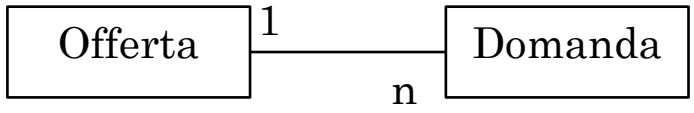
2017
Soluzioni compiti 2016

Giorgio Bruno

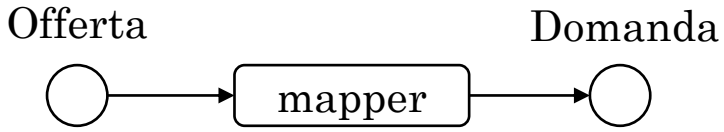
Dip. Automatica e Informatica
Politecnico di Torino
email: giorgio.bruno@polito.it

Complemento sui pattern

Estensioni del mapper

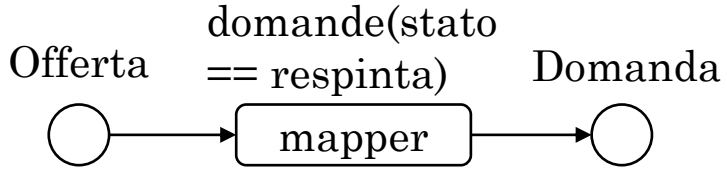


Domanda: stato (respinta, accettata)

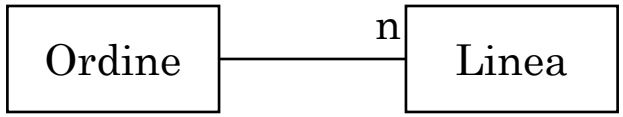


Il mapper emette le domande associate all'offerta.

mapper con condizione



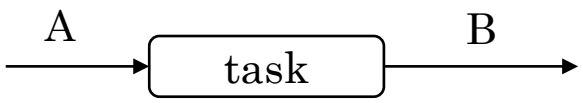
Il mapper emette soltanto le domande respinte che sono associate all'offerta.



Mapping implicito

Il task *accetta* modifica lo stato dell'ordine ed è seguito dal mapper che emette le linee associate all'ordine.
Il mapping può essere inglobato nel link di uscita.

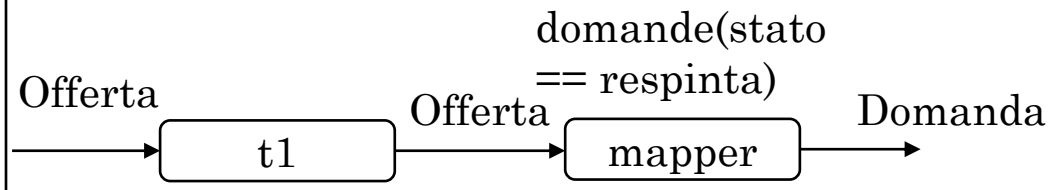
Il link di uscita non trasmette l'ordine ma le linee che sono associate all'ordine direttamente o tramite una relazione derivata.



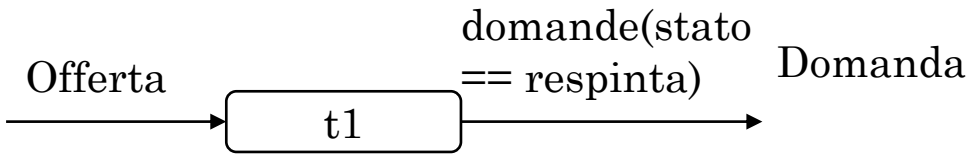
Forma generica
Il link non trasmette l'entità A ma le entità B connesse direttamente o indirettamente all'entità A; può esserci anche una sola entità B.

mapping link

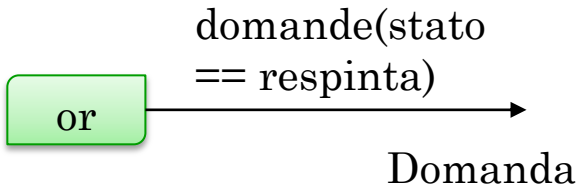
Mapping implicito con condizione



Il **mapper** emette soltanto le domande respinte che sono associate all'offerta.



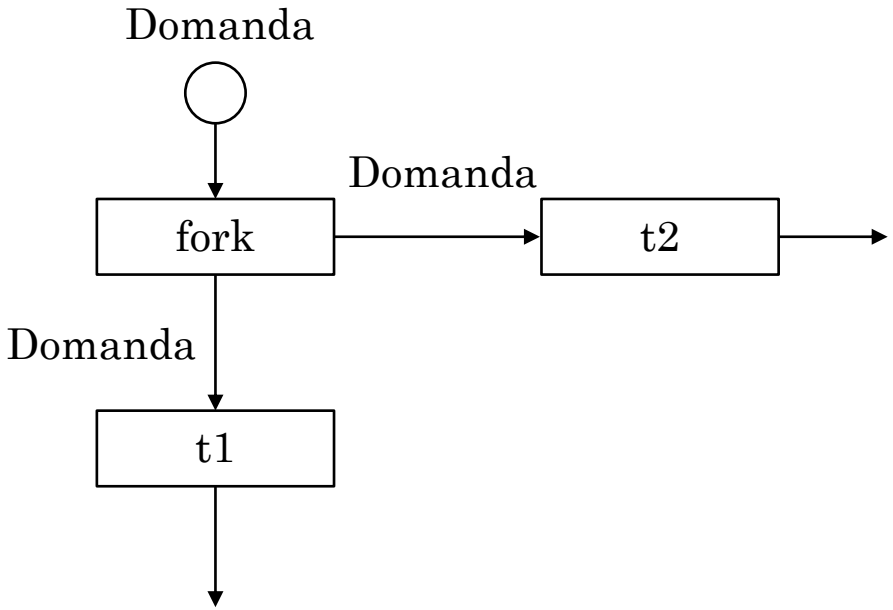
Il **mapping link** non trasmette l'offerta ma le domande respinte associate all'offerta.



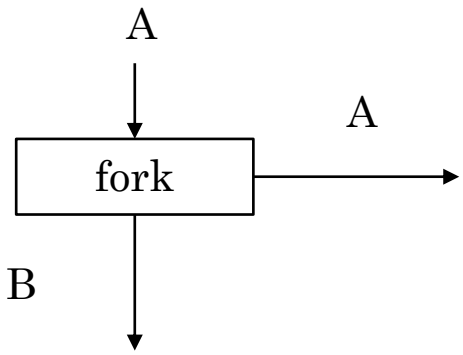
Un mapping link può anche essere associato ad un'interazione di input (o doppia).

Fork

Per seguire due o più percorsi in parallelo con la stessa entità si può usare una transizione fork. Il fork emette due token relativi alla stessa entità del token di input.

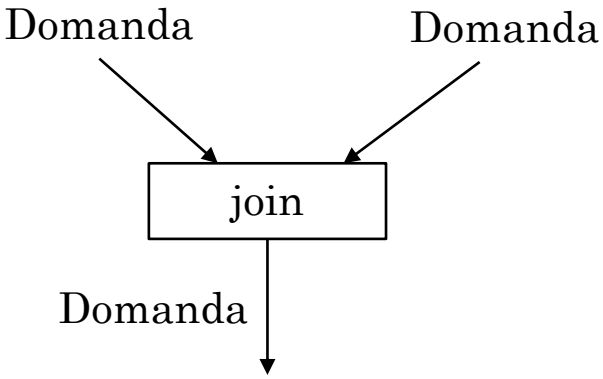


Attenzione: t1 e t2 devono accedere alla domanda in mutua esclusione.



Fork con mapping link

Join



Quando ha ricevuto due token relativi alla stessa domanda, il join emette un token relativo alla domanda.

Modelli B2B

Gestione Richieste di Preventivi (Tema 2015)

Il processo B2B Gestione Richieste di Preventivi consente ad un distributore di trattare le richieste di preventivi (RichiestaP) provenienti dai clienti. Una richiestaP porta un elenco di prodotti ciascuno dei quali si riferisce ad un tipo (Tipo) globale. Nel sistema informativo del distributore ai tipi sono associati dei fornitori. Ad ogni cliente è associato un account manager.

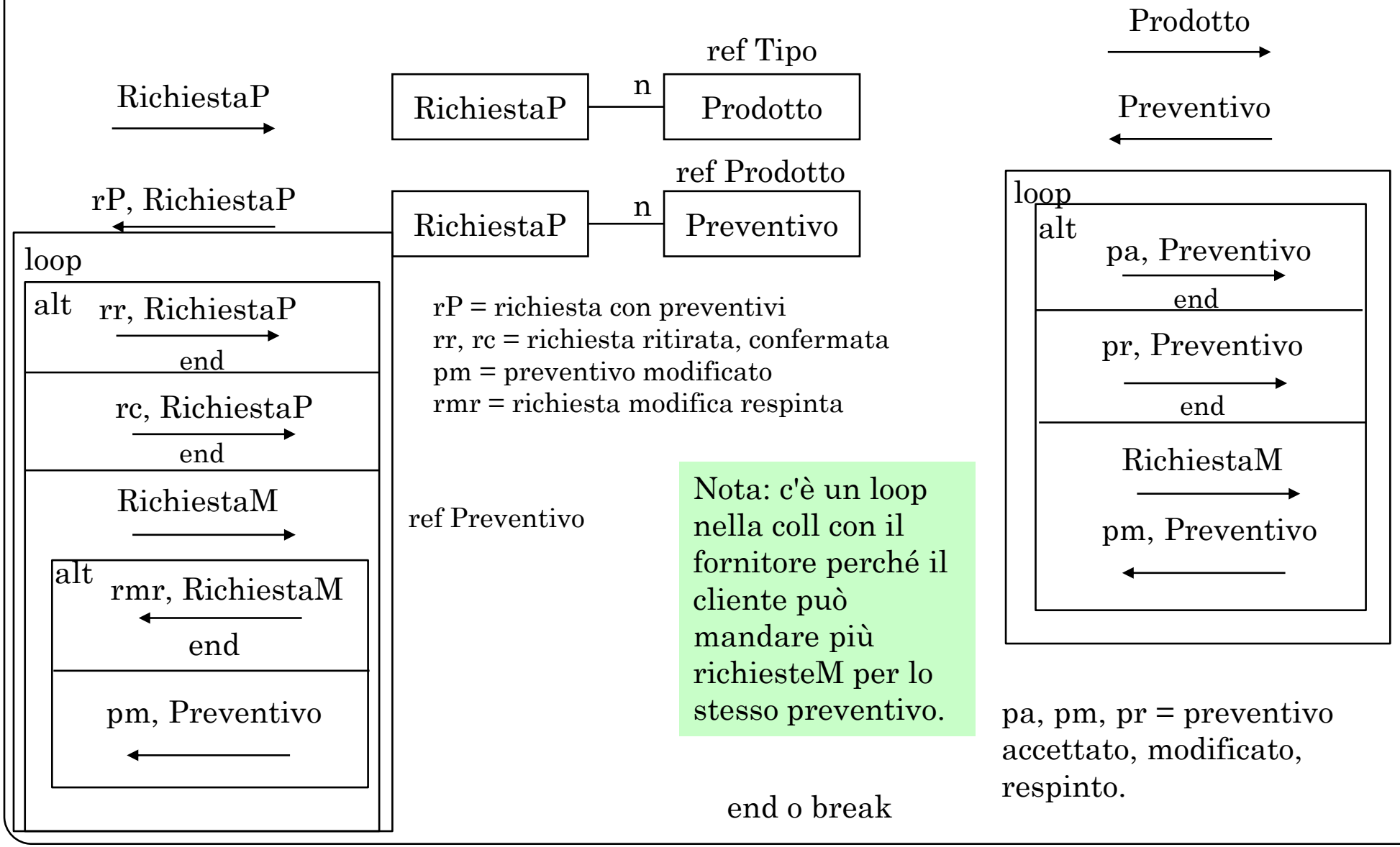
L'account manager assegna a ciascun prodotto di una richiestaP un fornitore idoneo (si usi un invariante) e il processo invia ciascun prodotto al fornitore corrispondente per ottenere un preventivo. Quando ha ricevuto tutti i preventivi, il processo invia al cliente la richiestaP con i preventivi (ciascuno relativo ad un prodotto): il cliente può confermare o ritirare la richiestaP oppure può mandare una richiesta di modifica (RichiestaM) per un preventivo. I primi due casi concludono la collaborazione e i fornitori sono informati dell'esito (accettato o respinto) dei loro preventivi.

