] STREAM-IF- WHENEVER- SOME(s, r, ψ_1, ψ_2)	[OBSERVE(s, r , TLV (STREAM = ψ_1 , IF = 1, WHENEVER= ψ_2 , ALL = 1)] [OBSERVE(s, r , TLV (STREAM = ψ_1 , IF = 1, WHENEVER- SOME(s, r, ψ_1, ψ_2))]
WHENEVER= ψ_2 , SOME = 1)]	[OBSERVE(s, r , TLV (STREAM = ψ_1 , IF = 1, WHENEVER= ψ_2 , SOME = 1)]

Table 1: Complete translation of FIPA performatives to μ ACP encodings. Each μ ACP expression is a finite sequence of messages of the form $\langle \text{hdr}, v, O, p \rangle$ where $v \in \{\text{PING}, \text{TELL}, \text{ASK}, \text{OBSERVE}\}$ and O encodes FIPA parameters via TLV fields. CID = correlation identifier; PROP = proposition; ACT = action.

FIPA Performatives	μ ACP Encoding $\tau(f)$
STREAM-IF- WHENEVER- NONE(s, r, ψ_1, ψ_2)	$[\text{OBSERVE}(s, r, \text{TLV}(\text{STREAM} = \psi_1, \text{IF} = 1, \text{WHENEVER} = \psi_2, \text{NONE} = \text{ALL}), \psi_2)]$
WHENEVER= ψ_2 , NONE = 1] STREAM-WHEN- WHENEVER- ALL(s, r, ψ_1, ψ_2)	$[\text{OBSERVE}(s, r, \text{TLV}(\text{STREAM} = \psi_1, \text{WHEN} = 1, \text{WHENEVER} = \psi_2, \text{NONE} = \text{ALL}), \psi_2)]$
WHENEVER= ψ_2 , ALL = 1] STREAM-WHEN- WHENEVER- SOME(s, r, ψ_1, ψ_2)	$[\text{OBSERVE}(s, r, \text{TLV}(\text{STREAM} = \psi_1, \text{WHEN} = 1, \text{WHENEVER} = \psi_2, \text{SOME} = \text{ALL}), \psi_2)]$
WHENEVER= ψ_2 , SOME = 1] STREAM-WHEN- WHENEVER- NONE(s, r, ψ_1, ψ_2)	$[\text{OBSERVE}(s, r, \text{TLV}(\text{STREAM} = \psi_1, \text{WHEN} = 1, \text{WHENEVER} = \psi_2, \text{NONE} = \text{SOME}), \psi_2)]$
WHENEVER= ψ_2 , NONE = 1] STREAM-REF-IF- WHEN- ALL(s, r, x, ψ_1, ψ_2) IF= ψ_1 , WHEN = ψ_2 , ALL=1] STREAM-REF-IF- WHEN- SOME(s, r, x, ψ_1, ψ_2) IF= ψ_1 , WHEN = ψ_2 , SOME=1] STREAM-REF-IF- WHEN- NONE(s, r, x, ψ_1, ψ_2)	$[\text{OBSERVE}(s, r, \text{TLV}(\text{STREAM} = x, \text{REF} = 1, \text{WHEN} = \psi_2, \text{ALL} = \text{IF}), \psi_2)]$ $[\text{OBSERVE}(s, r, \text{TLV}(\text{STREAM} = x, \text{REF} = 1, \text{WHEN} = \psi_2, \text{SOME} = \text{IF}), \psi_2)]$ $[\text{OBSERVE}(s, r, \text{TLV}(\text{STREAM} = x, \text{REF} = 1, \text{WHEN} = \psi_2, \text{NONE} = \text{IF}), \psi_2)]$

Table 1: Complete translation of FIPA performatives to μ ACP encodings. Each μ ACP expression is a finite sequence of messages of the form $\langle \text{hdr}, v, O, p \rangle$ where $v \in \{\text{PING}, \text{TELL}, \text{ASK}, \text{OBSERVE}\}$ and O encodes FIPA parameters via TLV fields. CID = correlation identifier; PROP = proposition; ACT = action.

FIPA Performatives	μ ACP Encoding $\tau(f)$
IF= ψ_1 , WHEN = ψ_2 , NONE=1)]	
STREAM-REF-IF- WHENEVER- ALL(s, r, x, ψ_1, ψ_2) IF= ψ_1 , WHENEVER = ψ_2 , ALL=1)]	[OBSERVE(s, r , TLV (STREAM = x , REF = 1, WHENEVER- ALL(s, r, x, ψ_1, ψ_2))]
STREAM-REF-IF- WHENEVER- SOME(s, r, x, ψ_1, ψ_2) IF= ψ_1 , WHENEVER = ψ_2 , SOME=1)]	[OBSERVE(s, r , TLV (STREAM = x , REF = 1, WHENEVER- SOME(s, r, x, ψ_1, ψ_2))]
STREAM-REF-IF- WHENEVER- NONE(s, r, x, ψ_1, ψ_2) IF= ψ_1 , WHENEVER = ψ_2 , NONE=1)]	[OBSERVE(s, r , TLV (STREAM = x , REF = 1, WHENEVER- NONE(s, r, x, ψ_1, ψ_2))]
STREAM-REF- WHEN- WHENEVER- ALL(s, r, x, ψ_1, ψ_2)	[OBSERVE(s, r , TLV (STREAM = x , REF = 1, WHEN- WHENEVER- ALL(s, r, x, ψ_1, ψ_2))]
WHEN= ψ_1 , WHENEVER = ψ_2 , ALL=1)]	[OBSERVE(s, r , TLV (STREAM = x , REF = 1, WHEN= ψ_1 , WHENEVER = ψ_2 , ALL=1))]
STREAM-REF- WHEN- WHENEVER- SOME(s, r, x, ψ_1, ψ_2)	[OBSERVE(s, r , TLV (STREAM = x , REF = 1, WHEN- WHENEVER- SOME(s, r, x, ψ_1, ψ_2))]
WHEN= ψ_1 , WHENEVER = ψ_2 , SOME=1)]	[OBSERVE(s, r , TLV (STREAM = x , REF = 1, WHEN= ψ_1 , WHENEVER = ψ_2 , SOME=1))]

