

# A Semantic Template Based Designer for Web Processes

Ranjit Mulye, John Miller, Kunal Verma, Karthik Gomadam, Amit Sheth  
*Large Scale Distributed Information Systems (LSDIS) Lab,*  
*Computer Science Department*  
*The University of Georgia, Athens GA 30602-7404*  
*ranjit@uga.edu, { jam, karthik, verma, amit }@cs.uga.edu*

## Abstract

The growing popularity of Service Oriented Computing based on Web services standards is creating a need for paradigms to represent and design business processes. Significant work has been done in the representation aspects with regards to WSBPEL. However, design and modeling of business processes is still an open issue. In this paper, we present a novel designer for business processes, which allows for intuitive modeling of Web processes, as well as using a template based approach for semi-automatically integrating partners either at design time or at deployment time. This work has been done as part of the METEOR-S project, which concentrates on adding semantics to the entire Web process lifecycle.

**Keywords:** Semantic Web services, Web Services Business Process Execution Language (WSBPEL), process designer.

## 1. Introduction

Many businesses are adopting Web service technology to expose their business applications, allowing them to have business collaboration both, within their organization and with business partners outside the organization. With the increasing popularity of Web Services, a new paradigm of business processes is gaining significant attention. The services offered by various businesses can be interconnected into a complex business process or workflow. The idea of building such types of combined processes has brought importance for modeling of business processes for Web service orchestration.

With the need to build complex business process, tools that would aid process developers in building such processes are gaining importance. Along with providing ease of process creation, tools are expected to be smarter and more sophisticated. In this paper, we present such a WS-BPEL [2] Web process designer tool – METEOR-S process designer. The

METEOR-S process designer offers a GUI based design interface to allow business process design.

A key element of any Web process is the partners that the process interacts with. Currently, the partner selection process is mostly static and naive, wherein partners are either selected ahead of time or the developer manually connects to a UDDI registry [1] and tries to search for services that fit his requirements. Though this approach would work, it does not guarantee that a partner that would be used in the resulting business process would be necessarily optimal. Secondly, since most of these searches would be keyword-based, the process developer may not be aware of similar services just because they are differently named. Also, chances are that the service may no longer be accessible and the process developer would not be aware of the same.

Consider a scenario wherein a process developer is designing a process to execute a purchase order. The process is invoked by a purchase order request by a buyer specifying the desired quantity of items he plans to purchase. This is followed by querying the supplier (partner) for availability of items. If the desired quantity of the order item is available in the inventory, then the order is placed with the supplier service. On successful completion of the order, an invoice is generated for the purchase order and returned to the caller of the process.

Generally, the order is fulfilled by procuring good from various suppliers. However, using static partner selection, the business process would be tied to a single supplier. This would mean that the process developer would have to write separate processes for each supplier he wishes to interact with. Also, this approach limits the amount of dynamic selection the process developer can achieve in selecting the supplier to choose. A better approach would be to either dynamically select a partner during the design phase of the process or assume a virtual supplier while designing the process and perform partner discovery at deployment time to achieve optimal benefits. Using this approach, the process developer can write a single process that can act as a blue print and may deploy multiple

instances of the process by selecting a dynamic partner depending on varying orders or needs.

The METEOR-S process designer uses semantic template (see figure 1) based partner discovery to realize the above mentioned functionality. Using our tool, the process developer would be able to semantically describe what she expects from the partner service and defer the partner selection process till either design time or deployment time.

The METEOR-S [3] research project, at the Large Scale Distributed Information Systems Lab, at the University of Georgia, is an effort towards developing a system to support development of a complete business processes lifecycle and to leverage the use of semantics in each step of the development cycle. METEOR-S is further discussed in detail in section 3.1. The process designer which is part of the METEOR-S project, addresses the issue of dynamic discovery of partners in a business processes.

The rest of the paper is organized as follows: Section 2 discusses the different ways in which Web processes can be modeled. Section 3 stresses the importance of use of semantics in Web process composition. The design and implementation of METEOR-S process designer is discussed in Section 4; whereas, section 5 talks about the related work. We sum up the conclusion and future work in section 6.