Nell'ultimo caso, l'account manager può decidere se accettare o no la richiesta di modifica. Se l'accetta, invia la richiesta al fornitore che risponde con il preventivo modificato; allora il processo invia al cliente il preventivo modificato e il cliente può rispondere in tre modi come in precedenza. Se l'account manager non accetta la richiesta di modifica, il processo la rimanda al cliente come respinta; la collaborazione è quindi conclusa. Inoltre informa i fornitori che i preventivi sono stati respinti.

Cliente processo

CM

processo Fornitore



rP = richiesta con preventivi
rr, rc = richiesta ritirata, confermata
pm = preventivo modificato
rmr = richiesta modifica respinta

ref Preventivo

Nota: c'è un loop nella coll con il fornitore perché il cliente può mandare più richiesteM per lo stesso preventivo.

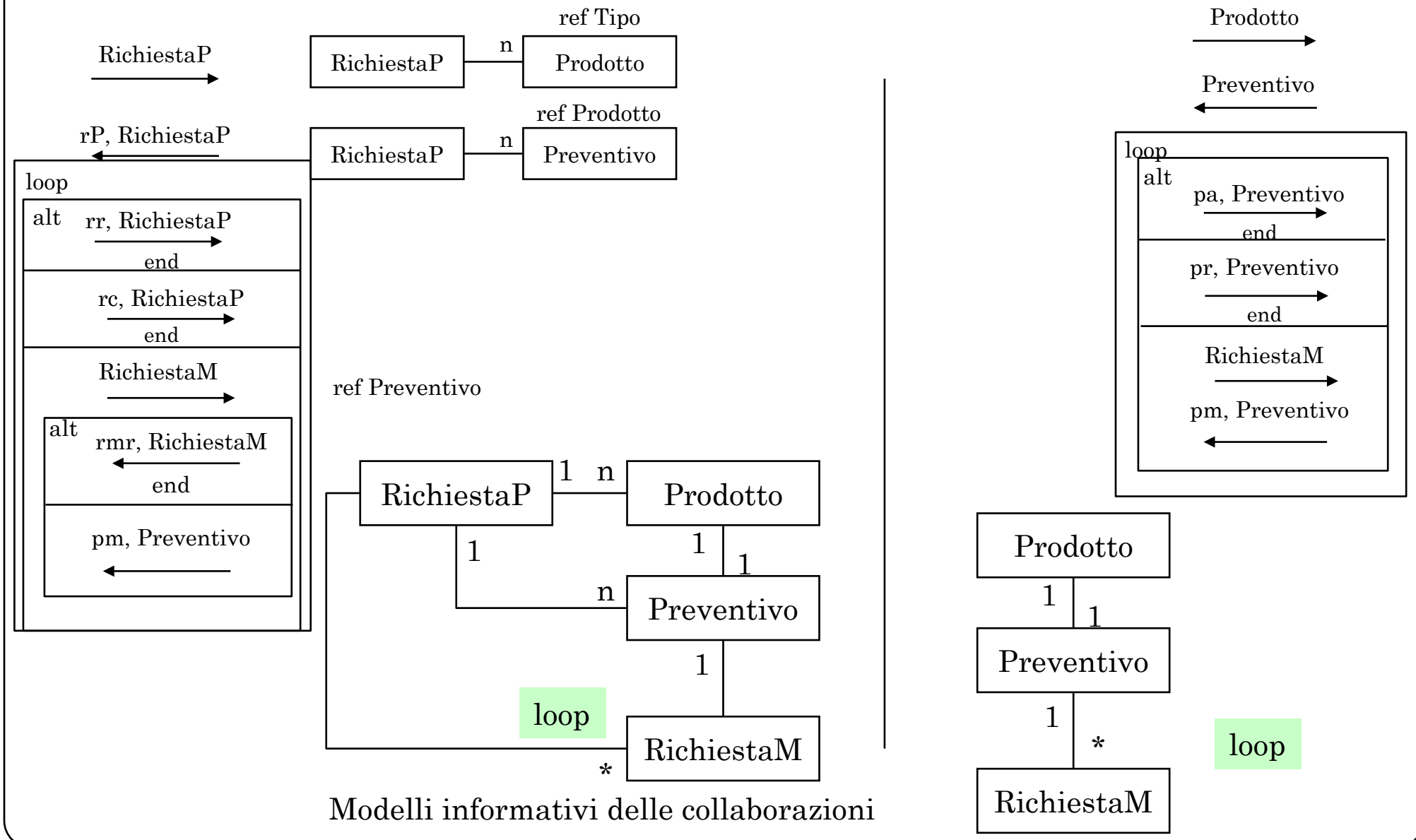
pa, pm, pr = preventivo accettato, modificato, respinto.

end o break

Modelli di collaborazione

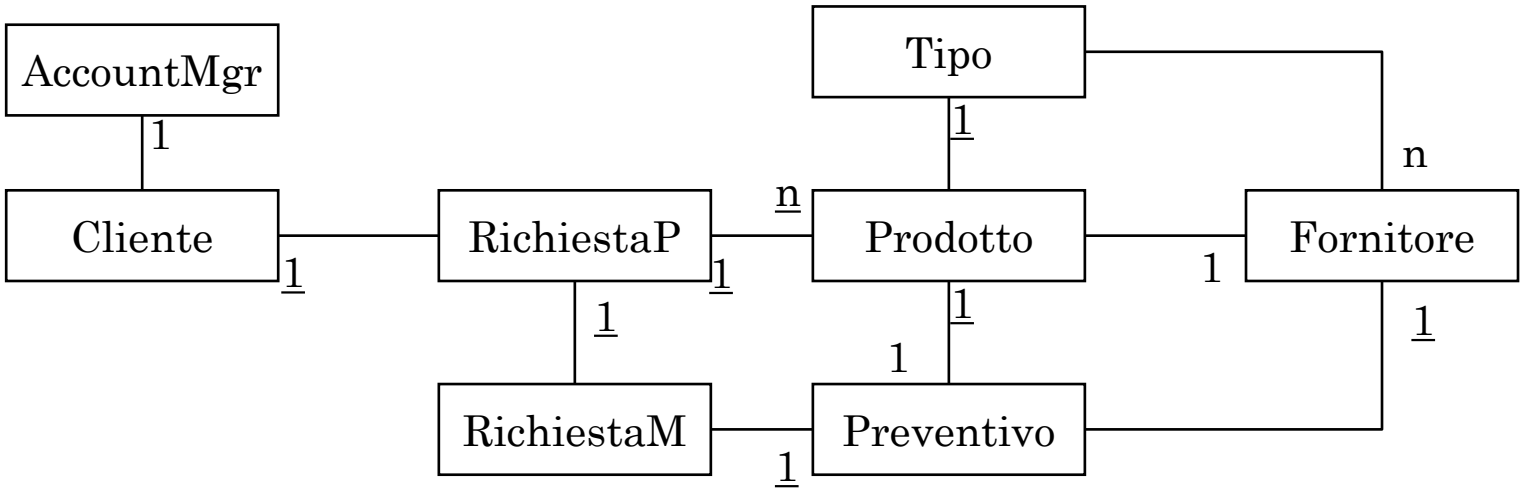
Cliente processo

processo Fornitore



Modelli informativi delle collaborazioni

MI



Invariante: prodotto.fornitore in prodotto.tipo.fornitori.

RichiestaM – AccountMgr = RichiestaM – Cliente – AccountMgr (può essere implicita)

RichiestaP – Preventivo = RichiestaP – Prodotto – Preventivo (anche implicita)

Note

Una richiestaM deve essere collegata a un preventivo e può essere collegata a una richiestaP e a un cliente. Il collegamento con la richiestaP lega la richiestaM all'entità che la precede nella collaborazione; il collegamento con il cliente indica il mittente della richiestaP. Questi collegamenti nel caso specifico non sono necessari in quanto si possono ottenere con relazioni derivate. Tuttavia il collegamento con la richiestaP risulta utile per ottenere i preventivi da respingere quando la richiestaM è respinta dall'accountMgr.

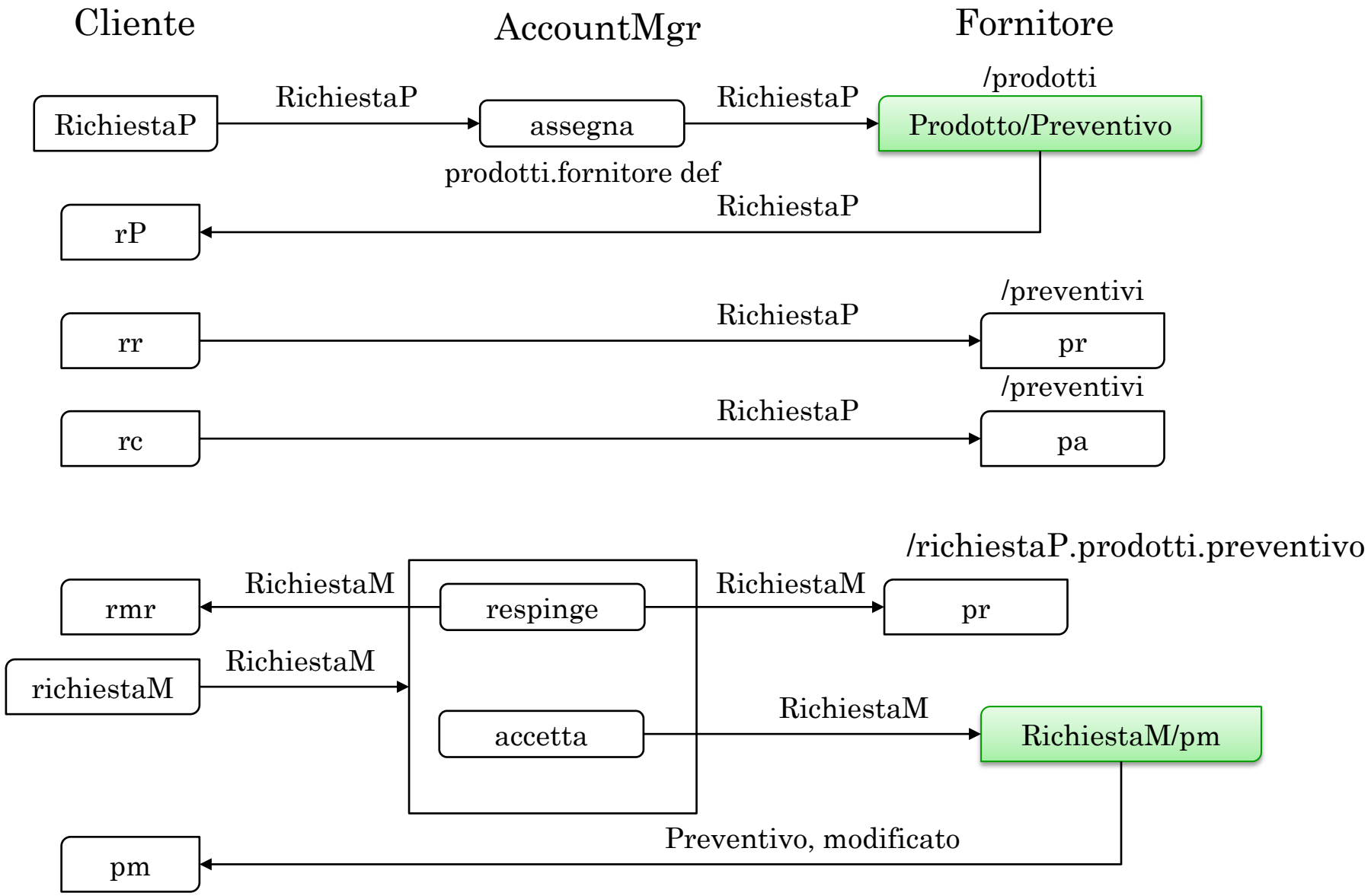
Il cliente si può ottenere con la relazione derivata

RichiestaM – RichiestaP – Cliente.

