



Estensioni del modello rendezvous

SCD

Anno accademico 2009/10
Sistemi Concorrenti e Distribuiti

Tullio Vardanega, tullio.vardanega@math.unipd.it

Corso di Laurea Magistrale in Informatica, Università di Padova 1/22



Estensioni del modello rendezvous

Limiti del modello base

- ❑ Il server può disporsi ad accettare messaggi da un solo canale alla volta
 - Il processo cliente può emettere un solo messaggio di richiesta per volta
- ❑ Una volta in attesa sul canale il server attende indefinitamente fino all'eventuale arrivo di un messaggio
 - Una volta emesso un messaggio di richiesta, il cliente resta indefinitamente sospeso in attesa della sua accettazione e del completamento delle azioni corrispondenti

Corso di Laurea Magistrale in Informatica, Università di Padova 2/22



Estensioni del modello rendezvous

Requisiti di estensione – 1

- ❑ Lato server (più critico)
 1. Poter attendere su più di un canale alla volta
 2. Limitare l'attesa a un tempo limite (*time-out*)
 3. Poter abbandonare immediatamente l'attesa su un canale che non abbia messaggi in coda
 4. Poter terminare quando nessun cliente fosse più in grado di emettere richieste
 - Il comportamento più naturale per un vero server

Corso di Laurea Magistrale in Informatica, Università di Padova 3/22



Estensioni del modello rendezvous

Requisiti di estensione – 2

- ❑ Lato cliente (meno critico)
 - Non è strettamente necessario poter emettere più richieste simultaneamente da parte di uno stesso cliente
 - Un processo cliente funzionalmente coeso ha una logica interna sequenziale!
 - 1. È desiderabile fissare un limite al tempo di attesa dell'accettazione di una richiesta
 - 2. È utile poter abbandonare l'attesa di accettazione qualora questa non fosse immediatamente disponibile

Corso di Laurea Magistrale in Informatica, Università di Padova 4/22



Estensioni del modello rendezvous

Estensioni di lato server – 1

- ❑ S-1. Attesa su più punti d'accesso
 - Il server può fornire più servizi, ciascuno dei quali viene fornito attraverso messaggi scambiati su uno specifico canale tipato (*entry*)

```
task Server is
  entry S1 (...);
  entry S2 (...);
end Server;
```

```
task body Server is
  ...
  begin
  loop
  select
  accept S1(...) do ... end S1;
  or
  accept S2(...) do ... end S2;
  end select;
  end loop;
end Server;
```

Corso di Laurea Magistrale in Informatica, Università di Padova 5/22



Estensioni del modello rendezvous

Estensioni di lato server – 2

- ❑ S-1. (continua)
 - Se nessuna richiesta fosse disponibile su alcun canale al momento della valutazione il server si pone in attesa
 - La valutazione avviene sempre simultaneamente su tutti i canali considerati
 - Viene così pienamente soddisfatto il requisito 1
 - Qualora punti d'accesso diversi avessero richieste in attesa, la scelta di uno tra essi è non deterministica
 - Concettualmente
 - Politica base FIFO per l'attesa di più richieste presso lo stesso canale
 - Politiche alternative (p.es. per urgenza) possono essere considerate

Corso di Laurea Magistrale in Informatica, Università di Padova 6/22

Estensioni del modello *rendezvous*

Estensioni di lato server – 3

□ S-1. (continua)

- Conviene aderire al modello di Dijkstra
- Prevedere "guardie" sui canali che esprimano condizioni logiche sull'opportunità funzionale di accettare particolari richieste

```

select
  Guard_1 => accept ...;
or
  Guard_2 => accept ...;
or
  ...
or
  Guard_N => accept ...;
end select;
                    
```

La guardia è una espressione Booleana, di tipo "when <condizione>", il cui verificarsi abilita la considerazione del canale

Le guardie entro un comando **select** sono valutate simultaneamente, una sola volta all'inizio del comando

Corso di Laurea Magistrale in Informatica, Università di Padova

7/22

Estensioni del modello *rendezvous*

Estensioni di lato server – 4

□ S-2.3. Limitare temporalmente l'attesa

- S-2. Fissare un tempo limite non nullo entro il quale il server è disposto ad attendere l'arrivo di richieste su uno dei canali considerati
 - Potendo esprimere il tempo di attesa come relativo o assoluto a seconda della necessità
- S-3. Abbandonare immediatamente l'attesa in assenza di richieste all'istante di valutazione
 - Equivalente a esprimere un tempo limite di attesa nullo

Corso di Laurea Magistrale in Informatica, Università di Padova

8/22

Estensioni del modello *rendezvous*

Estensioni di lato server – 5

□ S-2.

