Orchestration of Web Services

M.I. Capel

ETS Ingenierías Informática y
Telecomunicación
Departamento de Lenguajes y Sistemas Informáticos
Universidad de Granada
Email: manuelcapel@ugr.es
http://lsi.ugr.es/mcapel/

November, 6th 2019
Máster Universitario en Ingeniería Informática

1 Introduction

2 Exercise

Creation of a BPEL process

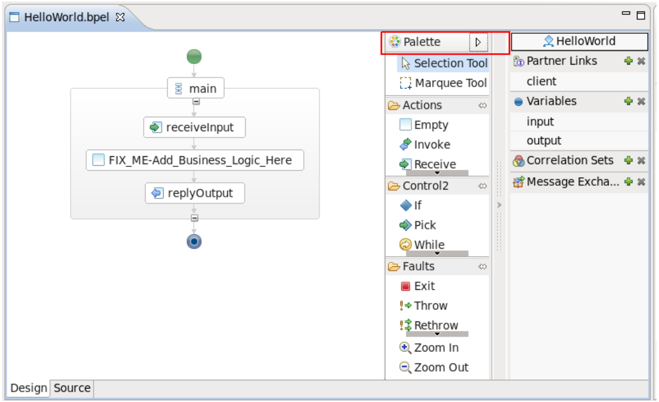


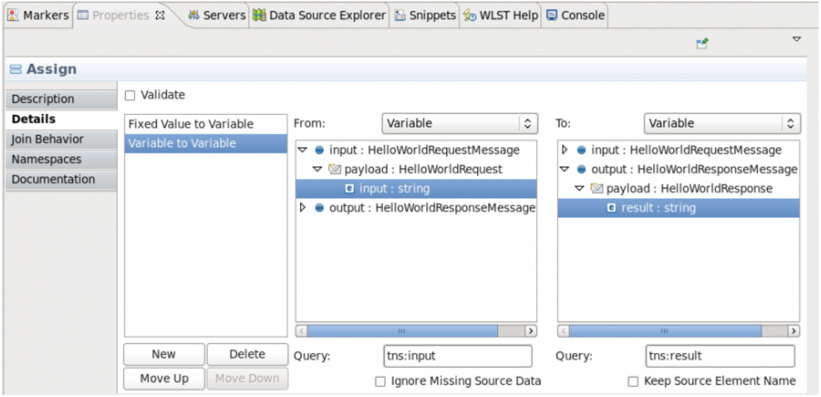
Figure: Palette View and BPEL process by the BPEL graphic editor

Creation of a BPEL process-II

Select template: *Synchronous BPEL Process* and change service's address to: `http://localhost:8080/ode/processes/HolaMundo`,
Tomcat executes on port 8080
Substitute the following in the widget shown:

- *Template*: `Synchronous BPEL Process`
- *Service Name*: `HolaMundoService` (autocompletado)
- *Port Name*: `HolaMundoPort`
- *Service Address*: `http://localhost:8080/ode/processes/HolaMundo`
- *Binding Protocol*: SOAP

BPEL Process actions



Validate

Fixed Value to Variable
Variable to Variable

From: Variable

- input : HelloWorldRequestMessage
  - payload : HelloWorldRequest
    - input : string
- output : HelloWorldResponseMessage

To: Variable

- input : HelloWorldRequestMessage
- output : HelloWorldResponseMessage
  - payload : HelloWorldResponse
    - result : string

Query: tns:input

Query: tns:result

Ignore Missing Source Data

Keep Source Element Name

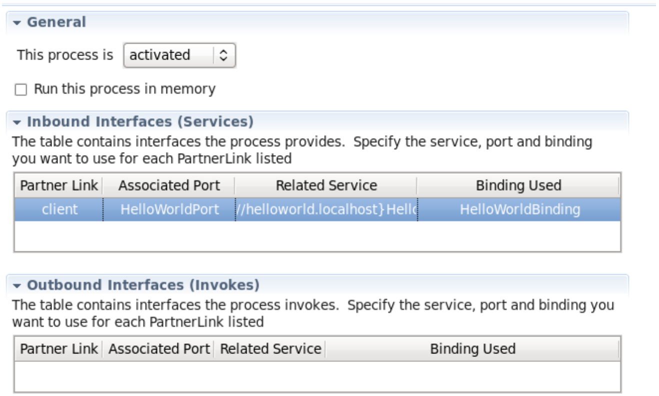
Figure: Assignment with copy of one variable inside an action of a bpel process