L'interazione rp deve portare i preventivi. Essi non sono direttamente collegati alla richiestaP ma si possono ottenere con la relazione derivata (anche implicita)

RichiestaP – Preventivo = RichiestaP – Prodotto – Preventivo

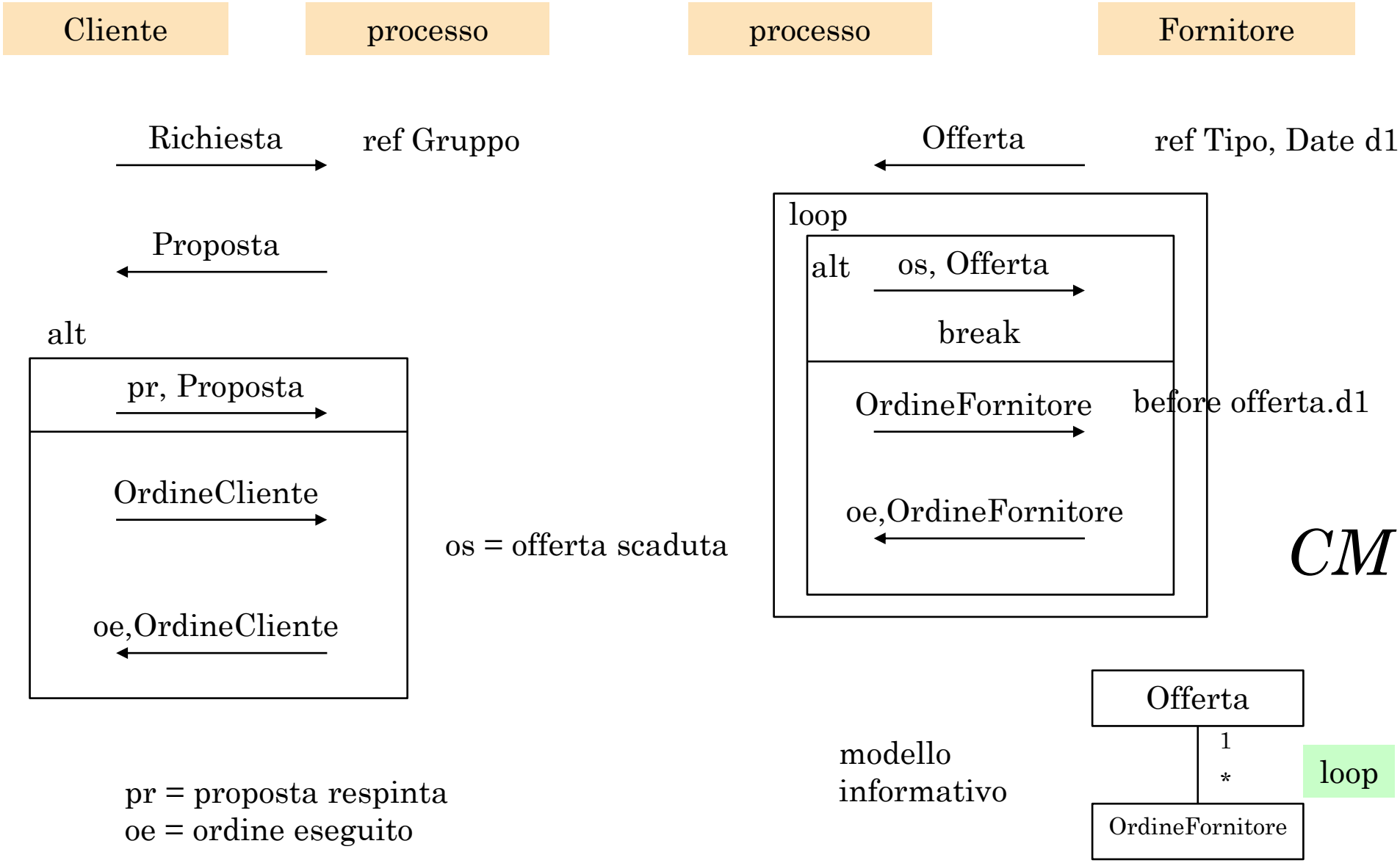
Processo

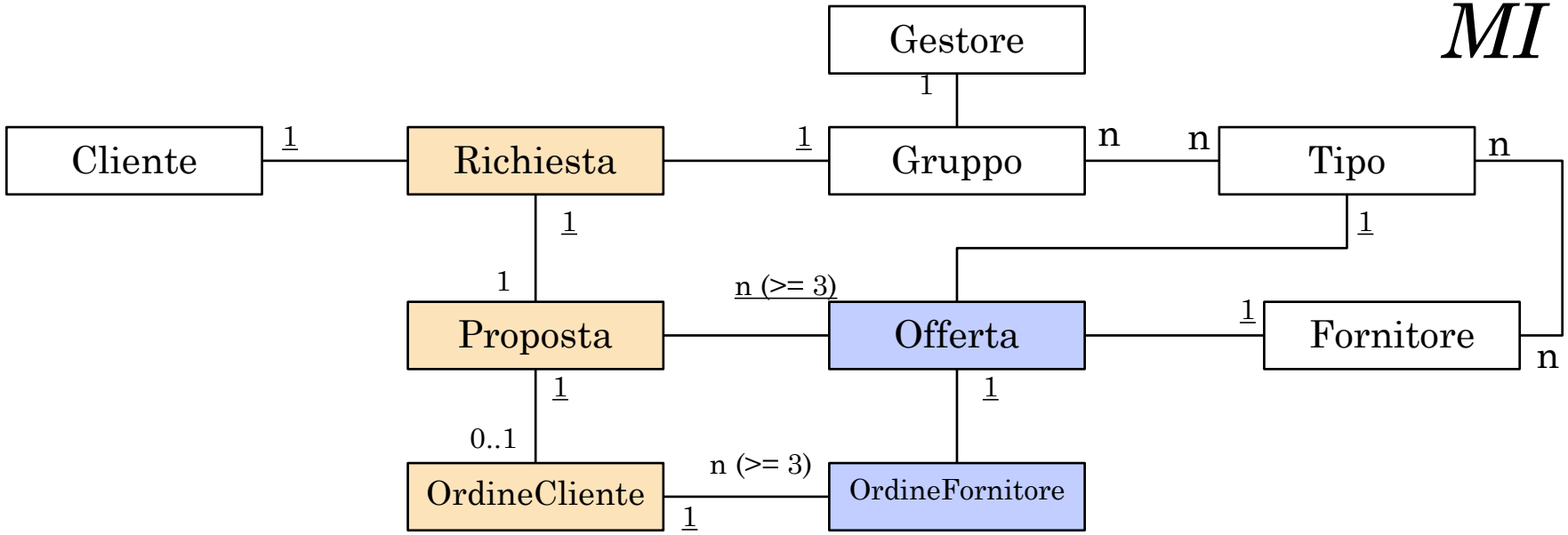

Cliente
AccountMgr
Fornitore
RichiestaP
RichiestaP
assegna
RichiestaP
/prodotti
Prodotto/Preventivo
prodotti.fornitore def
RichiestaP
rP
/preventivi
RichiestaP
rr
pr
/preventivi
RichiestaP
rc
pa
/richiestaP.prodotti.preventivo
RichiestaM
RichiestaM
rmr
respinge
pr
RichiestaM
richiestaM
RichiestaM
accetta
RichiestaM/pm
Preventivo, modificato
pm

Temi 2016

Es. 1

Il processo B2B opera in un'agenzia di intermediazione che tratta vari tipi di prodotti organizzati in gruppi (con relazione molti a molti). Ai tipi di prodotti sono associati vari fornitori. Ogni gruppo è associato ad un gestore (ruolo di staff).
Il processo serve le richieste dei clienti e le offerte dei fornitori. Una richiesta si riferisce ad un gruppo ed è quindi trattata dal gestore del gruppo. Un'offerta si riferisce ad un tipo di prodotto e ha una scadenza d1.
Il gestore tratta una richiesta generando una proposta che comprende almeno 3 offerte con il vincolo che il gruppo indicato dalla richiesta includa i tipi di tutte le offerte associate alla proposta (si definisca un invariante). Le offerte rimangono disponibili per altre proposte, finché non giungono a scadenza. Al contrario le richieste non sono più disponibili.
Il processo invia la proposta al mittente della richiesta, il quale può rifiutarla oppure può inviare un ordine cliente collegato alla proposta. Nel primo caso il gestore scrive respinta nello stato della proposta. Nel secondo caso il gestore produce un ordine fornitore per ogni offerta collegata alla proposta (un ordine fornitore è collegato ad un'offerta e all'ordine cliente) e il processo manda tali ordini ai fornitori. I fornitori rispondono con il messaggio ordine fornitore eseguito; il processo quando ha ricevuto tali messaggi per tutti gli ordini fornitori relativi allo stesso ordine cliente risponde al cliente con il messaggio ordine cliente eseguito.
Quando un'offerta scade il processo informa il fornitore che l'offerta è scaduta.


Cliente
processo
processo
Fornitore
Richiesta
ref Gruppo
Offerta
ref Tipo, Date d1
Proposta
loop
alt
os, Offerta
break
alt
pr, Proposta
OrdineFornitore
before offerta.d1
OrdineCliente
oe,OrdineFornitore
os = offerta scaduta
CM
oe,OrdineCliente
Offerta
1
*
loop
OrdineFornitore
modello
informativo
pr = proposta respinta
oe = ordine eseguito

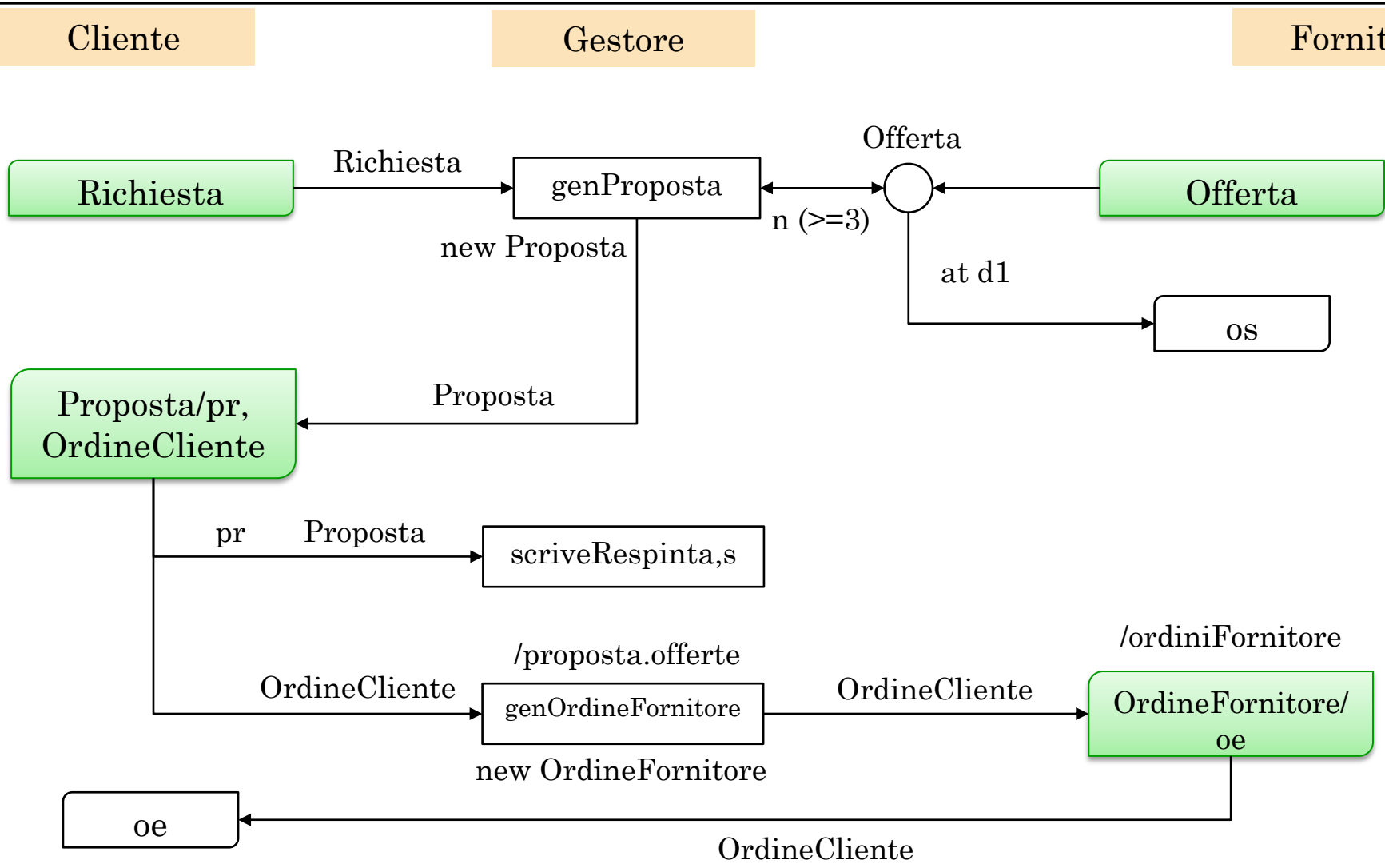


Offerta: Date d1. Proposta: stato (respinta).

