

Proprietà Non Funzionali nella composizione di Web Services

Aspetti semantici e realizzativi

Yuri Pirola

16 lug 2007

Outline

- 1 Formulazione del problema
 - Introduzione
 - Esempio
 - Composizione di WS
- 2 Rappresentazione della qualità di un servizio
 - Qualità di un servizio
 - Modello ontologico di PNF
- 3 QoS nella composizione di WS
 - Outline della soluzione
 - Modello di Zeng et al., 2004
 - Modello di Ardagna e Pernici, 2007
- 4 Riepilogo

Outline

- 1 Formulazione del problema
 - Introduzione
 - Esempio
 - Composizione di WS
- 2 Rappresentazione della qualità di un servizio
 - Qualità di un servizio
 - Modello ontologico di PNF
- 3 QoS nella composizione di WS
 - Outline della soluzione
 - Modello di Zeng et al., 2004
 - Modello di Ardagna e Pernici, 2007
- 4 Riepilogo

Composizione di WS

- **Web Services**

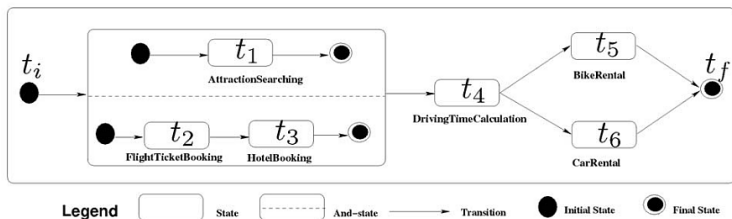
‘Espongono’ servizi, accessibili *programmaticamente*, via protocolli di comunicazione *standard*.

- **Web Services Composition**

Interconnessione di servizi web preesistenti al fine di creare servizi complessi.

Esempio

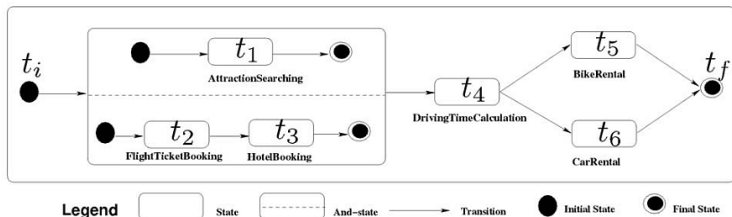
Prenotazione viaggi (Zeng et al., 2004 [1])



- Specifica a 2 livelli:
 - Astratto
 - Concreto

Realtà dei Web Services

Prenotazione viaggi



- Diversi provider forniscono servizi funzionalmente equivalenti.
 - Differiscono per la qualità del servizio offerto (QoS).
 - Esempio: costo, reputazione, tempo di esecuzione...
- Utenti diversi hanno esigenze diverse (ad es., massimo risparmio, massimo comfort, ...)

Composizione automatica di WS

- **Concretizzazione:** determinare la corrispondenza fra attività astratta (ad es. `PrenotazioneBigliettoAereo`) e provider (ad es. `Alitalia`).
- **Concretizzazione dinamica** del processo astratto:
 - richiede specifiche semantiche del funzionamento del servizio
 - discriminare fra le alternative in base alle specifiche non funzionali

Problemi da affrontare

- Problemi da affrontare:
 - rappresentare la qualità di un servizio
 - determinare la concretizzazione che **massimizza la qualità** percepita dall'utente soddisfacendo determinati **vincoli**

Outline

- 1 Formulazione del problema
 - Introduzione
 - Esempio
 - Composizione di WS
- 2 Rappresentazione della qualità di un servizio
 - Qualità di un servizio
 - Modello ontologico di PNF
- 3 QoS nella composizione di WS
 - Outline della soluzione
 - Modello di Zeng et al., 2004
 - Modello di Ardagna e Pernici, 2007
- 4 Riepilogo