WDSL file

We need to give one *port* and *“binding”* to communicate with the service
Substitute the following in the widget shown:

- *Name*: `HolaMundoPort`
- *Binding*: `HolaMundoBinding`
- *Address*: `http://localhost:8080/ode/processes/HolaMundo`
- *Protocol*: SOAP

Service deployment



General

This process is activated

☐ Run this process in memory

Inbound Interfaces (Services)

The table contains interfaces the process provides. Specify the service, port and binding you want to use for each PartnerLink listed

Partner Link	Associated Port	Related Service	Binding Used
client	HelloWorldPort	//helloworld.localhost}Hell	HelloWorldBinding

Outbound Interfaces (Invokes)

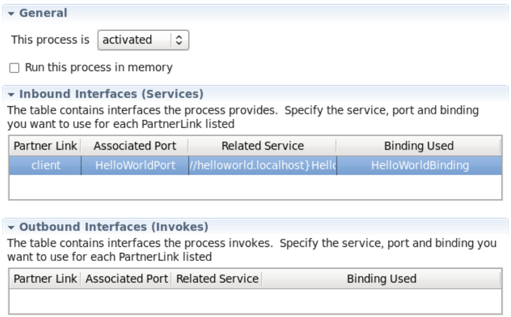
The table contains interfaces the process invokes. Specify the service, port and binding you want to use for each PartnerLink listed

Partner Link	Associated Port	Related Service	Binding Used

Figure: View of *deploy.xml* in the BPEL deployment editor

Service deployment -II

- In this example the *Partner Link* is called *client*
- We click on the field below column *Associated Port* and will appear a deplyment cursor for selecting *HolaMundoPort*
- Finally, click on *Related Service* and the file will be self-filled together with the other fields on the row



General

This process is activated

☐ Run this process in memory

Inbound Interfaces (Services)

The table contains interfaces the process provides. Specify the service, port and binding you want to use for each PartnerLink listed

Partner Link	Associated Port	Related Service	Binding Used
client	HelloWorldPort	/helloworld.localhost}Hello	HelloWorldBinding

Outbound Interfaces (Invokes)

The table contains interfaces the process invokes. Specify the service, port and binding you want to use for each PartnerLink listed

Partner Link	Associated Port	Related Service	Binding Used

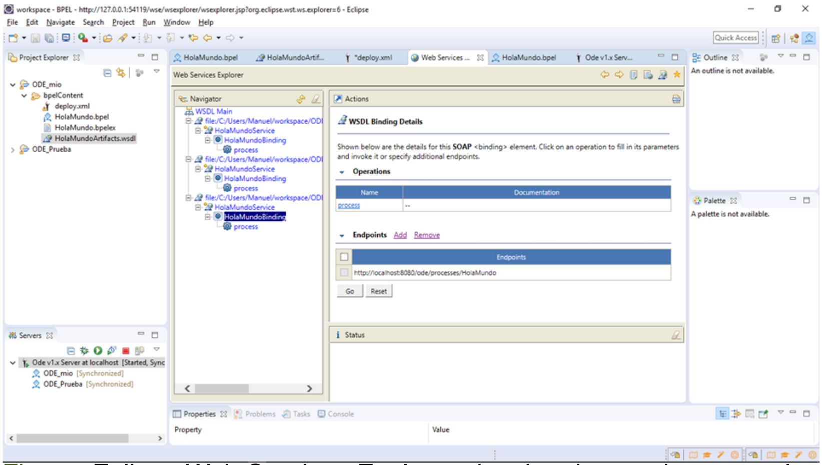
Figure: View of *deploy.xml* in the BPEL deployment editor

Service execution

- Go to the *Servers* view in your Eclipse and start up the ODE Server or right-click and select *Add and Remove*, a split menu that contains the project (*ODE_Prueba*) will show on the left part (*Available*), which will be added on the right column (*Configured*)
- Finally, select *Add* and *Finish*
- If the service *HolaMundo* was correctly deployed, we will see the following messages in the **Console** view:

```
11:44:49 INFO [BpelServerImpl] Registered process {http://holamundo.lo
11:44:49 INFO [DeploymentPoller] Deployment of artifact ODE_Prueba suc
[{http://holamundo.localhost}HolaMundo-1]
```

BPEL process execution with the Web Services Browser



WSDL Binding Details

Shown below are the details for this **SOAP** <binding> element. Click on an operation to fill in its parameters and invoke it or specify additional endpoints.

