



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

# **Validazione e Testing del Secondo Prototipo Progetto SFIDA-PMI**

Giuseppe Papuzzo, Danilo Cistaro

**RT-ICAR-CS-08-11**

**Dicembre 2008**



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](http://www.icar.cnr.it)

– Sezione di Napoli, Via P. Castellino 111, 80131 Napoli, URL: [www.na.icar.cnr.it](http://www.na.icar.cnr.it)

– Sezione di Palermo, Viale delle Scienze, 90128 Palermo, URL: [www.pa.icar.cnr.it](http://www.pa.icar.cnr.it)

# Indice

|  |           |
|--|-----------|
| <b>1 EXECUTIVE SUMMARY .....</b>   | <b>5</b>  |
| <b>2 INTRODUZIONE .....</b>  | <b>6</b>  |
| <b>3 AMBIENTE DI VALIDAZIONE .....</b>   | <b>7</b>  |
| 3.1 RIASSUNTO DELLE FUNZIONALITÀ DEL PROTOTIPO .....   | 7         |
| 3.2 SCENARIO DI RIFERIMENTO PER LA VALIDAZIONE .....   | 9         |
| <b>4 VALIDAZIONE E TESTING DELLE FUNZIONALITÀ.....</b>   | <b>12</b> |
| 4.1 VALIDAZIONE AMBIENTE DI AMBIENTE DI MODELLAZIONE A LIVELLO BUSINESS .....                                      | 12        |
| 4.1.1 Modellazione Processo OEM a livello Business.....  | 12        |
| 4.1.2 Modellazione Processo Fornitore a livello Business.....  | 13        |
| 4.2 VALIDAZIONE AMBIENTE DI MODELLAZIONE A LIVELLO PROCESSI ESEGUIBILI E AMBIENTE DISCOVERY<br>& COMPOSITION ..... | 13        |
| 4.2.1 Modellazione Processo Fornitore a livello eseguibile.....  | 13        |
| 4.2.2 Registrazione fornitori di Web-Services (b80).....   | 14        |
| 4.2.3 Ricerca e Composizione dei Web-Services (b80 / ACP).....   | 14        |
| 4.2.4 Generazione Vista Pubblica per il Fornitore .....  | 15        |
| 4.2.5 Modellazione Processo OEM a livello eseguibile .....   | 15        |
| 4.3 VALIDAZIONE AMBIENTE DI ESECUZIONE – ESECUZIONE NON SU GRID.....   | 16        |
| 4.3.1 Esecuzione Processo.....   | 16        |
| 4.4 VALIDAZIONE AMBIENTE DI ESECUZIONE – ESECUZIONE SU PIATTAFORMA GRID .....                                      | 18        |
| 4.4.1 Creazione di un Grid workflow per un processo automatizzato .....  | 18        |
| 4.4.2 Inclusione del processo Grid-Workflow nell'ambiente di modellazione .....                                    | 22        |
| 4.4.3 Esecuzione di un Grid-workflow tramite Shark.....  | 24        |
| 4.4.4 Tracciatura dell'esecuzione di un Grid workflow .....  | 25        |
| 4.5 VALIDAZIONE AMBIENTE DI PROCESS INTELLIGENCE.....  | 27        |
| 4.5.1 Importazione di log di esecuzione dei processi collaborativi.....  | 27        |
| 4.5.2 Analisi con gli strumenti di Process Intelligence.....   | 30        |
| <b>5 CONCLUSIONI .....</b>   | <b>39</b> |

## Indice delle Figure

|   |    |
|---|----|
| Figura 1 - Architettura del prototipo Beta.....                           | 7  |
| Figura 2: Il processo collaborativo di OR5.2.....                         | 10 |
| Figura 3: Schema workflow del processo <i>SELLGOODS</i> .....             | 10 |
| Figura 4: Il processo OEM a livello business.....                         | 13 |
| Figura 5: il processo del fornitore a livello eseguibile.....             | 14 |
| Figura 6: esportazione vista pubblica del fornitore.....                  | 15 |
| Figura 7: processo OEM a livello eseguibile.....                          | 16 |
| Figura 8: esecuzione del processo con Shark.....                          | 17 |
| Figura 9: il Grid-workflow “sell goods”.....                              | 22 |
| Figura 10: caricamento del processo Grid-workflow in Shark.....           | 24 |
| Figura 11: impostazione dei parametri di input e di QoS del processo..... | 24 |
| Figura 12: risultato del processo Grid su Sunflower.....                  | 25 |
| Figura 13: Sunflower – monitoraggio esecuzione.....                       | 25 |
| Figura 14: logs generati dall'esecuzione.....                             | 27 |
| Figura 15: ProM-Import4SFIDA.....   | 28 |
| Figura 16: logs in formato .mxml.....                                     | 30 |
| Figura 18: componente WFM.....  | 33 |
| Figura 19: Esecuzione su WFM.....   | 34 |
| Figura 20: ProM - componente PM.....                                      | 35 |
| Figura 21: ProM - schema di workflow scoperto.....                        | 36 |
| Figura 22: schema nodo R0.....  | 37 |
| Figura 23: astrazione processo.....                                       | 38 |

## **Indice delle Tabelle**

|   |   |
|---|---|
| Tabella 1: elenco delle funzionalità sviluppate in A4.1, A4.3 e A4.4..... | 8 |
| Tabella 2: Funzionalità al supporto dell'integrazione.....                | 9 |
| Tabella 3: Mappatura degli ambienti sui casi d'uso.....                   | 9 |

# **1 Executive Summary**

Questo documento ha lo scopo di illustrare i risultati delle attività di validazione e testing del Secondo Prototipo (Beta) per la modellazione e l'esecuzione dei processi di business collaborativi. Tali attività si collocano nell'ambito dell'Obiettivo Realizzativo OR4 ("Servizi di supporto alla gestione life-cycle dei processi collaborativi tra PMI"), che si ripropone di fornire tutti gli strumenti necessari per i servizi di supporto alla gestione del life-cycle dei processi collaborativi, dalla fase di modellazione dei processi, alla fase di attuazione, fino alla fase di negoziazione. E' importante osservare che le varie attività di sperimentazione descritte in questo documento sono state condotte in riferimento a più casi di studio dell'OR5.

Dopo aver richiamato l'architettura del prototipo ed avere illustrato lo scenario usato per la validazione, il documento descrive l'applicazione delle principali funzionalità del prototipo.

## 2 Introduzione

Questo documento è un risultato dell'Obiettivo Realizzativo OR4 ("Servizi di supporto alla gestione life-cycle dei processi collaborativi tra PMI"), e discute la validazione ed il test delle nuove funzionalità sviluppate nel Secondo Prototipo (Beta) per la modellazione e l'esecuzione dei processi di business collaborativi, realizzato nell'ambito dell'attività A4.2.

In sintesi, il risultato dell'attività A4.2 ("Ambiente adattativo di modellazione ed esecuzione dei processi sulla piattaforma a Grid") è un ambiente adattativo di modellazione ed esecuzione dei processi inter-aziendali, che in aggiunta agli strumenti già realizzati nella prima versione del prototipo (Alfa) in A4.1, integra gli strumenti di process intelligence e di composizione/negoziazione dei servizi realizzati in A4.3 e A4.4.

Inoltre, il prototipo Beta supporta l'esecuzione di processi automatizzati sulla piattaforma grid, permettendo di sfruttare, in particolare, il framework Sunflower, sviluppato nell'attività A2.2. Questo framework estende la piattaforma di griglia, con la capacità di gestire workflow dinamici e adattativi a servizi (Web/Grid service) in grado di auto-riconfigurarsi in risposta a variazioni del workload del sistema e/o guasti generati dal crash di nodi, tenendo conto delle risorse disponibili e della qualità del servizio offerto.

Il resto del documento è organizzato come segue. La sezione 3 richiama le funzionalità implementate nel prototipo Beta ed introduce lo scenario utilizzato per la loro validazione.

La sezione 4 è descrive quindi la validazione delle principali funzionalità implementate nel prototipo.

## 3 Ambiente di Validazione

### 3.1 Riassunto delle Funzionalità del Prototipo

La Figura 1 mostra l'architettura del prototipo Beta.

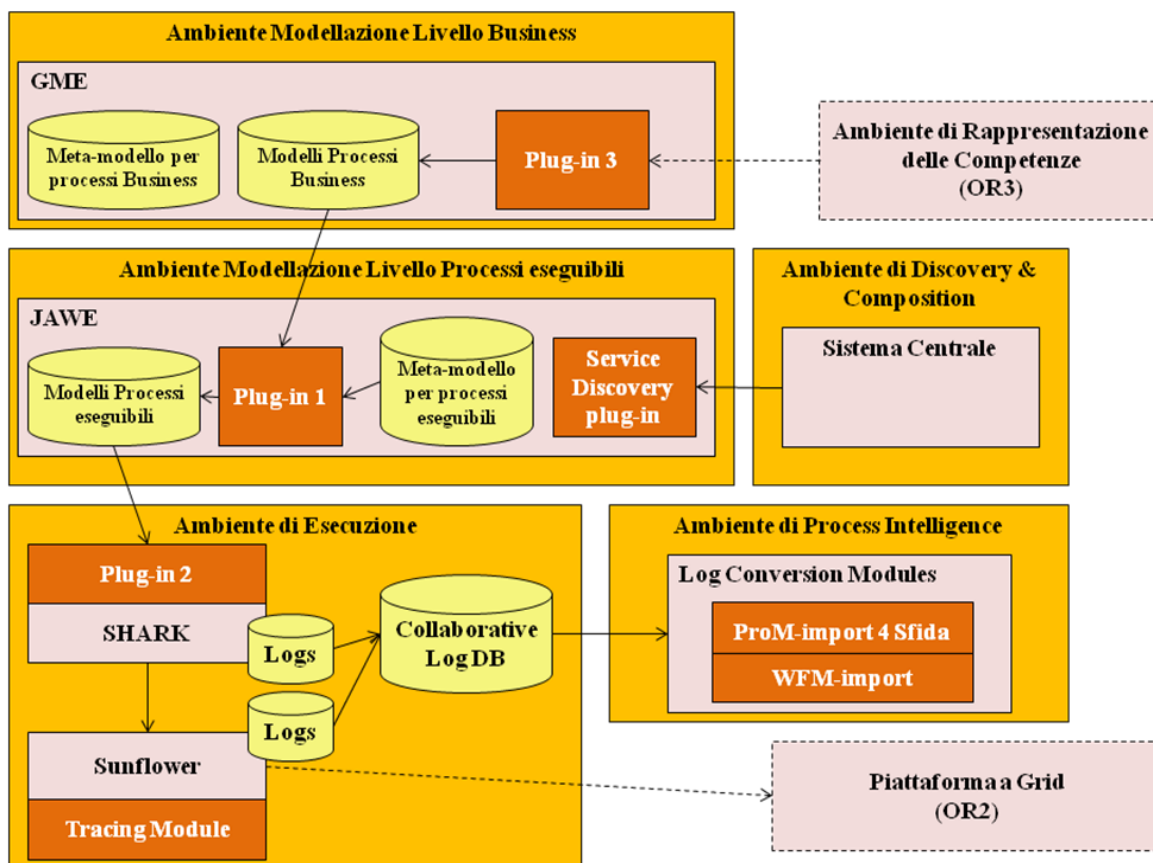


Figura 1 - Architettura del prototipo Beta.

Come descritto nel deliverable R4.2.2, il prototipo è composto dai seguenti ambienti:

1. **Ambiente di Modellazione a Livello Business :** questo ambiente offre gli strumenti per modellare i processi collaborativi a livello di business.
2. **Ambiente di Modellazione a Livello Processi Eseguibili :** questo ambiente offre gli strumenti per modellare i processi collaborativi a livello eseguibile.
3. **Ambiente di Esecuzione :** questo ambiente offre gli strumenti per eseguire i processi modellati con l'ambiente di Modellazione per Processi Eseguibili.
4. **Ambiente di Discovery & Composition :** questo ambiente permette di scoprire servizi basandosi su una descrizione in linguaggio naturale fornita dall'analista / modellatore. i.
5. **Ambiente di Process Intelligence:** permette di scoprire schemi di processo a partire dall'analisi dei log grazie a tecniche di Process Mining e di Workflow Mining. Questo ambiente, fornisce funzionalità innovative di *process mining* e di *process analysis*, di supporto all'analisi ed alla progettazione dei processi collaborativi

Il prototipo è quindi dotato di numerose funzionalità: la seguente tabella riassume TUTTE le funzionalità che esso offre, come già riportate nei deliverables R4.1.1, R4.3.1, R4.4.1, evidenziando l'ambiente in cui sono offerte:

| <b>ID</b> | <b>Nome Funzionalità</b>                               | <b>Ambiente</b>  |
|-----------|--|--|
| f.4.1.1   | Modellazione grafica di un processo a livello business | Ambiente di Modellazione a Livello Business            |
| f.4.1.2   | Modellazione grafica di un processo eseguibile         | Ambiente di Modellazione a Livello Processi Eseguibili |
| f.4.1.3   | Generazione di viste pubbliche / private               | Ambiente di Modellazione a Livello Processi Eseguibili |
| f.4.1.4   | Esportare / importare le viste pubbliche               | Ambiente di Modellazione a Livello Processi Eseguibili |
| f.4.1.5   | Gestione ciclo di vita del processo                    | Ambiente di Modellazione a Livello Processi Eseguibili |
| f.4.1.6   | Possibilità di definire varie tipologie di attività    | Ambiente di Modellazione a Livello Processi Eseguibili |
| f.4.1.7   | Esecuzione dei processi                                | Ambiente di Esecuzione                                 |
| f.4.1.8   | Generazione di log dall'esecuzione                     | Ambiente di Esecuzione                                 |
| f.4.3.1.1 | WF-pattern mining                                      | Ambiente di Process Intelligence                       |
| f.4.3.2.1 | Workflow discovery                                     | Ambiente di Process Intelligence                       |
| f.4.3.2.2 | Hierarchy discovery                                    | Ambiente di Process Intelligence                       |
| f.4.3.2.3 | Log clustering   | Ambiente di Process Intelligence                       |
| f.4.3.2.4 | Discriminant rule extraction                           | Ambiente di Process Intelligence                       |
| f.4.3.3.1 | Process abstraction                                    | Ambiente di Process Intelligence                       |
| f.4.3.3.2 | Process checking                                       | Ambiente di Process Intelligence                       |
| f.4.3.3.3 | Data&Knowledge management                              | Ambiente di Process Intelligence                       |
| f.4.3.3.4 | Log preprocessing                                      | Ambiente di Process Intelligence                       |
| f.4.3.3.5 | Log simulation   | Ambiente di Process Intelligence                       |
| f.4.3.3.6 | Schema editing   | Ambiente di Process Intelligence                       |
| f.4.3.3.7 | Process visualization                                  | Ambiente di Process Intelligence                       |
| f.4.4.1   | Registra Servizio                                      | Ambiente di Discovery & Composition                    |
| f.4.4.2   | Completa WSDL  | Ambiente di Discovery & Composition                    |
| f.4.4.3   | Salva in Repository                                    | Ambiente di Discovery & Composition                    |
| f.4.4.4   | Calcola Marcatura                                      | Ambiente di Discovery & Composition                    |
| f.4.4.5   | Interroga Repository                                   | Ambiente di Discovery & Composition                    |
| f.4.4.6   | Confronta Marcature                                    | Ambiente di Discovery & Composition                    |
| f.4.4.7   | Verifica Presenza Nuovi Servizi                        | Ambiente di Discovery & Composition                    |
| f.4.4.8   | Invoca Sistema Centrale                                | Ambiente di Discovery & Composition                    |
| f.4.4.9   | Aggiorna Informazioni                                  | Ambiente di Discovery & Composition                    |
| f.4.4.10  | Gestisce Tassonomie                                    | Ambiente di Discovery & Composition                    |
| f.4.4.11  | Riconduce/Riclassifica i servizi                       | Ambiente di Discovery & Composition                    |
| f.4.4.12  | Espone Risultati Ricerca                               | Ambiente di Discovery & Composition                    |
| f.4.4.13  | Cerca Servizi  | Ambiente di Discovery & Composition                    |
| f.4.4.14  | Elabora Risultati                                      | Ambiente di Discovery & Composition                    |