Qualità di un servizio

- La qualità di un servizio è descritta da un insieme di proprietà non funzionali

$$QoS \subseteq NFP$$

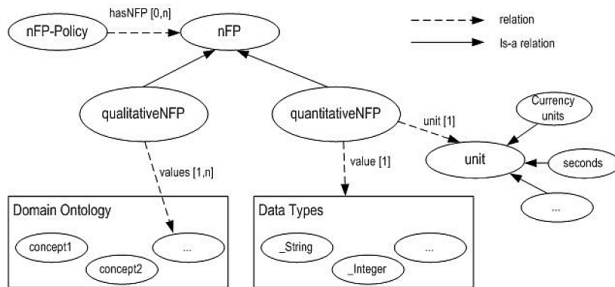
- Generiche:
 - Costo
 - Reputazione
 - Disponibilità
 - Tempo di esecuzione
 - ...
- Dipendenti dal dominio:
 - Risoluzione dell'immagine
 - Tipo di imballaggio
 - Categoria di un albergo
 - ...

Policy

- Un servizio, più livelli qualitativi:
 - Consegna espresso/tradizionale
 - Stampe ad alta/bassa risoluzione
- WS-Policy [2]:
 - stabilisce un *contratto* fra fornitore del servizio e utilizzatore (Service Level Agreement, SLA)
 - work in progress

QoS e ontologie

- Qualità del servizio:
 - concetto complesso, variegato e dipendente dal dominio
 - va rappresentato in modo processabile da un elaboratore
- Modello ontologico delle proprietà non funzionali [3].



Outline

- 1 Formulazione del problema
 - Introduzione
 - Esempio
 - Composizione di WS
- 2 Rappresentazione della qualità di un servizio
 - Qualità di un servizio
 - Modello ontologico di PNF
- 3 QoS nella composizione di WS
 - Outline della soluzione
 - Modello di Zeng et al., 2004
 - Modello di Ardagna e Pernici, 2007
- 4 Riepilogo

Concretizzazione automatica di un WS

Concretizzazione automatica di un WS basata su PNF

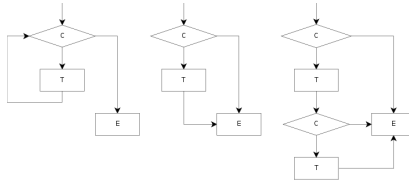
- **Vantaggio per il progettista:**
Web Services rappresentano uno scenario variabile, si adatta alla situazione.
- **Vantaggio per l'utente:**
Utilizzo di PNF aumenta la soddisfazione dell'utente (la qualità percepita si avvicina alla qualità attesa).

Concretizzazione automatica di un WS

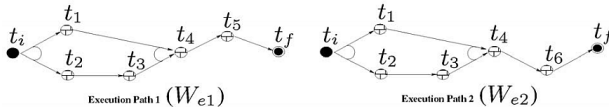
- **Input:**
Descrizione di un processo astratto
Specifiche dei servizi disponibili
- **Output:**
Concretizzazione del processo (=piano di esecuzione)
- **Outline della soluzione** (Zeng et al., 2004 [1]):
 - 1 Preprocessing dell'input
 - 2 Scelta della concretizzazione ottima ed Esecuzione del piano
 - 3 Ripianificazione in caso di errori e/o cambiamenti dell'ambiente

Preprocessing dell'input

- **Unfolding dei cicli:** 'svolgere' i cicli fino a un numero max di iterazioni



- Individuare i possibili *execution paths*



'Misurare' la qualità

- Restrizione a un insieme $Q = \{q_1, \dots, q_J\}$ di proprietà *quantitative*.
- Qualità di un servizio s : $q(s) = (q_1(s), q_2(s), \dots, q_J(s))$
- Qualità di un servizio composto concreto:

Criteria	Aggregation function
Price	$q_{pr}(p) = \sum_{i=1}^N q_{pr}(s_i, op(t_i))$
Duration	$q_{du}(p) = CPA(p, q_{du})$
Reputation	$q_{rep}(p) = \frac{1}{N} \sum_{i=1}^N q_{rep}(s_i)$
Success rate	$q_{rat}(p) = \prod_{i=1}^N (q_{rat}(s_i)^{z_i})$
Availability	$q_{av}(p) = \prod_{i=1}^N (q_{av}(s_i)^{z_i})$

Scelta della concretizzazione ottima

- Determinare il piano di esecuzione di un percorso di esecuzione:
 - **Approccio locale:** selezionare il miglior servizio per ciascun task.
Rapido, no vincoli sulla qualità complessiva.
 - **Approccio globale:** selezionare l'insieme di servizi che realizza al meglio la composizione.
Lento, soluzioni possibilmente migliori, vincoli globali.

