

# WST: A Tool for Verifying Web Services systems

**María Emilia Cambronero, Valentín Valero, Gregorio Díaz**

*Escuela Politécnica Superior de Albacete*

*Departamento de Sistemas Informáticos*

*Universidad de Castilla-La Mancha*

*Email: [emicp@dsi.uclm.es](mailto:emicp@dsi.uclm.es), [valentin@dsi.uclm.es](mailto:valentin@dsi.uclm.es), [gregorio@dsi.uclm.es](mailto:gregorio@dsi.uclm.es)*





# 1.Introduction

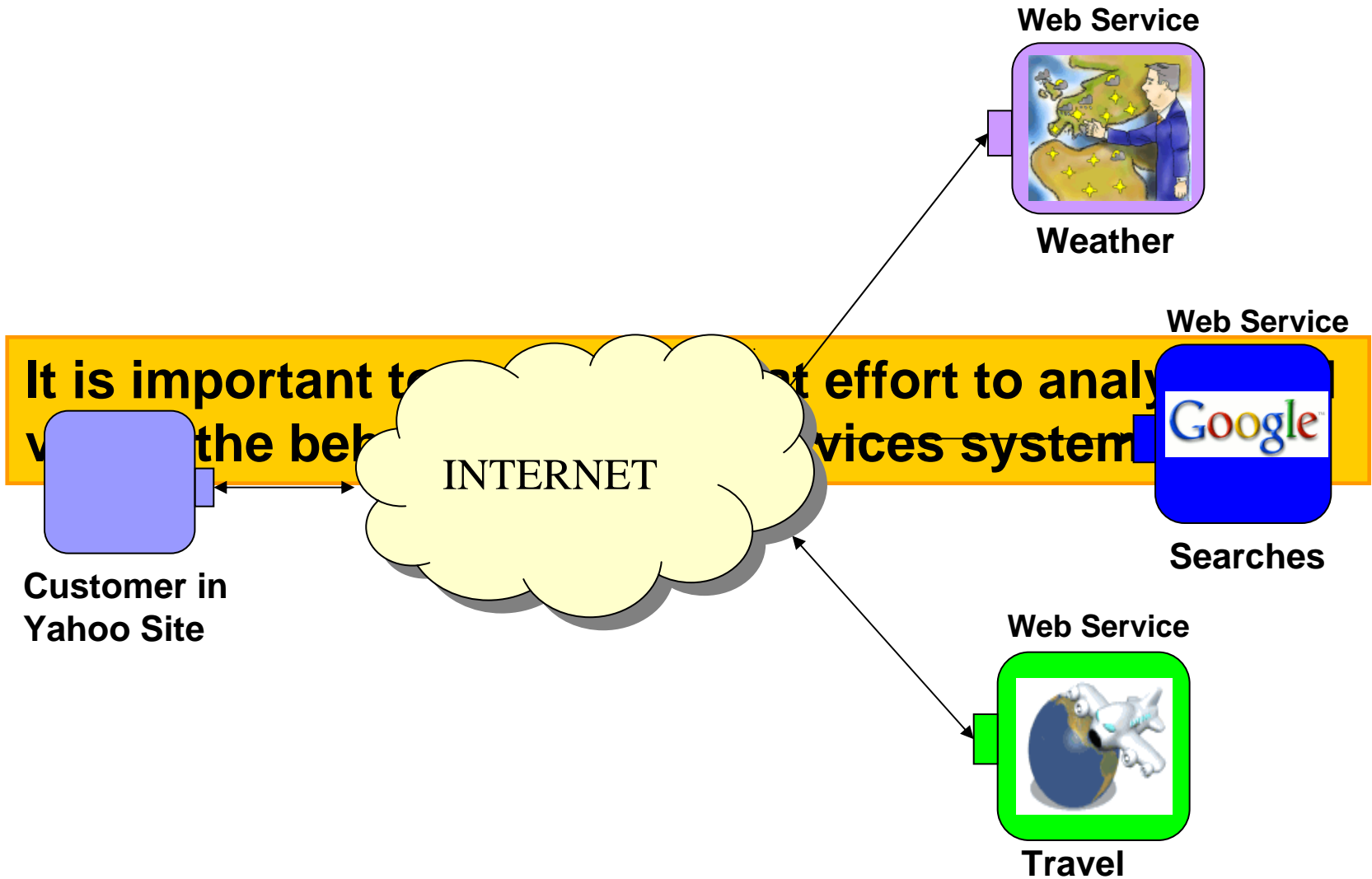
2.Web Services Translation tool (WST)