**Tabella 1: elenco delle funzionalità sviluppate in A4.1, A4.3 e A4.4**



In aggiunta a queste funzionalità, il prototipo Beta offre una serie di funzionalità necessarie per l'integrazione (si veda R4.2.1), ognuna delle quali estende uno specifico ambiente di quelli sopra indicati:

| ID     | Nome Funzionalità   | Ambiente esteso  |
|--------|---|--|
| f4.2.1 | Modellazione di sotto-processi eseguibili sulla piattaforma Grid                | Ambiente di Modellazione a Livello Processi Eseguibili |
| f4.2.2 | Esecuzione sotto-processi su piattaforma Grid                                   | Ambiente di Esecuzione                                 |
| f4.2.3 | Tracciatura delle attività e memorizzazione nei log di esecuzione dei processi. | Ambiente di Esecuzione                                 |
| f4.2.4 | Importazione di log MXML nel modulo di Workflow Mining                          | Ambiente di Process Intelligence                       |
| f4.2.5 | Memorizzazione dei log di esecuzione dei processi collaborativi                 | Ambiente di Esecuzione                                 |
| f4.2.6 | Conversione dei log di esecuzione di processi collaborativi nel formato MXML    | Ambiente di Esecuzione                                 |
| f4.2.7 | Integrazione di funzionalità di negoziazione e composizione di servizi          | Ambiente di Discovery & Composition                    |
| f4.2.8 | Integrazione di funzionalità per la modellazione delle competenze               | Ambiente di Modellazione a Livello Business            |

Tabella 2: Funzionalità al supporto dell'integrazione

## 3.2 Scenario di riferimento per la validazione

Vista la complessità del prototipo Beta, le sue funzionalità sono state validate in più di un caso utente. Inoltre essendo lo scopo di questo report mostrare la validazione del prototipo Beta nei reali casi d'uso, appare indispensabile far riferimento al testing e validazione dei vari ambienti piuttosto che alle singole sotto-funzionalità.

La seguente tabella riassume come è avvenuto il testing:

| Nome Ambiente   | Principale Caso d'uso | Altri casi d'uso |
|---|-----------------------|------------------|
| Ambiente di Modellazione a Livello Business             | OR5.2                 | OR5.4            |
| Ambiente di Modellazione a Livello Processi Eseguibili  | OR5.2                 | OR5.4, OR5.5     |
| Ambiente di Esecuzione – esecuzione non su Grid         | OR5.2                 | OR5.4            |
| Ambiente di Esecuzione – esecuzione su piattaforma Grid | OR5.3                 | OR5.5            |
| Ambiente di Discovery & Composition                     | OR5.2                 |                  |
| Ambiente di Process Intelligence                        | OR5.3                 |                  |

Tabella 3: Mappatura degli ambienti sui casi d'uso

Come si può notare il principale caso d'uso è quello di OR5.2 – collaborazione nell'industria automobilistica. Questo caso d'uso è incentrato esclusivamente sugli strumenti di OR4, coprendo le fasi di modellazione di un processo collaborativo fra un OEM ed una serie di fornitori sia a livello

business che a livello tecnico, la ricerca ed inserimento di servizi grazie all'ambiente di Discovery&Composition, e la sua esecuzione, senza tuttavia includere attività di tipo Grid.

Il seguente schema mostra una visione d'insieme del processo collaborativo:

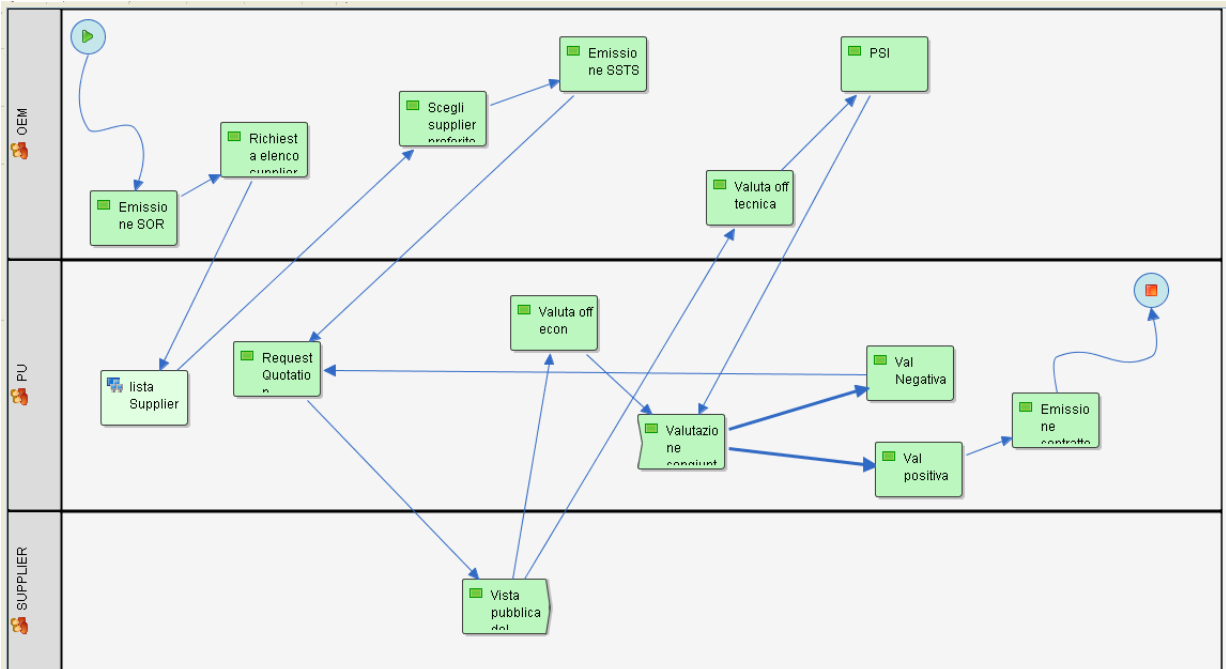


Figura 2: Il processo collaborativo di OR5.2

Le funzionalità legate all'ambiente di Process Intelligence ed esecuzione su piattaforma Grid sono state invece validate all'interno del caso d'uso OR5.3 – collaborazione nell'industria tessile. Questo caso d'uso sfrutta principalmente gli scenari di OR3, tuttavia per mostrare l'integrazione degli strumenti di OR3 con quelli di OR4 il caso considera non solo la pianificazione della supply-chain basata sulle competenze, ma anche l'avvio di processi collaborativi.

Lo schema concettuale del processo considerato, denominato *SellGoods*, illustrato nella seguente figura.

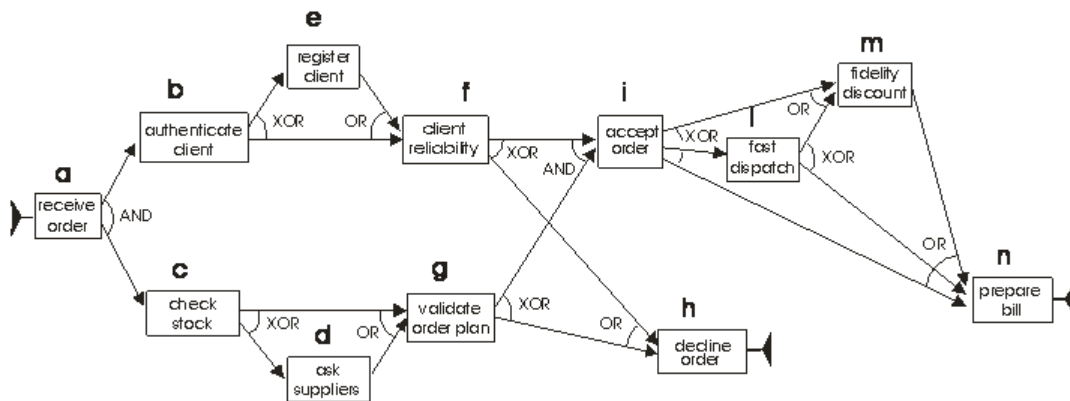


Figura 3: Schema workflow del processo SELLGOODS.

Per completare la panoramica, gli strumenti di modellazione ed esecuzione sono stati utilizzati anche all'interno del caso OR5.4 (industria degli elettrodomestici) ed in OR5.5 al fine di mostrare l'integrazione di OR4 con gli strumenti di OR1 ed OR2.

Si può far riferimento al documento R5.0.1 per maggiori dettagli sulle descrizione dei casi di studio.

## 4 Validazione e Testing delle funzionalità

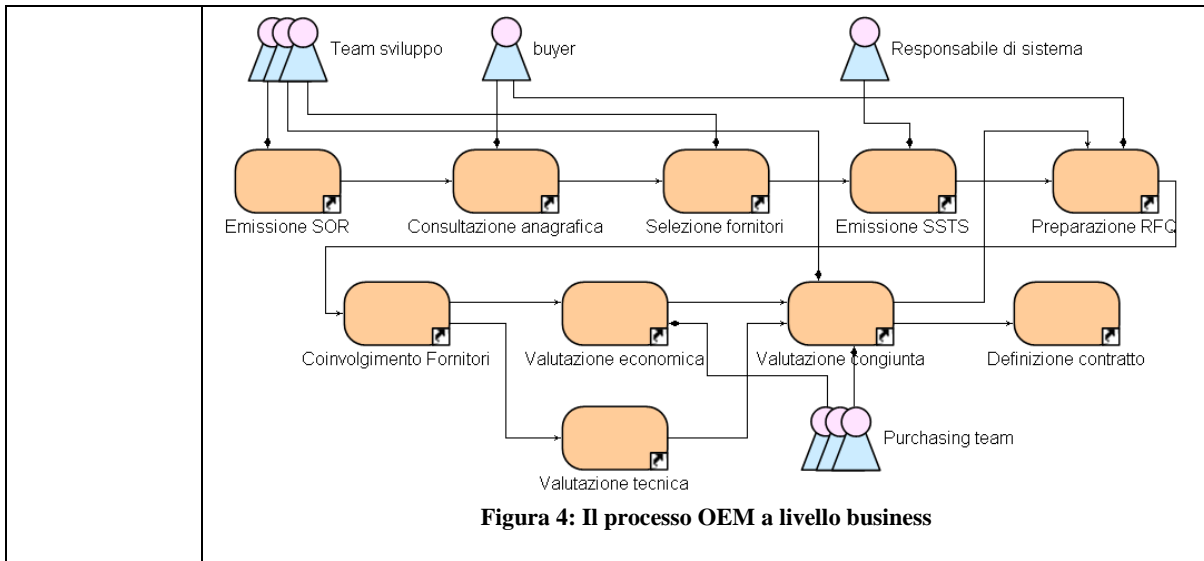
In questa sezione verranno riportate le tabelle descritte nel precedente capitolo che mostrano il corretto funzionamento del prototipo, negli scenari di validazione riportati in Tabella 3.

Per ogni gruppo di funzionalità è stato creato un apposito paragrafo.