Invariante
proposta.offerte.tipo in proposta.richiesta.gruppo.tipi.

oppure
richiesta.proposta.offerte.tipo in richiesta.gruppo.tipi

Nota:
Un ordineCliente è legato al mittente tramite la relazione derivata (anche implicita)
OrdineCliente – Proposta – Richiesta – Cliente.


Cliente
Gestore
Fornitore
Richiesta
Richiesta
genProposta
Offerta
n (>=3)
Offerta
new Proposta
at d1
os
Proposta/pr,
OrdineCliente
Proposta
pr
Proposta
scriveRespinta,s
/proposta.offerte
/ordiniFornitore
OrdineCliente
genOrdineFornitore
OrdineCliente
OrdineFornitore/
oe
new OrdineFornitore
oe
OrdineCliente

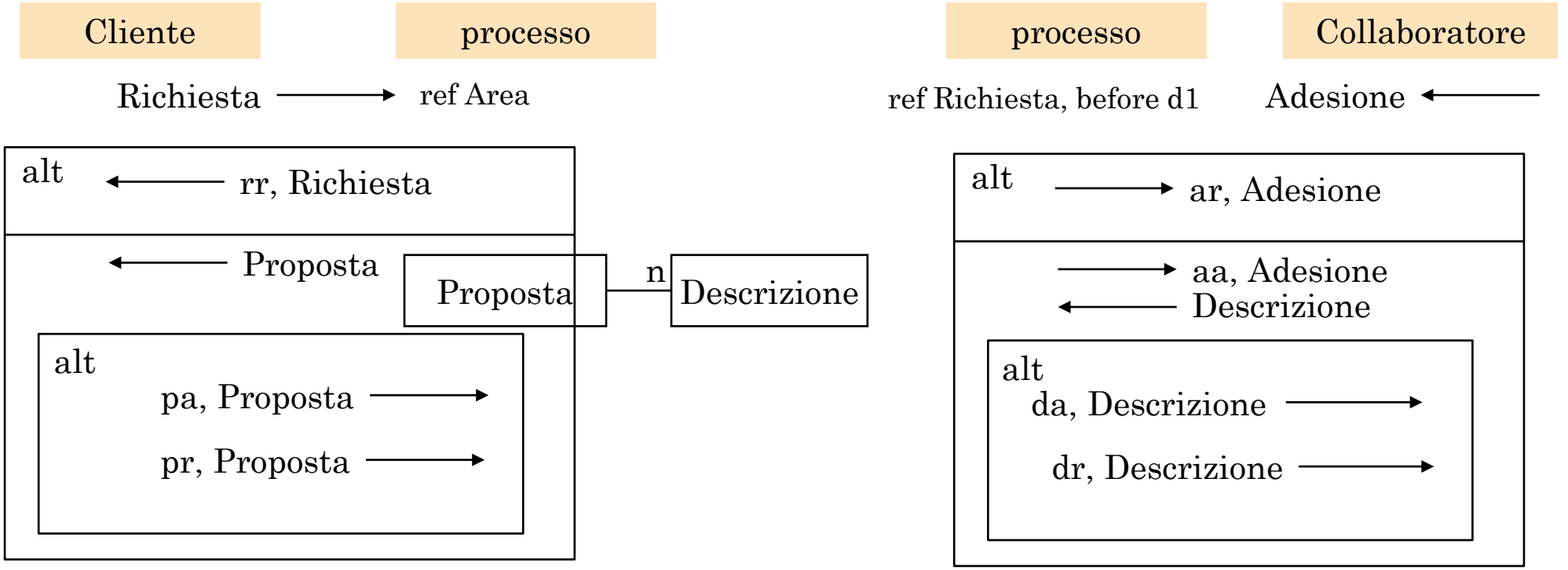
Processo

Es. 2

Il processo B2B opera in un'agenzia di supporto a progetti internazionali, il cui scopo è di mettere in contatto clienti e collaboratori. Il sistema informativo comprende i clienti, varie aree progettuali e vari collaboratori associate alle aree. Ogni area è trattata da un gestore (ruolo di staff).

Il processo riceve richieste di collaborazione dai clienti; una richiesta si riferisce ad un'area. Il gestore fissa la scadenza d1 della richiesta. Prima di d1 i collaboratori associati all'area della richiesta possono mandare un'adesione alla richiesta; si esprima con un invariante il vincolo che la richiesta a cui si riferisce un'adesione è relativa ad un'area trattata dal collaboratore. Al tempo d1 se ci sono meno di 3 adesioni il processo respinge ai mittenti sia le adesioni sia la richiesta. Altrimenti il gestore sceglie almeno 3 adesioni (scrivendo "accettata" nel loro stato) e respinge le altre. Il processo informa i mittenti con i messaggi adesione accettata o adesione respinta.

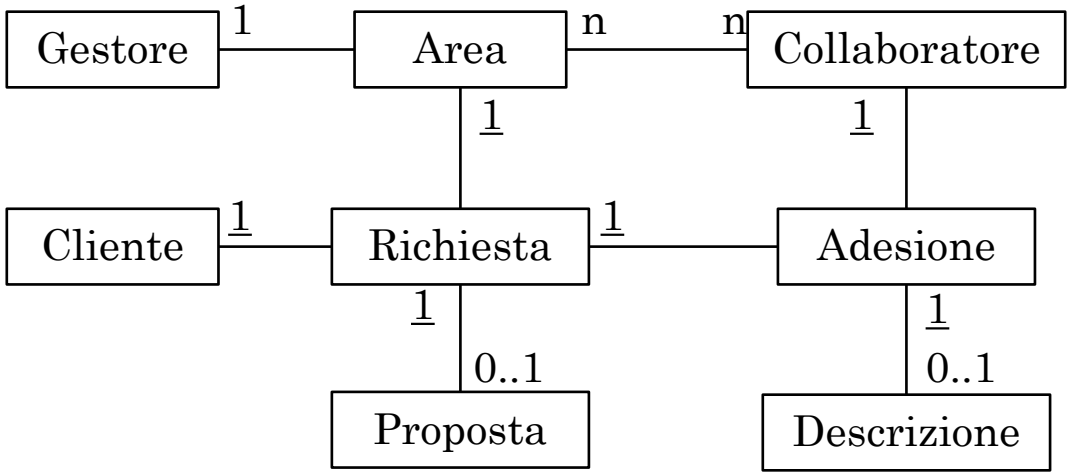
Ciascun collaboratore relativo ad un'adesione accettata risponde inviando la descrizione di ciò che intende svolgere; la descrizione è collegata all'adesione. Ricevute tutte le descrizioni il gestore produce una proposta di collaborazione (relativa alla richiesta). Il processo manda la proposta con le descrizioni al cliente, il quale può accettare o respingere la proposta. Il processo informa i collaboratori dell'esito delle loro descrizioni con i messaggi "descrizione accettata (da)" o "descrizione respinta (dr)".