- Un limite temporale di attesa non nullo consente al server di adempiere al suo compito base
 - Senza però diventare incapace di fare altro a causa di attese infinite
- La perdurante assenza di richieste sui suoi canali può suggerire al server che i clienti si trovino in una condizione di errore
 - Ciò impedisce all'eventuale anomalia di propagarsi al server

Corso di Laurea Magistrale in Informatica, Università di Padova

9/22

Estensioni del modello *rendezvous*

Esempio – 1

```

with Ada.Real_Time; use Ada.Real_Time;
task Sensor_Monitor is
  entry New_Period ( Period : Time_Span );
end Sensor_Monitor;
...
task body Sensor_Monitor is
  Current_Period : Time_Span := Milliseconds(10_000);
  Next_Cycle : Time := <time base> + Current_Period;
begin
  loop
    -- read sensor value and post it where required
    select
      accept New_Period (Period : Time_Span) do
        Current_Period := Period;
      end New_Period;
    delay until Next_Cycle;
  or
    delay until Next_Cycle;
    Next_Cycle := Next_Cycle + Current_Period;
  end select;
  end loop;
end Sensor_Monitor;
                    
```

Comportamento di base periodico, con lettura di sensore ogni 10 secondi

Capace di modificare dinamicamente l'ampiezza del periodo in caso di richiesta del cliente

L'effetto del cambiamento è ottenuto dalla presenza del blocco di comandi in ①

Corso di Laurea Magistrale in Informatica, Università di Padova

10/22

Estensioni del modello *rendezvous*

Esempio – 1: l'effetto

Sensor_Monitor.New_Period(...):

□ L'arrivo di una richiesta "New_Period" crea un nuovo riferimento T0' per i successivi periodi

- Interrompendo la periodicità precedente
- La soluzione data in ① comporta discontinuità temporale

Corso di Laurea Magistrale in Informatica, Università di Padova

11/22

Estensioni del modello *rendezvous*

Esempio – 2

```

task type Watchdog (Minimum_Distance : Duration) is
  entry All_is_Well;
end Watchdog;

task body Watchdog is
begin
  loop
    select
      accept All_is_Well;
      ... -- client is alive and well
    or
      delay Minimum_Distance;
      ... -- client may have failed, raise alarm
    end select;
  end loop;
end Watchdog;
                    
```

Il modello di Dijkstra applica anche all'alternativa di attesa temporale, che può pertanto ammettere una guardia

Corso di Laurea Magistrale in Informatica, Università di Padova

12/22

Estensioni del modello *rendezvous*

Estensioni di lato servernte – 6

□ S-3. Attesa nulla

- Il servernte può voler prestare attenzione solo a quei canali che abbiano già richieste in attesa al momento del controllo altrimenti effettuare azioni alternative
 - Questa modalità rende possibile l'attesa attiva, che resta però indesiderabile!

```
select
accept A;
or
accept B;
else
C;
end select;
```

Forma esplicita
(preferibile)

L'effetto desiderato
può essere ottenuto
in 2 modi alternativi

Forma implicita
per T=0.0

```
select
accept A;
or
accept B;
or
delay T;
end select;
```

Corso di Laurea Magistrale in Informatica, Università di Padova 13/22

Estensioni del modello *rendezvous*

Estensioni di lato servernte – 7

□ S-4. Terminazione in mancanza di clienti

- L'asimmetria del modello cliente-servernte può far sì che il servernte sopravviva al completamento dei suoi clienti
 - In questo caso è desiderabile che anche il servernte possa terminare
- La terminazione del servernte può essere trattata direttamente a programma
 - Vedi la soluzione alla terminazione nella versione bis del crivello di Eratostene (ove però non agiscono servernti "puri" ma degeneri)
- Trattandosi però di un requisito generale del modello esteso di *rendezvous* è desiderabile disporre di una soluzione generale
 - Basta consentire al servernte di aggiungere un'alternativa `terminate` a quelle da considerare in parallelo nel comando `select`

Corso di Laurea Magistrale in Informatica, Università di Padova 14/22

Estensioni del modello *rendezvous*

Estensioni di lato servernte – 8

□ S-4. (continua)