### 4.1 Validazione Ambiente di Ambiente di Modellazione a Livello Business

#### 4.1.1 Modellazione Processo OEM a livello Business

|                           |   |
|---------------------------|---|
| <b>ID</b>                 |   |
| <b>TITOLO</b>             | Modellazione grafica del processo OEM a livello business  |
| <b>OBIETTIVO</b>          | L'utente orientato al business sarà in grado di definire un generico processo come insieme di attività inserite in un determinato flusso.   |
| <b>SET-UP (input)</b>     | <ul style="list-style-type: none"> <li>• Disponibilità dell'ambiente GME su una macchina di test.</li> <li>• Disponibilità del meta-modello "Sfida Processo"</li> </ul>   |
| <b>STEP ESEGUITI</b>      | <ul style="list-style-type: none"> <li>• Aprire l'ambiente GME</li> <li>• Dal menu File si seleziona Register Paradigm per registrare il meta-modello SFIDA-Processo.</li> <li>• Dal menu File si seleziona New Project, ed in seguito si sceglie il "paradigm" registrato al punto precedente. Conseguentemente si seleziona la voce Create New; scegliere infine il path ed il nome del progetto (corrispondente al file .mga).</li> </ul> <p>Ora il nuovo processo è pronto per essere modellato.</p> <p>Di seguito sono elencati alcuni dei passi riprodotti per la modellazione dei processi; lo scopo è quello di fornire un'idea delle operazioni fondamentali con cui si modellano questo tipo di competenze aziendali.</p> <p><u>Creazione di una vista:</u></p> <ul style="list-style-type: none"> <li>• <i>Aggregate Tab</i> in the model browser</li> <li>• Tasto destro su <i>Root Folder</i> -&gt; <i>Insert Model</i> -&gt; selezionare il <i>diagram Process</i></li> <li>• Seleziona il modello appena creato; nell'attribute browser è possibile rinominarlo</li> <li>• Doppio click sul modello appena creato: nel part browser sono presenti gli elementi disponibili</li> <li>• Trascinare gli elementi desiderati nel <i>Model Editing Window</i>, e rinominarli nell'<i>Attribute Browser</i></li> </ul> |
| <b>RISULTATI (output)</b> | Il processo modellato a livello business è visibile nell'immagine seguente:   |



#### 4.1.2 Modellazione Processo Fornitore a livello Business

|                           |   |
|---------------------------|---|
| <b>ID</b>                 |   |
| <b>TITOLO</b>             | Modellazione grafica del processo del fornitore a livello business  |
| <b>OBIETTIVO</b>          | L'utente (a lato fornitore) orientato al business sarà in grado di definire un generico processo come insieme di attività inserite in un determinato flusso.            |
| <b>SET-UP (input)</b>     | <ul style="list-style-type: none"> <li>• Disponibilità dell'ambiente GME su una macchina di test.</li> <li>• Disponibilità del meta-modello "Sfida Processo"</li> </ul> |
| <b>STEP ESEGUITI</b>      | Gli step eseguiti sono analoghi a quelli riportati nella precedente scheda, ma riferiti allo specifico processo del fornitore   |
| <b>RISULTATI (output)</b> | <p>Il processo del fornitore modellato a livello business è visibile nell'immagine seguente:</p> <p><b>Figura 5: Il processo del fornitore a livello business</b></p>   |

### 4.2 Validazione Ambiente di Modellazione a Livello Processi Eseguibili e Ambiente Discovery & Composition

#### 4.2.1 Modellazione Processo Fornitore a livello eseguibile

|           |  |
|-----------|--|
| <b>ID</b> |  |
|-----------|--|

|                           |   |
|---------------------------|---|
| <b>TITOLO</b>             | Modellazione grafica lato fornitore di un processo eseguibile   |
| <b>OBIETTIVO</b>          | Il fornitore può modellare il proprio processo interno, mostrando all'OEM solo input ed output  |
| <b>SET-UP (input)</b>     | Disponibilità dell'ambiente JaWE customizzato   |
| <b>STEP ESEGUITI</b>      | Il processo è costituito dalla ricezione della Request for Quotation da parte dell'OEM e dalla successiva invocazione a un web-service di cui non ci conosce però l'end point – esso sarà trovato grazie al sistema di “discovery&composition” descritto nelle prossime due schede. |
| <b>RISULTATI (output)</b> | <p>Il processo del fornitore è visibile nella figura</p> <p><b>Figura 5: il processo del fornitore a livello eseguibile</b></p>   |

#### 4.2.2 Registrazione fornitori di Web-Services (b80)

<< set-up ambiente di discovery&composition >>

|                           |  |
|---------------------------|--|
| <b>ID</b>                 |  |
| <b>TITOLO</b>             | Registrazione di un fornitore di Web-Services  |
| <b>OBIETTIVO</b>          | Permettere a un fornitore di Web-Services di registrare il suo servizio nell'ambiente di “Discovery & Composition” |
| <b>SET-UP (input)</b>     |  |
| <b>STEP ESEGUITI</b>      |  |
| <b>RISULTATI (output)</b> |  |

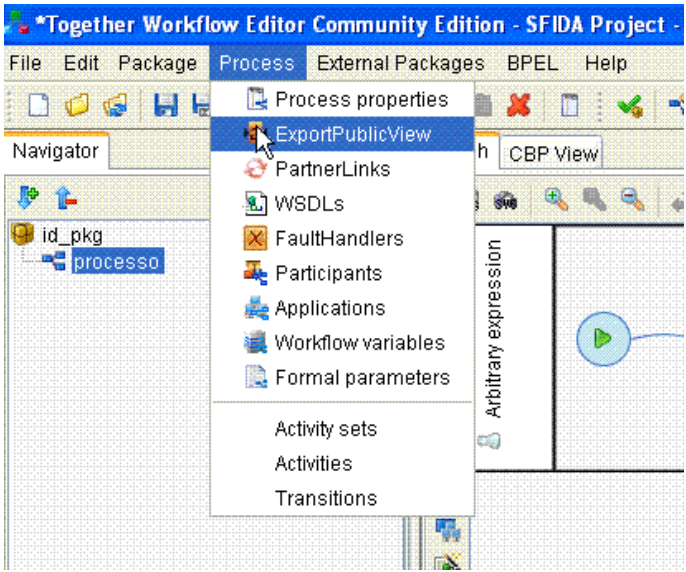
#### 4.2.3 Ricerca e Composizione dei Web-Services (b80 / ACP)

<< Dall'interno di JaWE, su caso CRF >>

|                       |  |
|-----------------------|--|
| <b>ID</b>             |  |
| <b>TITOLO</b>         | Ricerca e Composizione dei Web-Services  |
| <b>OBIETTIVO</b>      | Permettere al Supplier di ricercare il web-service adatto alle sue esigenze, e inserirlo nel processo eseguibile in Jawe |
| <b>SET-UP (input)</b> |  |
| <b>STEP ESEGUITI</b>  |  |
| <b>RISULTATI</b>      |  |

|          |  |
|----------|--|
| (output) |  |
|----------|--|

#### 4.2.4 Generazione Vista Pubblica per il Fornitore

|                           |   |
|---------------------------|---|
| <b>ID</b>                 |   |
| <b>TITOLO</b>             | Esportare le viste pubbliche  |
| <b>OBIETTIVO</b>          | Il processo .xpdL è esportato in un secondo file xpdL contenente solo le informazioni che si vogliono rendere pubbliche.  |
| <b>SET-UP (input)</b>     | <ul style="list-style-type: none"> <li>• Disponibilità dell'ambiente JaWE customizzato</li> <li>• Disponibilità del processo fornitore a livello eseguibile</li> </ul>  |
| <b>STEP ESEGUITI</b>      | <p>E' sufficiente aprire il file e salvare la vista pubblica selezionando il menu "Export Public View"</p>  <p>The screenshot shows the 'Process' menu open in the software. The 'ExportPublicView' option is highlighted. Other options in the menu include Process properties, PartnerLinks, WSDLs, FaultHandlers, Participants, Applications, Workflow variables, Formal parameters, Activity sets, Activities, and Transitions. The background shows a workflow diagram with a 'processo' element and an 'Arbitrary expression' box.</p> <p><b>Figura 6: esportazione vista pubblica del fornitore</b></p> |
| <b>RISULTATI (output)</b> | Viene generato un file XPD L, che è una black box rappresentante esclusivamente le variabili in input ed output, senza riportare i dettagli interni del processo.   |

#### 4.2.5 Modellazione Processo OEM a livello eseguibile

|                       |   |
|-----------------------|---|
| <b>ID</b>             |   |
| <b>TITOLO</b>         | Modellazione grafica lato OEM di un processo eseguibile   |
| <b>OBIETTIVO</b>      | I vari partecipanti dell'OEM, a seconda dei loro ruoli e compiti, potranno modellare la loro parte di processo, definendo esattamente input, output e risorse richieste per ogni attività.                    |
| <b>SET-UP (input)</b> | <ul style="list-style-type: none"> <li>• Disponibilità dell'ambiente JaWE customizzato</li> <li>• Disponibilità della vista pubblica del fornitore</li> </ul>   |
| <b>STEP ESEGUITI</b>  | <ul style="list-style-type: none"> <li>• Project Manager (PM) : crea da zero il file "oem.xpdL", in cui inserisce la prima attività, che riceve i requisiti funzionali come pura stringa di testo,</li> </ul> |



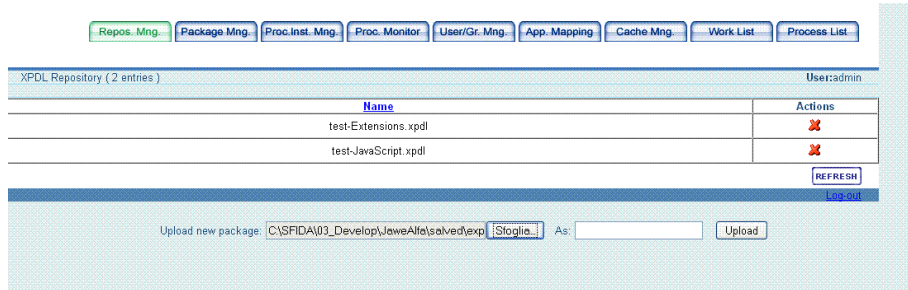


**SET-UP  
(input)**

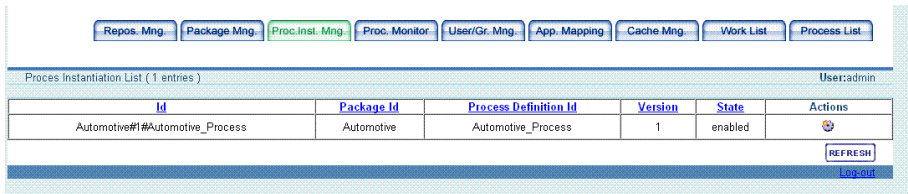
E' necessario aver modellato il processo a livello business (con Jawe) ed avere a disposizione, up and running, i web services necessari.

Il processo xpdL viene inviato al motore di esecuzione, tramite una form web, nei seguenti passi:

- upload fisico del file .xpdL dal menu "Repository Manager"

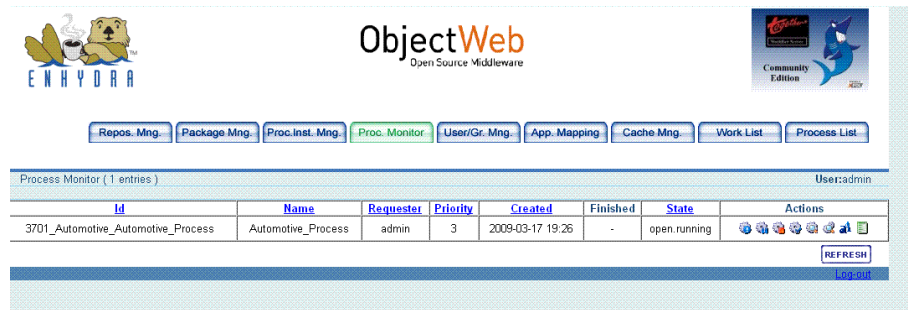


- load del package dal "Package Manager", in cui è verificata la correttezza lessicale dell'xpdL

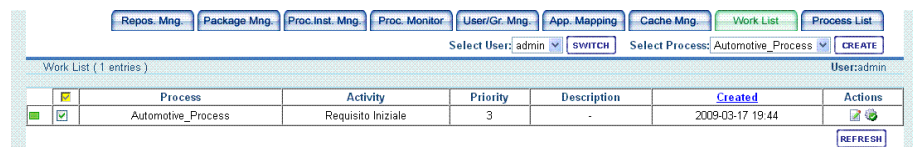


**STEP  
ESEGUITI**

Dal menu "Process Monitor" si inizia il processo e si visualizza e modifica il suo stato nel corso dell'esecuzione (avviato, in pausa, terminato, cancellato...)



Dal menu "sharkWorkList" si hanno le stesse possibilità riguardo le variabili , soprattutto si può inserire l'input. Nel caso del workflow relativo al fornitore, sarà egli stesso ad inserire i dati dopo aver eseguito separatamente sul suo sistema legacy il processo.



**Figura 8: esecuzione del processo con Shark**

**RISULTATI**

Come mostrato negli step precedenti, il processo è stato correttamente caricato

|                 |  |
|-----------------|--|
| <b>(output)</b> | ed eseguito nel motore di esecuzione Shark |
|-----------------|--|

## 4.4 Validazione Ambiente di Esecuzione – esecuzione su piattaforma Grid

Per le seguenti sezioni si ricorda (come descritto nel capitolo 3) che lo scenario di riferimento è quello del caso di studio OR5.3

### 4.4.1 Creazione di un Grid workflow per un processo automatizzato

|                       |   |
|-----------------------|---|
| <b>TITOLO</b>         | Creazione di un Grid workflow per un processo automatizzato.  |
| <b>OBIETTIVO</b>      | L'utente deve essere in grado di creare un workflow BPEL e deve avere a disposizione un'interfaccia grafica che gli permetta di modellare tutti gli elementi costituenti del linguaggio BPEL  |
| <b>SET-UP (input)</b> | Avere a disposizione: Eclipse, Oracle BPEL Designer.  |
| <b>STEP ESEGUITI</b>  | <p><b>FASE 1 (preliminare): Creazione dei Grid services da riusare e comporre per l'implementazione del workflow.</b></p> <p>In questa fase, sono stati realizzati vari moduli software, capaci di eseguire automaticamente i task previsti nel workflow del processo <i>SellGoods</i> schematizzato nella Figura 2. Questi moduli sono stati implementati mediante classi Java, ed esportati come web services.</p> <p>Di seguito è riportato un elenco dei task con una breve descrizione e una tabella contenente i port-types estratti dai WSDL dei servizi implementati.</p> <ol style="list-style-type: none"> <li>Registrazione dell'ordine ricevuto nella base di dati.</li> <li>Autenticazione del cliente e verifica dei dati ricevuti.</li> <li>Verifica del livello delle scorte di magazzino.</li> <li>Invio della richiesta ai fornitori, solo se tali scorte sono inferiori alla quantità ordinata dal cliente.</li> <li>Registrazione del cliente, solo se questi sta accedendo al sistema per la prima volta.</li> <li>Valutazione dell'affidabilità del cliente.</li> <li>Pianificazione, validazione e registrazione dell'ordine.</li> <li>Registrazione del rifiutato dell'ordine, se il cliente appare nella lista dei clienti non affidabili o se non è possibile soddisfarlo né con la merce presente in magazzino né con l'intervento dei fornitori.</li> <li>Accettazione e registrazione dell'ordine, solo se il cliente è affidabile e la merce è disponibile.</li> <li>Procedura di spedizione rapida, non è mai effettuata se la merce richiesta non è presente in magazzino.</li> <li>Calcolo di uno sconto fedeltà, non viene mai effettuato per clienti nuovi.</li> <li>Preparazione della documentazione per la spedizione.</li> </ol> <p>Si noti che il servizio "D" implica l'interazione con un servizio esterno all'azienda, esposto da un altro partner della filiera tessile, che permette di avere informazioni sulla fornitura di merce.</p> |

### Service "A": receivingOrder

```
- <wsdl:portType name="A">
  - <wsdl:operation name="receivingOrder" parameterOrder="clientName clientSurname clientAddress productName productAmount">
    <wsdl:input message="impl:receivingOrderRequest" name="receivingOrderRequest"/>
    <wsdl:output message="impl:receivingOrderResponse" name="receivingOrderResponse"/>
  </wsdl:operation>
</wsdl:portType>
```