3.WS-CDL

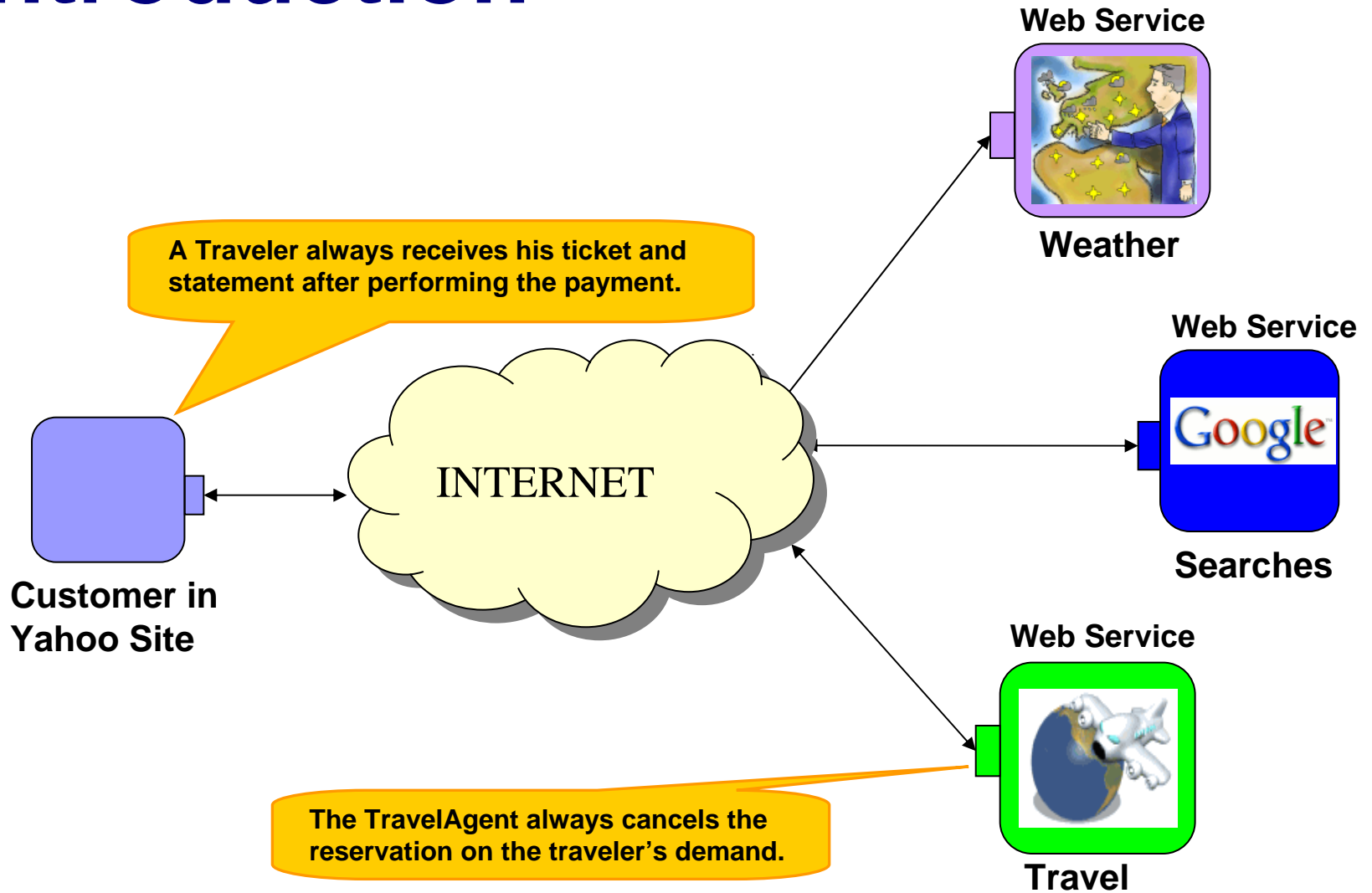
4.Translation from WS-CDL to TA

5.Conclusion and Future Work

# Introduction

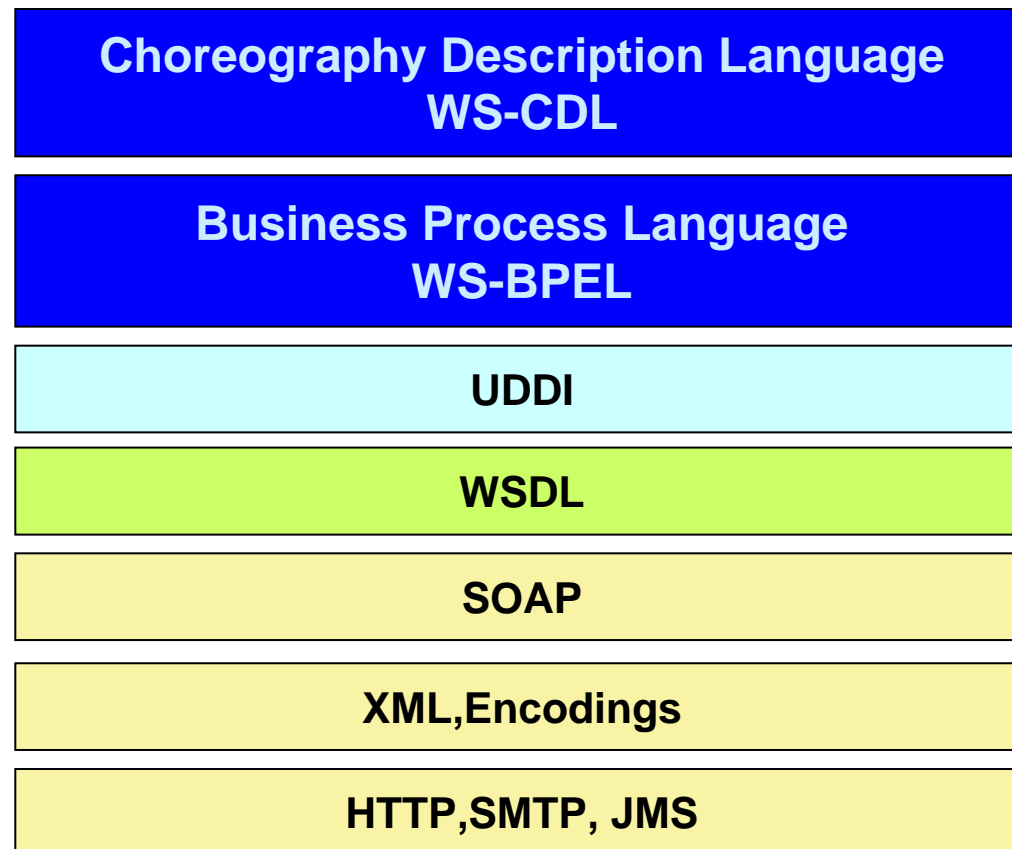


# Introduction

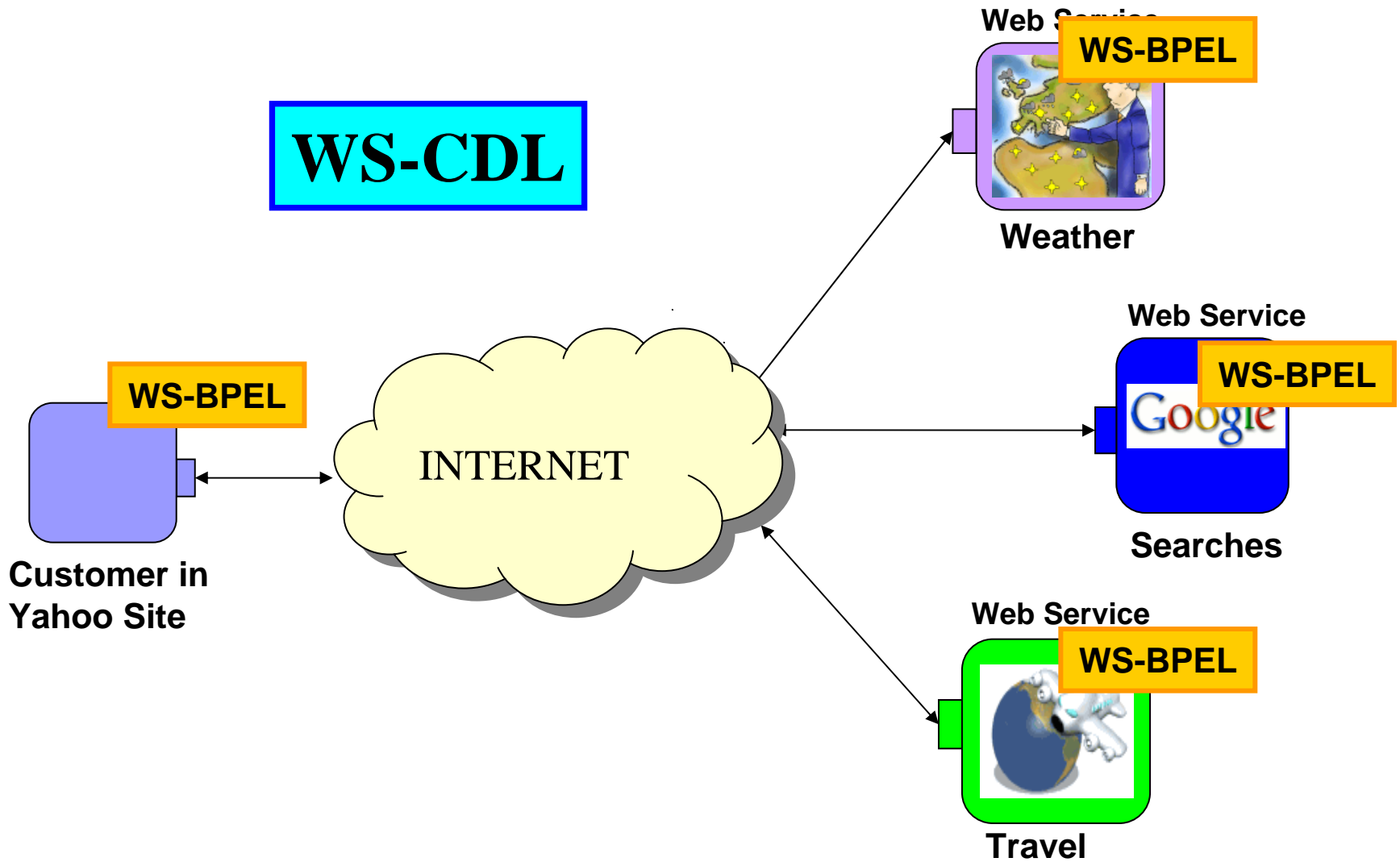