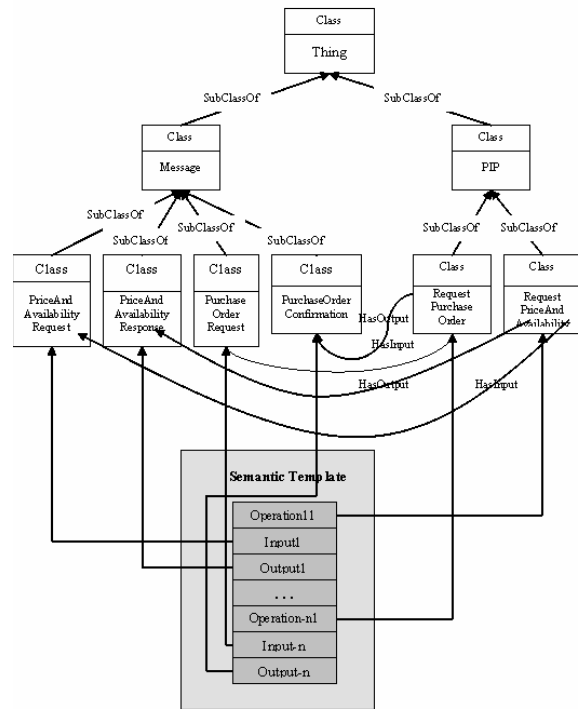
## 2. Web Processes

Different methodologies have been suggested for Web process creation. Systems such as SWORD [15] and SHOP2 [9] rely on AI planning for creating processes from high level goals. Also, automated composition of web services described in OWL-S [5, 28] process models have been proposed. While these systems provide complete automation, such systems have typically been limited to academic environments. Several tools have been created for semi-automatic composition of Web services [24, 25]. So far, the most successful and popular category of designers has been GUI based designers, that abstract the details of the process language syntax from the user. The METEOR-S designer fits in the GUI based designer category, with the caveat that it uses semantic templates to facilitate dynamic partner discovery.

### 2.1. Semantics in Web Services

One of the issues with search and discovery of Web Services is that currently services are mostly searched by their names. More often than not, due to words taking different meaning in different contexts, it becomes impossible to search for a service that

provides a desired operation just by searching on the service name. Also, keyword based search cannot handle the complexities of checking for matching services, in context of service orchestration. A key consideration of a service orchestration is to check for data mapping between outputs and inputs among successive services. These problems can be solved if service properties are described in an interoperable manner.



**Figure 1: Semantic Template using RosettaNet Ontology**

Considerable effort is being put to apply the Semantic Web [12] methodology to Web Services [16, 19] to achieve the above mentioned goals. The attempt is to describe the semantics of Web Services through use of ontology languages. The input and output messages of a Web Service can be annotated with concepts from an ontology to describe what they mean. Similarly, operations offered by the Service can be annotated to describe their functionality. Such annotations would allow search to be based more on “What can this service do functionally?” This would return better results, rather than a search on “What is this service named”? Also, the use of ontology languages facilitates the service properties to be machine processable.

Our tool makes use of the above mentioned data and functional semantics to dynamically search for partners. A process developer can specify the

requirements of a partner in the form of a semantic template. The template works as a requirement specification for desired partners. The semantic template consists of a set of operations with their inputs and outputs. All the operations and input/output messages are annotated by concepts from ontologies. A typical semantic template is shown in figure 1.

### 3. Architecture of the METEOR-S process designer

In this section, we discuss the architecture and implementation of the METEOR-S process designer.

#### 3.1 METEOR-S

As mentioned earlier, this process designer constitutes a part of the METEOR-S project at the LSDIS lab at University of Georgia. The METEOR (Managing End-To-End Operations) project at the LSDIS lab addresses the issues related to workflow process management for large-scale, complex workflow process applications in real-world multi-enterprise heterogeneous computing environments [8]. The follow-on project, called METEOR-S endeavors to define and support the complete lifecycle of Semantic Web processes [3]. An architectural overview of METEOR-S is presented in figure 4 in the appendix. The key steps in a life cycle of a Web process are

- Development and deployment of Semantic Web Services [7, 20, 21]
- Publication and Discovery of services [23]
- Composition[26]
- Constraint analysis [22]