### Service "B": authenticatingClient

```
- <wsdl:portType name="B">
  - <wsdl:operation name="authenticatingClient" parameterOrder="clientName clientSurname">
    <wsdl:input message="impl:authenticatingClientRequest" name="authenticatingClientRequest"/>
    <wsdl:output message="impl:authenticatingClientResponse" name="authenticatingClientResponse"/>
  </wsdl:operation>
</wsdl:portType>
```

### Service "C": checkStock

```
- <wsdl:portType name="C">
  - <wsdl:operation name="checkStock" parameterOrder="productName productAmount">
    <wsdl:input message="impl:checkStockRequest" name="checkStockRequest"/>
    <wsdl:output message="impl:checkStockResponse" name="checkStockResponse"/>
  </wsdl:operation>
</wsdl:portType>
```

### Service "D": askSupplies

```
- <wsdl:portType name="D">
  - <wsdl:operation name="askSupplies" parameterOrder="productName productAmount">
    <wsdl:input message="impl:askSuppliesRequest" name="askSuppliesRequest"/>
    <wsdl:output message="impl:askSuppliesResponse" name="askSuppliesResponse"/>
  </wsdl:operation>
</wsdl:portType>
```

### Service "E": RegisterClient

```
- <wsdl:portType name="E">
  - <wsdl:operation name="registerClient" parameterOrder="clientName clientSurname clientAddress">
    <wsdl:input message="impl:registerClientRequest" name="registerClientRequest"/>
    <wsdl:output message="impl:registerClientResponse" name="registerClientResponse"/>
  </wsdl:operation>
</wsdl:portType>
```

### Service "F": clientReliability

```
- <wsdl:portType name="F">
  - <wsdl:operation name="clientReliability" parameterOrder="clientName clientSurname orderId">
    <wsdl:input message="impl:clientReliabilityRequest" name="clientReliabilityRequest"/>
    <wsdl:output message="impl:clientReliabilityResponse" name="clientReliabilityResponse"/>
  </wsdl:operation>
</wsdl:portType>
```

### Service "G": validateOrder

```

- <wsdl:portType name="G">
  - <wsdl:operation name="validateOrder" parameterOrder="internalStock externalStock orderId">
    <wsdl:input message="impl.validateOrderRequest" name="validateOrderRequest"/>
    <wsdl:output message="impl.validateOrderResponse" name="validateOrderResponse"/>
  </wsdl:operation>
</wsdl:portType>

```

Service "H": rejectingOrder

```

- <wsdl:portType name="H">
  - <wsdl:operation name="rejectingOrder" parameterOrder="orderId">
    <wsdl:input message="impl.rejectingOrderRequest" name="rejectingOrderRequest"/>
    <wsdl:output message="impl.rejectingOrderResponse" name="rejectingOrderResponse"/>
  </wsdl:operation>
</wsdl:portType>

```

Service "I": acceptOrder

```

- <wsdl:portType name="I">
  - <wsdl:operation name="acceptOrder" parameterOrder="orderId">
    <wsdl:input message="impl.acceptOrderRequest" name="acceptOrderRequest"/>
    <wsdl:output message="impl.acceptOrderResponse" name="acceptOrderResponse"/>
  </wsdl:operation>
</wsdl:portType>

```

Service "L": fastDispatch

```

- <wsdl:portType name="L">
  - <wsdl:operation name="fastDispatch" parameterOrder="orderId">
    <wsdl:input message="impl.fastDispatchRequest" name="fastDispatchRequest"/>
    <wsdl:output message="impl.fastDispatchResponse" name="fastDispatchResponse"/>
  </wsdl:operation>
</wsdl:portType>

```

Service "M": fidelityDiscount

```

- <wsdl:portType name="M">
  - <wsdl:operation name="fidelityDiscount" parameterOrder="clientName clientSurname">
    <wsdl:input message="impl.fidelityDiscountRequest" name="fidelityDiscountRequest"/>
    <wsdl:output message="impl.fidelityDiscountResponse" name="fidelityDiscountResponse"/>
  </wsdl:operation>
</wsdl:portType>

```

Service "N": prepareBill

```

- <wsdl:portType name="N">
  - <wsdl:operation name="prepareBill" parameterOrder="orderId sconto">
    <wsdl:input message="impl.prepareBillRequest" name="prepareBillRequest"/>
    <wsdl:output message="impl.prepareBillResponse" name="prepareBillResponse"/>
  </wsdl:operation>
</wsdl:portType>

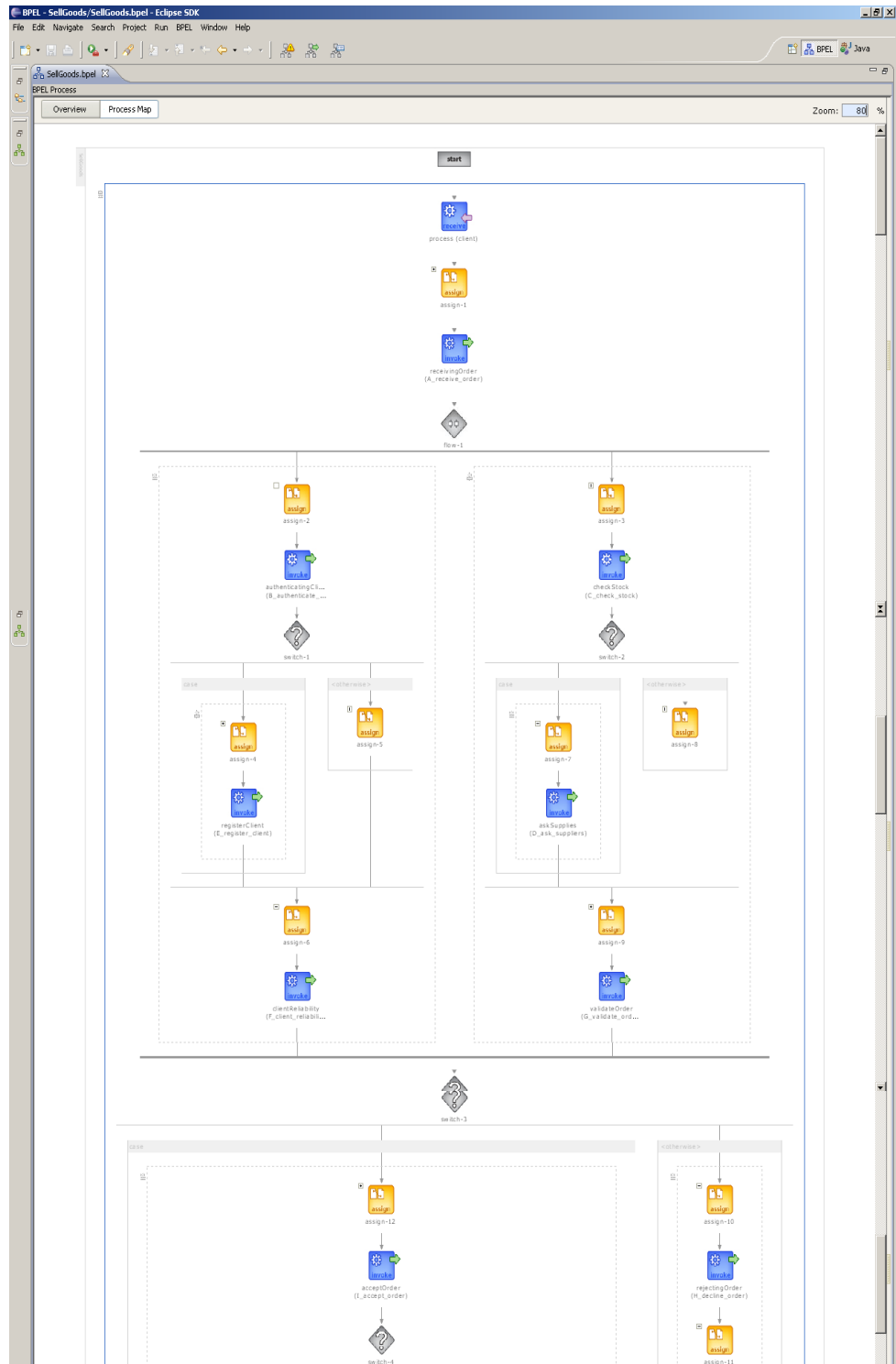
```

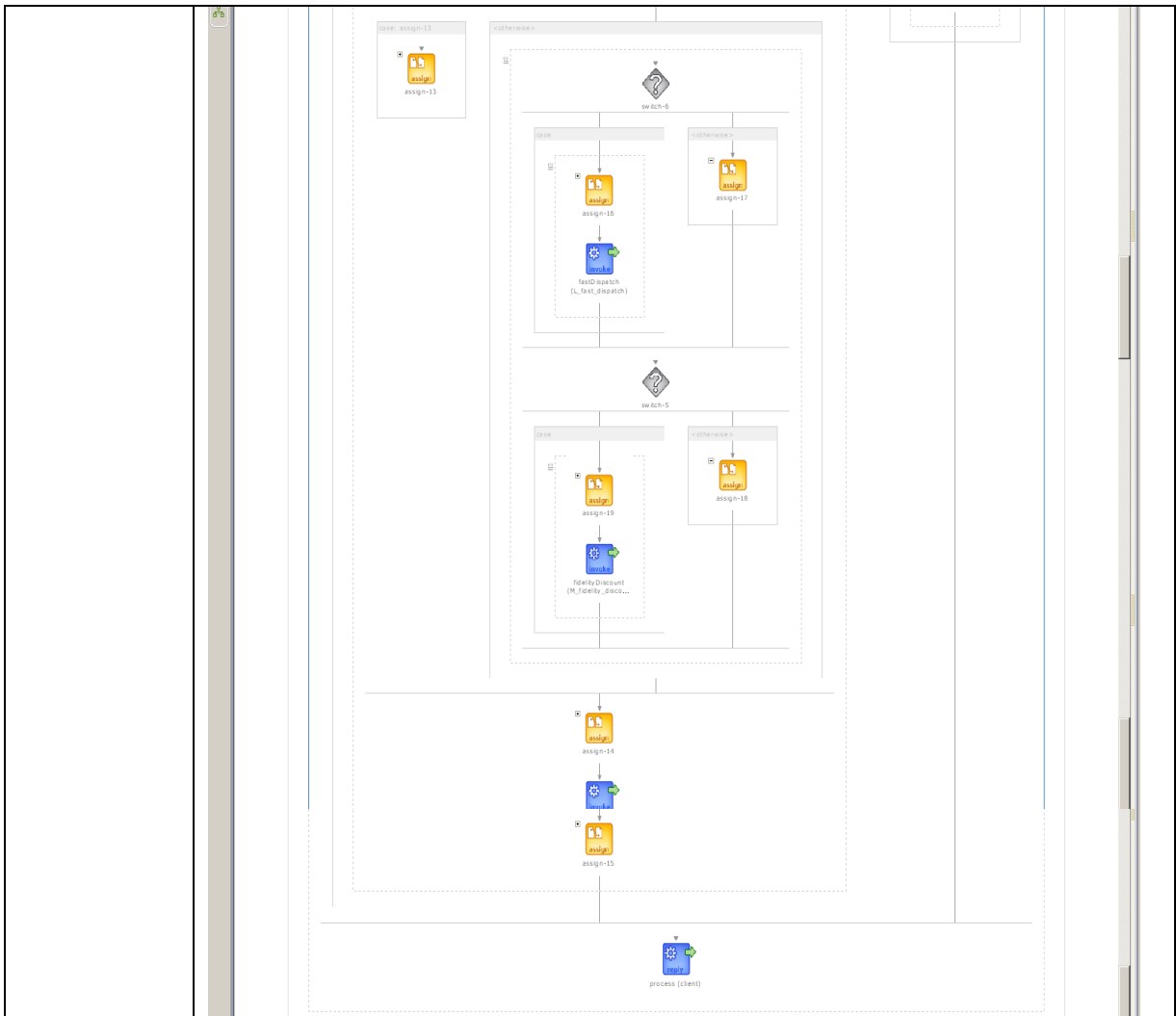
## ***FASE 2. Definizione di uno grid workflow BPEL4WS***

Aprò Oracle BPEL Designer.

Creo uno schema BPEL4WS (denominato "bpel\_SellGoods\_1.0.jar"), che specifica come riusare i grid services descritti precedentemente per eseguire i singoli task del

workflow di Figura 2. Una rappresentazione grafica di tale schema è mostrata nella seguente figura.





**Figura 9: il Grid-workflow “sell goods”**

Questo grid workflow può essere caricato sulla Sunflower Console, in modo da poterlo usare per l'esecuzione del processo.

|                           |  |
|---------------------------|--|
| <b>RISULTATI (output)</b> | Un nuovo Grid workflow BPEL4WS è pronto per essere eseguito con il sistema Sunflower (vedi test successivo). |
|---------------------------|--|

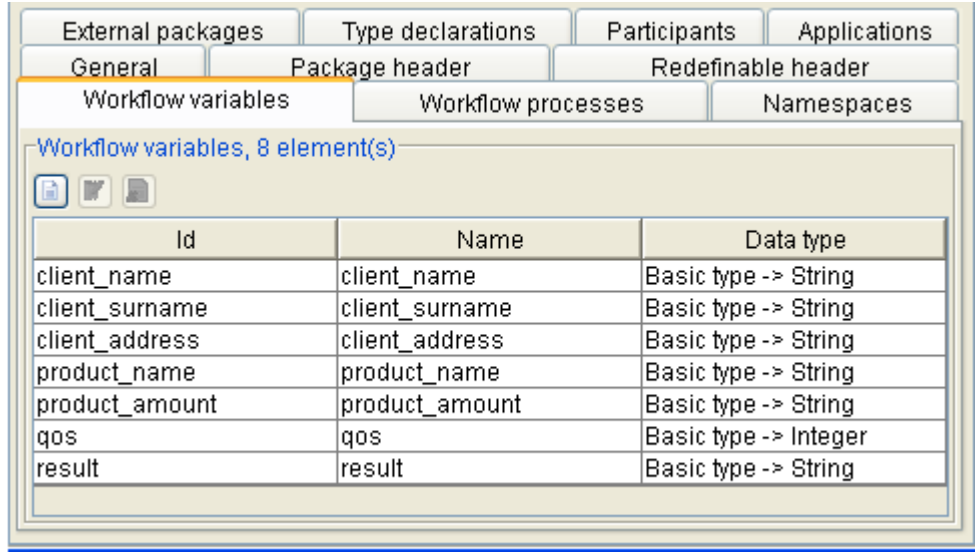
### 4.4.2 Inclusion del processo Grid-Workflow nell'ambiente di modellazione

|                  |  |
|------------------|--|
| <b>ID</b>        |  |
| <b>TITOLO</b>    | Inclusione processo Grid-Workflow in ambiente modellazione   |
| <b>OBIETTIVO</b> | Incorporare il processo Grid-Workflow creato in Sunflower nell'ambiente di modellazione processi a livello eseguibile (Jawe) |
| <b>SET-UP</b>    | <ul style="list-style-type: none"> <li>• Disponibilità di un processo Grid-Workflow in Sunflower</li> </ul>                  |

