Consiglio Nazionale delle Ricerche
Istituto di Calcolo e Reti ad Alte Prestazioni

Manuale Sunflower

Un sistema autonomico per la gestione di workflow basati sul modello a coreografia per Grid e Cloud

Sabrina Celia, Danilo Cistaro,
Giuseppe Papuzzo

RT-ICAR-CS-10-11 **Dicembre 2010**

Consiglio Nazionale delle Ricerche, Istituto di Calcolo e Reti ad Alte Prestazioni (ICAR)
– Sede di Cosenza, Via P. Bucci 41C, 87036 Rende, Italy, URL: *www.icar.cnr.it*
– Sezione di Napoli, Via P. Castellino 111, 80131 Napoli, URL: *www.na.icar.cnr.it*
– Sezione di Palermo, Viale delle Scienze, 90128 Palermo, URL: *www.pa.icar.cnr.it*

LABORATORIO PUBBLICO PRIVATO OPENKNOWTECH

Progetto di Ricerca**: DM21301**

OpenKnowTech "*Laboratorio di Tecnologie per la Integrazione, Gestione e Distribuzione di Dati, Processi e Conoscenze*"

Obiettivo Realizzativo OR2

Open Grid

Manuale Sunflower

Un sistema autonomico per la gestione di workflow basati sul modello a coreografia per Grid e Cloud

A cura di: *ICAR-CNR*

Data 31/12/2010

SOMMARIO

Manuale Sunflower

1 *Installazione della Sunflower Console e del Sunflower Distribute Engine*

La Sunflower Console (SC) consente di gestire l'intero ciclo di vita delle istanze di workflow, dalla fase di creazione fino alla visualizzazione del risultato. Mentre il Sunflower Distribute Engine (SDE) si occupa dei eseguire le istanze di workflow in coreografia.
Prima di installare la SC su un PC è necessario soddisfare i seguenti prerequisiti:

1. Installazione della Java JDK 6 nella cartella /var/lib/jdk1.6.X_XX e settaggio delle relative variabili d'ambiente JAVA_HOME e PATH .
   (http://www.oracle.com/technetwork/java/javase/downloads/index.html)
2. Installazione di Apache Tomcat 6.0 nella cartella /var/lib/apache-tomcat-6.0.XX e settaggio della relativa variabile d'ambiente CATALINA_HOME .
   (http://tomcat.apache.org/)
3. Installazione di MySQL da package scaricato dal sito MySQL o con il gestore dei pacchetti APT o YUM .
   (http://www.mysql.com/)

Per installare la SC:

4. Scaricare il file Sunflower_1.0.zip dalla sezione download e copiare il file "SunflowerConsole.war" in esso contenuto nella cartella "$CATALINA_HOME/webapps". Dopo pochi secondi inizierà il deploy automatico del file *.war, nel caso ciò non accada riavviare il server Apache Tomcat.

Terminata la fase di installazione della SC è possibile accedervi browser internet all'indirizzo : http://localhost:8080/SunflowerConsole/ . La figura 1 mostra la GUI della SC attraverso cui è possibile effettuare le successive fasi di installazione e configurazione del SDE.

Figura 1: GUI di configurazione di Sunflower.

Selezionando la scheda "Download SDE" dalla SC è possibile scaricare il software e la documentazione necessaria per installare il SDE. In questa scheda inizialmente troveremo abilitato per il download solo il link "Sunflower Distribute Engine For Sunflower Console" (SDE4SC), mentre il secondo link "SunflowerDistributeEngine4Services.zip" (SDE4S) risulterà disabilitato fino a quando non verrà installato e configurato SDE4SC sulla macchina dove è presente la Sunflower Console. Nella scheda sono indicati i passi di installazione per il SDE4SC:

- Passo 1:
  Scaricare il "SunflowerDistributeEngine4SunflowerConsole.zip" in una cartella temporanea.

- Passo 2:
  Estrarre il contenuto del "SunflowerDistributeEngine4SunflowerConsole.zip" in "/var/lib/SunflowerDistributedEngine" ed eseguire "startup.sh" in essa contenuta.

SDE4SC come prerequisito richiede Java JDK 6, ma essendo installato sulla stessa macchina dove è presente la SC questo risulta già soddisfatto. La prima volta che si avvia il SDE4SC, questo creerà automaticamente i file di configurazione necessari per attivare il download del SDE4S.

Il SDE4SC fungerà da bridge di comunicazione tra la SC e i SDE4S che verranno installati successivamente, per questa ragione è necessario caricare prima di attivare il download del SDE4S i dati relativi all'end-point del SDE4SC.

Ritornando nella scheda "Download SDE" sarà ora possibile scaricare il "Sunflower Distributed Engine 4 Service". Questo va installato sui PC dove sono presenti i Web/Grid Service che si vuole utilizzare nell'esecuzione dei workflow. Nella scheda sono indicati i passi di installazione per il SDE4S (prerequisito, installazione della Java JDK 6 come al passo 1 dei prerequisiti della SC):

- Passo 1:
  Scaricare o copiare il file "SunflowerDistributeEngine4Service.zip" in una cartella temporanea.

- Passo 2 : (per sistemi linux)
  Estrarre il contenuto del "SunflowerDistributeEngine4Services.zip" in "/var/lib/SunflowerDistributedEngine/" ed eseguire "startup.sh" in essa contenuta.

- Passo 2 : (per sistemi windows)
  Estrarre il contenuto del "SunflowerDistributeEngine4Services.zip" in "C:\SunflowerDistributeEngine\" ed eseguire "startup.bat" in essa contenuta.

Il SDE è basato sul piattaforma P2P JXTA, in caso di problemi di configurazione e firewall fare riferimento alla documentazione contenuta al seguente indirizzo https://jxta.dev.java.net .

Al termine dell'installazione, della SC e del SDE4SC, sarà possibile eseguire alcuni workflow di test presenti nella scheda "Execute Workflow" della SC. Questi workflow di prova non utilizzano alcun servizio ma servono solo a testare la corretta installazione del sistema. A questo punto per poter utilizzare workflow più complessi che richiedo l'impiego di servizi, è necessario come primo passo associare tali servizi al framework Sunflower. Per far ciò è necessario che i WSDL dei servizi, che useremo per la creazione dei workflow astratti con "Eclipse BPEL Designer", siano presenti nel WSDL Repository della SC. La prossima sezione mostrerà i passi da eseguire per effettuare la registrazione dei servizi reali come WSDL astratti nel WSDL Repository.

N.B.: Nel primo accesso alla scheda "Execute Workflow" verrà richiesto di inserire la password dell'utente "root" del database "MySQL", dopo averla inserita la SC configurerà automaticamente MySQL con tutti i database e tabelle di cui necessita.