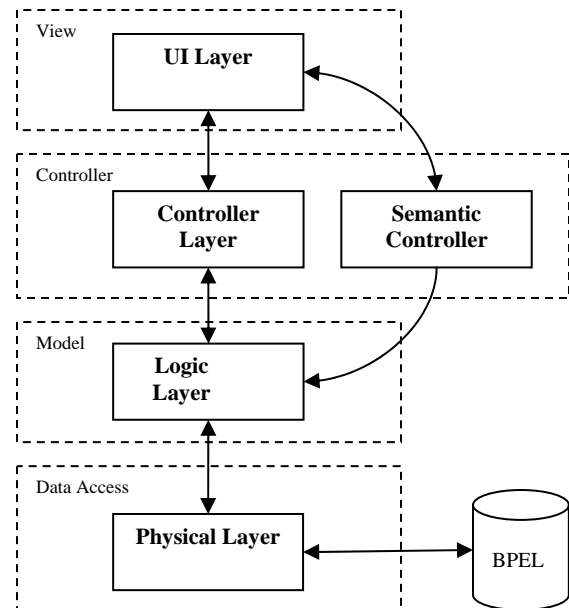
METEOR-S makes use of semantics in all of the above phases. This paper addresses the step of process composition.

#### 3.2 Architecture

The METEOR-S process designer follows the Model View Controller (MVC) [27] architecture. In the MVC paradigm input handling, modeling of the functional logic and the visual feedback of the state of the system are decoupled and handled separately. The “View” handles and manages the graphical output to the UI. The controller interprets the user inputs and acts as a link between the model and the view. The “Model” manages the behavior of the application logic.

The METEOR-S process designer consists of four main components, they are – UI Layer (View), Controller Layer (Controller), Logic Layer (Model) and

the Physical layer (Data Access). The layered structure is shown in figure 2.

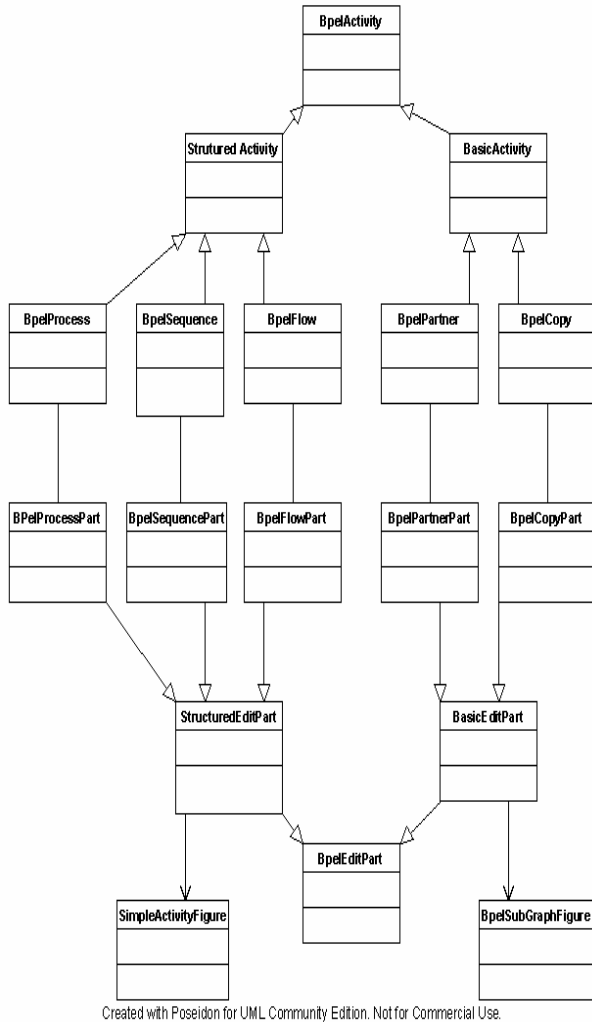


**Figure 2. METEOR-S Semantic Process Designer Architecture**

The UI Layer, as described earlier, takes care of reflecting the current state of the process and that of each contained BPEL element to the user. The Logic Layer represents the in-memory model of the entire process. The UI Layer reflects the current state of the model classes to the user. The Controller Layer is responsible for handling editing of the BPEL element properties and chaining the Logic and UI layer. A part of the overall UML diagram for the architecture is shown in figure 3.

#### 3.2.1 Physical Layer

The main task of this module is to generate output in the form of a WSBPEL [2] process file from the in-memory model and to parse a WSBPEL process file when an existing process is to be opened for editing. The physical layer components get invoked only during loading an existing process into the designer or saving a process after design completion. The physical layer also generates a skeleton process WSDL [13] for the web process. The WS-BPEL engine uses the process WSDL to expose the operations of the process to allow other clients or processes to access the process as a web service in itself.