# Introduction

Web Services: SET OF PROTOCOLS & STANDARDS



# Introduction





1. Introduction

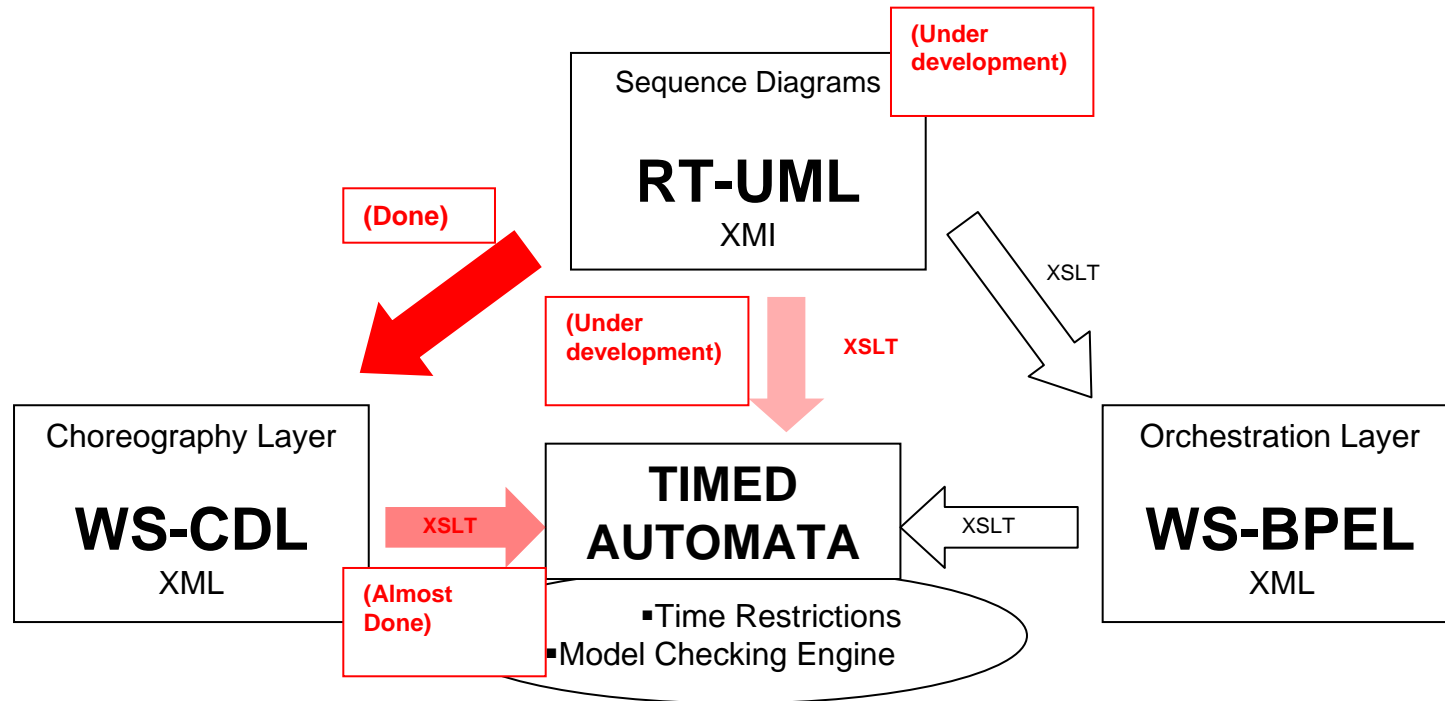
2. Web Services Translation tool (WST)