Ottimizzazione globale

- È un problema di ottimizzazione multiobiettivo vincolata!
- Funzione obiettivo: combinazione lineare delle dimensioni qualitative normalizzate

$$score(s_i) = \sum_{j=1}^J w_j \cdot \hat{q}_j(s_i)$$

- Può essere formulato come problema di Programmazione Lineare Intera

Formulazione matematica

Variabili: $\{y_{i,l}, x_l\}$

- $y_{i,l} = 1$ se $s_{i,l}$ realizza t_l
- x_l tempo di inizio del task t_l

Funzione Obiettivo: $\max(\text{score}(ep_l))$

Vincoli:

- di Allocazione
- Lineari (costi, durata, reputazione, ...)
- Non Lineari ma riconducibili a lineari (disponibilità, tasso di successo, ...)

Ripianificazione del processo

- Ripianificazione del processo:
 - in caso di errori/variazioni al valore di QoS di $s_{i,l}$
 - necessaria solo per approccio globale
 - computazionalmente costosa (potrebbe essere ripetuta molte volte!)
 - ⇒ eseguita se il guadagno/perdita attese superano una certa soglia

Caratteristiche del metodo

- Flessibile rispetto alle dimensioni qualitative scelte.
- Permette pianificazioni locali o globali:
 - specifiche di vincoli globali
- Individua soluzioni globalmente ottime:
 - gestione 'grossolana' dei cicli
 - concetto di *critical* e *hot path*
- Ripianificazione potenzialmente onerosa.

Altre soluzioni

Proposte altre soluzioni (Ardagna and Pernici, 2007 [4]):

- Formulato come problema di PLI:
 - più raffinata della precedente
 - funzione obiettivo: media pesata degli *score* di tutti gli *ep* in base alla probabilità di verificarsi
 - vincoli di dipendenza tra WS
- Peeling dei cicli
- **Negoziazione** dei vincoli:
 - Rilassamento di alcuni vincoli se impossibili da soddisfare
 - Concretizzare processi altrimenti impossibili
 - maggiore complessità computazionale (in teoria NP)

Outline

- 1 Formulazione del problema
 - Introduzione
 - Esempio
 - Composizione di WS
- 2 Rappresentazione della qualità di un servizio
 - Qualità di un servizio
 - Modello ontologico di PNF
- 3 QoS nella composizione di WS
 - Outline della soluzione
 - Modello di Zeng et al., 2004
 - Modello di Ardagna e Pernici, 2007
- 4 Riepilogo

Conclusioni





Concretizzazione dinamica

- Distingue descrizione astratta (progettista) da grounding (applicazione).
- Richiede la descrizione semantica delle funzionalità
- Primo passo verso la composizione automatica

Utilizzo di PNF nella composizione dei servizi web

- Discrimina alternative funzionalmente equivalenti
- Adatta il servizio alle esigenze dell'utente (espresse come requisiti NF)
- Richiede la descrizione semantica delle Proprietà Non Funzionali

Riferimenti essenziali

-  [1] Zeng, Benatallah, et al.
QoS-Aware Middleware for Web Services Composition,
IEEE Trans. on Software Eng., 2004.
-  [2] WS-Policy.
Web Service Policy 1.2 - Framework, W3C, 2006.
-  [3] Comerio, De Paoli, Maurino, and Palmonari.
NFP-aware Semantic Web Services Selection, to appear.
-  [4] Ardagna, Pernici.
Adaptive Service Composition in Flexible Processes, IEEE
Trans. on Software Eng., 2007.