**Figure 3: UML sample of METEOR-S Process Designer Architecture**

### 3.2.2 Logical Layer

The Logical Layer forms the core of the designer. The model classes in this layer are responsible for holding the current state of each element of a business process and that of the process as a whole, while the process developer works on the design of the process. For each business element or activity defined in the WSBPEL specification, we have designed a model class that reflects the structure and properties as outlined in the WSBPEL specification. The model classes have been grouped according to their functionality. The groups are summarized in table 1 below. Each group shares some common behavior.

### 3.2.3 Controller Layer

A controller class is designed for each model class that needs to be presented in the UI. The controller class, as the name suggests, acts as an agent between the model and the UI presentation. Any changes in the logical layer are reflected in the UI with the help of the controller classes. Also, it handles user input in the form of either creation and deletion of new process elements or modifications done to element properties.

**Table 1: Functional groups of BPEL elements**

<i>Definitions Group</i>	<i>Basic Activities Group</i>
<ul style="list-style-type: none"> <li>• Namespace</li> <li>• Variables</li> <li>• Partners</li> <li>• Correlation Sets</li> </ul>	<ul style="list-style-type: none"> <li>• Invoke</li> <li>• Receive</li> <li>• Reply</li> <li>• Wait</li> <li>• No-op</li> <li>• Assignment</li> </ul>
<i>Structured Activities Group</i>	<i>Extensions/Special Activities Group</i>
<ul style="list-style-type: none"> <li>• Sequence</li> <li>• Switch</li> <li>• While</li> <li>• Pick</li> <li>• Flow</li> <li>• Links</li> <li>• Scope</li> </ul>	<ul style="list-style-type: none"> <li>• Correlation</li> <li>• Compensation Handlers</li> <li>• Fault Handlers</li> <li>• Transaction Elements</li> </ul>

### 3.2.4. UI Layer

METEOR-S process designer uses the Graphical Editing Framework (GEF) [10] which is part of the Eclipse project [29] to design the front-end GUI for the designer. Since GEF follows a Model-View-Controller architecture, it fits in well with the overall design of the system. The view objects in the UI layer give a visual representation of the process state to the user. The views of the various elements are color coded according to the group they belong to (refer to table 1). This helps in making the process designer more intuitive.

## 4. Features and Working

The METEOR-S process designer provides some useful features to aid process developers build WS-BPEL processes.

The METEOR-S process designer offers an easy to use GUI to process developers to rapidly build web processes. Processes developers are offered support for dragging and dropping of process elements on a process “canvas”. This, combined with ease of element selection and deletion offer a simple to use GUI. Selecting a particular element opens up a property sheet that allows the user to modify element properties. This approach helps in hiding the unnecessary syntax details from the developer. The METEOR-S process designer uses color coded element icons for better understanding of the process. The process developer can quickly get an idea of the element type just by looking at the color of the icon. Container elements such as sequence, flow, etc can be nested to any depth and thus can have a process with arbitrary complexity. Figure 4 in the appendix shows a process created using the METEOR-S process designer. As can be seen, the GUI is easy to comprehend and one can easily make out as to which process element is contained within another element.

#### **4.1. Semantic Selection of Process Partner**

To support capabilities of dynamic partner selections, METEOR-S process designer makes use of semantic Web Services described in section 3.

For the semantic selection process to work, it is assumed that semantically annotated services have been developed and published by Web Service developers. The Web Service designer annotates the data and operations of a Web Service before publishing them to a UDDI registry. METEOR-S uses OWL ontologies to annotate the operations offered by a Web service and the parameters and return types of those operations. This can be done using the WSDL-S [21] development module of METEOR-S and is discussed in more detail in [7].

The selection process can be thought of being constituted of two phases. First is the template generation phase where the process developer specifies his requirements for a partner in terms of semantic template. In the second phase the METEOR-S process designer tries to find a matching partner for the semantic template specified. The following sections discuss the two phases in detail.

In the first phase, the process developer generates a semantic template with a simple to use GUI. The developer can add operations to the template and specify the input and output messages for each operation. Each of these entities is annotated by ontology concepts. The developer can either create a new template or she has a choice to load an existing template.