Operations

Name	Documentation
process	--

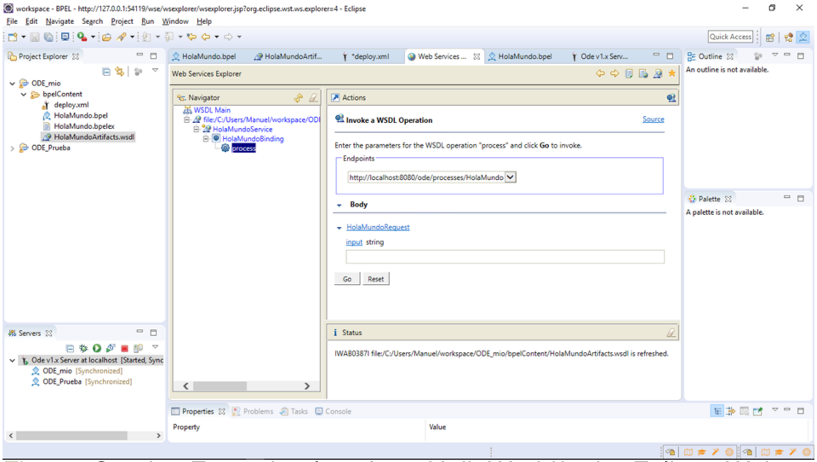
Endpoints Add Remove

☐	Endpoints
☐	http://localhost:8080/ode/processes/HolaMundo

Go Reset

Figure: Eclipse Web Services Explorer showing the service-example *HolaMundoService*

BPEL process execution with the Web Services Browser -II



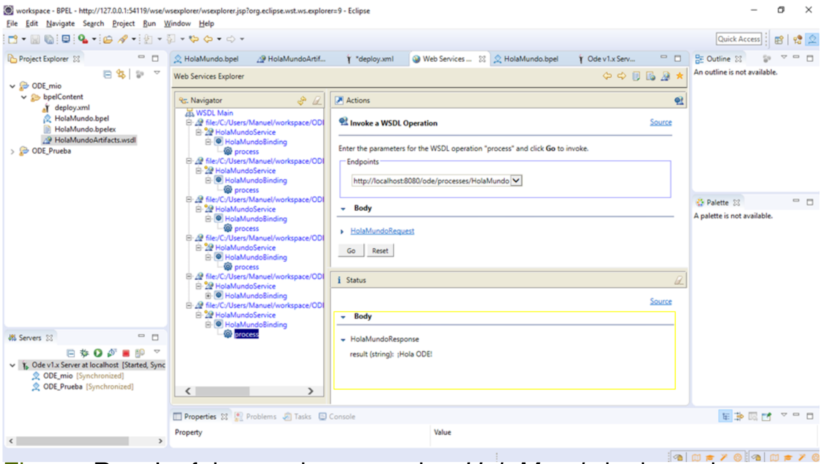
Enter the parameters for the WSDL operation "process" and click **Go** to invoke.

Endpoints: http://localhost:8080/ode/processes/HolaMundo

IWAB0387I file:/C:/Users/Manuel/workspace/ODE_mio/bpelContent/HolaMundoArtifacts.wsdl is refreshed.

Figure: Service Execution Interface *HelloWorld* in the Eclipse Web Services Explorer

BPEL process execution with the Web Services Browser -III



Invoke a WSDL Operation — Source

Enter the parameters for the WSDL operation "process" and click **Go** to invoke.

Endpoints: http://localhost:8080/ode/processes/HolaMundo

Body

HolaMundoRequest

Go Reset

Status — Source

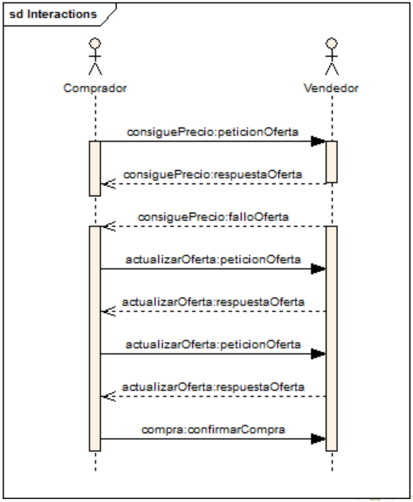
Body

HolaMundoResponse

result (string): ¡Hola ODE!