2 *Deposito dei WSDL astratti*

L'idea di base del WSDL Repository è quelle di fornire gli strumenti necessari per gestire, in modo semplice e intuitivo, un UDDI locale di servizi per Sunflower. Attraverso la scheda "WSDL Repository" della SC riportata in figura 2, gli utenti possono riorganizzare in categorie e registrare i prototipi dei servizi che intendono usare nella creazione dei workflow BPEL.

Per aumentare il livello di astrazione rispetto ad un classico UDDI si è scelto di utilizzare il paradigma della *struttura a directory*, così da utilizzare la tipica struttura gerarchica ad albero per facilitare l'organizzare in categorie dei prototipi dei servizi. Attraverso questa visione, il nome di ogni directory esprime un concetto, che diventa sempre più specifico quanto più si avvicina ad una foglia. Mentre il nome di ogni file rappresenta un concetto finale, non scomponibile ulteriormente, concretizzato come prototipo di servizio alias WSDL astratto. Infine, nella nostra visione, il contenuto di un file foglia è il descrittore del servizio in *formato WSDL* secondo lo standard W3C (http://www.w3.org/TR/wsdl).
Nel WSDL Repository una volta prescelto un ramo/directory/concetto o una foglia/file/WSDL, su di esso si potranno eseguire le quattro operazioni fondamentali di creazione e modifica:

1. "Add Dir", aggiunge una directory/concetto nella posizione corrente.
2. "Add WSDL", aggiunge una foglia/WSDL astratto nella posizione corrente.
3. "Del Dir", cancella la directory corrente.
4. "Del WSDL", cancella il file corrente.

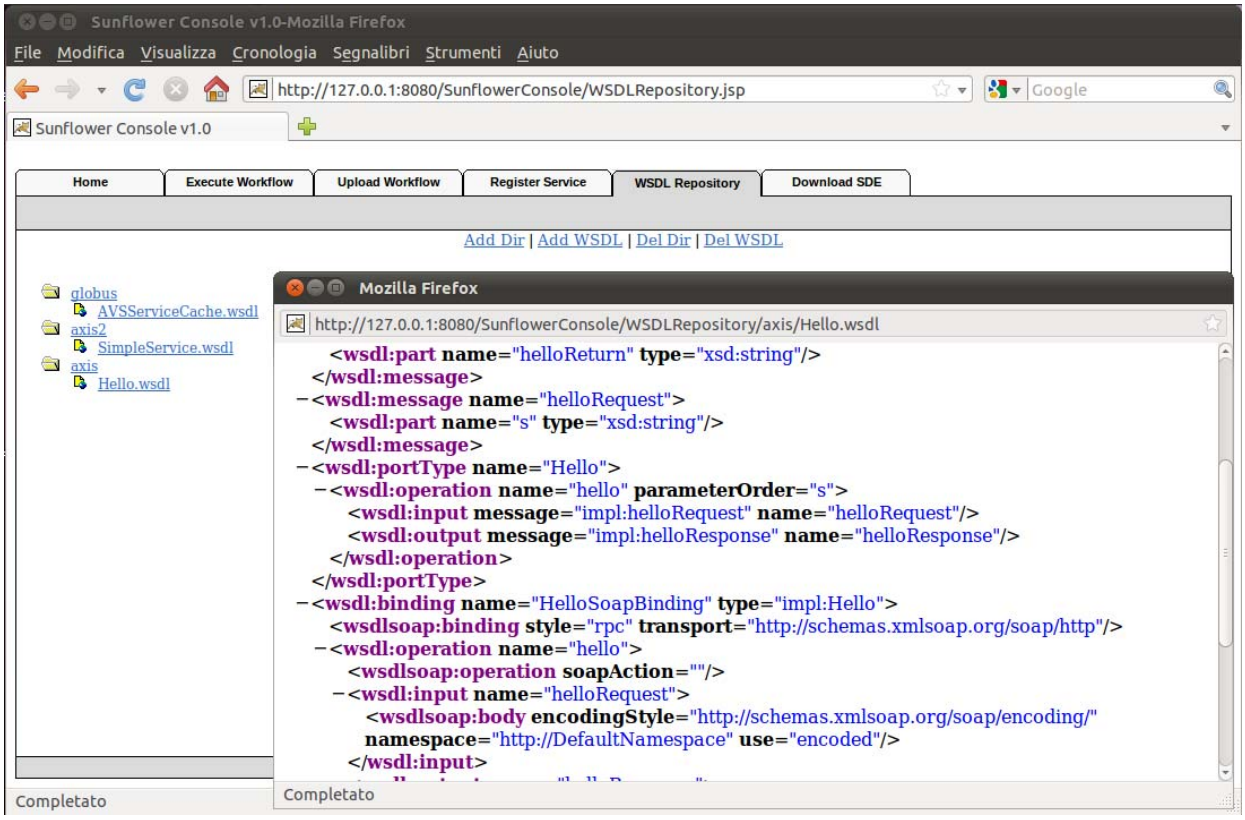

Figura 2: WSDL Repository.

La struttura dati così creata non fa riferimento a nessun servizio realmente implementato, ma crea la struttura dove andare a classificare e registrare i servizi creati. Un utente che vuole creare un servizio, deve prima di tutto selezionare una foglia da cui ricavare il WSDL, in secondo luogo implementare il servizio in base alle specifiche del WSDL, in fine registrare il servizio creato alla relativa classe di servizio. Per semplificare le operazioni di creazione del WSDL astratto il metodo *Add WSDL* importa automaticamente il *WSDL di un servizio reale* come *WSDL astratto*, e richiede

come parametri l'URL del WSDL servizio reale da importare e il nome da assegnare alla foglia nell'ontologia a directory. Questo passaggio consente di creare, partendo da un WSDL di servizio reale, un prototipo di WSDL di servizio astratto. Questo prototipo di servizio verrà successivamente usato per registrare il servizio reale da cui è stato generato e tutti servizi ad esso equivalenti al medesimo WSDL astratto. Inoltre, nella fase di creazione di un workflow BPEL attraverso Eclipse BPEL Designer, questi WSDL astratti verranno usati come sostituti dei servizi reali così da creare dei *workflow astratti* basati su servizi astratti.

3 *Registrazione di un servizio reale*

I WSDL astratti contenuti nel Repository consentono la creazione di workflow astratti, tuttavia tali informazioni non sono sufficienti per l'esecuzione del workflow. Nelle varie fasi del ciclo di vita di un workflow, infatti, sono necessarie informazioni riguardanti servizi reali e peer.
La registrazione di un servizio reale e del suo nodo SDE di competenza è completamente automatizzata, e richiede semplicemente la compilazione dei campi di una Form accessibile nella scheda "Register service" della SC (vedi figura 3).