3. WS-CDL

4. Translation from WS-CDL to TA

5. Conclusion and Future Work

# WST tool





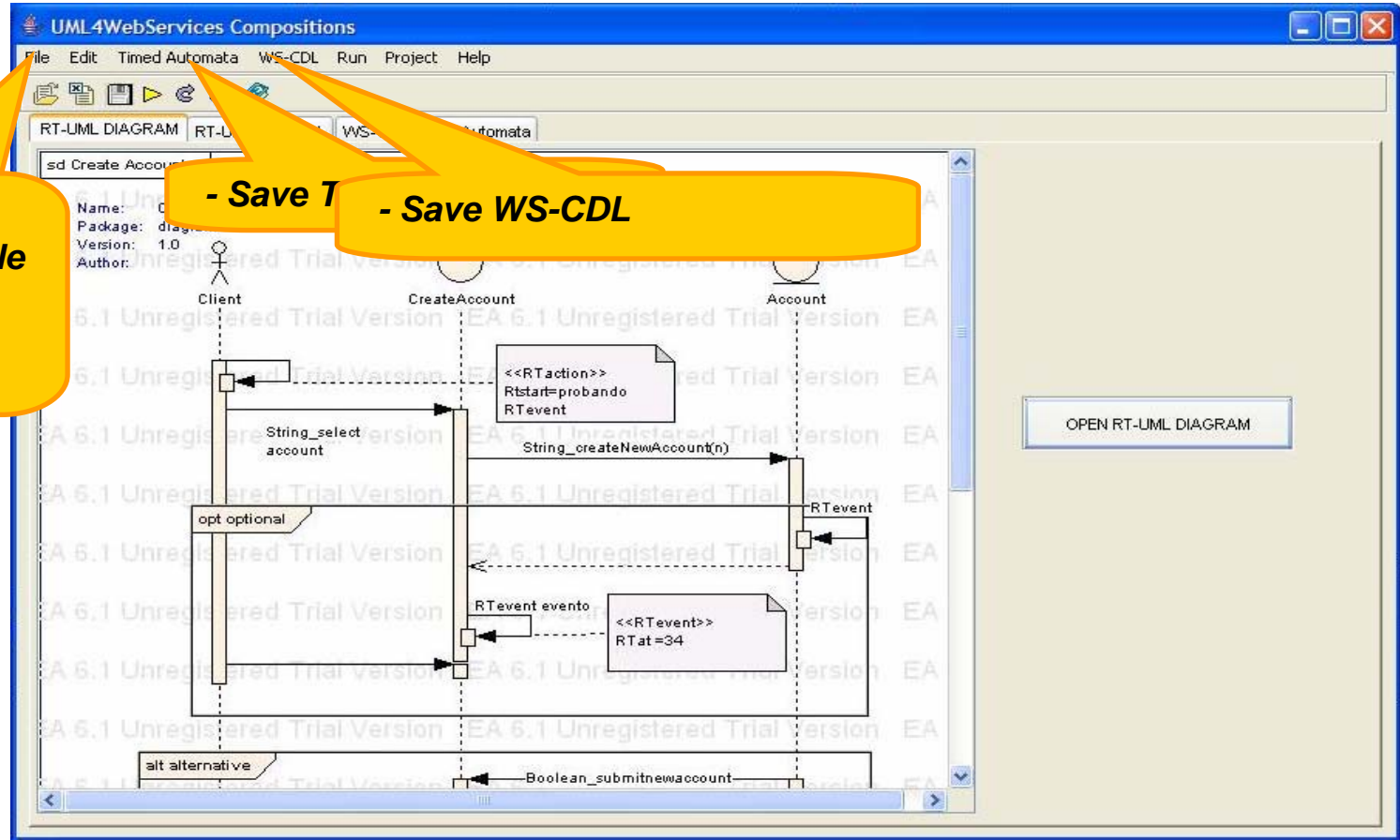
# WST tool

- Open XMI file
- Open WS-CDL file
- Save XMI file
- Save WS-CDL
- Exit

- Save T

- Save WS-CDL

OPEN RT-UML DIAGRAM



# WST tool

The screenshot shows the UML4WebServices Compositions application window. The title bar reads "UML4WebServices Compositions". The menu bar includes "File", "Edit", "Timed Automata", "WS-CDL", "Run", "Project", and "Help". The toolbar contains icons for file operations and execution. The main workspace is divided into three tabs: "RT-UML DIAGRAM", "RT-UML2WS-CDL", and "WS-CDL2TimedAutomata".

The "RT-UML DIAGRAM" tab is active, displaying XML code for an XMI document. The code includes headers for XML encoding, DOCTYPE, and namespaces, followed by XMI header and documentation tags. The main content is an XMI model element for "EA Model".

The "WS-CDL2TimedAutomata" tab is also active, displaying XML code for a WS-CDL document. The code includes headers for encoding and version, followed by a series of role and token declarations. The roles declared are "ClientRoleType", "AccountRoleType", and "CreateAccountRoleType", each with a description and a behaviour. The tokens declared are "ClientRef", "AccountRef", and "CreateAccountRef", each with an information type.

A blue speech bubble with the text "Execute the translation process" points to a blue arrow icon located between the two XML editors. A vertical line is drawn between the two editors to separate the source and target code.

WS-CDL documents generation, from the initial model (RT-UML diagram).

