



A CONJOIN APPROACH TO IOT AND SOFTWARE ENGINEERING USING TIBCO B.W

Tummala Bindu Madhavi ^{*1}, N.L.S.Sindhura ², M.Rithvik ³

^{*1, 2, 3} Assistant Professor, S.R.K. Institute of Technology, Vijayawada, India

Abstract:

This Paper represents a motivational approach of integration of two domains of two different streams and this paper constitutes the two parallel roads in one platform. Perhaps it's a challenging task to make this sort of Integrations but with the development of new tools like TIBCO B.W this has come to reality. This Paper is a moxie to represent an application into its physical level of representation to its virtual level of journey. In order to conjoin two topologies into a single topology this paper constitutes an involuntary approaches to make a small application of software which is already existed into reality. The future scope of this paper might be a successful application that turns an IOT application into the software boom which comes into reality.

Keywords: Conjoin; Moxie; Virtual; Physical; Elucidate; Eradicate; Evaluate.

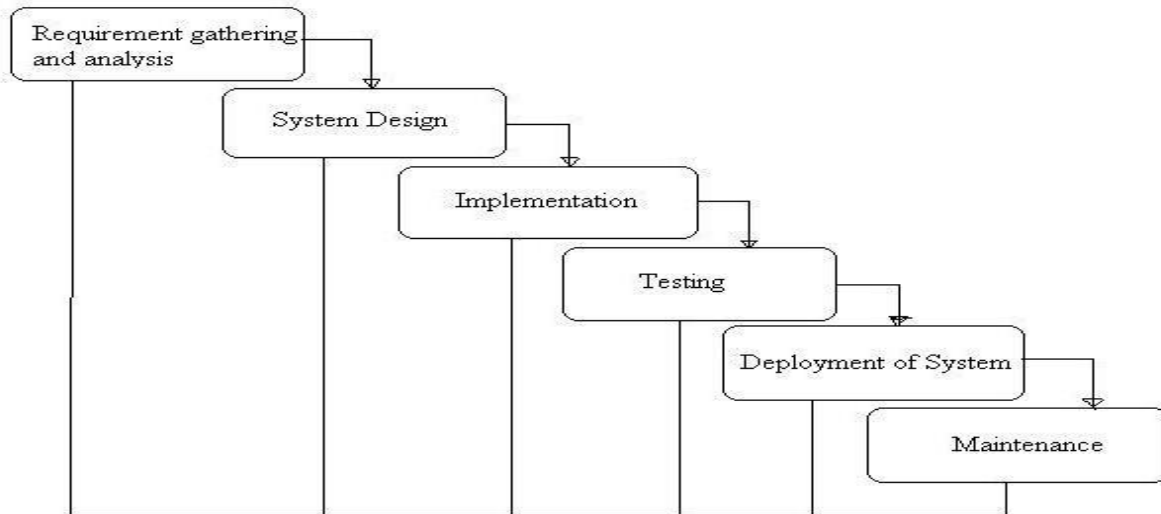
Cite This Article: Tummala Bindu Madhavi, N.L.S.Sindhura, and M.Rithvik. (2017). "A CONJOIN APPROACH TO IOT AND SOFTWARE ENGINEERING USING TIBCO B.W." *International Journal of Engineering Technologies and Management Research*, 4(11), 36-42. DOI: 10.5281/zenodo.1066790.

1. Introduction To Software Engineering

Software is more than just a program code. A program is an executable code, which serves some computational purpose. Software is considered to be collection of executable programming code, associated libraries and documentations. Software, when made for a specific requirement is called software product.

1.1. Waterfall Model

In this model software testing starts only after the development is complete. In **waterfall model phases** do not overlap.

General Overview of "Waterfall Model"

As waterfall model suggests various phases in the evaluation of software. This Integration is possible with those phases to make into an integrated technology that brings up a revolution in the field of software. This approach is going to start with a minute bit of taking the requirements. The requirement was to take this as an imitative to combine with other technologies that lapse its role.

And the appropriate coding and design was done at the physical level of an IoT Sensory device to make this combination.

2. Introduction To IOT

IOT the internet of things is a combo of several things into a single platform as sensory devices connective as all devices are combined at a single level a distributed service that reconstructs all the domains in one platform can be done with the help of a tool called TIBCO B.W that makes all technologies and all security issues to get resolved upto some extent. As devices are interconnected security can be resolved in a centralized server and this server can be passed to that server

3. Materials and Methods

The most important material used in this Conjoin approach to eradicate the errors that are occurred during the process of making this sort of combinational approach can be elucidated in the below diagram.