Phase two comprises of the core searching mechanism for dynamic partner discovery. Once the process designer has generated/selected a semantic template for a particular partner, she can trigger the discovery process to find a matching partner. The METEOR-S process designer extracts the semantic information from the template and passes it to the discovery module. The discovery module performs the semantic search and returns results that match the template. The process developer can then choose a particular service from the result set to act as a partner for the particular process instance. This phase is explained in figure 3. The process developer can either discover the partner service at design time or differ it till deployment time. Provision to allow runtime discovery is discussed as part of future work in section 6.

#### **4.2. Design Time Discovery vs. Deployment Time Discovery**

As discussed earlier a process developer can either choose to discovery partner services at design time or may do so at deployment time of the process. The advantages of deployment time discovery is that the level of dynamism achieved is greater as compared to design time discovery. Chances are that between the time the process is designed and later gets deployed, some services that yield a better match may be available. However, with deployment time discovery, one has to look into the issue of data mapping of the inputs and outputs of the new found partner with that of what the process expects. This is discussed in further details in [11].

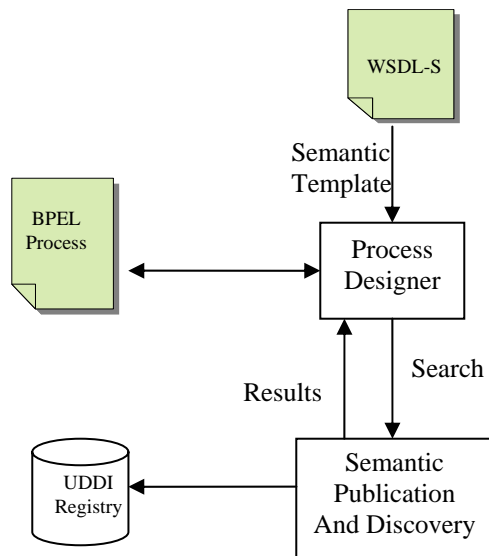
### **5. Related Work**

In this paper we have presented approach to model business processes using an easy to use process designer. We also introduced an approach for dynamic partner search using features of the semantic Web. This section discusses some of the related work in the field.

Searching for Web Services using non-semantics approaches like similarity searches, have been proposed [17] and have been evaluated to do better than conventional keyword search. Use of semantics in achieving automation in the process of Web processes creation and Web Service interactions has been proposed [15, 9, 5] using ontologies to describe Web Service entities. The work in [15, 9, 5] follows fully automated composition approach. In these approaches, the process developer has no control over the exact steps of the Web process. The METEOR-S process designer uses a semi automatic composition

approach that requires human intervention. Though the amount of work needed on the part of the developer in case of fully automated process design is less, current business may not be comfortable with the approach. Businesses would expect to have a control over the entire process design. In the METEOR-S process designer the process designer has freedom to write the complete process as per his design and we provide dynamism at a final granularity of partner selection. Though this involves more work on the part of the process developer, she has the complete freedom over the entire process design.

In the area of process designer without dynamic selection, different approaches to model of Web processes have been proposed [14, 6] either using existing methodologies like UML activity graph [18] or similar workflow languages.



**Figure 3. Phase II – Dynamic partner discovery**

## 6. Conclusion and Future Work

In this paper, we presented an approach to model Web processes that allows a process designer to take advantage of advancement in the area of semantic Web while designing Web process. This approach proposes a template based searching mechanism to allow dynamic searching of business process partners at design or deployment time of a Web process. We have implemented a model based designer to verify the usefulness of our approach. The designer also offers a user friendly GUI to the process developer to construct

business processes. The designer gives complete freedom to the process developer over the design and structure of the web process she plans to design.. This work has been done as part of the METEOR-S project at the University of Georgia.

Future work in this area involves inbuilt support for browsing Web process WSDL and searching for them from a UDDI. Adding dynamic data mapping capabilities to further enhance deployment time binding is proposed. The process designer can be adapted to further achieve run-time discovery of partner services. This can be done by use of proxy services that would perform the discovery and invocation of partner services. Mechanism to hook the designer into a BPEL engine to reflect real-time feedback has been suggested as a value addition.