Execute the translation process

# WST tool

The screenshot shows the UML4WebServices Compositions tool interface. The window title is "UML4WebServices Compositions". The menu bar includes "File", "Edit", "Timed Automata", "WS-CDL", "Run", "Project", and "Help". The toolbar contains icons for file operations and execution. The main workspace is divided into two panes: "WS-CDL" on the left and "TIMED AUTOMATA" on the right. The "WS-CDL" pane displays XML code for a package containing information types and role types. The "TIMED AUTOMATA" pane displays the corresponding Timed Automata code, including locations, transitions, and synchronization labels. A blue callout bubble with the text "Execute the translation process" points to a blue arrow icon located between the two panes. A black callout bubble with the text "TA documents generation, from WS-CDL documents." is positioned over the WS-CDL code.

**TA documents generation, from WS-CDL documents.**

**Execute the translation process**

```
<?xml version="1.0"?>
<package>
  <informationType name="purchaseOrder"
    type="tns:PurchaseOrderMsg"/>
  <informationType name="purchOrderAccp"
    type="tns:PurchaseOrderAccp"/>
  <informationType name="purchaseOrderAck"
    type="tns:PurchaseOrderAck"/>
  <informationType name="purchaseOrderID"
    type="tns:PurchaseOrderID"/>
  <informationType name="purchaseOrderType"
    type="tns:PurchaseOrderType"/>
  <tokenLocator tokenName="tns:purchaseOrderID"
    informationType="tns:purchaseOrderID"
    query="/PO/orderId"/>
  <tokenLocator tokenName="tns:purchOrderAccp"
    informationType="tns:purchOrderAccp"
    query="/PO/orderId"/>
  <roleType name="Customer">
    <behavior name="Cust4IntSell"
      interface="tns:CustIntSellPT"/>
    <behavior name="Cust4Carr"
      interface="tns:Cust4CarrPT"/>
  </roleType>
  <roleType name="InternetSeller">
    <behavior name="IntSell4Cust"
      interface="tns:IntSell4Cust"/>
  </roleType>
</package>
```

```
<?xml encoding="UTF-8"?>
<!-- Uppaal Team//DTD Flat System -->
<!-- channel, customerchannel, ce
-->
<!-- PurchaseOrderCustomer -->
<!-- PurchaseOrderCustomer -->
</location>
<location id="identDeliverProductOrderCustomer">
  <name>identDeliverProductOrderCustomer</name>
</location>
<init ref="idhandlePurchaseOrderCustomer"/>
<transition>
  <source ref="idhandlePurchaseOrderCustomer"/>
  <target ref="identDeliverProductOrderCustomer"/>
  <label kind="synchronisation">sellerchannel!</label>
</transition>
<transition>
  <source ref="identDeliverProductOrderCustomer"/>
  <target ref="idhandlePurchaseOrderCustomer"/>
  <label kind="synchronisation">customerchannel?</label>
  <label kind="guard">
</label>
Clock1<=24
```



1. Introduction

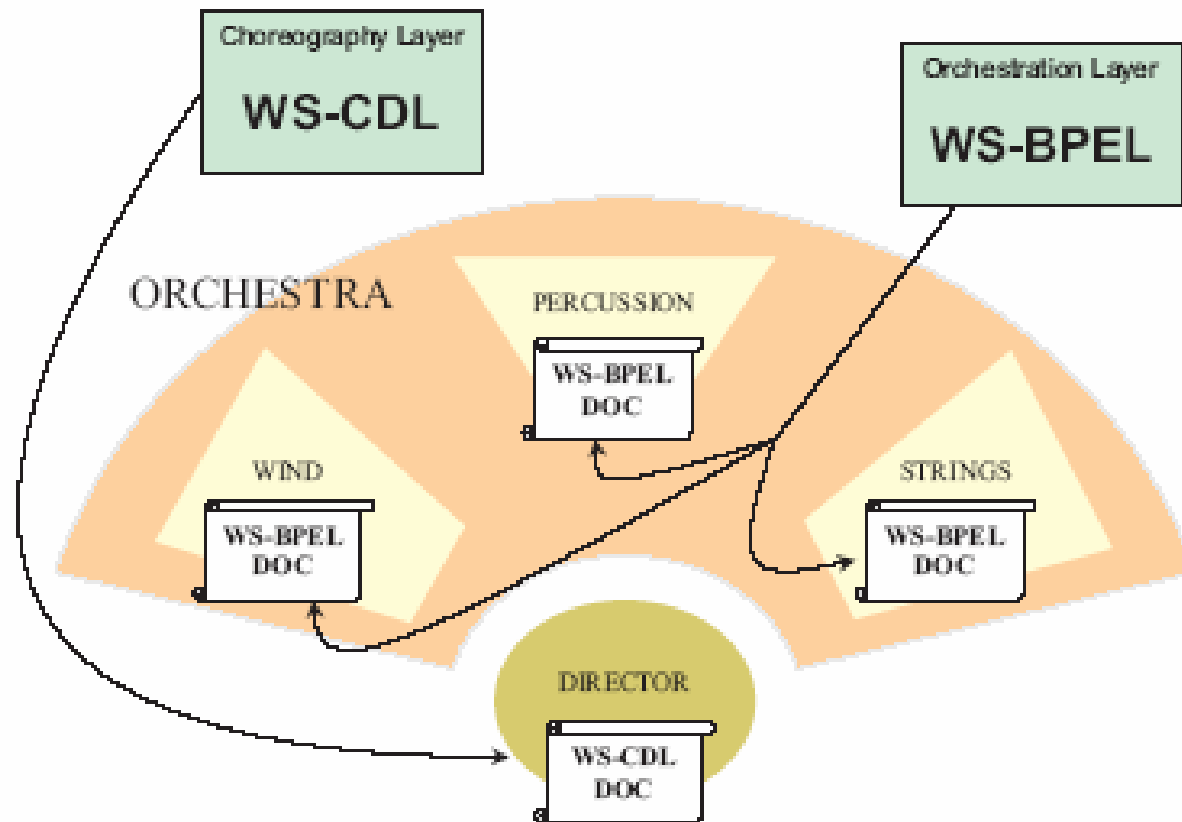
2. Web Services Translation tool (WST)

3. WS-CDL

4. Translation from WS-CDL to TA

5. Conclusion and Future Work

# WS-CDL



# WS-CDL

- Participant, Role and Relations types.
- Information types, Variables and Tokens.
- Choreographies (life-line, Exception and Finalizer blocks).
- Channels.
- Work Units.
- Activities and Ordering Structures.
- Interaction Activity.
- Semantics.