La Form contiene tre campi che vanno a formare il contenuto di una tupla che lega assieme un servizio reale, un servizio astratto e un nodo SDE.

- URL del WSDL relativo al servizio reale da registrare
  (es.: http://150.145.63.130:8080/axis/DataServerCache.jws?wsdl).
- URL del WSDL astratto nel WSDL Repository, da cui è stato generato il sevizio reale
  (es.:
  http://127.0.0.1:8080/SunflowerConsole/WSDLRepository/Dati/DataServerCache.wsdl).
- Locazione del file "client.adv", generato al primo avvio del nodo SDE sul PC su cui risiede il servzio reale
  (es.: /var/lib/SunflowerDistributeEngine/client.adv per linux,
  C:\ SunflowerDistributeEngine \client.adv per windows).

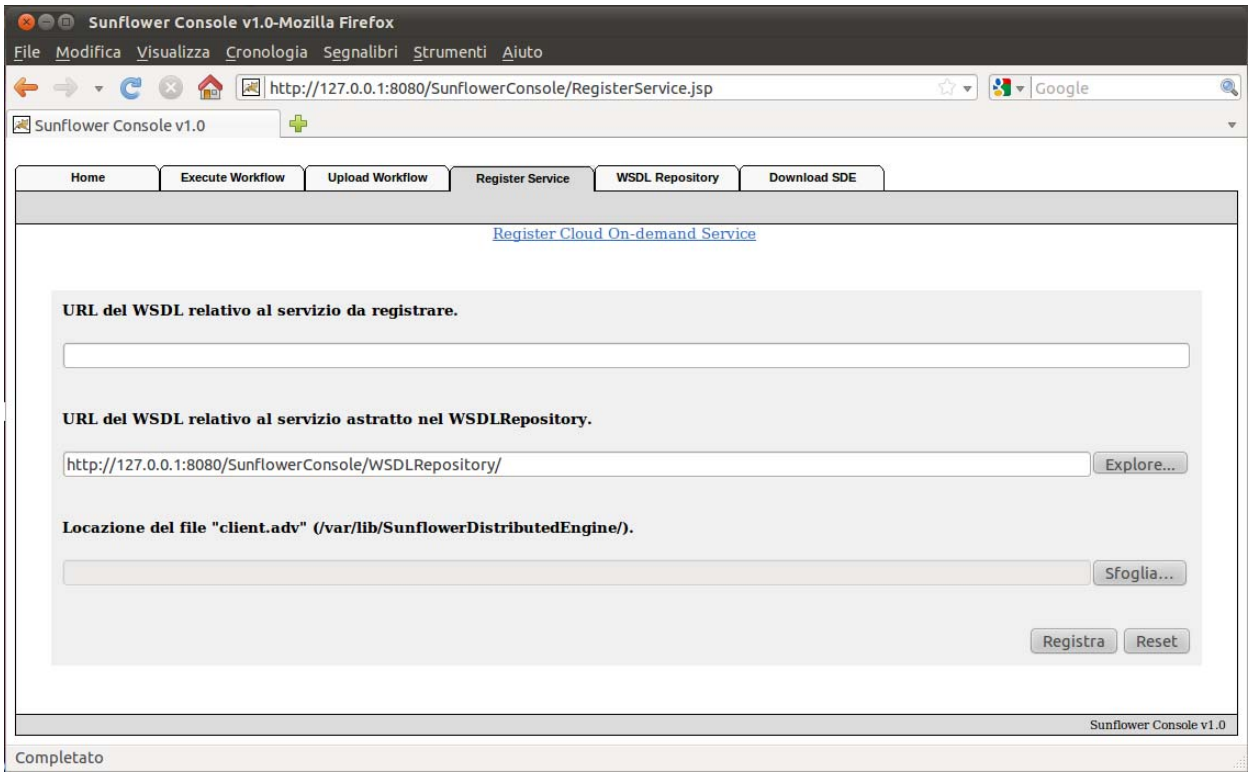

Figura 3: Registrazione dei servizi reali e associazione al relativo servizio astratto e Peer.

Naturalmente il nodo del SDE da associare al servizio reale è quello presente sul PC dove risiede il servizio, in modo da garantire al SDE il monitoraggio del sistema e il rispetto della QoS imposta dall'utente. Si possono verificare però casi particolari in cui il nodo SDE registrato non sia quello dove risiede il servizio, in questi casi il sistema non può garantire il monitoraggio e la QoS ma garantisce solo l'esecuzione. Per maggiori dettagli su questi casi particola far riferimento alla documentazione.
N.B.: Il contenuto del link "Register Cloud On-demamd Service" verrà discusso in seguito.

4 *Creazione e deploy di un workflow BPEL*

La creazione dei workflow BPEL avviene mediante l'impiego del plug-in "Eclipse BPEL Desiger", sviluppato per Eclipse nel progetto "BPEL Designer Project". Grazie a questo designer è possibile creare codice in base allo standard XML-BPEL senza dover toccare direttamente codice XML, componendolo attraverso operazioni di drag & drop e configurando le singole attività mediante wizzard.

Prima di procedere all' installazione dell'Eclipse BPEL Designer è necessario soddisfare i seguenti prerequisiti:

1. Installazione della Java JDK 6 nella cartella /var/lib/jdk1.6.X_XX e settaggio delle relative variabili d'ambiente JAVA_HOME e PATH .
   (http://www.oracle.com/technetwork/java/javase/downloads/index.html)
2. Installazione di Eclipse Classic 3.6.X nella cartella /home/$USER/eclipse .
   (http://www.eclipse.org/downloads/)

Per installare Eclipse BPEL Designer 0.5 :

3. Avviare Eclipse e selezionare "Help" → "Install New Software" , nella finestra di dialogo inserire nella casella "Work with" il seguente URL http://download.eclipse.org/technology/bpel/update-site , selezionare il plug-in BPEL per l'installazione e cliccare su "Next", per completare l'installazione seguire le istruzioni della finestra di dialogo.

Terminata la fase d'installazione, Eclipse con il plug-in BPEL Designer si presenterà con l'interfaccia grafica (perspective) riportata in figura 4.