rr = richiesta respinta
pa, pr = proposta accettata, respinta

aa, ar = adesione accettata, respinta
da, dr = descrizione accettata, respinta

CM



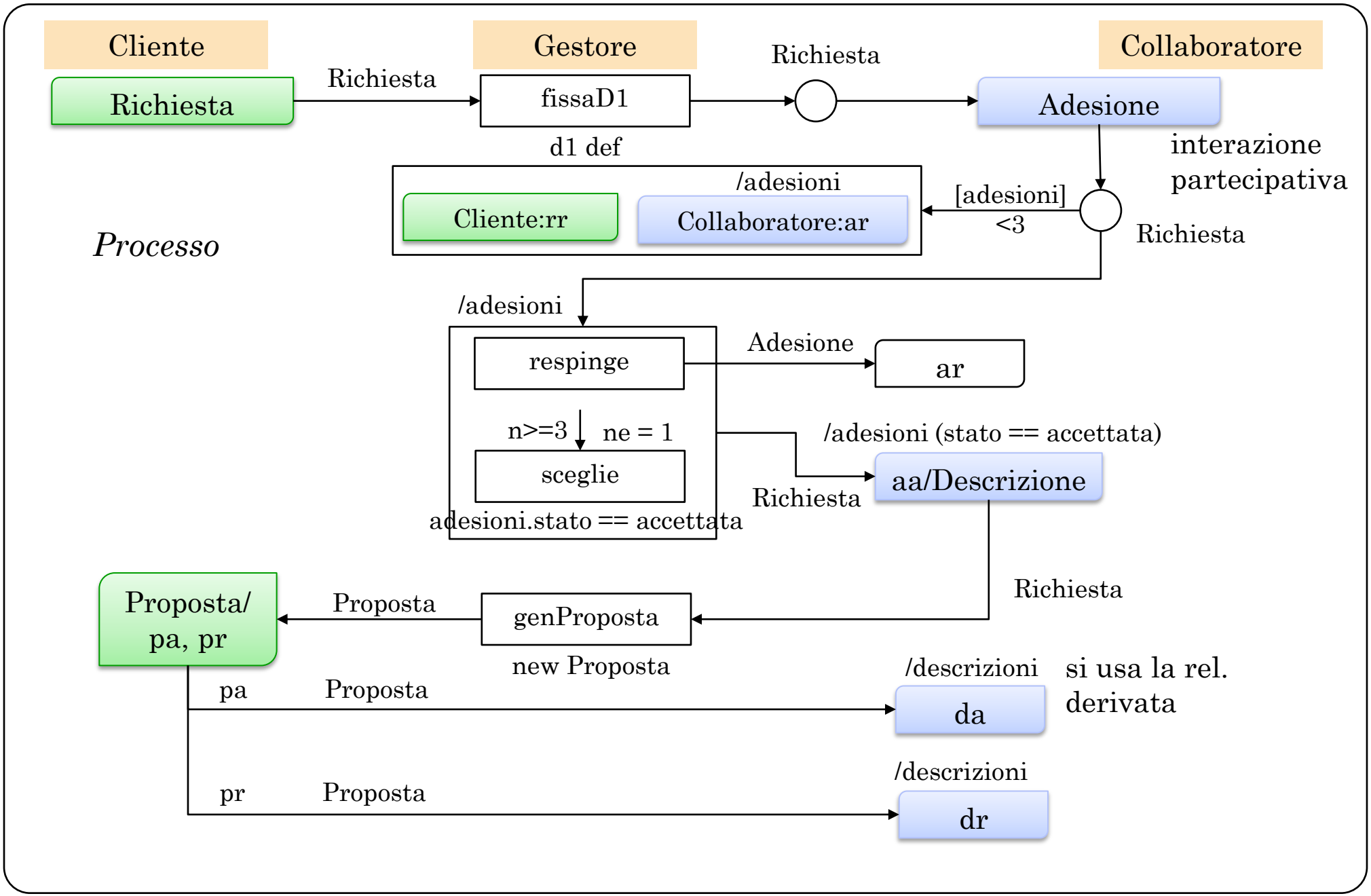
MI

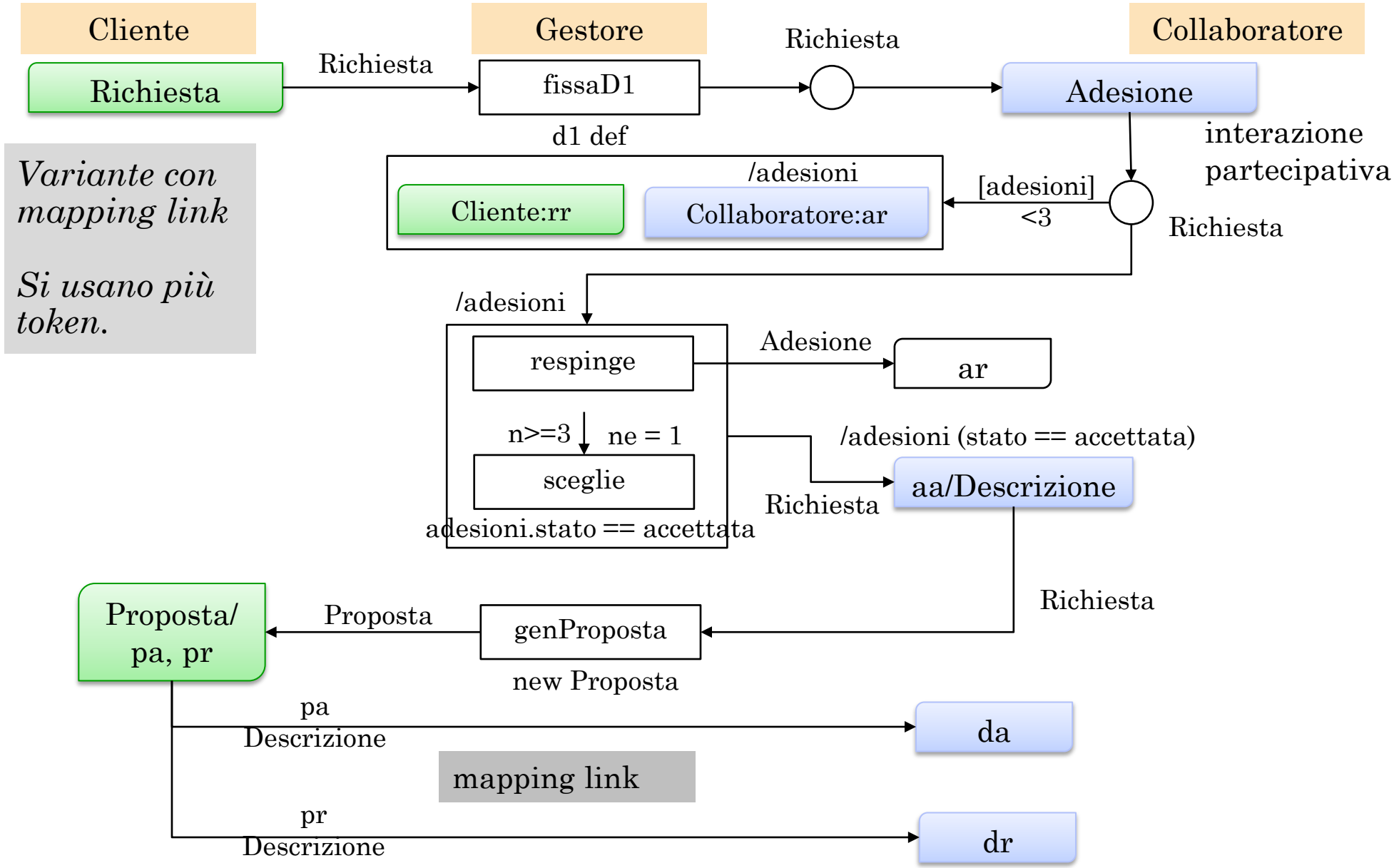
Richiesta: Date d1.
Adesione: stato (accettata)

Invariante: adesione.richiesta.area in adesione.collaboratore.aree

Relazione derivata (anche implicita):
Proposta – Descrizione = Proposta – Richiesta – Adesione – Descrizione

Nota: L'invariante può essere espresso anche in altri modi in base al circuito che collega le classi Richiesta, Area, Collaboratore, Adesione.
L'espr. "richiesta.area in richiesta.adesioni.collaboratore.aree è errata".


Cliente
Gestore
Collaboratore
Richiesta
Richiesta
fissaD1
Richiesta
Adesione
d1 def
interazione partecipativa
/adesioni
Cliente:rr
Collaboratore:ar
[adesioni] <3
Richiesta
Processo
/adesioni
respinge
Adesione
ar
n>=3
ne = 1
sceglie
adesioni.stato == accettata
/adesioni (stato == accettata)
aa/Descrizione
Richiesta
Proposta/ pa, pr
Proposta
genProposta
new Proposta
Richiesta
/descrizioni
si usa la rel. derivata
pa
Proposta
da
/descrizioni
pr
Proposta
dr

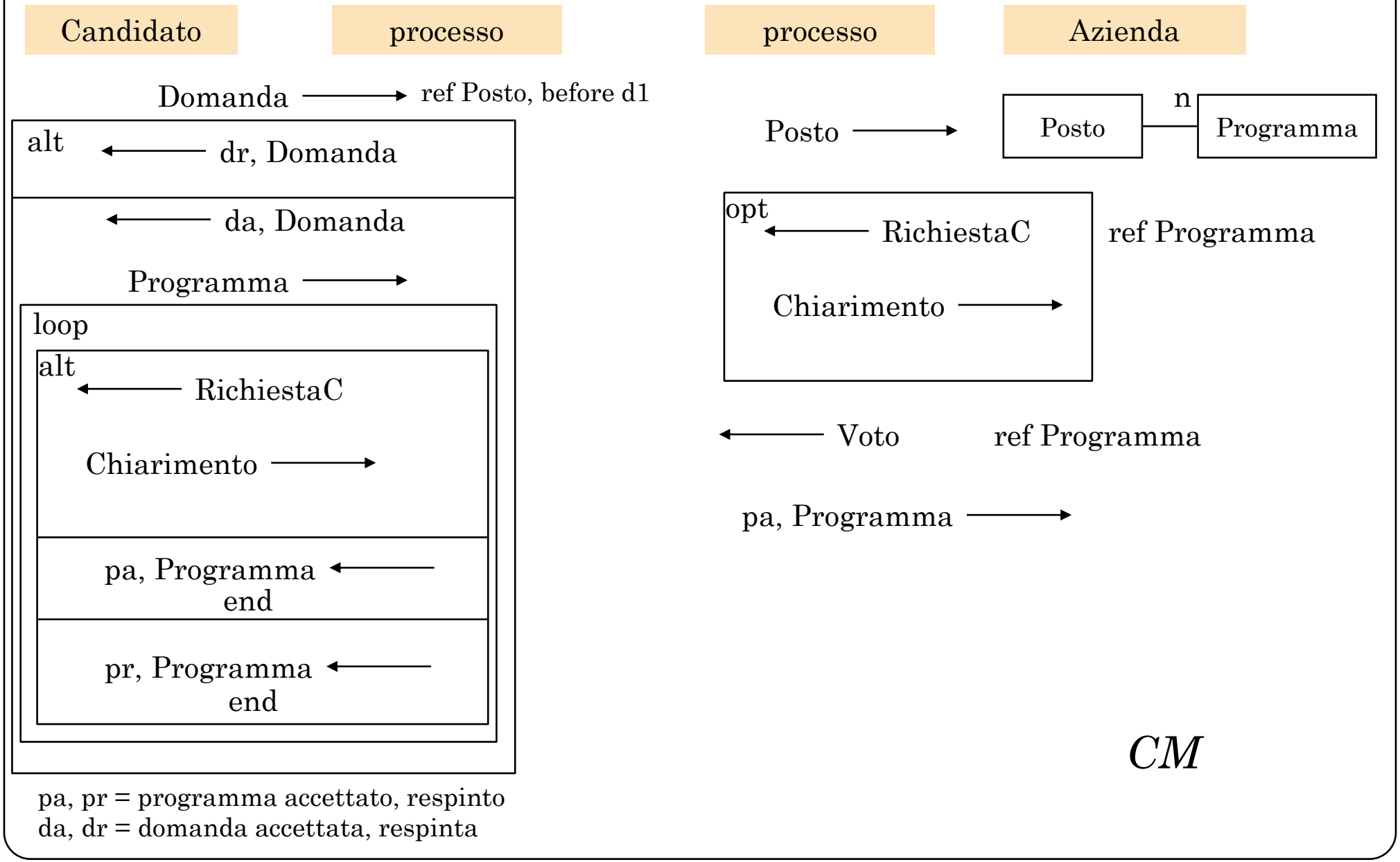


si usa la rel. derivata

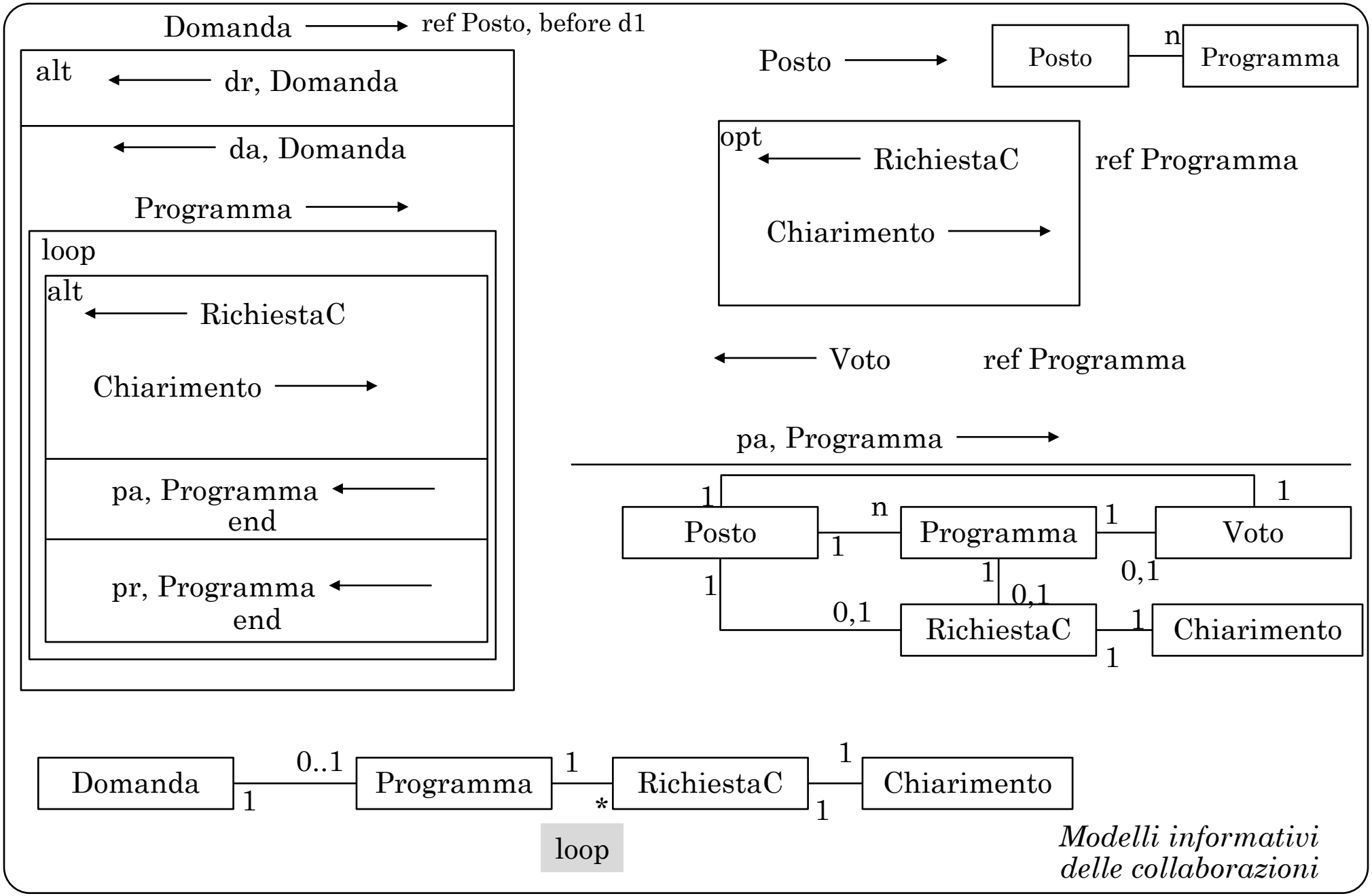
Es. 3

Il processo B2B opera in un centro di gestione interaziendale che ha lo scopo di gestire concorsi per posti direzionali vacanti. Un posto vacante è inserito nel processo da una collaborazione non considerata; il posto è relativo ad una responsabilità. Alle responsabilità sono associate varie aziende e vari candidati; una responsabilità è seguita da un gestore (ruolo di staff).