1. Introduction

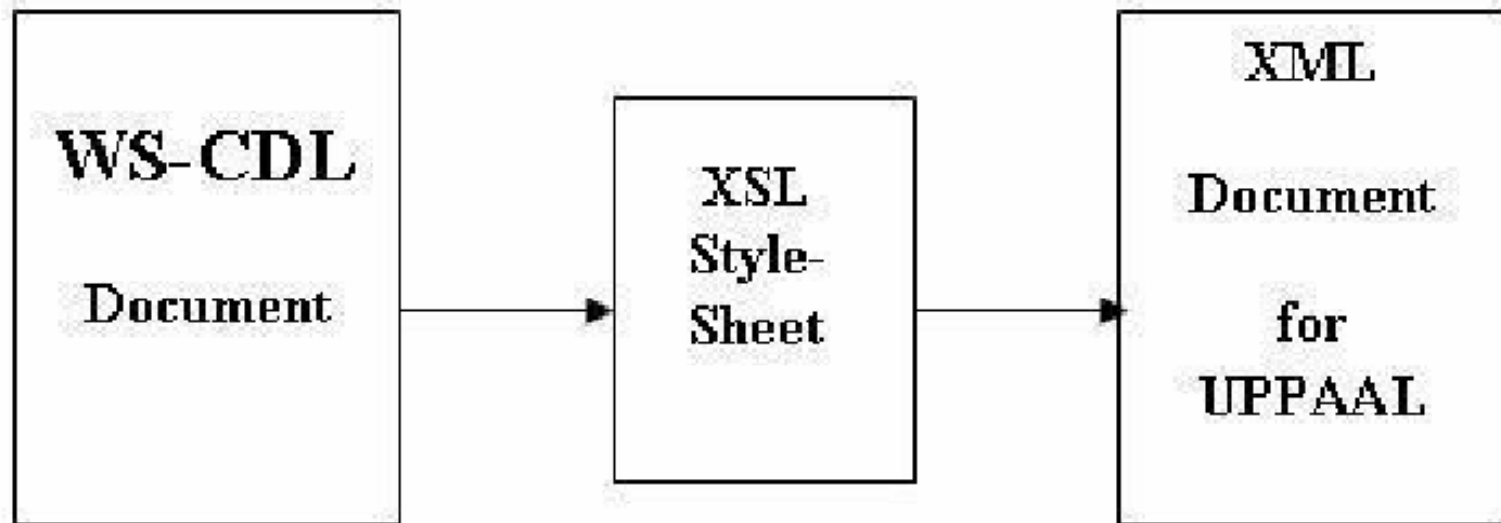
2. Web Services Translation tool (WST)

3. WS-CDL

4. Translation from WS-CDL to TA

5. Conclusion and Future Work

# From WS-CDL to TA



**XSLT (XML Stylesheets Language) for Transforming XML documents into other XML Documents.**



# From WS-CDL to TA

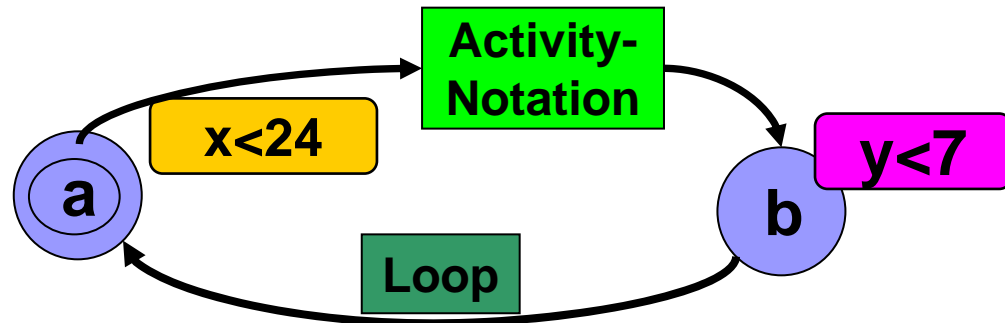
<b>WS-CDL</b>	<b>Timed Automata</b>
<b>Role</b>	<b>Template</b>
<b>Relation Type</b>	<b>Channel<sup>+</sup></b>
<b>Participant Type</b>	<b>Process<sup>+</sup></b>
<b>Channel Type</b>	<b>Channel</b>
<b>Variables</b>	<b>Variables</b>
<b>Choreography</b>	<b>Choreography<sup>+</sup>   Activity</b>
<b>Activity</b>	<b>Work Unit   Sequence   Parallelism   Choice</b>
<b>Sequence</b>	<b>Activity<sup>+</sup></b>
<b>Parallelism</b>	<b>Activity<sup>+</sup></b>
<b>Choice</b>	<b>Activity<sup>+</sup></b>

# From WS-CDL to TA

```
<roleType name="name">  
</roleType>
```

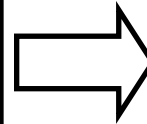
```
<workunit name="ncname"  
  guard="xsd:x<24"?  
  repeat="xsd:y<7"?  
  block="true|false"? >  
  Activity-Notation  
</workunit>
```