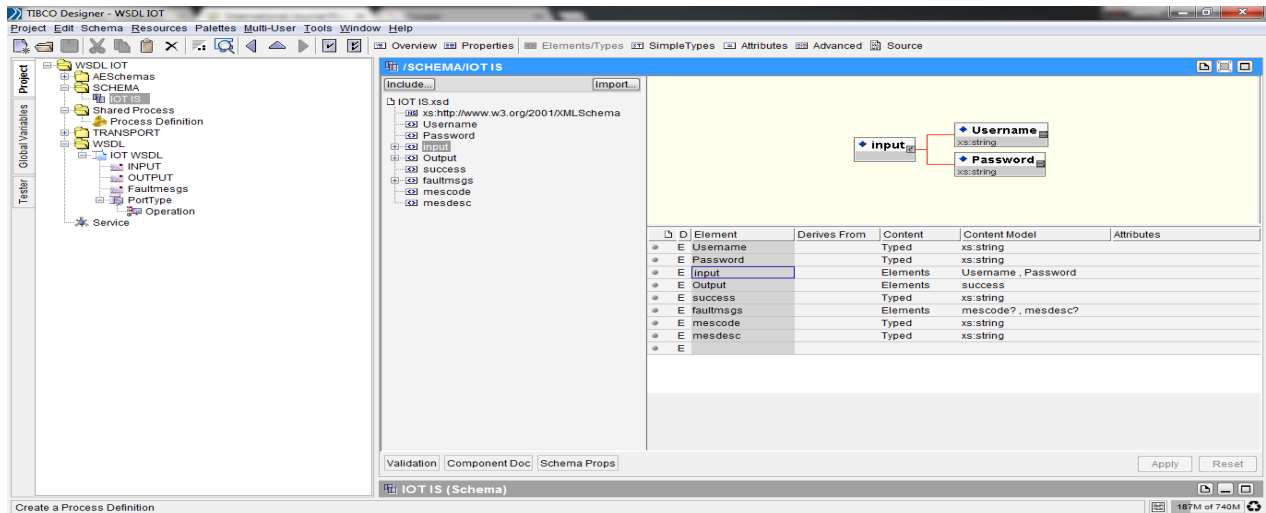
The first and primary thing that is going to be taken from the software engineering is that the requirement essentials that are essential to build a model or a design.

Even though there are few models that are evolved now a days Waterfall model being the oldest model to bring up this into a platform as a newest model in this conjoin approach. The tool that is

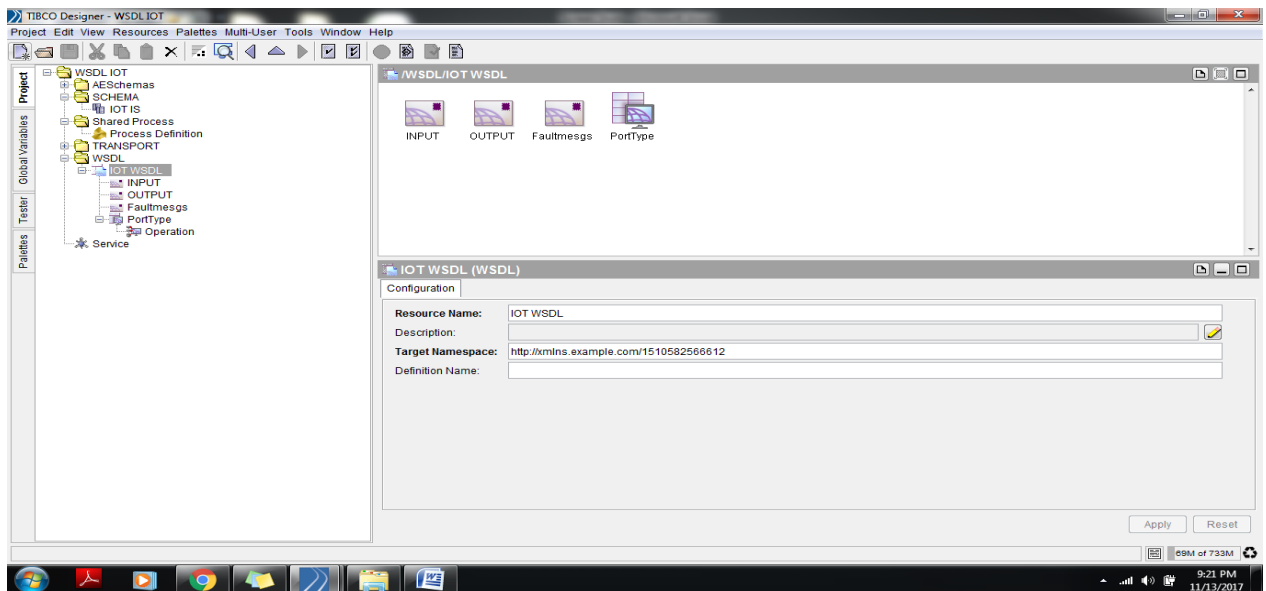
used to build this model is TIBCO B.W where Xpath functions play a key role in providing a service.