Figure: Result of the service execution *HolaMundo* in the web services browser

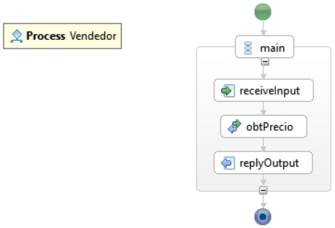
Orchestration of bargaining between buyer / seller services


sd Interactions
Comprador
Vendedor
consiguePrecio:peticionOferta
consiguePrecio:respuestaOferta
consiguePrecio:falloOferta
actualizarOferta:peticionOferta
actualizarOferta:respuestaOferta
actualizarOferta:peticionOferta
actualizarOferta:respuestaOferta
compra:confirmarCompra

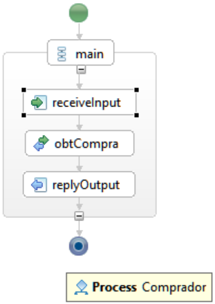
Problem description

- Orchestrate, in a simplified way, the bargaining between a buyer and a seller of a requested product, according to the interaction diagram shown in the figure
- The buyer starts asking for a price from the seller and the seller responds with a price for the product or an exception if he does not know the item that is demanded or is not available in the store
- The buyer continues to ask the seller for a price and enters a repetitive behavior with updates (from the price of the item) until he decides to buy the item when he considers that the best price is offered
- In this exercise it is requested to develop the complete description of the orchestration that has been described previously between the buyer and the seller

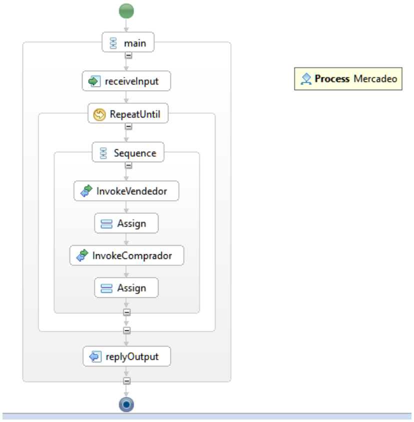
BPEL process “Vendedor”



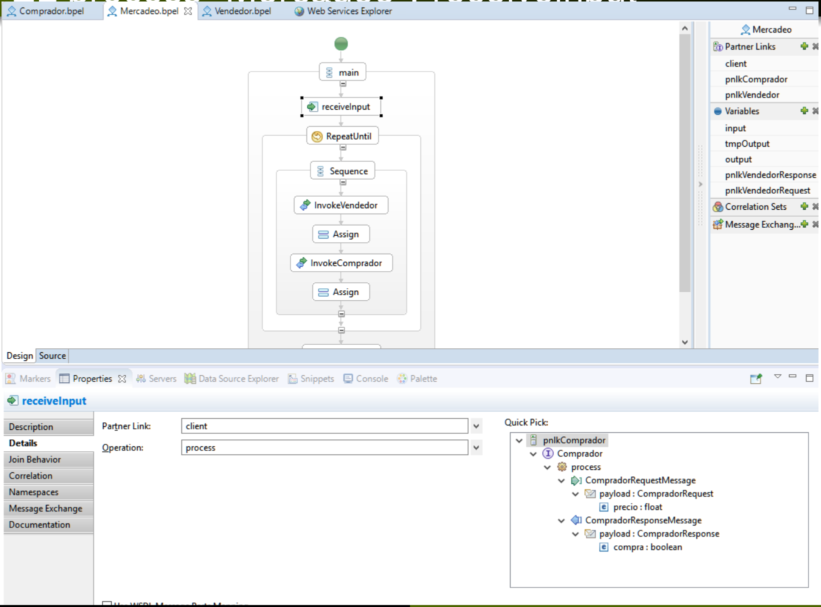
BPEL process “Comprador”



BPEL process “Mercadeo” (bargaining)