## References

- [1] Universal Description, Discovery and Integration (UDDI), <http://www.uddi.org/>
- [2] Web Services Business Process Execution Language (WSBPEL), [http://www.oasis-open.org/committees/tc\\_home.php?wg\\_abbrev=wsbpel](http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=wsbpel)
- [3] METEOR-S: Semantic Web Services and Processes, <http://lsdis.cs.uga.edu/projects/METEOR-S/>
- [4] Narayanan S. and McIlraith S., "Simulation, Verification and Automated Composition of Web Services" International World Wide Web Conference, May 2002,
- [5] Traverso P., and Pistore M., "Automated Composition of Semantic Web Services into Executable Processes", Third International Semantic Web Conference (ISWC2004), Nov. 2004,
- [6] W.M.P. van der Aalst and A.H.M. ter Hofstede. YAWL: Yet Another Workflow Language. QUT Technical report, FIT-TR-2002-06, Queensland University of Technology, Brisbane, 2002
- [7] Rajasekaran P., Miller J., Verma K., Sheth A., "Enhancing Web Services Description and Discovery to Facilitate Composition," Proceedings of the 1st International Workshop on Semantic Web Services and Web Process Composition, July 2004
- [8] METEOR: Cardoso J., Sheth A., et. al., "Quality of Service for Workflows and Web Service Processes", Journal of Web Semantics, 2004
- [9] Nau D., Cao Y., Lotem A., and Muñoz-Avila H, "SHOP: Simple Hierarchical Ordered Planner", Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence, Pages: 968 - 975, 1999
- [10] GEF: Graphical Editing Framework, <http://www.eclipse.org/gef/>

- [11] Verma K., Akkiraju R., Goodwin R., Doshi P., and Lee J., "On Accommodating Inter Service Dependencies in Web Process Flow Composition", 2004 AAAI Spring Symposium Series, March, 2004
- [12] Berners-Lee, T., Hendler, J., Lassila, O., "The Semantic Web", Scientific American, May 2001.
- [13] WSDL: Web Service Description Language, <http://www.w3.org/TR/wsdl>
- [14] Zarras A., Vassiliadis P., and Issarny V., "Model-Driven Dependability Analysis of Web Services", International Symposium on Distributed Objects and Applications, October 2004
- [15] Ponnekanti S., and Fox A., "SWORD: A Developer Toolkit for Web Service Composition", In Eleventh World Wide Web Conference (WWW2002, Web Engineering Track), Honolulu, Hawaii, May 2002.
- [16] Dogac A., "Exploiting Semantic of Web Services through ebXML Registries"
- [17] Dong X., Halevy A., Madhavan J., Nemes E., and Zhang J., "Similarity Search for Web Services", 30th VLDB Conference, August - September 2004
- [18] UML: Unified Modeling Language, Object Management Group, <http://www.uml.org/>
- [19] Paolucci M., Sycara K., and Kawamura T., "Delivering Semantic Web Services." In Proceedings WWW2003
- [20] Sivashanmugam K., Verma K., Sheth A., and Miller J., "Adding Semantics to Web Services Standards", 1st International Conference on Web Services, June 2003
- [21] Miller J., Verma K., Rajasekaran P., Sheth A., Aggarwal R., and Sivashanmugam K., "WSDL-S: A Proposal to W3C WSDL 2.0 Committee", <http://lsdis.cs.uga.edu/projects/WSDL-S/wsdl-s.pdf>
- [22] Aggarwal R., Verma K., Miller J., and Milnor W., "Constraint Driven Web Service Composition in METEOR-S," Proceedings of the 2004 IEEE International Conference on Services Computing (SCC 2004), Shanghai, China (September 2004) pp. 23-32.
- [23] Verma K., Sivashanmugam K., Sheth A., Patil A., Oundhakar S., and Miller J., "METEOR-S WSDI: A Scalable Infrastructure of Registries for Semantic Publication and Discovery of Web Services", Journal of Information Technology and Management, Special Issue on Universal Global Integration, Vol. 6, No. 1 (2005) pp. 17-39. Kluwer Academic Publishers.
- [24] Aggarwal R., Verma K., Miller J., and Milnor W., "Dynamic Web Service Composition in METEOR-S" , Technical Report, LSDIS Lab, Computer Science Dept., UGA, May 2004.
- [25] Sirin e., Parsia b., and Hendler J., "Composition-driven filtering and selection of semantic web services", AAAI Spring Symposium on Semantic Web Services, 2004.
- [26] Sivashanmugam K., Miller J., Sheth A., and Verma K., "Framework for Semantic Web Process Composition," International Journal of Electronic Commerce (IJEC), Special Issue on Semantic Web Services and Their Role in Enterprise Application Integration and E-Commerce, Vol. 9, No. 2 (Winter 2004-5) pp. 71-106. M.E. Sharpe, Inc.
- [27] Reenskaug T., "The Model-View-Controller (MVC) Its Past and Present", [http://heim.ifi.uio.no/~trygver/2003/javazone-jaoo/MVC\\_pattern.pdf](http://heim.ifi.uio.no/~trygver/2003/javazone-jaoo/MVC_pattern.pdf)
- [28] OWL-S: Semantic Markup for Web Services, <http://www.daml.org/services/owl-s/1.0/owl-s.html>
- [29] Eclipse: Ppen platform for tool integration, <http://www.eclipse.org/>