(input)

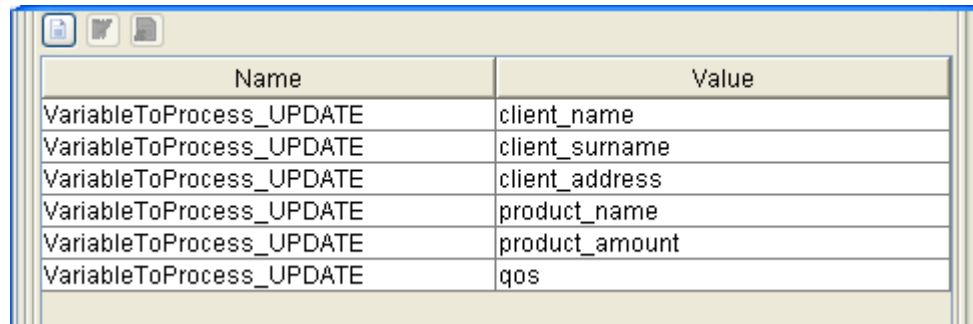
STEP  
ESEGUITI

- 1) Creazione di sei variabili in input, corrispondenti a quelle richieste dal processo GRID, che includono anche la quality of service, più una settima per memorizzare il risultato



| Id             | Name           | Data type             |
|----------------|----------------|-----------------------|
| client_name    | client_name    | Basic type -> String  |
| client_surname | client_surname | Basic type -> String  |
| client_address | client_address | Basic type -> String  |
| product_name   | product_name   | Basic type -> String  |
| product_amount | product_amount | Basic type -> String  |
| qos            | qos            | Basic type -> Integer |
| result         | result         | Basic type -> String  |

- 2) Creazione di un'attività per impostare i valori di tali variabili in fase di esecuzione

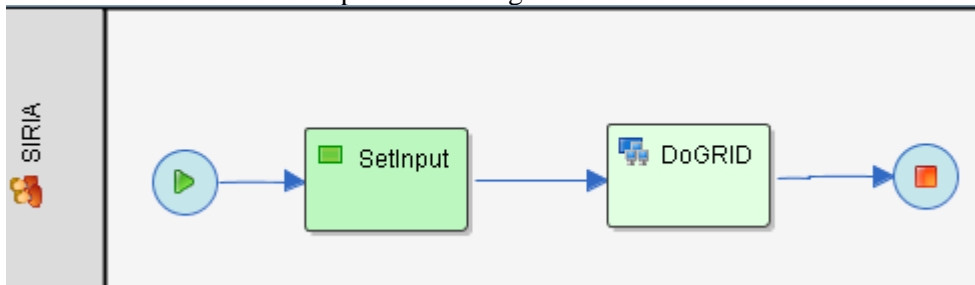


| Name                     | Value          |
|--------------------------|----------------|
| VariableToProcess_UPDATE | client_name    |
| VariableToProcess_UPDATE | client_surname |
| VariableToProcess_UPDATE | client_address |
| VariableToProcess_UPDATE | product_name   |
| VariableToProcess_UPDATE | product_amount |
| VariableToProcess_UPDATE | qos            |

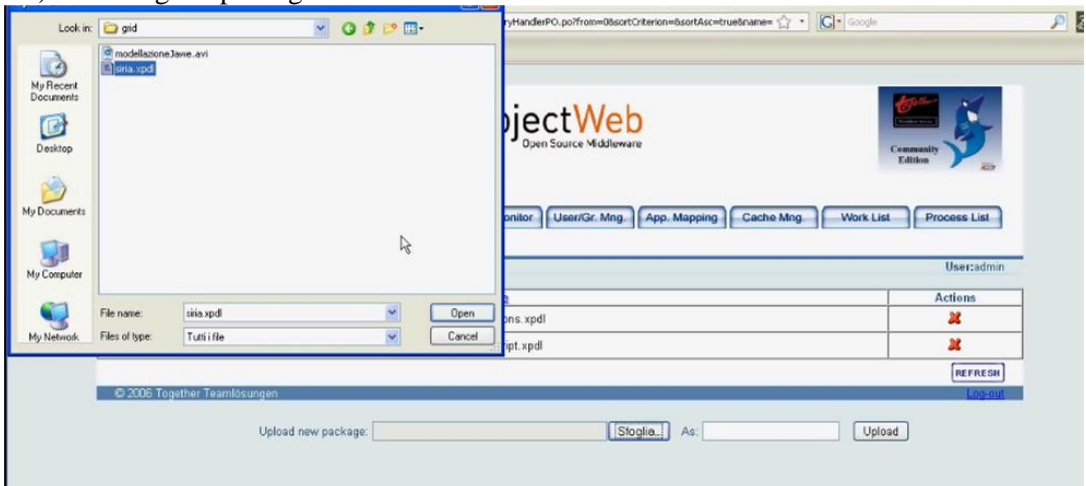
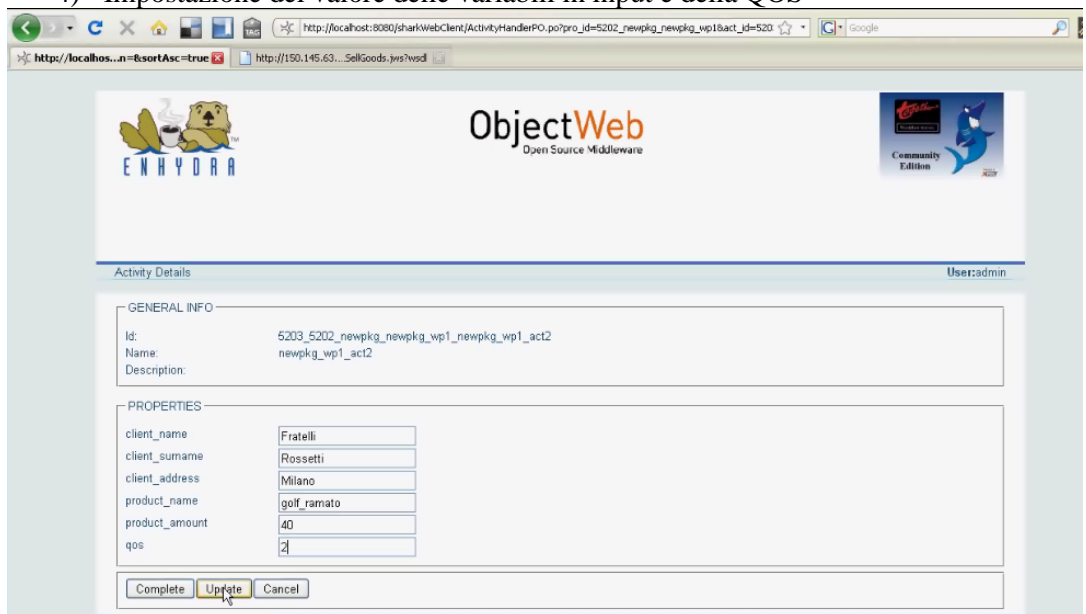
- 3) Creazione di un'attività GRID per chiamare il processo con le variabili predefinite e ricevere il risultato

RISULTATI  
(output)


Si ottiene un file XPDL che può essere eseguito dal motore shark



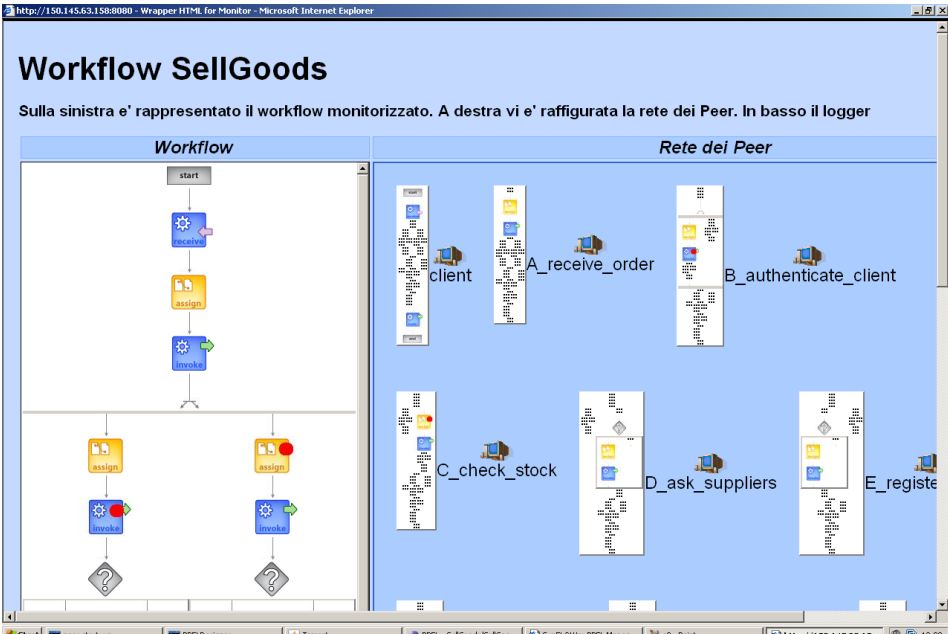
### 4.4.3 Esecuzione di un Grid-workflow tramite Shark

|                       |   |
|-----------------------|---|
| <b>TITOLO</b>         | Esecuzione di un Grid workflow (Sunflower) tramite Shark  |
| <b>OBIETTIVO</b>      | Poter lanciare l'esecuzione di un Grid-workflow per Sunflower dal motore di esecuzione Shark  |
| <b>SET-UP (input)</b> | <ul style="list-style-type: none"> <li>• Un Grid workflow, già registrato sul sistema Sunflower</li> <li>• Un processo eseguibile per Shark</li> </ul>  |
| <b>STEP ESEGUITI</b>  | <ol style="list-style-type: none"> <li>1) Caricamento del processo tramite l'interfaccia web di Shark (vedasi caso 4.3.1)</li> <li>2) Loading del package</li> </ol>  <p style="text-align: center;"><b>Figura 10: caricamento del processo Grid-workflow in Shark</b></p> <ol style="list-style-type: none"> <li>3) Avvio del processo in Shark</li> <li>4) Impostazione del valore delle variabili in input e della QoS</li> </ol>  <p style="text-align: center;"><b>Figura 11: impostazione dei parametri di input e di QoS del processo</b></p> <ol style="list-style-type: none"> <li>5) Esecuzione del processo in Sunflower</li> </ol> |



|                           |   |
|---------------------------|---|
| <b>RISULTATI (output)</b> | <p>Il risultato del processo Grid su Sunflower è memorizzato nella variabile definita al punto 1 e contiene l'indirizzo web di un file testuale con il risultato del processo.</p>  <p style="text-align: center;"><b>Figura 12: risultato del processo Grid su Sunflower</b></p> |
|---------------------------|---|

#### 4.4.4 Tracciatura dell'esecuzione di un Grid workflow

|                       |   |
|-----------------------|---|
| <b>TITOLO</b>         | Possibilità di monitoraggio dell'esecuzione decentralizzata e visualizzazione del risultato   |
| <b>OBIETTIVO</b>      | Monitorare gli eventi relativi all'esecuzione dei task di un grid workflow, e memorizzarli in un opportuno repository di log.   |
| <b>SET-UP (input)</b> | Nessuno.  |
| <b>STEP ESEGUITI</b>  | <p>Nella finestra di pop-up, che appare all'avvio dell'esecuzione di un workflow, si può seguire l'esecuzione sui vari BPEL Engine dei frammenti di workflow. La seguente figura evidenzia come Sunflower consenta all'utente di seguire lo stato di esecuzione del workflow <i>SellGoods</i>, in relazione alla rete di peer impiegati per svolgere i vari task.</p>  <p style="text-align: center;"><b>Figura 13: Sunflower – monitoraggio esecuzione</b></p> |

Durante l'esecuzione del processo, il modulo di tracciatura integrato nello sistema registra, all'interno del database MySQL "workflowlog" (che implementa il modulo *Collaborative Log DB*), una serie di informazioni sull'esecuzione dei singoli task del processo.

Al termine dell'elaborazione, in base al tipo restituito, il risultato è automaticamente visualizzato nella finestra di pop-up oppure è eseguito un redirect verso la posizione che contiene i dati.

**RISULTATI  
(output)**

Al termine dell'elaborazione il risultato prodotto dal processo è restituito all'utente, mentre gli eventi di log sono stati registrati nell'apposito repository.

Ad esempio, nella figura seguente sono mostrati alcuni eventi (tabella *event*) registrati per una singola istanza di esecuzione (identificata dal codice 144) del workflow *SellGoods*. Per ogni evento, la tabella memorizza gli identificatori del processo, dell'istanza di esecuzione e del task associati all'evento, il tempo di occorrenza (*timestamp*), lo stato di esecuzione del task (*runningState*) ed il tipo di evento. Questo ultimo campo permette di distinguere fra invocazioni di grid services (*invoke*) e altri tipi di eventi legati al coordinamento del flusso di esecuzione delle attività. Nel primo caso, la tabella permette anche di registrare il grid service ed il metodo (operation) effettivamente invocate. In ogni caso, per ogni evento viene memorizzato anche l'identificativo del peer coinvolto (*originatorID*).