BPEL process “Mercadeo”: receiveinput



receiveInput

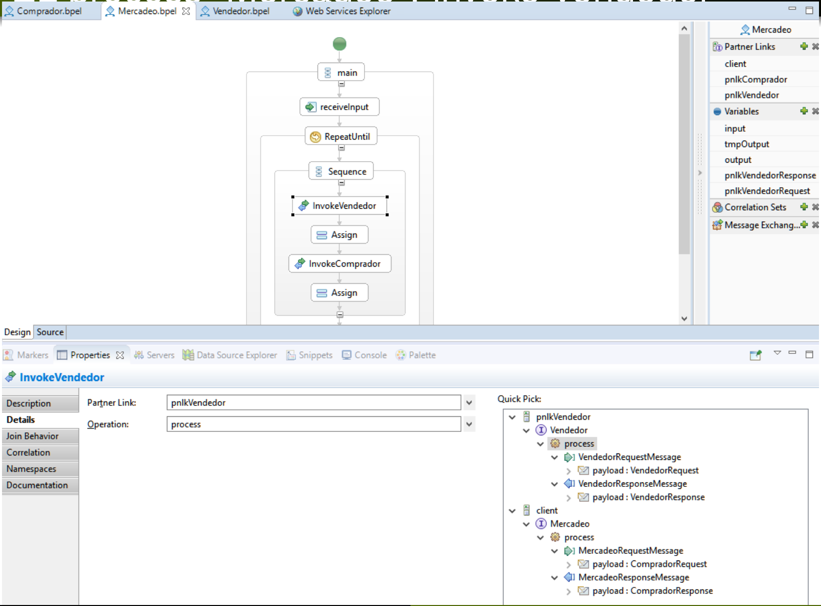
Partner Link: client

Operation: process

Quick Pick:

- pnlkComprador
  - Comprador
    - process
      - CompradorRequestMessage
        - payload : CompradorRequest
          - precio : float
      - CompradorResponseMessage
        - payload : CompradorResponse
          - compra : boolean

BPEL process “Mercadeo”: invoke-vendedor



InvokeVendedor

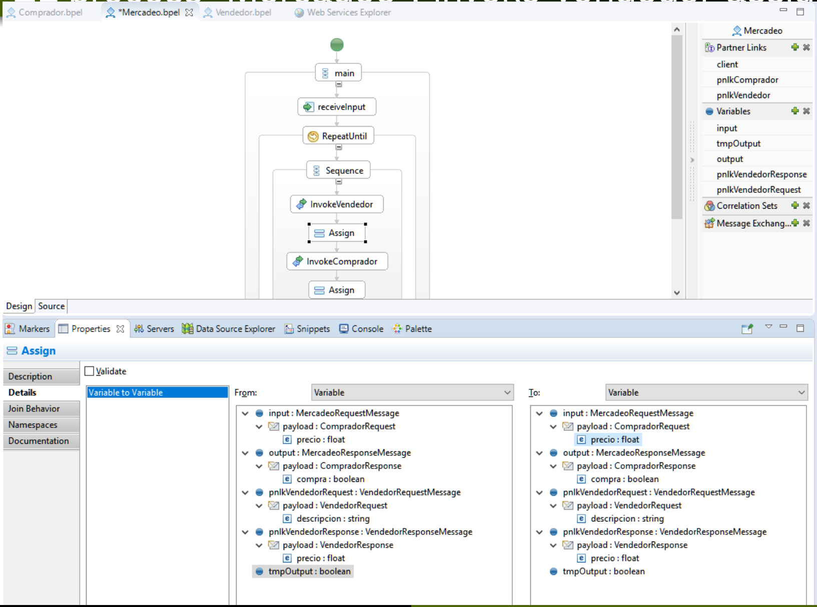
Partner Link: pnlkVendedor

Operation: process

Quick Pick:

```
pnlkVendedor
  Vendedor
    process
      VendedorRequestMessage
        payload : VendedorRequest
      VendedorResponseMessage
        payload : VendedorResponse
client
  Mercadeo
    process
      MercadeoRequestMessage
        payload : CompradorRequest
      MercadeoResponseMessage
        payload : CompradorResponse
```

BPEL process “Mercadeo”: invoke-vendedor-assign



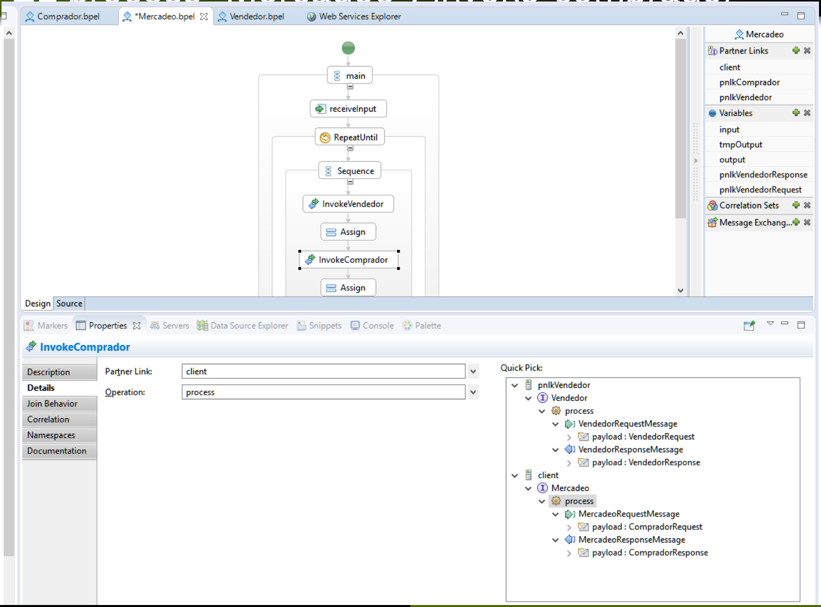
From:

```
input : MercadeoRequestMessage
  payload : CompradorRequest
    precio : float
output : MercadeoResponseMessage
  payload : CompradorResponse
    compra : boolean
pnlkVendedorRequest : VendedorRequestMessage
  payload : VendedorRequest
    descripcion : string
pnlkVendedorResponse : VendedorResponseMessage
  payload : VendedorResponse
    precio : float
tmpOutput : boolean
```