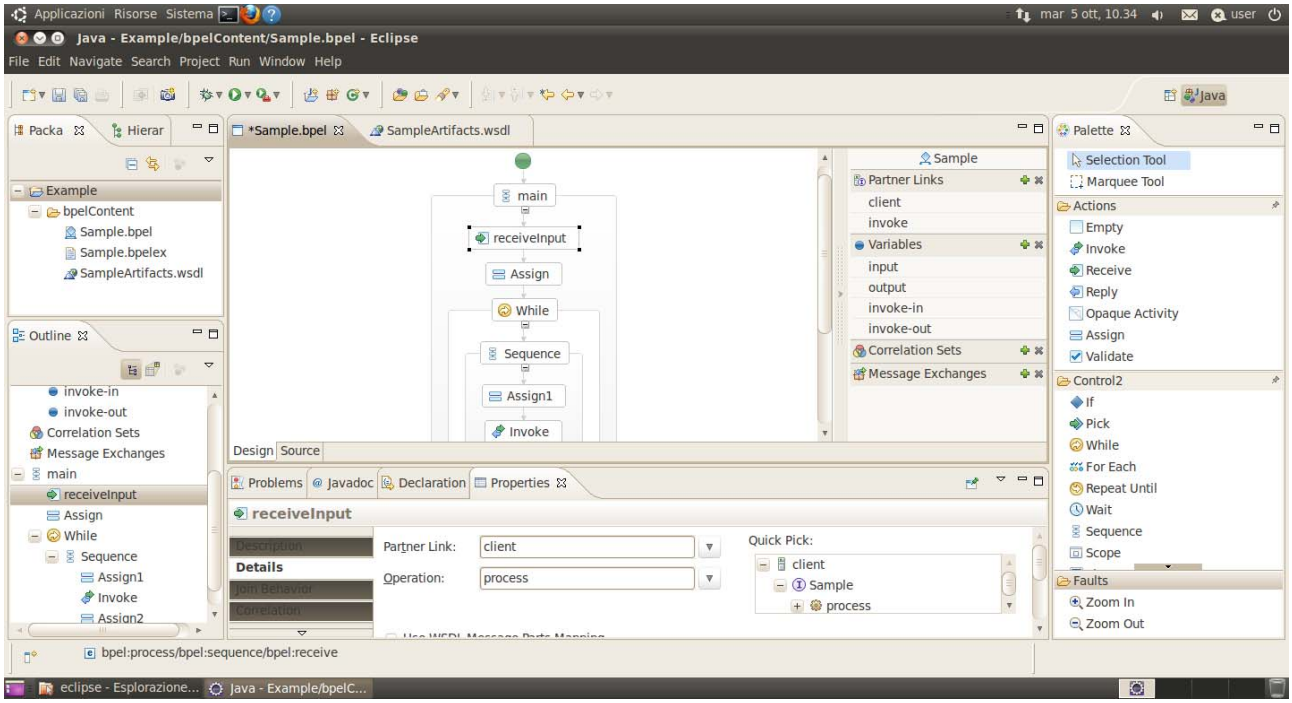

Figura 4: Eclipse BPEL Designer.

All'URL http://www.eclipse.org/bpel/ sono disponibili la documentazione, il tutorial e i video tutorial per la creazione di workflow BPEL. Per rendere il workflow creato compatibile con Sunflower, bisogna discostarsi leggermente da tale documentazione, più precisamente nel punto raggiungibile al seguente link http://www.eclipse.org/bpel/users/howto/wsdl.php o selezionando "Users" dal menu di sinistra e poi "Add a WSDL" dalla scheda dei contenuti. Nel punto evidenziato in figura 5, è possibile importare un qualsiasi WSDL di servizio per la creazione di "Partner Link".

Nel nostro caso i WSDL dei servizi da usare nei workflow BPEL devono essere selezionati tra quelli disponibili nel WSDL Repository che contenente i WSDL dei servizi astratti, accessibile all'indirizzo http://localhost:8080/SunflowerConsole/WSDLRepository.jsp .

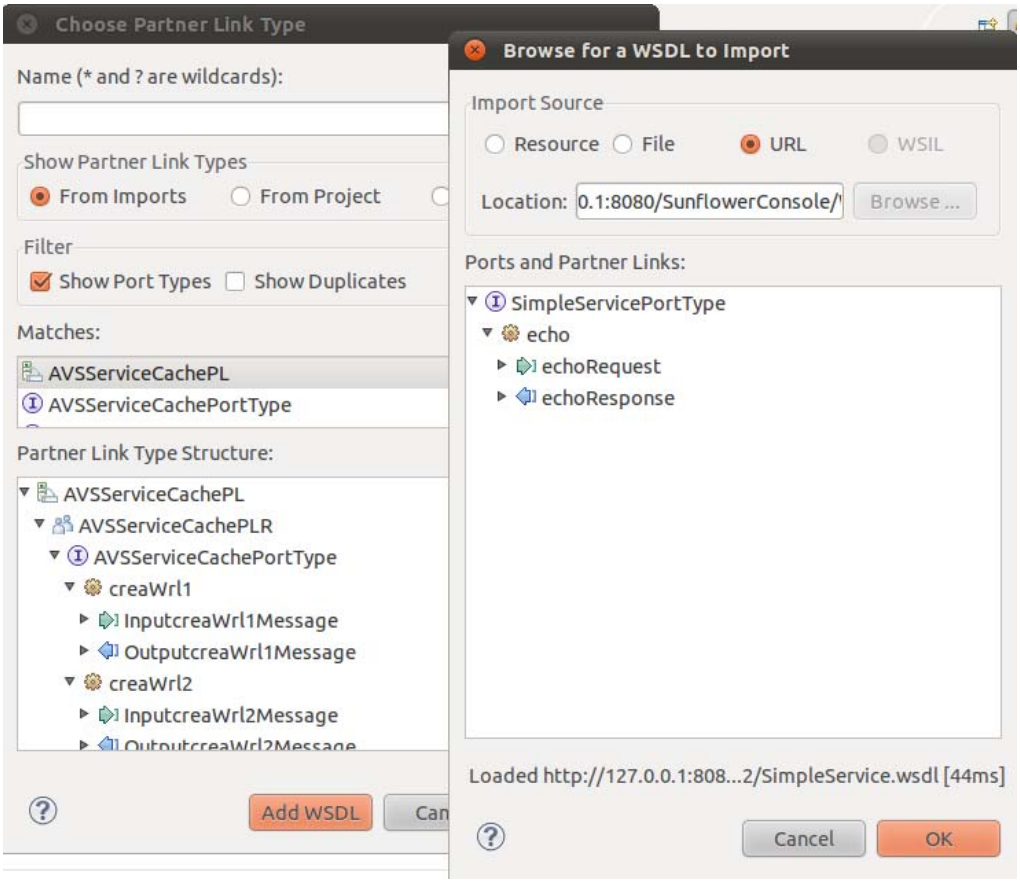

Figura 5: Creazione di un Partner Link che utilizza un WSDL astratto.