The screenshot shows a MySQL Query Browser window with the following table data:

| id   | runningState | processid | processInstancelid | taskid | servicelid | operation                | originatorid | timestamp           | eventType  |
|------|--------------|-----------|--------------------|--------|------------|--------------------------|--------------|---------------------|------------|
| 1021 | complete     | 17        | 144                | 208    | 0          | main                     | 13           | 2009-02-11 17:06:07 | structured |
| 1022 | complete     | 17        | 144                | 224    | 0          | receiveInput-1-          | 13           | 2009-02-11 17:06:08 | synchro    |
| 1023 | complete     | 17        | 144                | 161    | 0          | assign-1-1-              | 13           | 2009-02-11 17:06:09 | assign     |
| 1024 | complete     | 17        | 144                | 180    | 5          | A_receive_order-1-       | 13           | 2009-02-11 17:06:16 | invoke     |
| 1025 | complete     | 17        | 144                | 197    | 0          | flow-1-1-                | 13           | 2009-02-11 17:06:16 | structured |
| 1026 | complete     | 17        | 144                | 197    | 0          | flow-1-1-                | 13           | 2009-02-11 17:06:18 | structured |
| 1027 | complete     | 17        | 144                | 197    | 0          | flow-1-1-                | 13           | 2009-02-11 17:06:20 | structured |
| 1028 | complete     | 17        | 144                | 201    | 0          | flow-sequence-2-1-       | 13           | 2009-02-11 17:06:24 | structured |
| 1029 | complete     | 17        | 144                | 199    | 0          | flow-sequence-1-1-       | 13           | 2009-02-11 17:06:24 | structured |
| 1030 | complete     | 17        | 144                | 173    | 0          | assign-3-1-              | 13           | 2009-02-11 17:06:25 | assign     |
| 1031 | complete     | 17        | 144                | 172    | 0          | assign-2-1-              | 13           | 2009-02-11 17:06:25 | assign     |
| 1032 | complete     | 17        | 144                | 194    | 7          | C_check_stock-1-         | 14           | 2009-02-11 17:06:30 | invoke     |
| 1033 | complete     | 17        | 144                | 181    | 6          | B_authenticate_client-1- | 15           | 2009-02-11 17:06:32 | invoke     |
| 1034 | complete     | 17        | 144                | 242    | 0          | switch-2-1-              | 13           | 2009-02-11 17:06:33 | structured |
| 1035 | complete     | 17        | 144                | 242    | 0          | switch-2-1-              | 13           | 2009-02-11 17:06:34 | structured |
| 1036 | complete     | 17        | 144                | 240    | 0          | switch-1-1-              | 13           | 2009-02-11 17:06:35 | structured |
| 1037 | complete     | 17        | 144                | 182    | 0          | case-1-                  | 13           | 2009-02-11 17:06:37 | structured |
| 1038 | complete     | 17        | 144                | 240    | 0          | switch-1-1-              | 13           | 2009-02-11 17:06:38 | structured |
| 1039 | complete     | 17        | 144                | 214    | 0          | otherwise-2-             | 13           | 2009-02-11 17:06:38 | structured |
| 1040 | complete     | 17        | 144                | 178    | 0          | assign-8-1-              | 13           | 2009-02-11 17:06:42 | assign     |
| 1041 | complete     | 17        | 144                | 226    | 0          | sequence-1-              | 13           | 2009-02-11 17:06:42 | structured |
| 1042 | complete     | 17        | 144                | 215    | 0          | otherwise-2-             | 13           | 2009-02-11 17:06:43 | structured |
| 1043 | complete     | 17        | 144                | 174    | 0          | assign-4-1-              | 13           | 2009-02-11 17:06:46 | assign     |
| 1044 | complete     | 17        | 144                | 243    | 0          | switch-2-1-              | 13           | 2009-02-11 17:06:47 | structured |
| 1045 | complete     | 17        | 144                | 179    | 0          | assign-9-1-              | 13           | 2009-02-11 17:06:48 | assign     |
| 1046 | complete     | 17        | 144                | 196    | 9          | E_register_client-1-     | 15           | 2009-02-11 17:06:52 | invoke     |
| 1047 | complete     | 17        | 144                | 204    | 11         | G_validate_order_plan-1- | 13           | 2009-02-11 17:06:55 | invoke     |
| 1048 | complete     | 17        | 144                | 227    | 0          | sequence-1-              | 13           | 2009-02-11 17:06:56 | structured |
| 1049 | complete     | 17        | 144                | 202    | 0          | flow-sequence-2-1-       | 13           | 2009-02-11 17:06:58 | structured |
| 1050 | complete     | 17        | 144                | 183    | 0          | case-1-                  | 13           | 2009-02-11 17:06:58 | structured |
| 1051 | complete     | 17        | 144                | 241    | 0          | switch-1-1-              | 13           | 2009-02-11 17:07:03 | structured |
| 1052 | complete     | 17        | 144                | 176    | 0          | assign-5-1-              | 13           | 2009-02-11 17:07:04 | assign     |
| 1053 | complete     | 17        | 144                | 203    | 10         | F_client_reliability-1-  | 15           | 2009-02-11 17:07:07 | invoke     |
| 1054 | complete     | 17        | 144                | 200    | 0          | flow-sequence-1-1-       | 13           | 2009-02-11 17:07:08 | structured |
| 1055 | complete     | 17        | 144                | 198    | 0          | flow-1-1-                | 13           | 2009-02-11 17:07:11 | structured |
| 1056 | complete     | 17        | 144                | 244    | 0          | switch-3-1-              | 13           | 2009-02-11 17:07:13 | structured |
| 1057 | complete     | 17        | 144                | 244    | 0          | switch-3-1-              | 13           | 2009-02-11 17:07:14 | structured |

|  |
|--|
| Figura 14: logs generati dall'esecuzione |
|--|

## 4.5 Validazione Ambiente di Process Intelligence

### 4.5.1 Importazione di log di esecuzione dei processi collaborativi

|                       |  |
|-----------------------|--|
| <b>TITOLO</b>         | Importazione di log di esecuzione dei processi collaborativi   |
| <b>OBIETTIVO</b>      | I dati registrati nel repository dei log di processi collaborativi possono essere importati nell'ambiente di Process Intelligence, al fine di analizzarli con i vari strumenti di analisi e di mining in esso presenti.<br>Ciò implica la conversione dei dati nel formato MXML usato da tali strumenti.   |
| <b>SET-UP (input)</b> | Un insieme di log di eventi, registrati dall'ambiente di esecuzione.   |
| <b>STEP ESEGUITI</b>  | Apro il tool <i>ProMImport4Sfida</i> (plugin del sistema open-source <i>ProMimport</i> ).<br>Nella schermata iniziale di <i>ProM-Import4SFIDA</i> è possibile visualizzare ed eventualmente modificare i parametri di connessione al database dei log, ed il criterio utilizzato per tradurre il contenuto del database in una serie di tracce di esecuzione. Il campo <i>DataAttribute</i> è stato impostato true per inserire nel log MXML i valori associati all'esecuzione dei singoli task e dell'intero workflow. Effettuiamo la stessa scelta anche per il campo <i>FilterInstanceBy</i> , allo scopo di selezionare solo un sottoinsieme dei dati presenti nel database. |

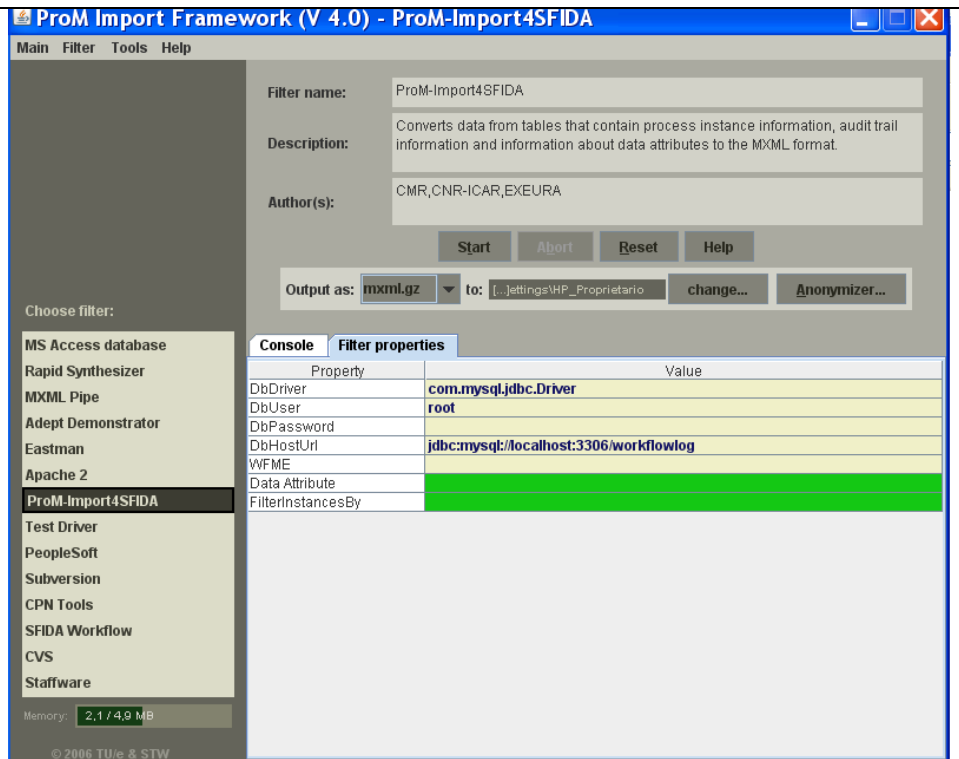
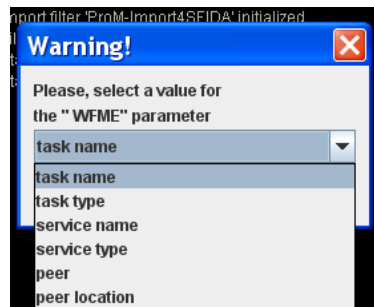
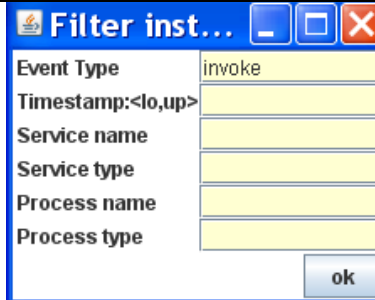


Figura 15: ProM-Import4SFIDA

Specifico il criterio per l'etichettatura delle attività del processo, tramite la finestra seguente. Ad esempio, decidiamo di considerare il nome logico dei task, esplicitamente riportati nel modello BPEL4WS.



Seleziono, eventualmente, un sottoinsieme degli eventi registrati, tramite la seguente finestra. Ad esempio, seleziono solo gli eventi che si riferiscono all'invocazione di grid services. In tal modo, nelle analisi dei log saranno trascurati tutti gli eventi relativi ad operazioni di ASSIGN, o costrutti BPEL per il controllo del flusso fra le attività elementari.

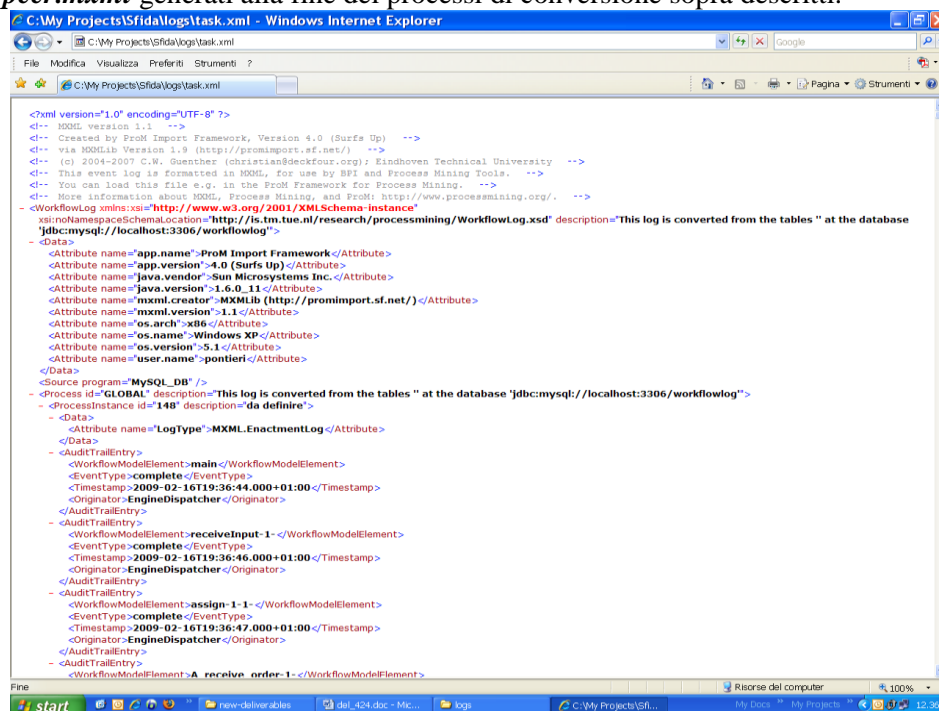


Il tool genera automaticamente un file MXML, che decidiamo di chiamare *task.mxml*, che contiene tutti i dati selezionati nei passi precedenti, riorganizzati come tracce di un processo.

Con un procedimento simile, a partire dagli stessi dati, possono essere generati altri file MXML. Ad esempio, è possibile creare un log *peer.mxml* in cui le tracce riportano come etichette di attività gli indirizzi dei peer associati ad ogni evento occorso durante l'esecuzione di una singola istanza del processo. A tale scopo basta sufficiente impostare WFME="peer".

Le figure seguenti mostrano una porzione dei file *task.mxml* e, rispettivamente, *peer.mxml* generati alla fine dei processi di conversione sopra descritti.