Automata  
Template

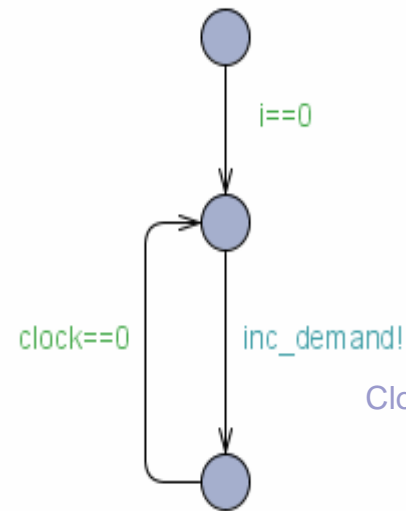


# From WS-CDL to TA

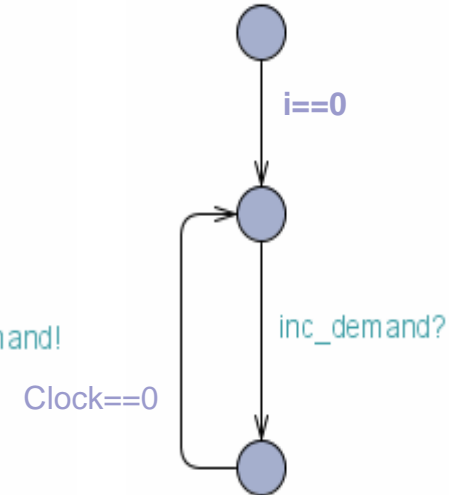
```
<role DemandMS>
<role ProductivityMS>
<choreography>
  <variable boolean inc_demand,clock x>
  <sequence>
    <workunit>
      <guard i==0>
      <repeat clock==0>
        <interaction>
          <from: DemandMS to: ProductivityMS>
          <exchange action=request>
            <record inc_demand:=false>
          </exchange>
        </interaction>
      </workunit>
    </sequence>
  </choreography>
```



Template  
DemandMS



Template  
ProductivityMS





1.Introduction

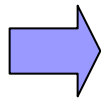
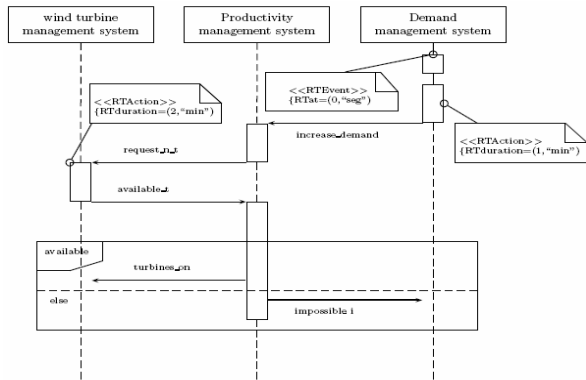
2.Web Services Translation tool (WST)

3.WS-CDL

4.Translation from WS-CDL to TA

5.Conclusion and Future Work

# Conclusions

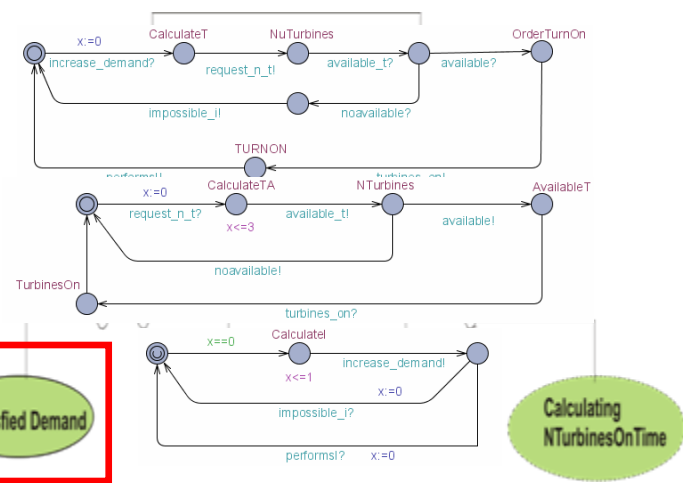
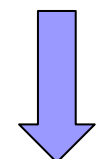


```

<interaction name= "" channelVariable="tns:turbines_on-channel"
operation="TurnOnTurbines" align="true" initiate="true">
  <participate relationshipType="tns:ProductivityMSWindTurbineMS"
fromRole="tns:ProductivityMS

toRole="tns:WindTurbineMS"/>
  <exchange name="request"
informationType="tns:turbineonType" action="request">
    <send variable= "cdl:getVariable("tns: turbineon", "", "")"/>
    <receive variable = "cdl:getVariable("tns:turbineon", "", "")"
recordReference="record-the-channel-info" />
  </exchange>
</interaction>

```



```

<interaction name= "" channelVariable="tns:turbines_on-channel"
operation="TurnOnTurbines" align="true" initiate="true">
  <participate
relationshipType="tns:ProductivityMSWindTurbineMS"
fromRole="tns:ProductivityMS

```

```

toRole="tns:WindTurbineMS"/>
  <exchange name="request"
informationType="tns:turbineonType" action="request">
    <send variable= "cdl:getVariable("tns: turbineon", "", "")"/>
    <receive variable = "cdl:getVariable("tns:turbineon", "", "")"

recordReference="record-the-channel-info" />
  </exchange>