Per poter eseguire il deploy del workflow, creato con Eclipse BPEL Designer, questo va opportunamente inserito in un package per predisporlo alla fase di upload. Per far ciò in Eclipse si seleziona "File" → "Export" → "Archive file" → selezionare la cartella del progetto BPEL contenete il workflow creato → selezionare un nome per l'archivio da esportare → "Save in zip format".

Dopo aver creato il package contenete il workflow BPEL è necessario eseguire il deploy in modo renderlo disponibile per l'esecuzione nell'elenco di "Execute Workflow", ciò può essere fatto accedendo alla scheda "Deploy Workflow " e seguendo le istruzioni in essa contenute (figura 6).

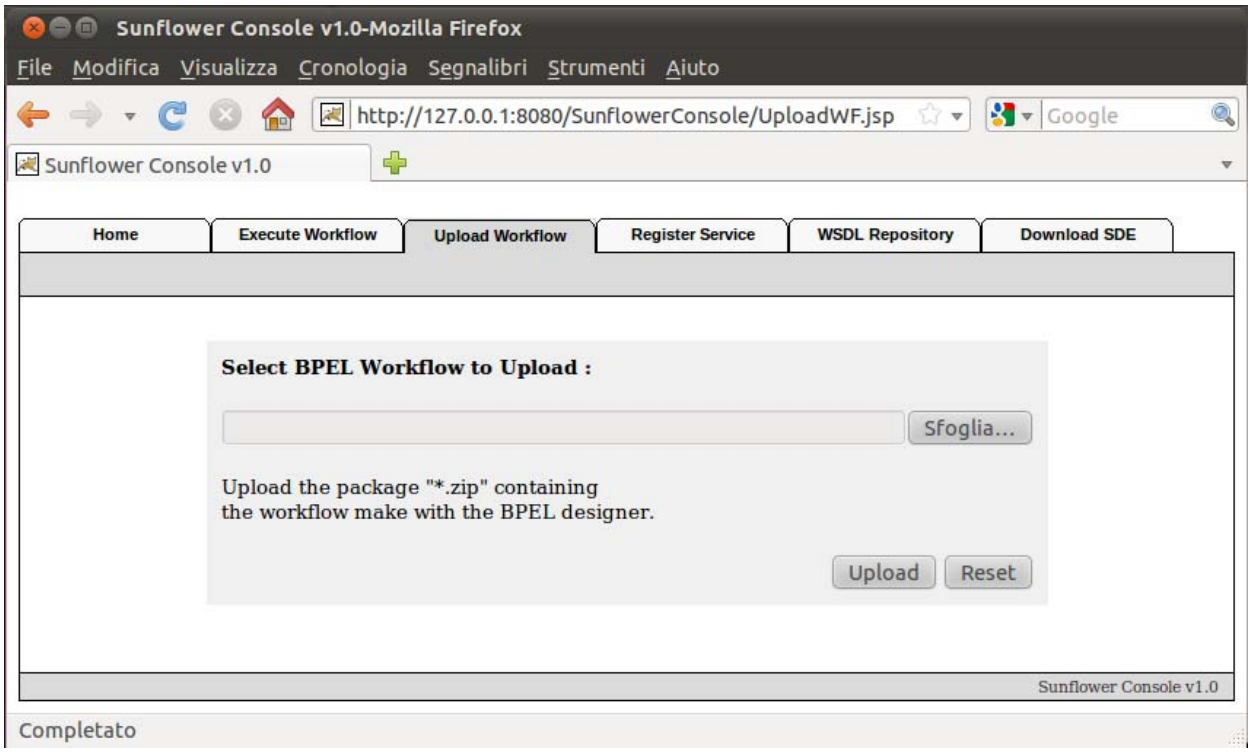

Figura 6: Upload e Deploy di un workflow BPEL.

Nella fase di upload il package contenete il workflow è inviato alla SC, mentre nella fase di deploy si effettua la verifica il contenuto del package. Se questo contiene un workflow BPEL, verrà inserito nell'elenco dei workflow disponibili, altrimenti sarà restituito l'errore.

5 *Inizializzazione ed esecuzione di un workflow BPEL*

Tutti i workflow BPEL per i quali la fase di Upload e Deploy è andata a buon fine, vengono aggiunti alla lista presente nella scheda "Execute Workflow"(figura 7). Qui è possibile selezionare il workflow da eseguire in base al nome assegnato al package esportato da Eclipse.

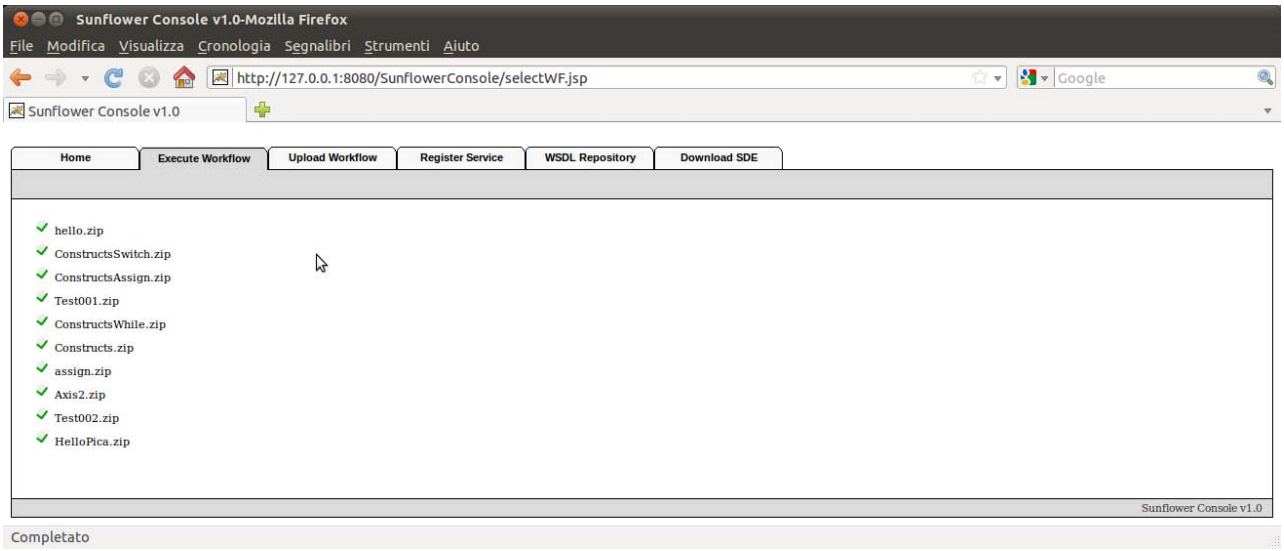

Figura 7: Elenco dei workflow BPEL disponibili per l'esecuzione.