**RISULTATI**  
(output)



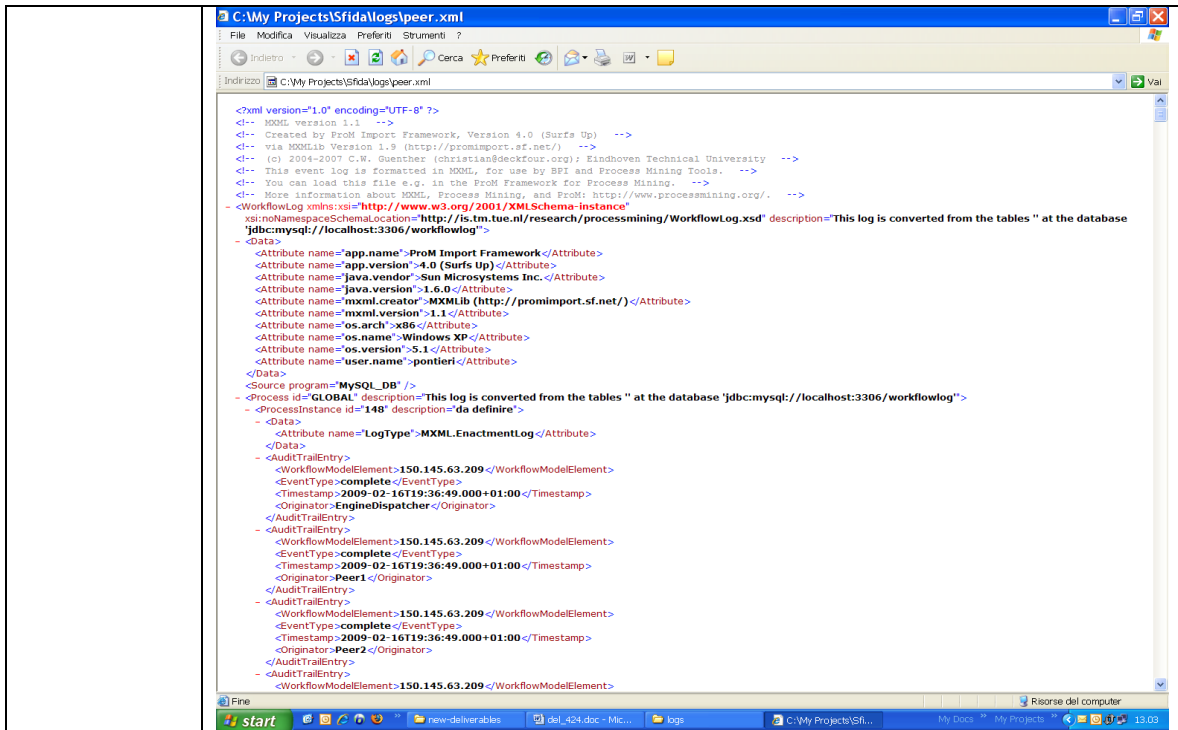
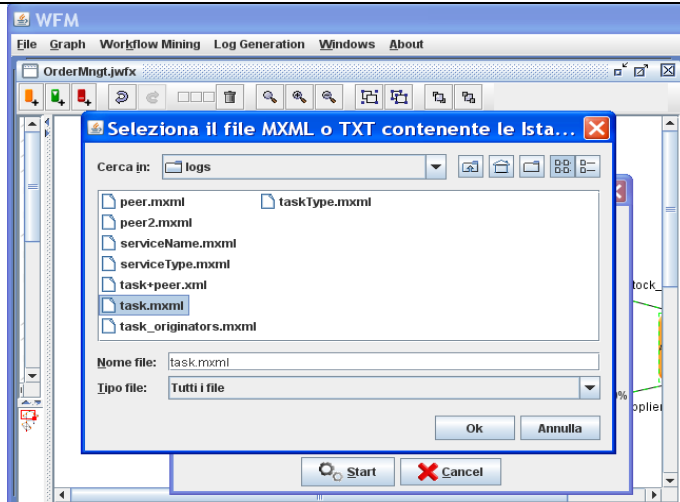


Figura 16: logs in formato .mxml

Come descritto nella sottosezione successiva, entrambi i file possono essere forniti in input agli strumenti di Process Intelligence per analizzare in che modo il processo *SellGoods* sia stato effettivamente eseguito.

#### 4.5.2 Analisi con gli strumenti di Process Intelligence

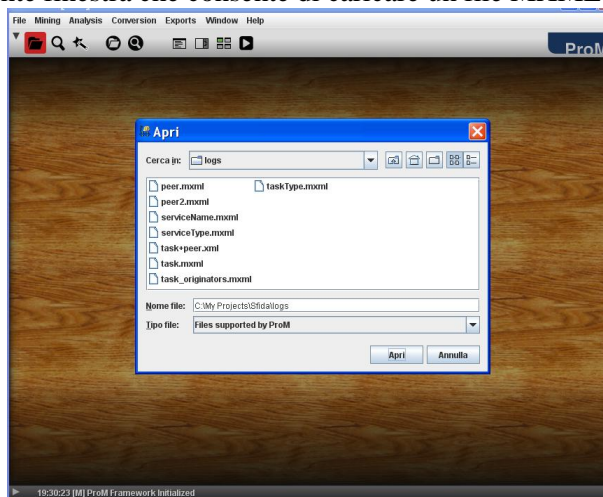
|                       |  |
|-----------------------|--|
| <b>TITOLO</b>         | Analisi con gli strumenti di Process Intelligence  |
| <b>OBIETTIVO</b>      | Gli eventi registrati in un log MXML possono essere analizzati con i vari strumenti di analisi e di mining disponibili nell'ambiente di Process Intelligence.  |
| <b>SET-UP (input)</b> | Un log MXML.   |
| <b>STEP ESEGUITI</b>  | <p><b>Fase 1.a Caricamento di un file MXML nella componente WFM</b></p> <p>Apro la componente di workflow mining WFM.<br/> Carico o disegno un grafo workflow per il processo da analizzare.<br/> Apro il menù <i>Workflow Mining</i> → <i>Execution Patterns</i> → <i>Frequent Patterns Algorithms</i> e premo il tasto <i>Load</i>.<br/> Appare la seguente finestra che consente di caricare un file MXML, convertendolo automaticamente nel formato interno utilizzato dagli algoritmi di workflow mining. Ciò consente di applicare tali algoritmi per scoprire pattern di esecuzione del processo che si riscontrano frequentemente nel log.</p> |



**Fase 1.b Caricamento di un file MXML nella componente PM.**


Analogamente, per applicare gli strumenti di process mining e process analysis, carico il file nella componente PM (ProM esteso), come descritto di seguito.

Apro il tool PM. Seleziono la funzione *File* → *Open Supported File*.  
 Appare la seguente finestra che consente di caricare un file MXML.



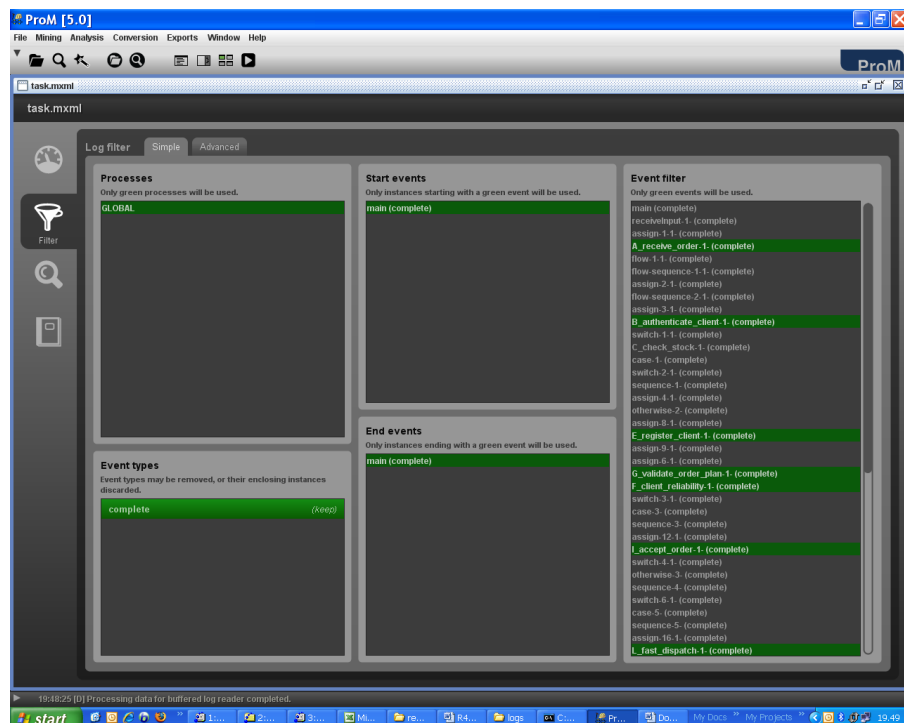
Una volta caricato il file, il tool mostra alcune informazioni riassuntive sul suo contenuto (elenco delle attività, elenco degli esecutori, numero di istanze di esecuzione,..)



Dal pannello dei filtri (simbolo ) è possibile impostare una varietà funzioni di pre-processing, che consentono di ristrutturare il contenuto del file di log in vari modi, prima di applicare gli strumenti di mining e di analisi.

### ***Fase 2. Pre-processing dei dati di log (nella componente PM).***

La seguente figura mostra come selezionando direttamente i nomi delle attività presenti nel pannello Event filter, sia possibile filtrare gli eventi che si riferiscono alle attività logiche del processo descritte nella seguente figura, trascurando tutte le altre attività presenti nello schema BPEL4WS (corrispondenti a costrutti di controllo del flusso ed a operazioni ASSIGN)



**Figura 17: ProM - EventFilter**

Precisiamo comunque che il framework ProM (e quindi la stessa componente PM



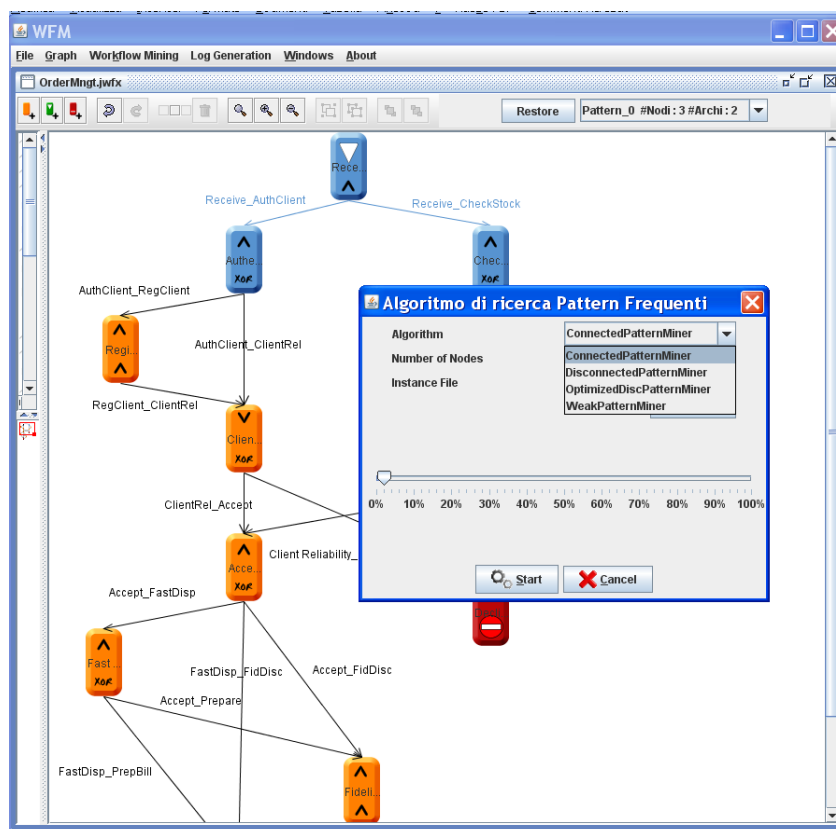
che ne è un'estensione) offre una vasta gamma di filtri che permettono di effettuare manipolazioni anche piuttosto complesse.

### **Fase 3.a: Analisi dei dati di log con gli strumenti di workflow mining**

La componente WFM implementa una serie di algoritmi di workflow mining che permettono di scoprire i sottografi del workflow corrente, che siano sufficientemente frequenti in un dato file di log. A tale scopo, basta eseguire la seguente procedura.

Apri il menù *Workflow Mining* → *Execution Patterns* → *Frequent Patterns Algorithms* usato per caricare il file di log (con il tasto *Load*).

Compare la seguente finestra.



**Figura 18: componente WFM**

Seleziono uno degli algoritmi di workflow mining disponibili.

Imposto la soglia minima di supporto.

Premo il tasto *Start*.

La figura seguente mostra la finestra di output che raccoglie i vari pattern estratti da un log di esecuzione del processo, oltre allo schema workflow usato come riferimento per l'analisi. In particolare, nella figura è evidenziato in blu uno dei pattern, selezionato interattivamente dalla lista dei pattern scoperti.

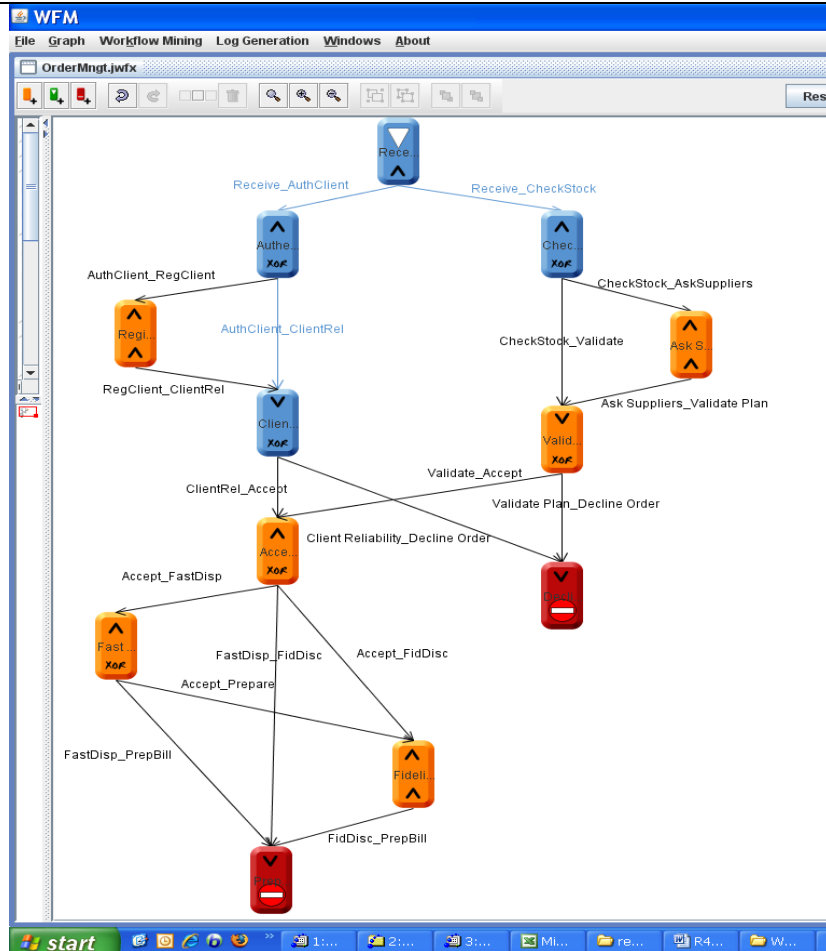


Figura 19: Esecuzione su WFM

***Fase 3.b: Analisi dei dati di log con gli strumenti di process mining***

La componente PM offre una vasta gamma serie di strumenti di Process Mining (menù *Mining*) e Process Analysis (menù *Analysis*).