4. Results and Discussions

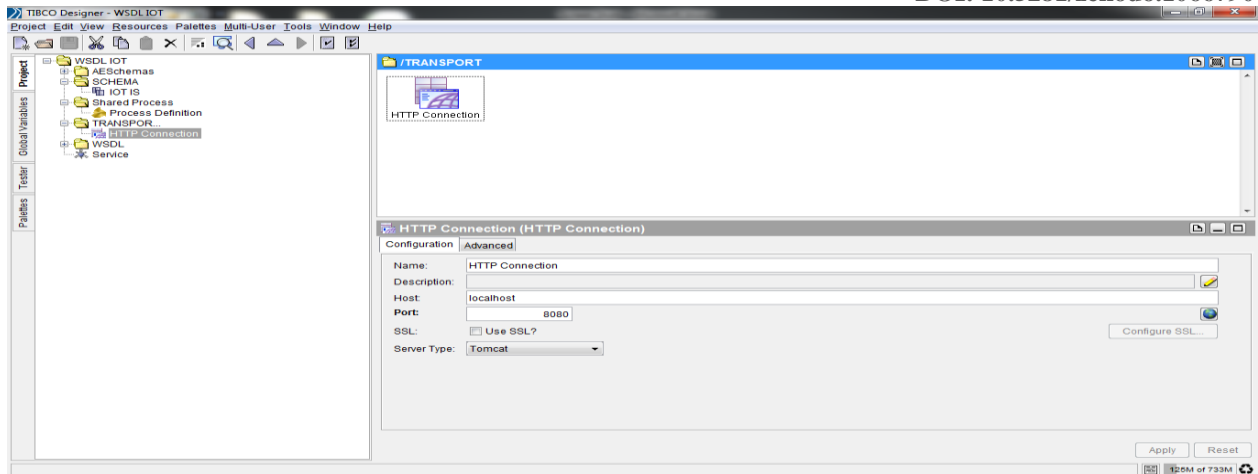
The below result explains the generation of a schema that takes an initial part of our design. The schema is defined with an input that constitute of a username and password and an output that constitutes of a result of input and output.