Table 1: Complete translation of FIPA performatives to μ ACP encodings. Each μ ACP expression is a finite sequence of messages of the form $\langle \text{hdr}, v, O, p \rangle$ where $v \in \{\text{PING}, \text{TELL}, \text{ASK}, \text{OBSERVE}\}$ and O encodes FIPA parameters via TLV fields. CID = correlation identifier; PROP = proposition; ACT = action.

FIPA Performatives	μ ACP Encoding $\tau(f)$
STREAM-REF- WHEN- WHENEVER- NONE(s, r, x, ψ_1, ψ_2)	$[\text{OBSERVE}(s, r, \text{TLV}(\text{STREAM} = x, \text{REF} = 1,$ $\psi_2,$ $\text{NONE}=1)]$
WHEN= ψ_1 , WHENEVER = ψ_2 , ALL=1)	
STREAM-IF- WHEN- WHENEVER- ALL($s, r, \psi_1, \psi_2, \psi_3$)	$[\text{OBSERVE}(s, r, \text{TLV}(\text{STREAM} = \psi_1, \text{IF} = 1,$ $\psi_3,$ $\text{ALL}=1)]$
WHEN= ψ_2 , WHENEVER = ψ_3 , SOME=1)	$[\text{OBSERVE}(s, r, \text{TLV}(\text{STREAM} = \psi_1, \text{IF} = 1,$ $\psi_3,$ $\text{SOME}=1)]$
STREAM-IF- WHEN- WHENEVER- NONE($s, r, \psi_1, \psi_2, \psi_3$)	$[\text{OBSERVE}(s, r, \text{TLV}(\text{STREAM} = \psi_1, \text{IF} = 1,$ $\psi_2,$ $\text{NONE}=1)]$
WHEN= ψ_2 , WHENEVER = ψ_3 , NONE=1)	
STREAM-REF-IF- WHEN- WHENEVER- ALL($s, r, x, \psi_1, \psi_2, \psi_3$)	$[\text{OBSERVE}(s, r, \text{TLV}(\text{STREAM} = x, \text{REF} = 1,$ $\psi_2,$ $\text{NONE}=1)]$

Table 1: Complete translation of FIPA performatives to μ ACP encodings. Each μ ACP expression is a finite sequence of messages of the form $\langle \text{hdr}, v, O, p \rangle$ where $v \in \{\text{PING}, \text{TELL}, \text{ASK}, \text{OBSERVE}\}$ and O encodes FIPA parameters via TLV fields. CID = correlation identifier; PROP = proposition; ACT = action.

FIPA Performatives	μ ACP Encoding $\tau(f)$
IF= ψ_1 , WHEN = ψ_2 ,	
WHENEVER= ψ_3 , ALL = 1]	
STREAM-REF-IF- WHEN-	[OBSERVE(s, r , TLV (STREAM = x , REF = 1,
WHENEVER-	SOME($s, r, x, \psi_1, \psi_2, \psi_3$)
SOME($s, r, x, \psi_1, \psi_2, \psi_3$)	
IF= ψ_1 , WHEN = ψ_2 ,	
WHENEVER= ψ_3 , SOME = 1]	
STREAM-REF-IF- WHEN-	[OBSERVE(s, r , TLV (STREAM = x , REF = 1,
WHENEVER-	NONE($s, r, x, \psi_1, \psi_2, \psi_3$)
NONE($s, r, x, \psi_1, \psi_2, \psi_3$)	
IF= ψ_1 , WHEN = ψ_2 ,	
WHENEVER= ψ_3 , NONE = 1)]	

Table 2: TLV field type definitions for μ ACP encoding

TLV Type	Description
PROP	Propositional content
ACT	Action specification
DONE	Action completion marker
QUERY	Query content
QUERY_REF	Query reference
CALL	Call for proposals
PROPOSAL	Proposal content
ACCEPT	Acceptance marker
REJECT	Rejection marker
SUBSCRIBE	Subscription request
NOTIFY_ON	Notification trigger
AGREE	Agreement marker
REFUSE	Refusal marker
CANCEL	Cancellation marker
FAIL	Failure reason
CONFIRM	Confirmation marker
DISCONFIRM	Disconfirmation marker
PROP	Propagation marker
PROC	Procedure specification
FORWARD	Forwarding marker
ERR	Error information
CID	Correlation identifier
VAL	Value content
NOTIFY	Notification content
UNSUBSCRIBE	Unsubscription marker
BROKER	Brokerage marker
PAYLOAD	Message payload
IF	Conditional marker
REF	Reference marker
WHEN	Temporal condition
WHENEVER	Recurring condition
STREAM	Stream specification
ALL	Universal quantifier
SOME	Existential quantifier
NONE	Negation marker