La figura seguente mostra, ad esempio come sia possibile invocare il plugin DWS per indurre una gerarchia di schemi workflow che descrive gli eventi registrati nel file *task.xml*

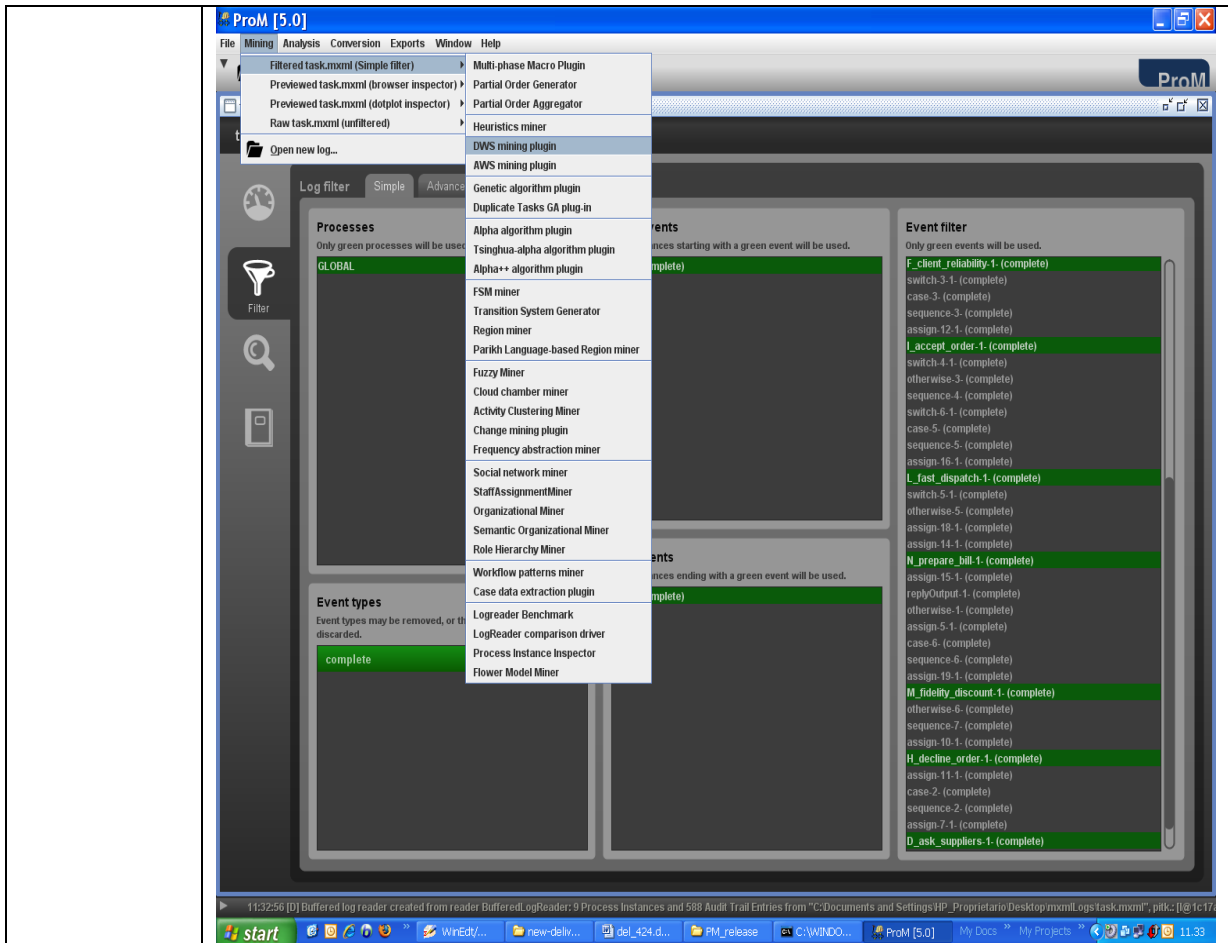


Figura 20: ProM - componente PM

La figura successiva illustra l'output del plugin: la gerarchia di schemi workflow ottenuta, nella parte sinistra della finestra, ed una rappresentazione grafica della radice, sulla parte destra.

Nella parte inferiore sono mostrate infine le regole discriminanti, scoperte nello stesso log, che hanno consentito di separarne le tracce in quattro distinti gruppi.

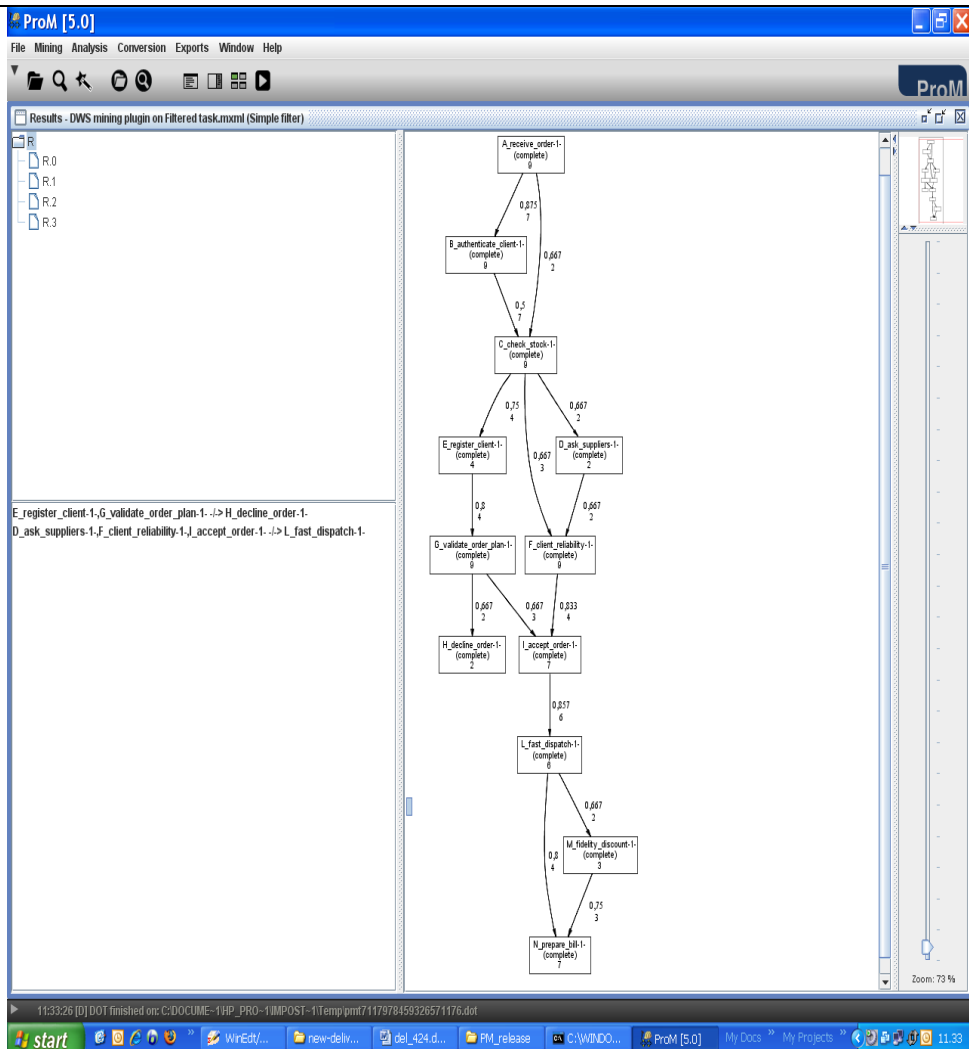


Figura 21: ProM - schema di workflow scoperto

Selezionando un qualunque nodo della gerarchia, è possibile visualizzarne lo schema workflow. La seguente figura mostra, ad esempio, lo schema del nodo R0

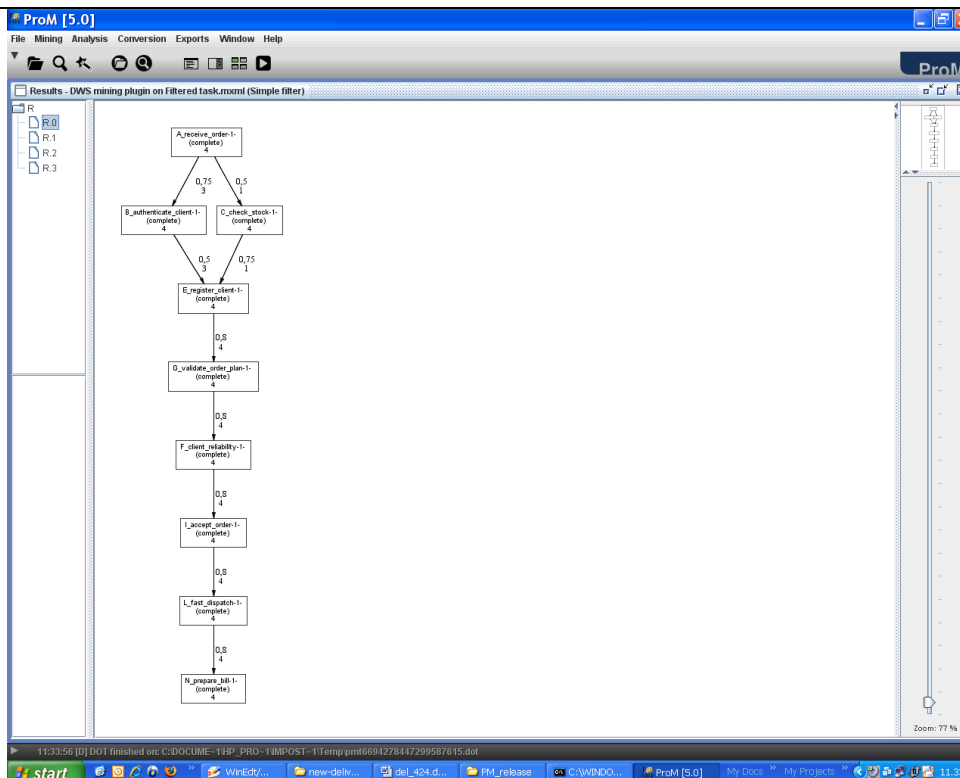


Figura 22: schema nodo R0

**RISULTATI (output)**

Un'importante categoria di risultati prodotti dall'ambiente di Process Intelligence è costituito invece dai pattern di esecuzione frequenti che possono essere scoperti con gli strumenti di workflow mining presenti nella componente WFM.

Un altro importante tipo di risultati ottenibili con l'ambiente di Process Intelligence è costituita dai modelli di processo che possono essere scoperti con gli strumenti di Process Mining.

In particolare, in aggiunta ai modelli workflow (semplici o gerarchici) mostrati finora, il plugin AWS permette di scoprire un modello di processo tassonomico, costituito da un insieme di schemi workflow a vari livelli di astrazione.

La seguente figura mostra, ad esempio, la tassonomia (navigabile) scoperta per il log task.xml, fornendo dettagli sullo schema radice (selezionato per default) e sulle proprietà di astrazione trovate.

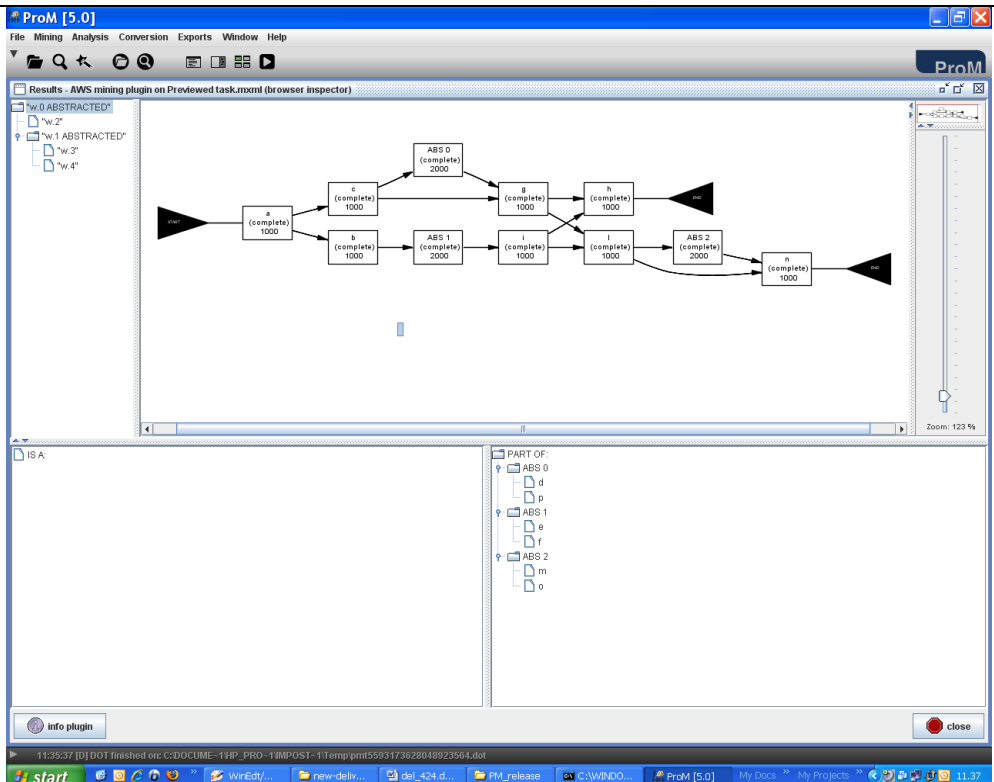
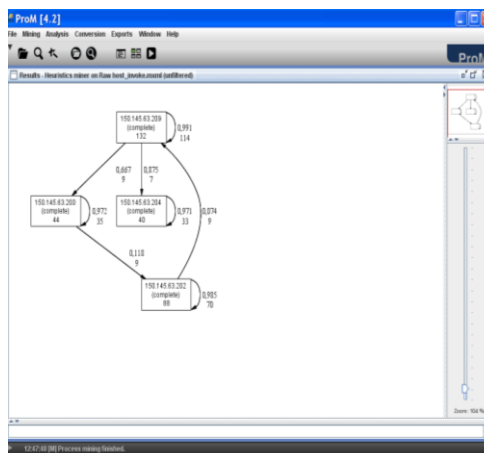


Figura 23: astrazione processo

La figura seguente mostra invece uno schema workflow indotto con il plugin (built-in) HeuristicMiner, a partire dal log peer.mxml  
 E' interessante notare l'applicazione delle tecniche di process mining su tale tipo di log (sequenze dei peer coinvolti nell'esecuzione delle varie istanze di processo) permette di studiare le modalità tipiche di interazione fra i componenti organizzativi/implementativi utilizzati per eseguire le attività di un processo.



## **5 Conclusioni**

I test di valutazione effettuati hanno permesso di verificare che il Prototipo Beta risponde correttamente alle specifiche funzionali anche all'interno di casi di studio reali.

In particolare si sono utilizzati gli scenari di OR5.2 e OR5.3 per mostrare la validazione, ma va ricordato che questo Prototipo è stato utilizzato anche nel caso di studio di OR5.4 e in parte anche per OR5.5. Il suo corretto funzionamento non è chiaramente legato ad uno specifico scenario ma possiamo concludere che si presta a svariati casi di business legati alla modellazione ed esecuzione di processi collaborativi.