To:

```
input : MercadeoRequestMessage
  payload : CompradorRequest
    precio : float
output : MercadeoResponseMessage
  payload : CompradorResponse
    compra : boolean
pnlkVendedorRequest : VendedorRequestMessage
  payload : VendedorRequest
    descripcion : string
pnlkVendedorResponse : VendedorResponseMessage
  payload : VendedorResponse
    precio : float
tmpOutput : boolean
```

BPEL process “Mercadeo”: invoke-comprador



InvokeComprador

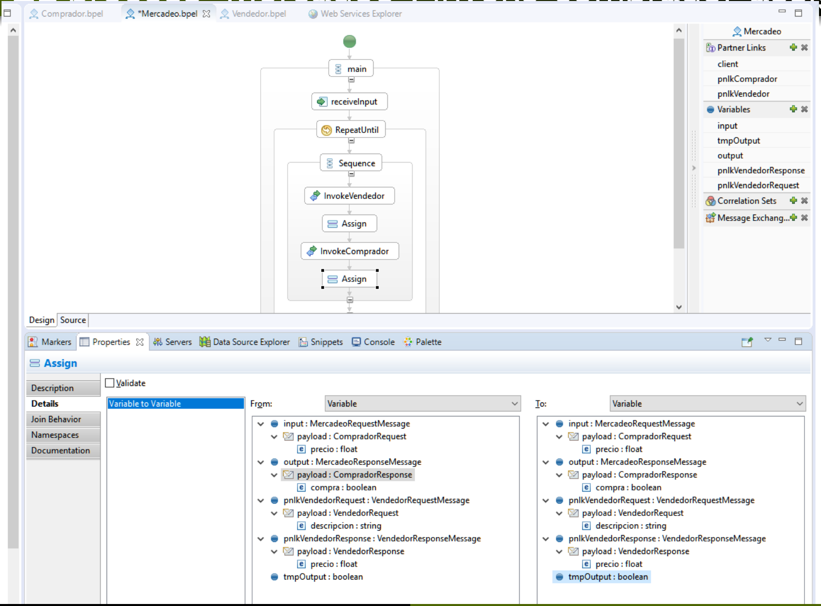
Partner Link: client

Operation: process

Quick Pick:

- pnlkVendedor
  - Vendedor
    - process
      - VendedorRequestMessage
        - payload : VendedorRequest
      - VendedorResponseMessage
        - payload : VendedorResponse
- client
  - Mercadeo
    - process
      - MercadeoRequestMessage
        - payload : CompradorRequest
      - MercadeoResponseMessage
        - payload : CompradorResponse

BPEL process “Mercadeo”: invoke-comprador-assign



From: Variable	To: Variable
input : MercadeoRequestMessage	input : MercadeoRequestMessage
payload : CompradorRequest	payload : CompradorRequest
precio : float	precio : float
output : MercadeoResponseMessage	output : MercadeoResponseMessage
payload : CompradorResponse	payload : CompradorResponse
compra : boolean	compra : boolean
pnlkVendedorRequest : VendedorRequestMessage	pnlkVendedorRequest : VendedorRequestMessage
payload : VendedorRequest	payload : VendedorRequest
descripcion : string	descripcion : string
pnlkVendedorResponse : VendedorResponseMessage	pnlkVendedorResponse : VendedorResponseMessage
payload : VendedorResponse	payload : VendedorResponse
precio : float	precio : float
tmpOutput : boolean	tmpOutput : boolean