Il gestore corrispondente fissa la scadenza d1 del posto; entro d1 i candidati possono inviare una domanda per il posto (si esprima con un invariante il vincolo che la responsabilità relativa al posto al quale si riferisce la domanda sia inclusa nelle responsabilità alle quali è interessato il mittente della domanda). Il gestore deve accettare almeno 3 domande e respingere le altre; i candidati sono informati con i messaggi domanda accettata o domanda respinta. I mittenti delle domande accettate inviano un programma di lavoro. Quando ha ricevuto tutti i programmi, il processo manda il posto con i programmi alle aziende interessate alla responsabilità relativa al posto. Queste possono chiedere chiarimenti sul programma di un candidato e il processo passa la richiesta di chiarimento al candidato che risponde con un chiarimento, poi passato dal processo all'azienda. Un'azienda può chiedere al più un chiarimento poi esprime un voto su un programma; un candidato può però ricevere più richieste di chiarimento sullo stesso programma da parte di aziende diverse. Ricevuti tutti i voti, il gestore accetta il programma con il massimo dei voti. Il processo informa i candidati dell'esito dei programmi con i messaggi programma accettato o programma respinto; comunica inoltre alle aziende il programma accettato con il messaggio programma accettato.

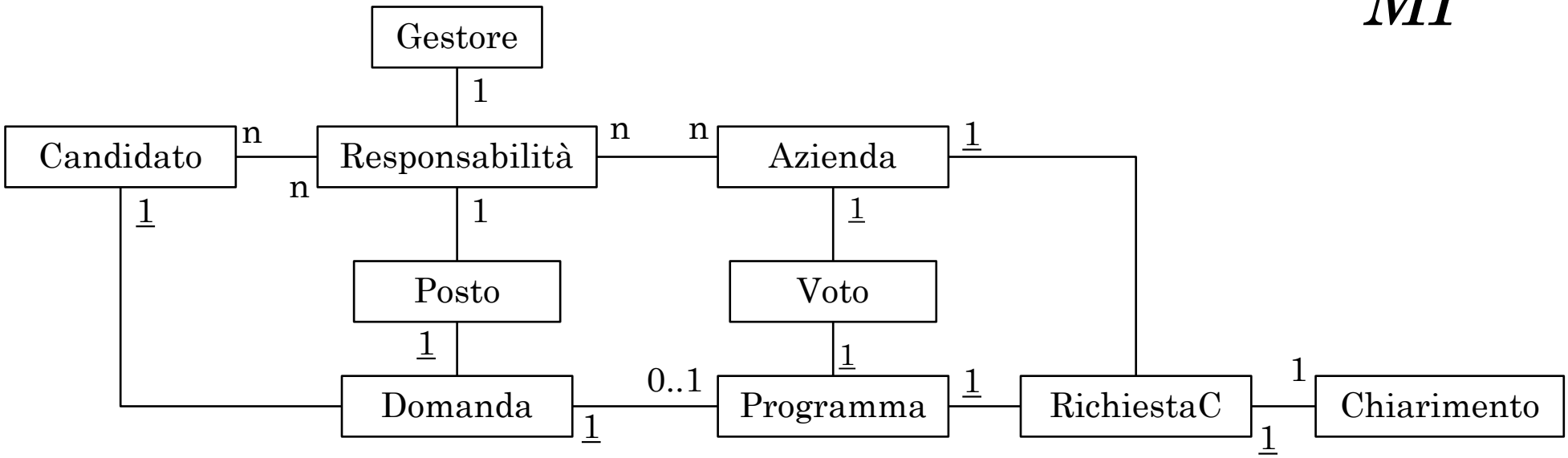


pa, pr = programma accettato, respinto
da, dr = domanda accettata, respinta



Modelli informativi delle collaborazioni

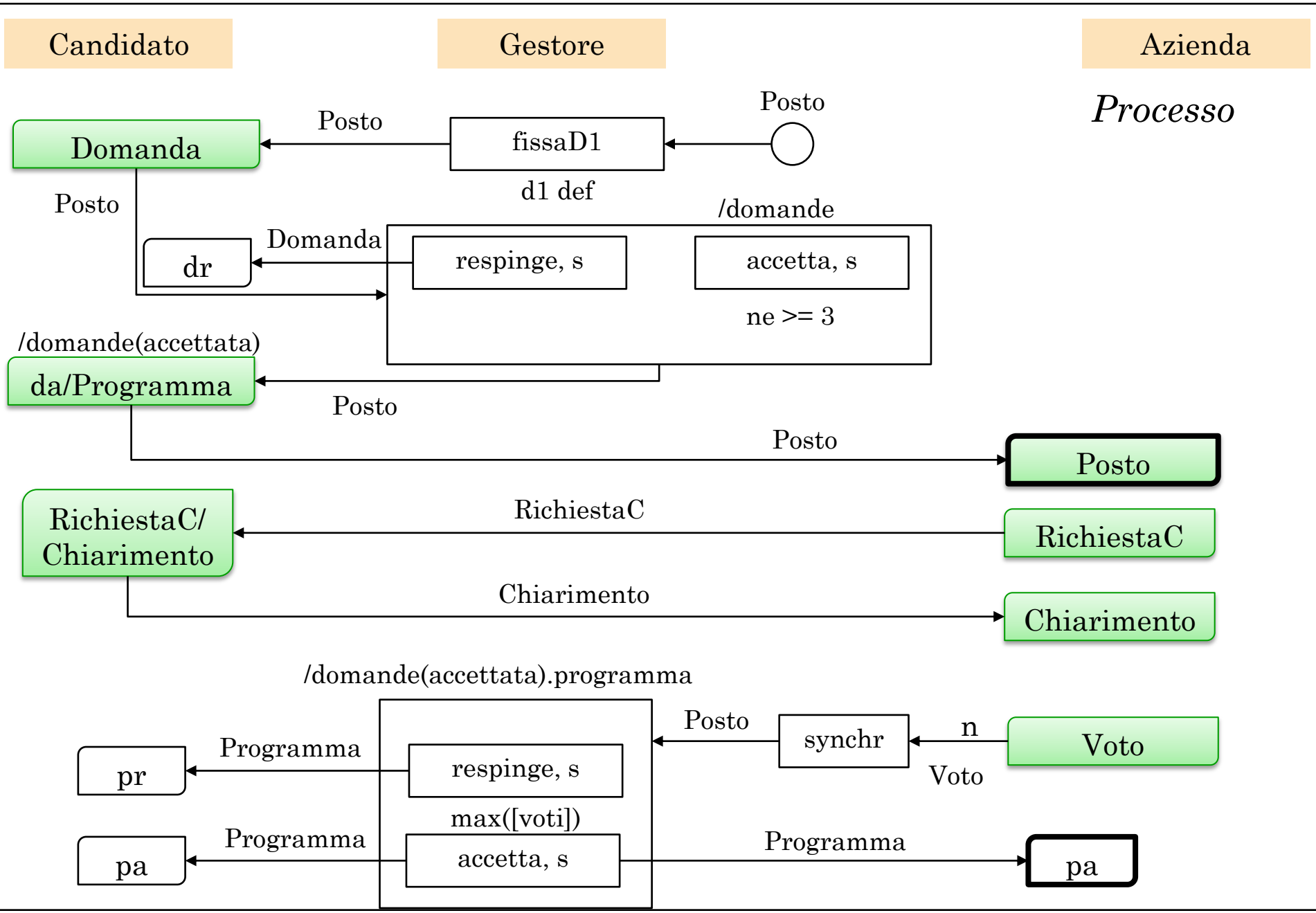
MI



Posto: Date d1.
Domanda: stato (accettata, respinta); boolean accettata = stato == accettata.
Programma: stato (accettato, respinto).

Invariante: domanda.posto.responsabilità in domanda.candidato.responsabilità

Relazione derivate (anche implicite)
Posto – Programma = Posto – Domanda – Programma
Voto – Posto = Voto – Programma – Domanda – Posto
La relazione diretta Voto – Posto è ridondante.


Candidato
Gestore
Azienda
Processo
Domanda
Posto
fissaD1
Posto
d1 def
Posto
/domande
dr
Domanda
respinge, s
accetta, s
ne >= 3
/domande(accettata)
da/Programma
Posto
Posto
Posto
RichiestaC/
Chiarimento
RichiestaC
RichiestaC
Chiarimento
Chiarimento
/domande(accettata).programma
Posto
synchr
n
Voto
Voto
pr
Programma
respinge, s
max([voti])
pa
Programma
accetta, s
Programma
pa

Note

Un programma è collegato direttamente ad una domanda e indirettamente al mittente (il candidato relativo alla domanda).

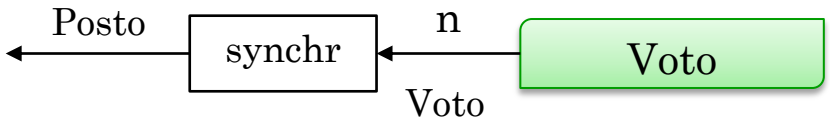
Una richiesta di chiarimento entra nell'istanza di collaborazione che contiene il programma al quale si riferisce.

Un chiarimento entra nell'istanza di collaborazione che contiene la richiesta di chiarimento alla quale si riferisce.

Un voto è collegato direttamente ad un programma e al mittente (un'azienda) e indirettamente ad un posto.

L'interazione pa con le aziende entra in tutte le istanze di collaborazione che contengono il programma indicato nel payload.

Sincronizzatore

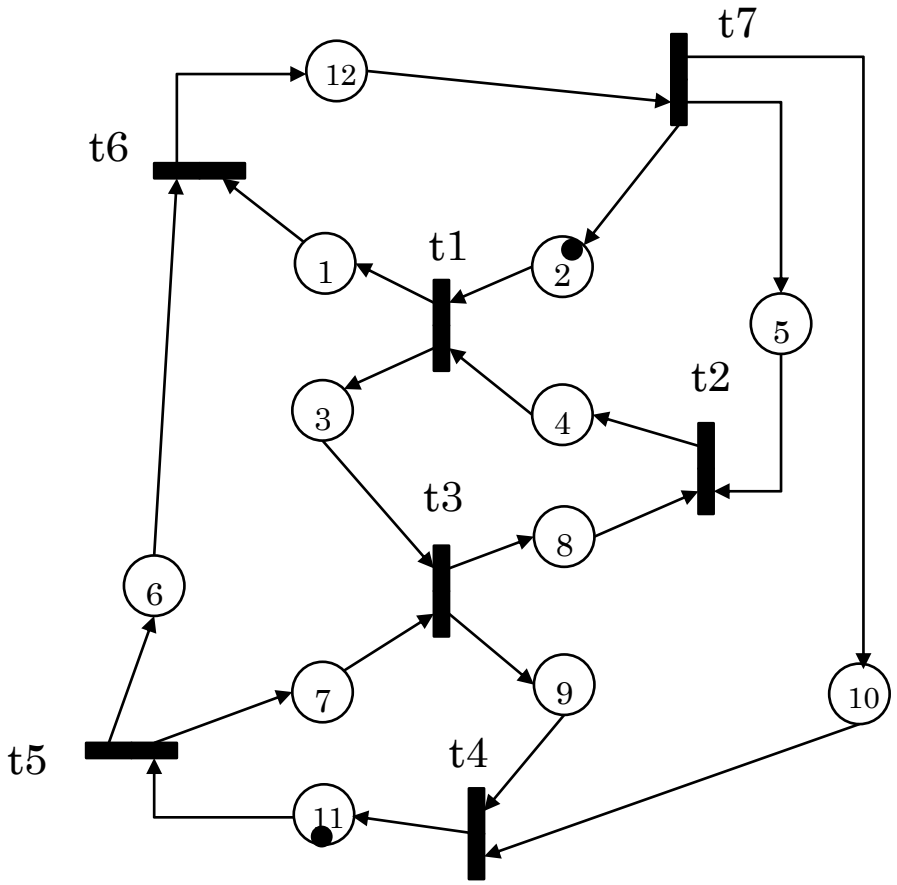


n = [posto.responsabilità.aziende]

I voti sono raggruppati per posto (in base alla relazione derivata) e il loro numero è pari al numero delle aziende collegate alla responsabilità relativa al posto come indicato dal parametro n.