# Appendix

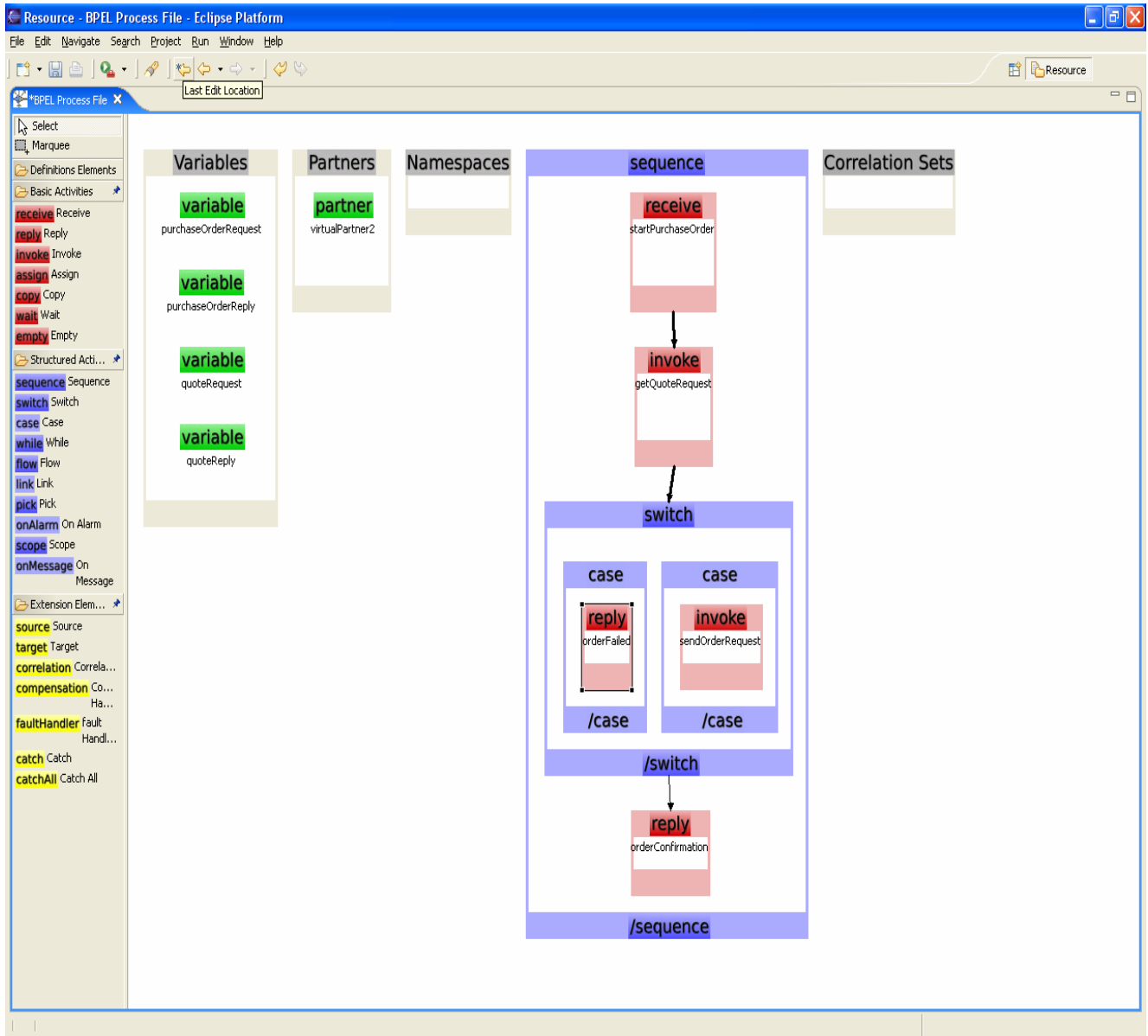


Figure 4. Snapshot of the BPEL Designer with Web process design in progress