- Un servernte sospeso su comando `select` con alternativa `terminate` aperta viene considerato completo allorché
 - Il *master* da cui esso dipende ha completato la propria esecuzione
 - Ogni altro processo dipendente da quello stesso *master* è
 1. Già terminato, oppure
 2. A sua volta sospeso su un comando `select` con alternativa `terminate` aperta
- La condizione 1 assicura che non vi possano essere nuove richieste di servizio in arrivo
- La condizione 2 applica la regola transitivamente

Corso di Laurea Magistrale in Informatica, Università di Padova 15/22

Estensioni del modello *rendezvous*

Ultime volontà ☺

□ La semantica di terminazione S4 va arricchita con un meccanismo che consenta al processo terminante di effettuare azioni esplicite di finalizzazione sul suo *scope*

- Le ultime volontà

□ Alcuni tipi esportano un metodo di finalizzazione che viene invocato dal "supporto a tempo di esecuzione" quando un oggetto di quel tipo esce di *scope*

□ La terminazione di un processo la cui regione dichiarativa contenga istanze di tali tipi comporta l'invocazione automatica dei corrispondenti metodi di finalizzazione



Corso di Laurea Magistrale in Informatica, Università di Padova 16/22

Estensioni del modello *rendezvous*

Esempio – 3

□ Il crivello di Eratostene (versione Ter)

- Vogliamo aggiungere controllo di terminazione alle istanze dei processi di tipo Sieve_T
- In caso di terminazione, vogliamo anche che il processo terminante ce ne fornisca notifica

Corso di Laurea Magistrale in Informatica, Università di Padova 17/22

Estensioni del modello *rendezvous*

Estensioni di lato cliente

□ Per il lato cliente avevamo solo 2 esigenze

- C-1. Porre un limite temporale non nullo all'attesa di servizio
 - Equivalente al requisito S.2 di lato servernte e soddisfatto nello stesso modo
 - Il limite riguarda solo la durata massima dell'attesa fino all'inizio della sincronizzazione
 - Nessuna relazione con la durata effettiva della sincronizzazione!
- C-2. Lasciare l'attesa qualora il servernte non fosse immediatamente disponibile
 - Equivalente al requisito S.3 del lato servernte e soddisfatto nello stesso modo
 - Sta alla realizzazione del modello gestire il caso in cui più clienti desiderino simultaneamente conoscere la disponibilità istantanea di uno stesso servernte

Corso di Laurea Magistrale in Informatica, Università di Padova 18/22

Estensioni del modello *rendezvous*

Usi del modello cliente-servente

- Un server è un'entità **reattiva** capace di garantire **mutua esclusione**
 - Eseguendo una sola alternativa `accept` alla volta
- L'esecuzione della **sincronizzazione** rappresenta la **sezione critica**
- La risorsa condivisa deve però essere visibile **soltanto** al processo servente

```

task body Buffer (..) is
  .. the shared resource
begin
  ..
  loop
  select
  when ..
  accept Put (..) do .. end Put;
  .. -- local housekeeping
  or
  when ..
  accept Get (..) do .. end Get;
  .. -- local housekeeping
  or
  terminate;
  end select;
  end loop;
end Buffer;
    
```

task type Buffer (..) is
entry Put (..);
entry Get (..);
end Buffer;

Corso di Laurea Magistrale in Informatica, Università di Padova 19/22

Estensioni del modello *rendezvous*

Abusi del modello cliente-servente

- Programmazione incauta può causare situazioni di stallo
 - Il modello *rendezvous*, anche nella sua forma estesa, **non** è capace di impedirne strutturalmente il rischio

```

task T1 is
  entry A;
end T1;

task body T1 is
begin
  T2.B;
  accept A;
end T1;

task T2 is
  entry B;
end T2;

task body T2 is
begin
  T1.A;
  accept B;
end T1;
    
```

Corso di Laurea Magistrale in Informatica, Università di Padova 20/22

Estensioni del modello *rendezvous*

Una buona prassi

- I processi dovrebbero essere usati soltanto per realizzare entità attive **oppure server**
- Le entità attive non dovrebbero possedere canali ma solo inviarvi messaggi
- I *server* "puri" accettano richieste di accesso ma non ne fanno alcuna

Corso di Laurea Magistrale in Informatica, Università di Padova 21/22

Estensioni del modello *rendezvous*

Stati d'esecuzione di processo

Corso di Laurea Magistrale in Informatica, Università di Padova 22/22