PN

Es. 1



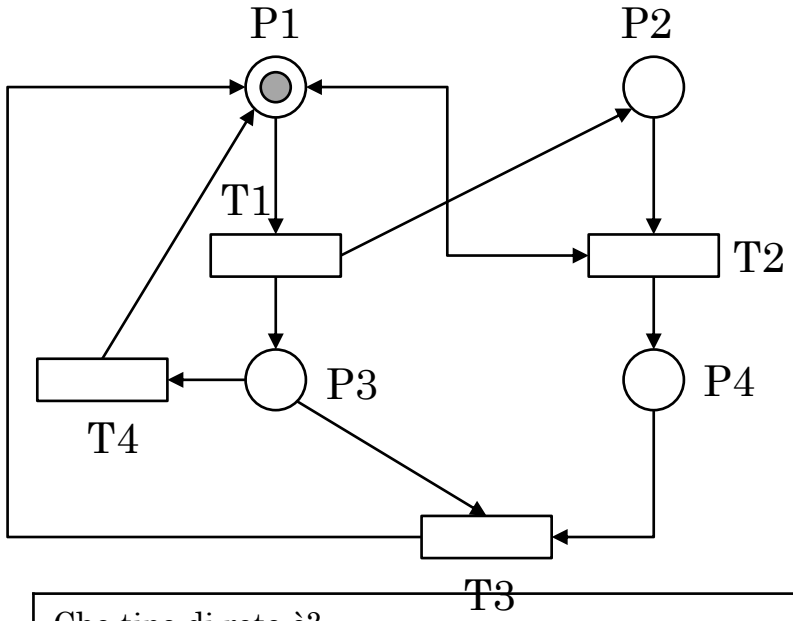
8 circuiti
0 [1, 12, 2]
1 **[1, 12, 5, 4]**
2 [1, 12, 10, 11, 7, 8, 4]
3 [2, 3, 9, 11, 6, 12]
4 **[3, 8, 4]**
5 [3, 9, 11, 6, 12, 5, 4]
6 [6, 12, 10, 11]
7 [7, 9, 11]

Punti delle domande: 1, 1, 2, 2, 2

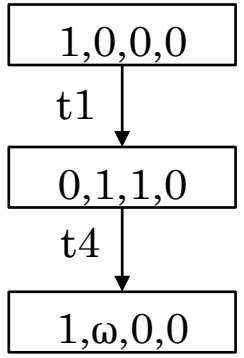
Tabella

Quanti sono i circuiti?	8
Quali sono i circuiti di base?	[1, 12, 2] [3, 8, 4] [7, 9, 11]
Ci sono circuiti privi di token? se sì, quanti e quali sono?	2 circuiti senza token [3, 8, 4] [1, 12, 5, 4]
Si renda la rete live e safe con una sola variazione della marcatura iniziale; quali sono i posti marcati nella nuova marcatura?	[**2**, 4, **11**]
Quanto vale il tempo ciclo con la marcatura scelta nell'ipotesi che tutte le transizioni abbiano durata pari a 1 tranne t4 che ha durata 2. Qual è il circuito che lo determina?	Il tempo ciclo vale 5 e il circuito è [6,12,10,11]

Es. 2



subset [1, 2] [+-, +-, +, +, t]
subset [1, 3] [+-, +-, +-, +-, st]
subset [1, 2, 3] [+-, +-, +-, +-, st]
subset [1, 2, 4] [+-, +-, +-, +, t]
subset [1, 3, 4] [+-, +-, +-, +-, st]
the net is live



Che tipo di rete è?	AC in p1, p2 e in p3,p4
La rete è live? Ci sono sifoni che non contengono trappole marcate inizialmente; se sì quali?	Sì; no
Ci sono sifoni uguali a trappole; se sì, quali?	(1,3), (1,2,3) (1,3,4)
La rete ha dei deadlock o no? Se sì con quale marcatura?	no
La rete è bounded? Se no in quali posti e perché?	è unbounded in p2 e p4
La rete è safe o no e perché?	no, perché unbounded
La rete è reversibile o no e perché?	no; T3 consuma token da P4 ma è preceduta da T1 che aggiunge un token a P2
Nel grafo delle marcature come sono scritte le marcature che si ottengono con due scatti di transizione da M0?	(1,ω,0,0)

Testing

Es. 1

```
static int wbt1 (int a, int b,
int c, int d) {
int r = -1;
if (b >= c) return r;              1
else {
  if (a > b && a < c) {            2
     if (c > d) r += c;            3
     else r += d;
     r += 100;
     if (c > d) return c;          4
  } else {
     if (a > d) return a;          5
  }
  return d;
}
}
```

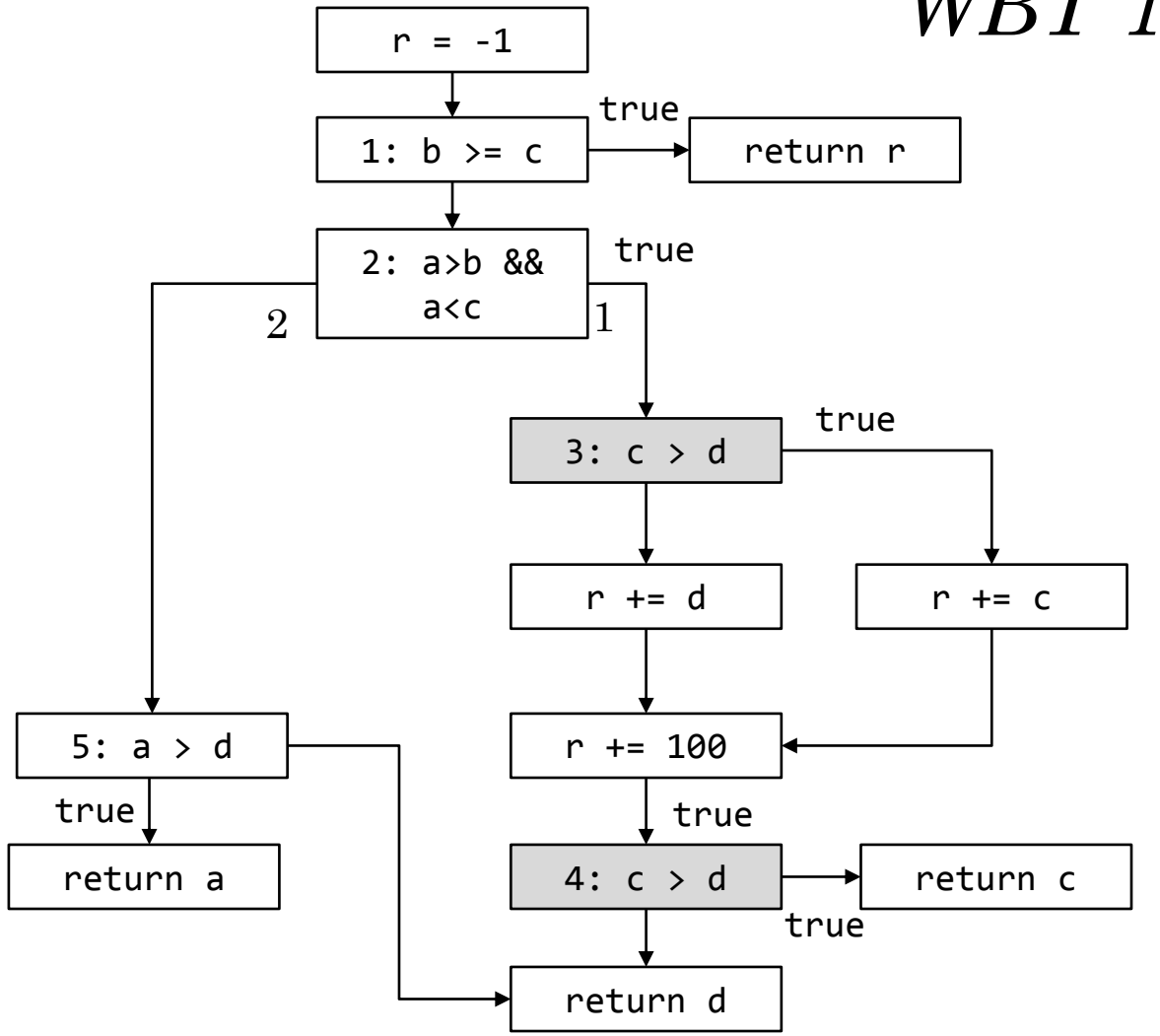
N. min di test per la copertura dei criteri seguenti; si spieghi il valore.
Nodi
Link (edge)
Percorsi
Condizioni multiple
N. min test per tutti i criteri:

Nota: si numerino le condizioni in ordine di comparsa nel programma.

Punti delle domande: 1, 1, 2, 1, 2 + 1 per il grafo

WBT 1

N. min di test per la copertura dei criteri seguenti; si spieghi il valore.
Nodi 4 1T 1F2T(3T4T,3F,4F) 1F2F5T
Link (edge) **5**: i 4 dei nodi più un percorso con il link da a>d a return d 1F2F5T
Percorsi **5**: come sopra (data la correlazione c>d)
Condizioni multiple **3** casi di test: a<=b; a>b && a<c; a>=c 1F2T; 1F2F; 1F2F
N. min test per tutti i criteri **5**: bastano i casi di test relativi ai percorsi; la condizione multipla non necessita di casi supplementari.



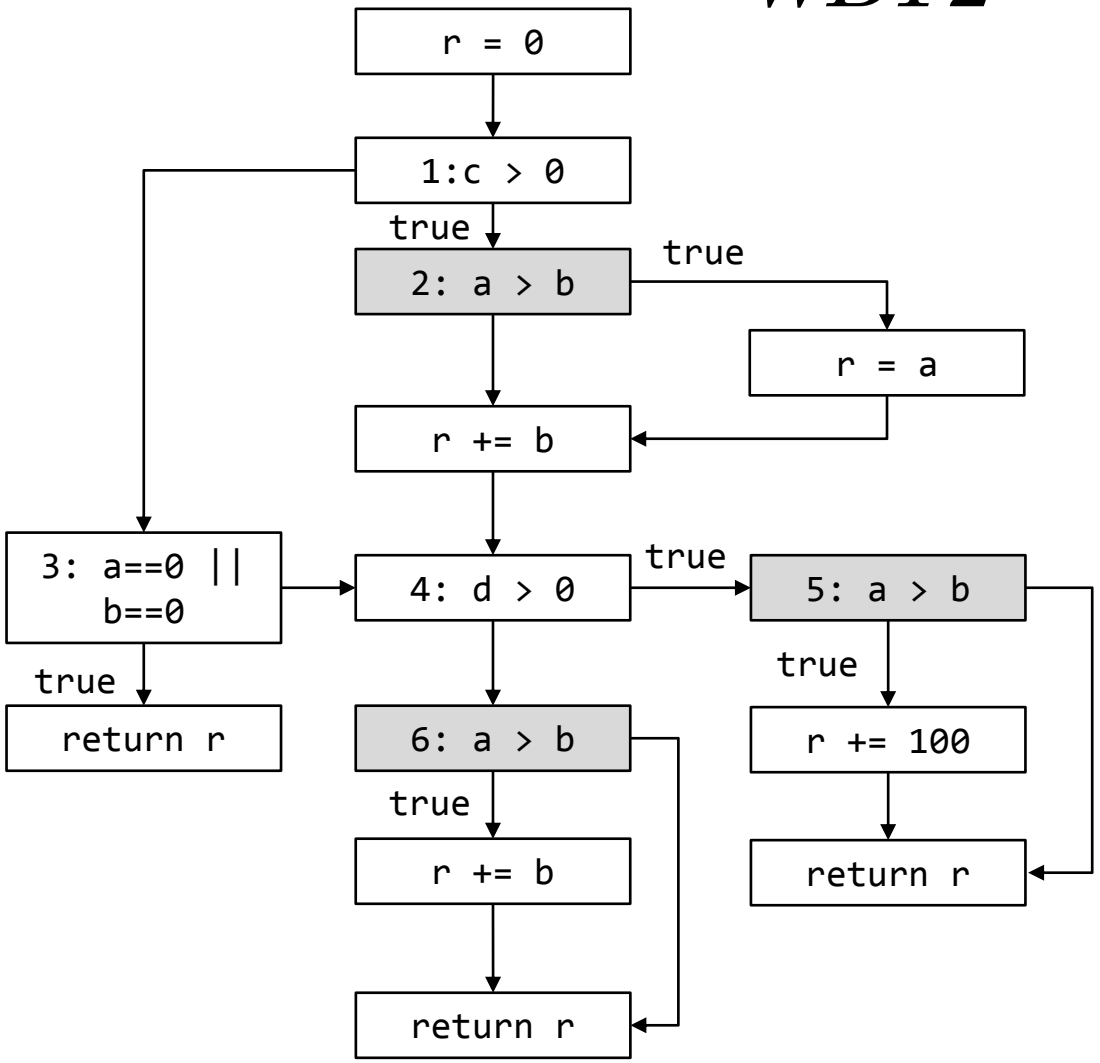
correlazione

Es. 2

```
static int wbt2 (int a, int b,
int c, int d) {
int r = 0;
if (c > 0) {                        1
 if (a > b) r = a;                  2
 r += b;
} else {
 if (a==0 || b==0) return r;        3
}
if (d > 0) {                        4
  if (a > b) r += 100;              5
  return r;
} else {
  if (a > b) r += b;                6
  return r;
}
}
```