Selezionato un workflow BPEL l'inizializzazione, dei parametri relativi alle variabili di input e agli attributi che descrivono la QoS, può essere eseguita attraverso la compilazione della form mostrata in figura 8. Nella parte superiore della form sono presenti le caselle relative alle variabili di ingresso del workflow che devono essere inizializzate con i valori che si vogliono assegnare alle variabili. In basso è presente una tabella che riporta per riga i servizi coinvolti nel workflow e per colonna i vincoli di QoS da rispettare. Gli attributi attualmente presenti per descrivere la QoS sono: frequenza della CPU, percentuale di utilizzo della CPU, memoria fisica disponibile e memoria libera disponibile. Tali attributi riportano per ogni servizio un valore di default, che può essere modificato in base alle esigenze di esecuzione con valori scelti dall'utente, ma sempre con valori compatibili alle caratteristiche presenti nelle macchine disponibili. Associato ad ogni servizio è disponibile una check-box che permette di rendere il servizio vincolato o non vincolato ad un nodo SDE (Switch/No Switch). Nel caso di nodo vincolato è possibile usare solo quel servizio e non copie replicate, quindi la sostituzione del servizio in caso di QoS bassa è disabilitata. In tal caso il workflow userà il servizio anche se la sua QoS non è rispettata. Questa caratteristica rende il sistema compatibile con servizi state-full, dove parte dell'elaborazione parziale è memorizzata nel servizio stesso. Da notare che solitamente il risultato parziale del workflow è conservato nel Token e nelle cache dei servizi.

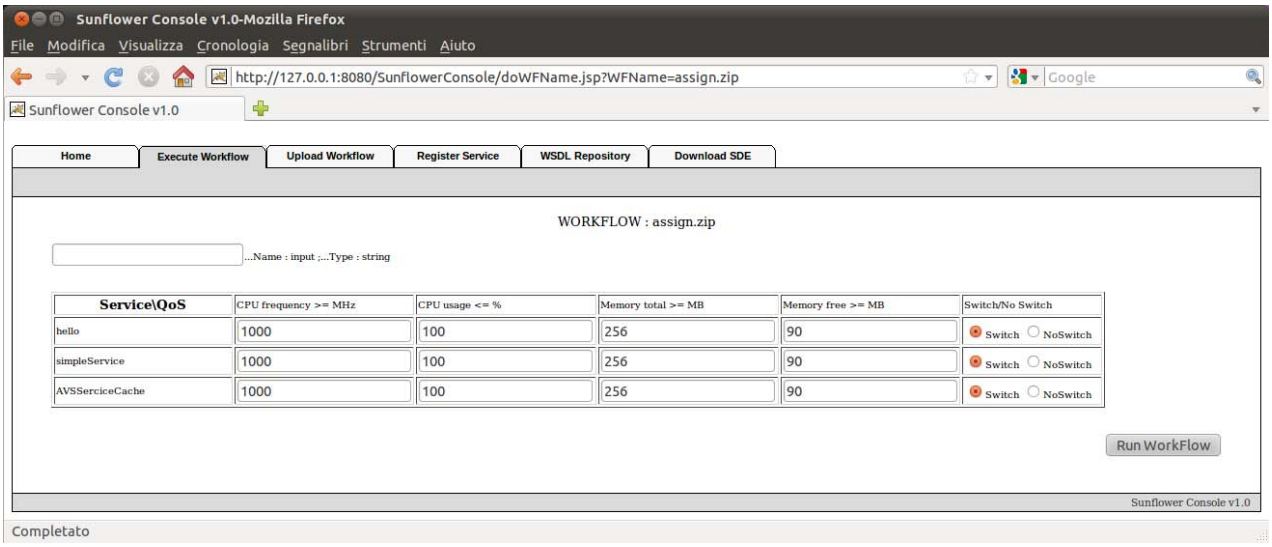

Figura 8: Form per l'inserimento dei parametri relativi alle variabili e alla QoS.

Terminata la fase d'inizializzazione del workflow è possibile avviarne l'esecuzione mediante il bottone "Run workflow". Prima di iniziare l'esecuzione vera e propria, il codice XML-BPEL presente nel package è tradotto in base al modello a coreografia basato su Rete di Petri. Successivamente, ogni frammento di codice BPEL è assegnato a un nodo SDE in base al servizio reale scelto per concretizzare del workflow astratto, ed il Token è inizializzato con i parametri di input del workflow. In fine, Token e frammenti XML-BPEL sono inviata ai rispettivi nodi SDE per l'esecuzione.

6 *Monitoring e visualizzazione del risultato*

Sunflower fornisce un'interfaccia grafica che consente di monitorare l'esecuzione del workflow attraverso due rappresentazioni grafiche, il primo è quello definito con Eclipse BPEL Designer e il secondo quello è quello risultante dalla decomposizione per l'esecuzione in coreografia (vedi figura 9). Nel riquadro di sinistra è riportato il workflow completo generato dal codice BPEL classico. Nel riquadro di destra sono mostrati i nodi con i relativi frammenti di codice BPEL che devono eseguire in coreografia. Il flusso di esecuzione del workflow può essere seguito attraverso i simboli che appaiono di volta in volta sui vari costrutti BPEL, il cui significato è riportato di seguito:

- Il pallino arancione, rappresenta il Token ed indica l'ultima operazione eseguita correttamente.
- Il pallino rosso, indica un errore irreversibile che richiede la procedura di "recovery del workflow", dopo pochi secondi il monitoring verrà ridiretto sulla nuova istanza di workflow per continuare l'esecuzione.
- Una "R" rossa sui costrutti di "invoke", indica che il servizio associato all'invoke non era in grado di rispettare la QoS richiesta ed è stata eseguita con successo la procedura di Routing per sostituirlo.

- Le lettere “RC” rosse sul costrutto di “invoke”, indica che il servizio associato all’invoke non era in grado di rispettare la QoS richiesta ed è stata eseguita con successo la procedura di Routing per sostituirlo, ma l’unico servizio disponibile era sul “Cloud”.