```

```

<timeout time-to-complete=
"cdl:minor(cdl:getVariable("tns:z", "", ""),"00:04")" />?

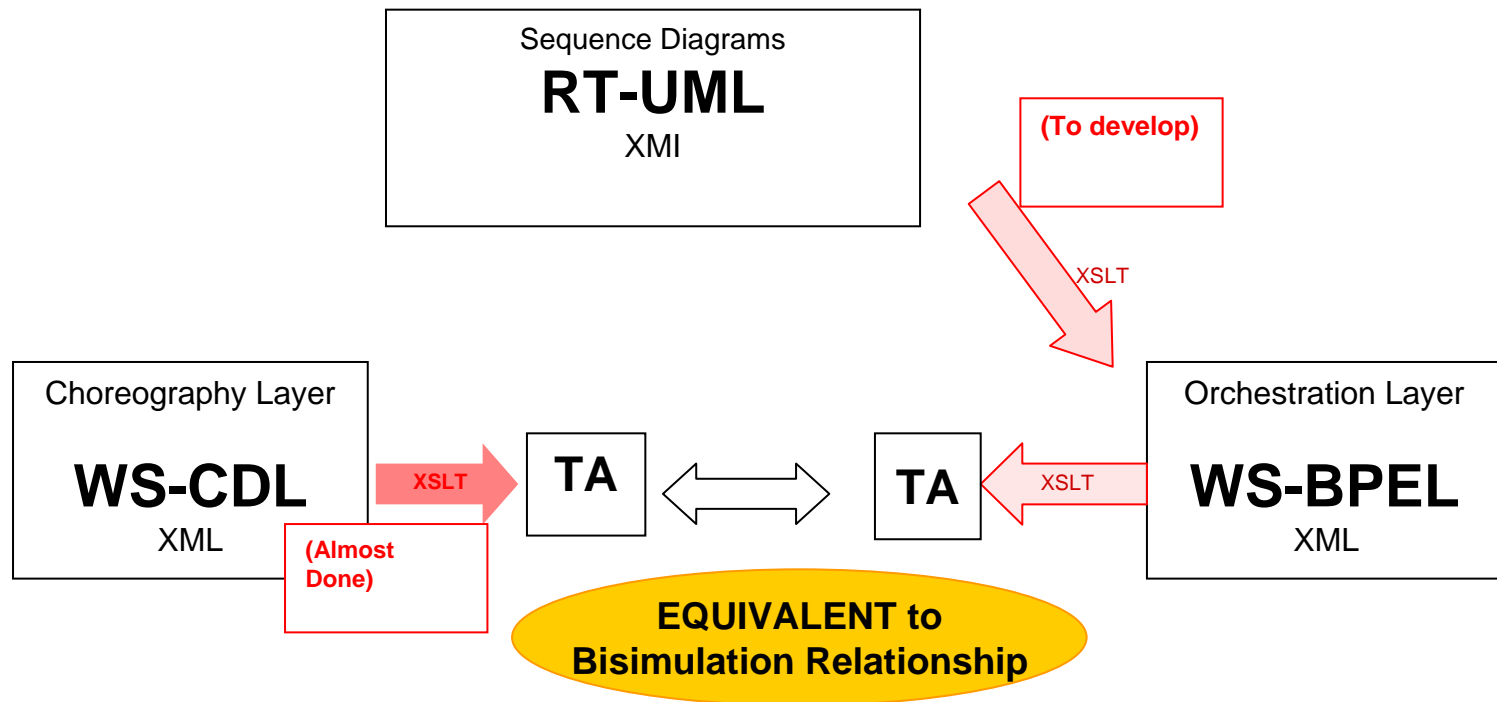
```

```

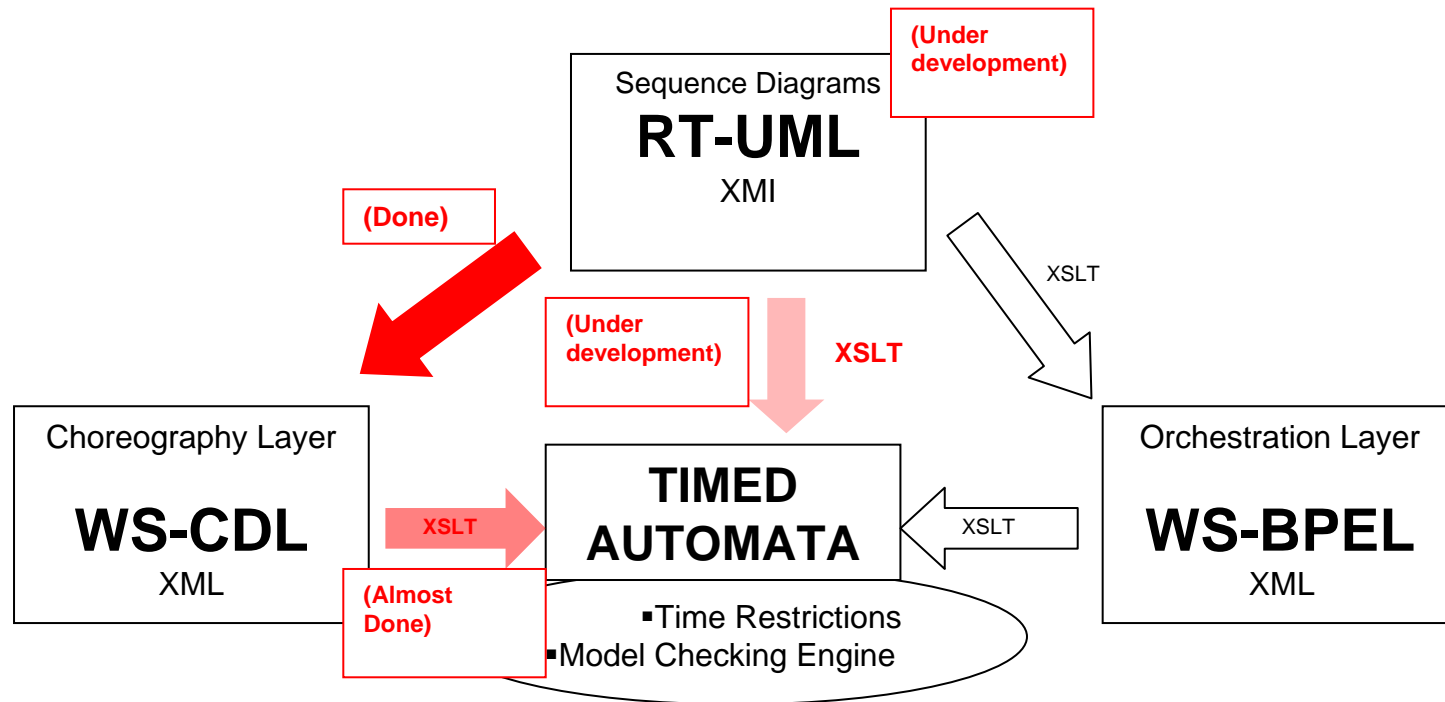
</interaction>

```

# Future Work



# Future Work



# WST: A Tool for Verifying Web Services systems

**María Emilia Cambroneró, Valentín Valero, Gregorio Díaz**

*Escuela Politécnica Superior de Albacete*

*Departamento de Sistemas Informáticos*

*Universidad de Castilla-La Mancha*

*Email: [emicp@dsi.uclm.es](mailto:emicp@dsi.uclm.es), [valentin@dsi.uclm.es](mailto:valentin@dsi.uclm.es), [gregorio@dsi.uclm.es](mailto:gregorio@dsi.uclm.es)*