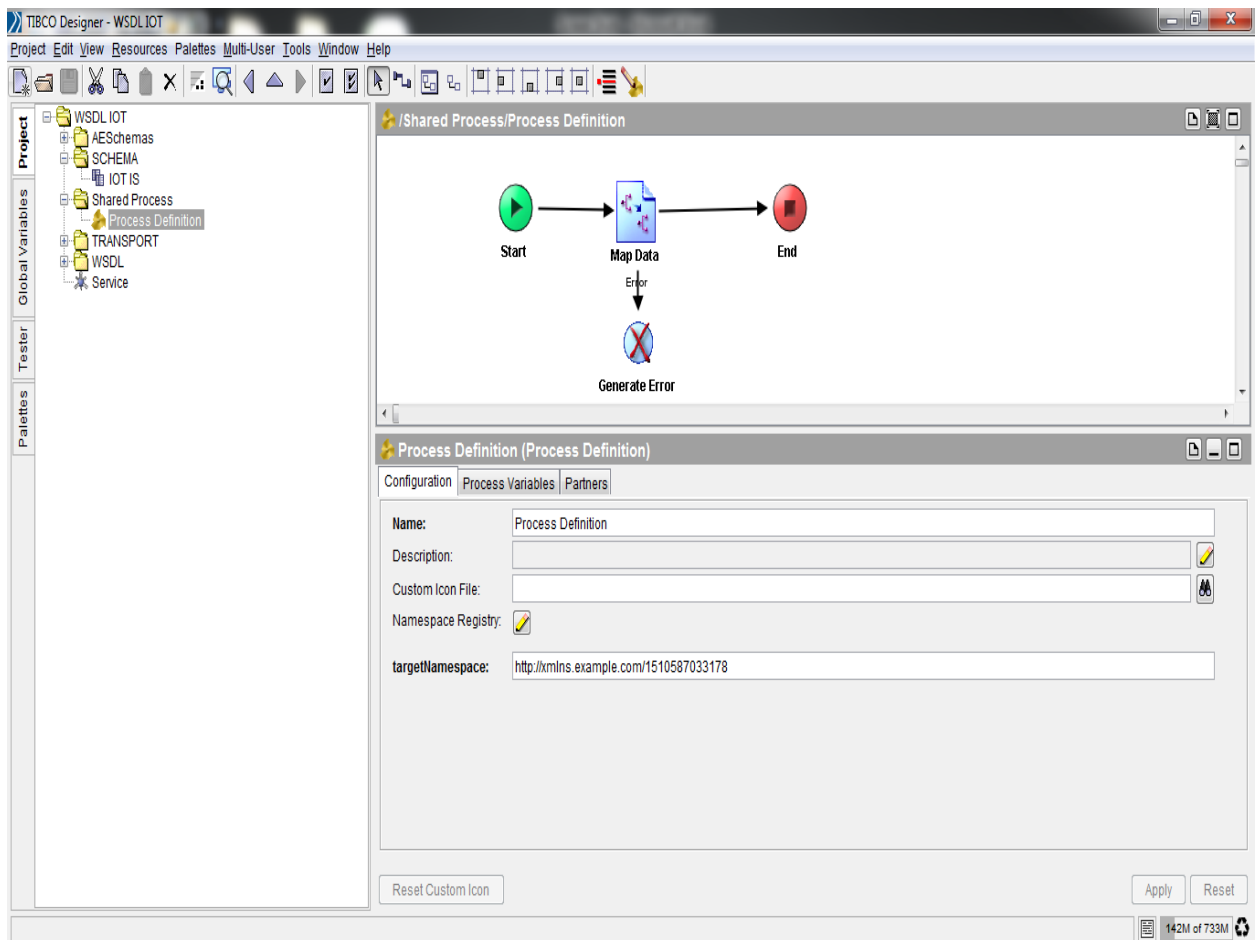
The below result explains about a WSDL file that is WEB SERVICE DESCRIPTOR LANGUAGE which constitute of a message port and the operation. This operation suggests a process



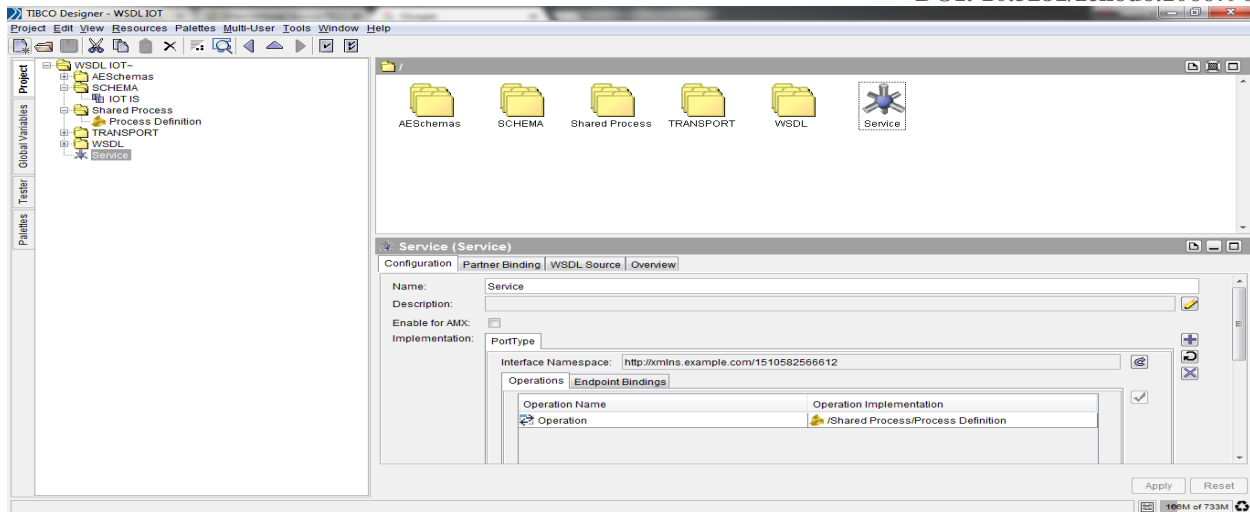
This figure shows the transport layer connections along with port numbers eg.8080.



This is the process that is exactly going to take place when the user gives a username and a password to make it into one as a contactination and thereby moving towards the application layer.



This is the final service that is generated by the addition of an operation with the WSDL files as a SOA Architecture thereby passing this service to all the domains.