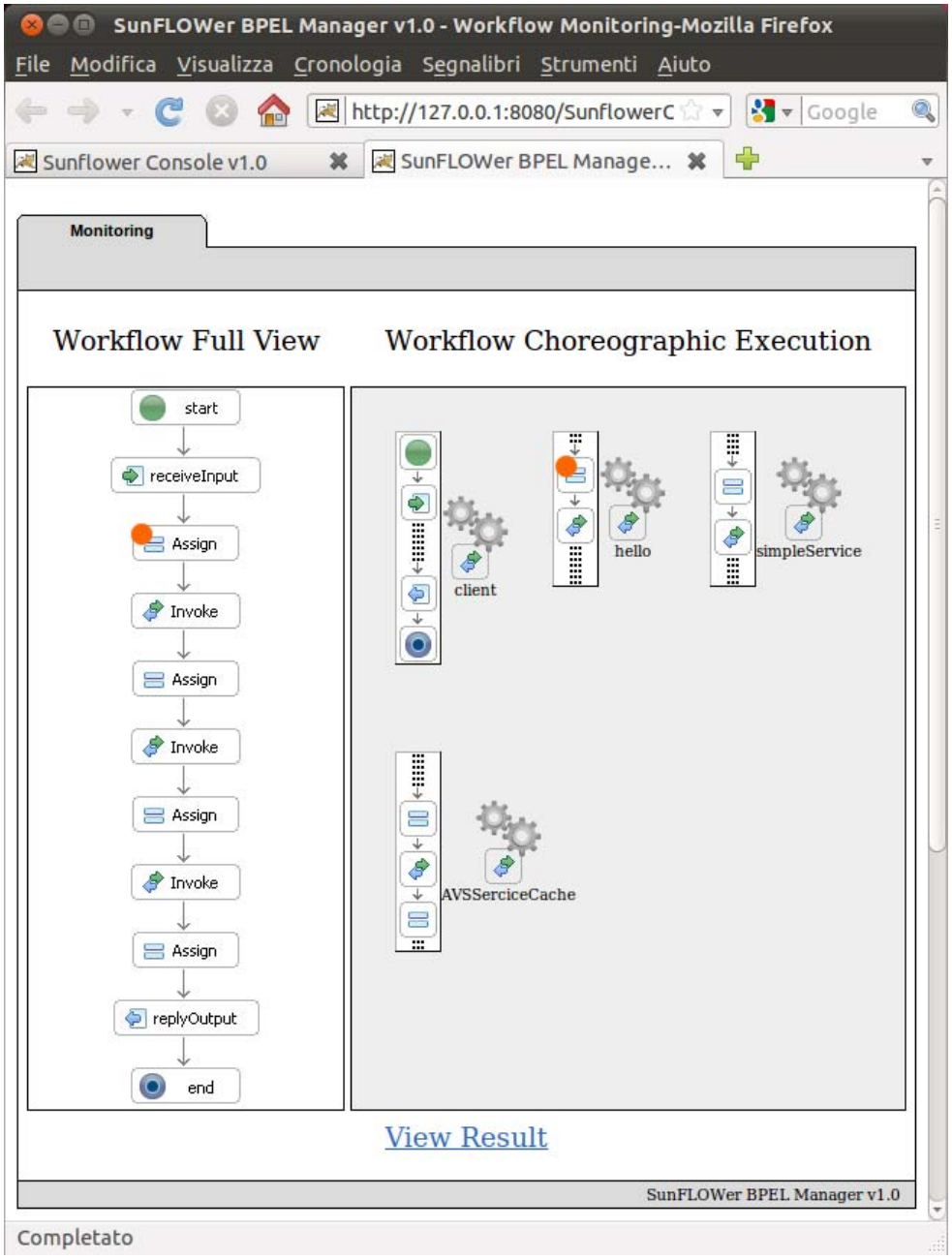

Figura 9: Monitoring di un workflow in Sunflower.

Quando il Token raggiunge il costrutto “end”, l’esecuzione del workflow BPEL è completa ed è possibile visualizzare il risultato cliccando sul link “View Result”. Il risultato verrà visualizzato in base alle impostazioni predefinite del browser, mentre nel caso in cui il risultato sia un URL il

monitoring ridirigerà la connessione verso URL remoto per poi visualizzarlo sempre in base alle impostazioni del brower.

7 *Sunflower & Eucalyptus Cloud*

Sunflower, grazie ad algoritmi di scheduling statico e dinamico basati su QoS è in grado di bilanciare il carico di lavoro dei workflow BPEL distribuendolo sui vari SDE. Sfortunatamente, per quanto si possa ottimizzare l'allocazione sui servizi reali, nel caso di colli di bottiglia e picchi di carico se il numero di servizi non è sufficiente per soddisfare le richieste, il sistema comincerà a rifiutare ulteriori workflow e a terminare con errore quelli già in esecuzione. Per superare questi picchi di carico e colli di bottiglia in tempi rapidi e a basso costo, si è pensato di integrare in Sunflower un meccanismo di "service on-demand" basato su "Eucaluptus Cloud" (http://www.eucalyptus.com/).

Prima di creare un service on-demand per Sunflower è necessario soddisfare i seguenti prerequisiti:

1. Installazione di Eucaliptus 2.0.2 e Euca2ools 1.3.1 .
   (http://open.eucalyptus.com/downloads)
2. Studio del tutorial "Eucalyptus Getting Started" e "Eucalyptus Image Management" per acquisire i concetti e la nomenclatura usati successivamente in questo documento.
   (http://open.eucalyptus.com/wiki/EucalyptusGettingStarted_v2.0
   http://open.eucalyptus.com/wiki/EucalyptusImageManagement_v2.0)

Per creare un service on-demand per Sunflower bisogna:

3. Selezionare e installare tra le "Images" di virtual machine (VM) presenti nello "Store" quella più adatta al tipo di servizio che si deve creare.
4. Recuperare l'ID della VM Image installata (emi-12345678) e creare un'istanza da essa (euca-run-instances –k … -t … emi-12345678).
5. Accede con ssh all'istanza di VM creata ed eseguire tutti i passi necessari per installare il servizio da utilizzare in Sunflower (ssh –X –i … root@ip_instance).
6. Salvare l'immagine del disco virtuale modificato e aggiungerlo come indicato in Managing Eucalyptus Images alle VM Images, recuperando l'ID della VM Image creata.

Per registrare un service on-demand su Sunflower, bisogna accedere alla scheda Register Service della SC e cliccare su "Register Cloud On-demand Service" (vedi figura 10). La Form in essa contenuta ha vari campi, che vanno a formare il contenuto di una tupla che lega assieme un servizio on-demand, un servizio astratto e un certificato per Euca2ool.

- URL del WSDL relativo al servizio on-demand da registrare
    1. Protocollo usato dal servizio (http / https)
    2. ID della WM Image contenete il servizio (emi-XXXXXXXX)
    3. Porta usata dal servizio (80 / 443 / 8080)
    4. Path del WSDL servizio creato

(es.: http://emi-12345678:8080/axis/Hello.jws?wsdl).