N. min di test per la copertura dei criteri seguenti; si spieghi il valore.
Nodi
Link (edge)
Percorsi
Condizioni multiple
N. min test per tutti i criteri:

WBT2



N. min di test per la copertura dei criteri seguenti; si spieghi il valore.
Nodi 3: 1T2T(4T5T,4F6T); 1F3T
Link (edge) 5: si aggiungono ai precedenti 1T2F4T5F;1F3F4F6F
Percorsi 9: se c>0 è true servono 4 percorsi date le correlazioni; se è false 5 1T2T(4T5T,4F6T) 1T2F(4T5F,4F6F) 1F3T; 1F3F(4T(5),4F(6)
Condizioni multiple 4 per la condizione doppia
N. min test per tutti i criteri: 11: l'or doppio chiede 3 casi di test per l'uscita true mentre per i percorsi ne basta 1

Nota: è un errore accorpare i 3 "return r".

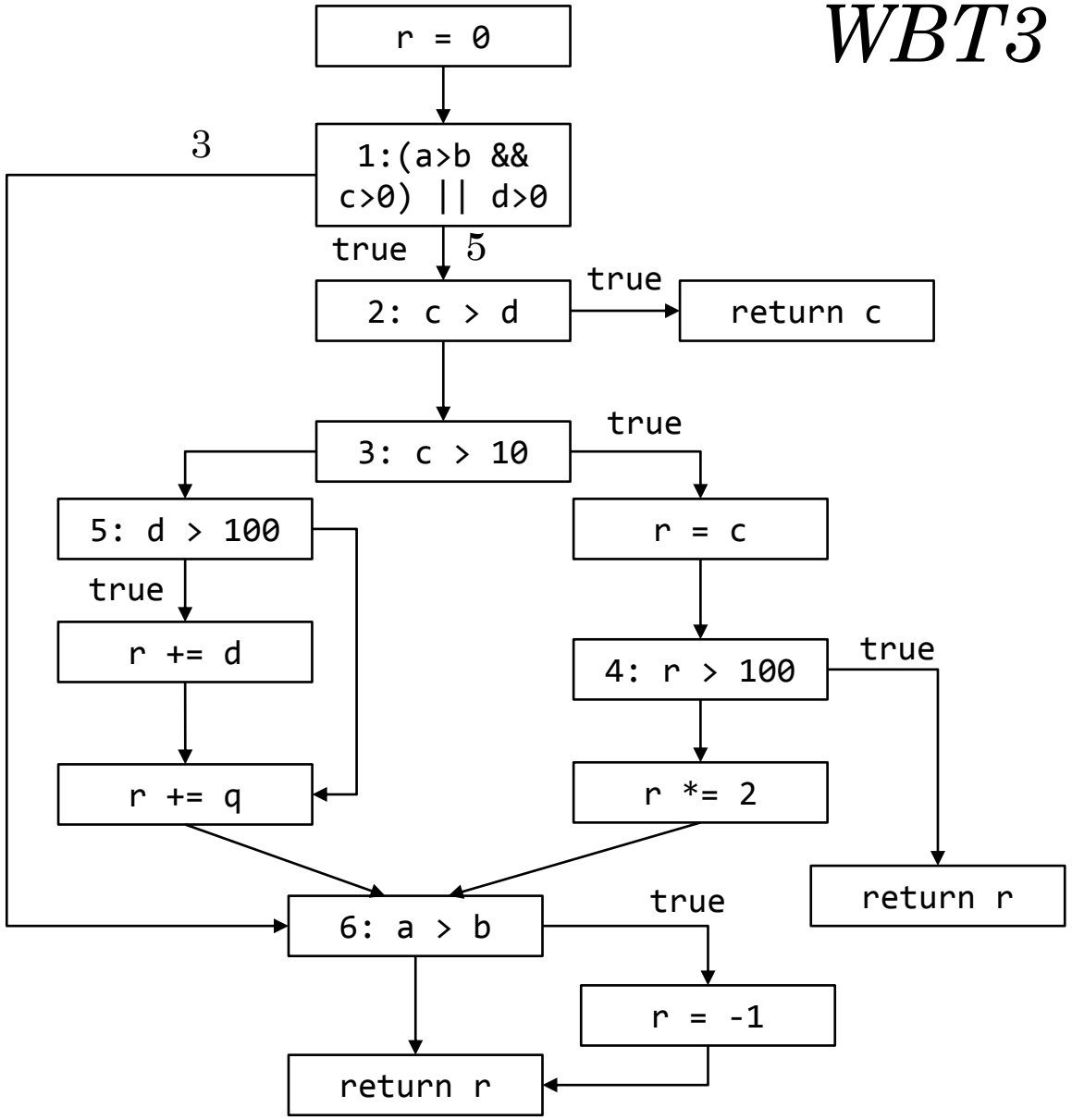
Es. 3

```
static int wbt3 (int a, int b,
int c, int d) {
int r = 0;
if ((a>b && c>0) || d>0) {          1
 if (c > d) return c;              2
 if (c > 10) {                     3
   r = c;
   if (r > 100) return r;          4
   r *= 2;
 } else {
   if (d > 100) r += d;            5
   r += q;
 }
}
if (a > b) r = -1;                 6
return r;
}
```

Per il metodo seguente, si definisca il control flow graph e si risponda alle domande. Si considerino scorrelate tutte le condizioni [1 p.to per le prime 4 domande, 2 per l'ultima; 0 p.ti se manca la spiegazione].

Si considerino scorrelate tutte le condizioni.

WBT3



N. min di test per la copertura dei criteri seguenti; si spieghi il valore.
Nodi 4 1T2T;1T2F3T(4T,4F6T); 1T2F3F5T6F
Link (edge) 6 mancano 2 sequenze contenenti 1F e 5F
Percorsi 1+3+4+2=10 1T2T; 1T2F3T(4T,4F(6)) 1T2F3F(5)(6) 1F(6)
Condizioni multiple 8 la condizione tripla ha 5 casi per l'uscita true e 3 per la false
N. min test per tutti i criteri: 11: l'uscita false della condizione multipla chiede 3 casi di test, ma per i percorsi ne bastano 2.

Domande

Es.1

Domanda	Vero	Falso
Il modello unificato del ciclo di vita del software include il modello incrementale e quello evolutivo ma non quello waterfall.		X
Una relazione recursiva in un modello informativo presuppone sempre che gli oggetti collegati svolgano ruoli reciprocamente diversi, ad esempio se x1 è predecessore di x2, x2 è successore di x1.		X
Due transizioni, t1 e t2, di una rete di Petri si dicono in serie se c'è un posto che ha t1 come unico input e t2 come unico output.		X
Un modello dataflow (DFD) è composto soltanto da attori esterni, da attività e da collegamenti che trasportano i dati da un attore ad un'attività, da un'attività ad un attore o da un'attività ad un'altra attività.		X

Es.2

Domanda	Vero	Falso
Nello sviluppo agile attenersi al piano prestabilito è considerato un principio assoluto.		X
La durata di un task in un diagramma PERT/CPM prescinde dall'impegno effettivo richiesto agli esecutori del task.	X	
Una rete di Petri in cui tutti gli elementi (posti e transizioni) hanno almeno 1 input e almeno 1 output è anche fortemente connessa.		X
Il metodo di Fagan è un metodo per la validazione del software.		X

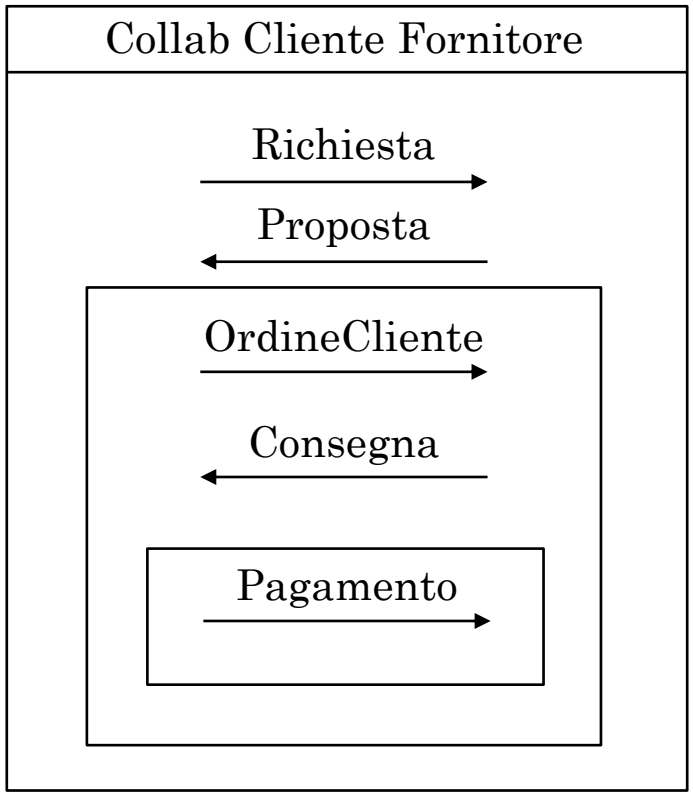
Metamodello

Metamodello

Si definisca il metamodello relativo ai modelli di collaborazione con le caratteristiche seguenti.

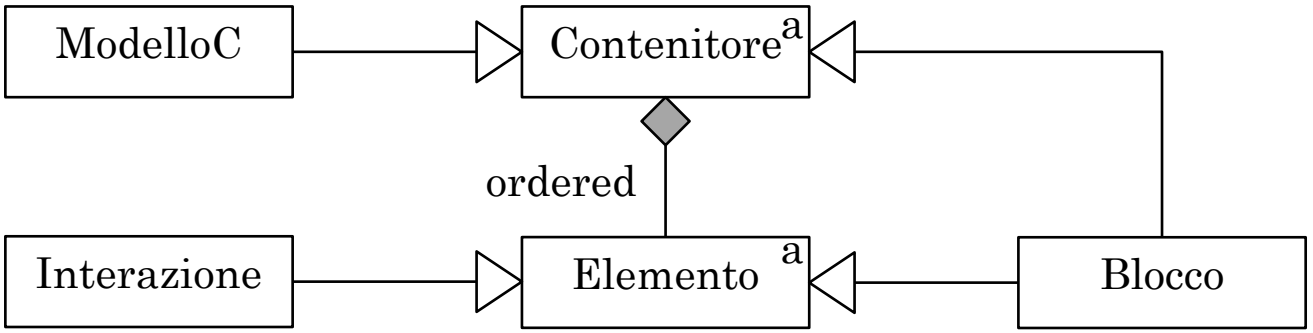
Un modello di collaborazione ha un nome (stringa) e contiene una sequenza di elementi che possono essere interazioni oppure blocchi. Un'interazione ha un'etichetta (stringa) e una direzione che può essere LR (left to right) o RL (right to left). Un blocco contiene una sequenza di elementi.

La figura presenta un esempio di modello di collaborazione.



Metamodello

Un contenitore contiene una sequenza di elementi che sono interazioni o blocchi. Un blocco e un modelloC sono contenitori.



Un elemento appartiene ad un blocco oppure ad un modelloC. Quindi un blocco è contenuto in un altro blocco o in un modelloC (che non è contenuto in null'altro).

Elemento e Contenitore sono abstract.
Attributi
ModelloC: String nome.
Interazione: String etichetta; direzione (LR, RL).