WSDL File thus generated is as follows:

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<!--Created by TIBCO WSDL-->
```

```
<wsdl:definitions xmlns:tns="http://xmlns.example.com/1510585941559"
xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
xmlns:ns0="http://www.tibco.com/schemas/WSDL IOT/SCHEMA/Schema.xsd"
xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/" name="Untitled"
targetNamespace="http://xmlns.example.com/1510585941559">
```

```
<wsdl:types>
```

```
<xs:schema xmlns="http://www.tibco.com/schemas/WSDL
IOT/SCHEMA/Schema.xsd" xmlns:xs="http://www.w3.org/2001/XMLSchema"
targetNamespace="http://www.tibco.com/schemas/WSDL IOT/SCHEMA/Schema.xsd"
elementFormDefault="qualified" attributeFormDefault="unqualified">
```

```
<xs:element name="Output">
```

```
<xs:complexType>
```

```
<xs:sequence>
```

```
<xs:element ref="ns0:success"/>
```

```
</xs:sequence>
```

```
</xs:complexType>
```

```
</xs:element>
```

```
<xs:element name="Password" type="xs:string"/>
```

```
<xs:element name="Username" type="xs:string"/>
```

```
<xs:element name="faultmsgs">
```

```
<xs:complexType>
```

```
<xs:sequence>
```

```
<xs:element ref="ns0:mescode" minOccurs="0"/>
```

```

        <xs:element ref="ns0:mesdesc" minOccurs="0"/>
    </xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="input">
    <xs:complexType>
        <xs:sequence>
            <xs:element ref="ns0:Username"/>
            <xs:element ref="ns0:Password"/>
        </xs:sequence>
    </xs:complexType>
</xs:element>
<xs:element name="mescode" type="xs:string"/>
<xs:element name="mesdesc" type="xs:string"/>
<xs:element name="success" type="xs:string"/>
</xs:schema>
</wsdl:types>
<wsdl:service name="Service">
    <wsdl:port name="PortTypeEndpoint1" binding="tns:PortTypeEndpoint1Binding">
        <soap:address
location="http://localhost:8080/Service.serviceagent/PortTypeEndpoint1"/>
    </wsdl:port>
</wsdl:service>
<wsdl:portType name="PortType">
    <wsdl:operation name="Operation">
        <wsdl:input message="tns:INPUT"/>
        <wsdl:output message="tns:OUTPUT"/>
        <wsdl:fault name="fault1" message="tns:Faultmesgs"/>
    </wsdl:operation>
</wsdl:portType>
<wsdl:binding name="PortTypeEndpoint1Binding" type="tns:PortType">
    <soap:binding style="document"
transport="http://schemas.xmlsoap.org/soap/http"/>
    <wsdl:operation name="Operation">
        <soap:operation style="document"
soapAction="/Service.serviceagent/PortTypeEndpoint1/Operation"/>
        <wsdl:input>
            <soap:body use="literal" parts="input"/>
        </wsdl:input>
        <wsdl:output>
            <soap:body use="literal" parts="part1"/>
        </wsdl:output>
        <wsdl:fault name="fault1">
            <soap:fault use="literal" name="fault1"/>
        </wsdl:fault>
    </wsdl:operation>

```

```
</wsdl:binding>
<wsdl:message name="INPUT">
  <wsdl:part name="input" element="ns0:input"/>
</wsdl:message>
<wsdl:message name="OUTPUT">
  <wsdl:part name="part1" element="ns0:Output"/>
</wsdl:message>
<wsdl:message name="Faultmesgs">
  <wsdl:part name="part1" element="ns0:faultmsgs"/>
</wsdl:message>
```

5. Conclusions and Recommendations

This sort of approach being done in the physical layer as well as the network layer can be extended to a base of its application that is done in the IOT using the Aurdino boards this service can be programmed as an application in building an innovative model that makes everyone use this board and sensory devices in the provision of a high secured data.

Acknowledgements

We would like to thank our **Dr.D.Hatitha, Head of the Department, Department of C.S.E, and S.R.K. Institute of Technology** for providing us a path in the collection of this research.

We would like to thank all the faculty of **C.S.E Department, S.R.K. Institute of Technology** in making this visionary Project into a Paper.

References

- [1] TIBCO B.W Tutorials.
- [2] A Modernization approach to software engineering-M.Rithvik, Ch.Ambedkhar.
- [3] A Doctrinal approach to information security-IJRASET-T.Bindu Madhavi.

*Corresponding author.

E-mail address: rithvikmadugula@ gmail.com