- URL del WSDL astratto nel WSDL Repository, da cui è stato generato il sevizio installato nel disco immagine della VM
  (es.: http://127.0.0.1:8080/SunflowerConsole/WSDLRepository/axis/Hello.wsdl).
- Locazione del file "euca2-admin-x509.zip", scaricabile dall'interfaccia web di Eucalyptus o tramite il comando "euca_conf --get-credentials euca2-admin-x509.zip", contenete le credenziali per accedere a Eucalyptus mediante Euca2ools.

N.B.: L' "URL del WSDL relativo al servizio on-demand da registrare" è identico all' "URL del WSDL relativo al servizio reale da registrare" in cui l'IP è sostituito dall'ID della VM Image che contiene il servizio.

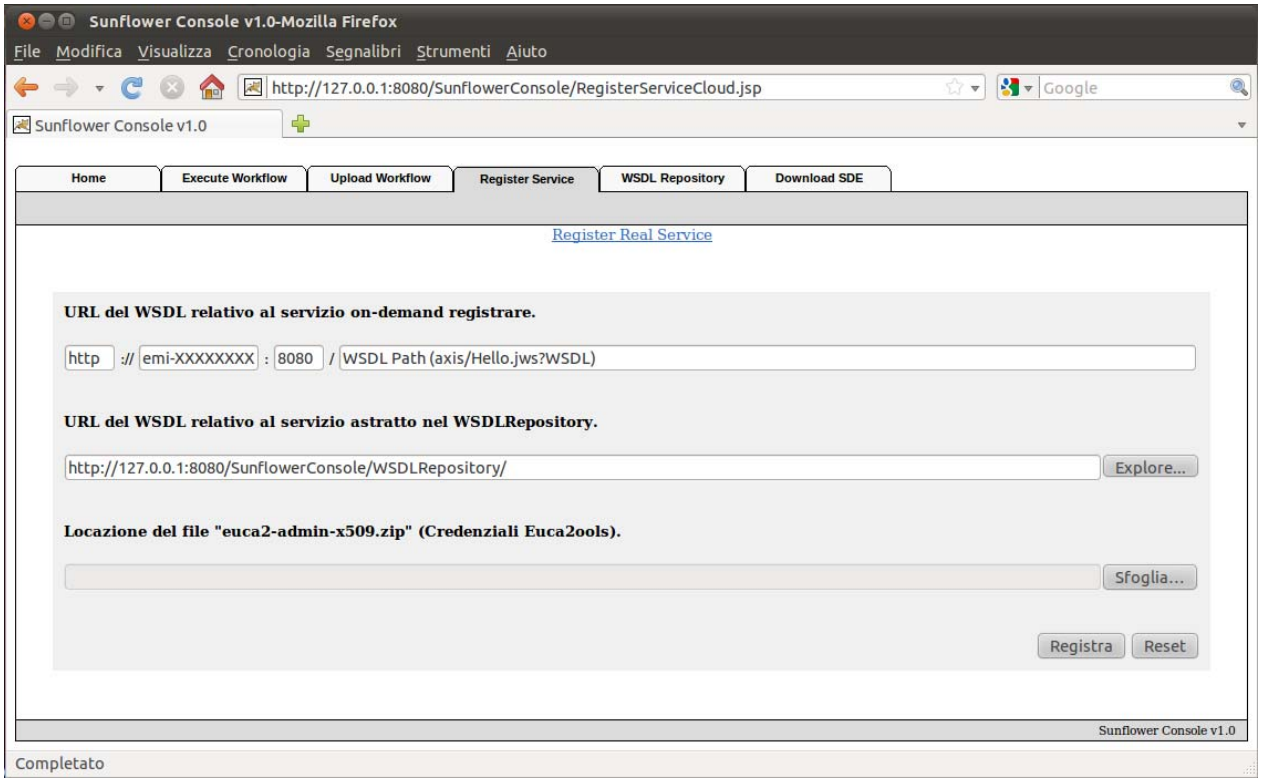

Figura 10: Registrazione di un servizio on-demand con il relativo servizio astratto e certificato.

Come si può notare, la registrazione di un servizio on-demand Eucalyptus è molto simile alla registrazione di un servizio reale, mentre la logica di utilizzo è fortemente differente. Analizzando la Form si vede che l'URL del WSDL del servizio non è più un indirizzo reale, ma un URL che in fase di utilizzo andrà opportunamente risolto per farlo puntare al servizio on-demand. Vediamo i passi di come questa risoluzione avviene:

1. Quando non sono più disponibili servizi reali associati al servizio astratto richiesto, si verifica se esiste un servizio on-demand del tipo richiesto.

2. Se esiste, si recupera L'ID della VM Image e la SC tramite le librerie Euca2ool provvedere a creare un'istanza di VM.
3. Si recupera l'IP dell'istanza di VM attiva, e si sostituisce nell'URL del WSDL del servizio on-demand al posto dell'ID della VM image.
4. L'URL del servizio on-demand creato è restituito e utilizzato come un normale servizio reale.

Un'altra importante differenza sta nel fatto che il servizio on-demand non è gestito direttamente da un suo SDE, ma sfrutta altri SDE per l'esecuzione e Eucalyptus per la gestione della QoS. Infatti, se per i servizi reali bisogna registrare il SDE di competenza, per i servizi on-demand bisogna registrare le credenziali Euca2ools per l'accesso ad Eucalyptus. Grazie alle credenziali di accesso, Euca2ools è in grado di istanziare una VM con la QoS richiesta dall'utente per il servizio, ed Eucalyptus provvederà a mantenere tale QoS per tutto il ciclo di vita dell'istanza di VM. Venuta a mancare la necessità di monitorare direttamente la QoS del servizio da parte del SDE, i frammenti di codice BPEL relativi al servizio on-demand possono essere tranquillamente accorpati con quelli associati ad un altro servizio reale, per poi invocare il servizio on-demand non più come servizio locale ma come servizio remoto. Tale strategia per i servizi on-demand presenta due vantaggi, il primo è quello di mantenere l'istanza di VM più leggera possibile, la seconda è quella di evitare problemi di comunicazione tra gli SDE dei servizi reali e un eventuale SDE installato in una istanza di VM.

Il tempo di vita di un'istanza di VM è determinato dalla durata del workflow, infatti quando questo termina lancia un messaggio di garbage che distrugge l'istanza del workflow eseguito, e nel caso di servizi on-demand creati durante la sua esecuzione provvede a terminarne l'istanza di VM (euca-terminate-instance i-xxxxxxxx).