

1 Augmented Chess: Assignment 9

Group Members

Jacob Holm Mortensen	jmorte14@student.aau.dk
Martin Raunkjær Andersen	marand13@student.aau.dk
Thomas Gwynfryn McCollin	tmccoll14@student.aau.dk

1.1 Question 1

Consider a Service Oriented Architecture as described in "Introduction to Service Oriented Architecture" for your Web application. Which services would be used by your Web application under this architecture? (identify modular components of your Web application). Which advantages would bring to implement your application based on services? (Does your Web application share any of the goals in Section 1.1.1 or would be enhanced by considering these goals?)

If we were to use a SOA, our architecture would likely look like this:

Where the **Presentation** service would consist of the chessboard and the presentation logic required to dynamically redraw the board when pieces are moved, as well as a communication layer that would contain an open http connection with the rest of the webservice, allowing the presentation layer to request and receive dynamic response from the logic layer.

The **Communication** service would keep track of a list of persistent http **sockets**, corresponding to each user currently using the presentation layer. Whenever a user performs a move in the presentation layer, a request goes to this layer, calling the **Game Logic** service for validation of the move and returning the current state of the game.

The **Game Logic** receives moves and computes the new state of the game, writing it to the **Data** service in the process.

The **Data** service stores the state of the games in progress.

Since our application is a game, and therefore susceptible to cheating if game logic is available to the user, it would make a great deal of sense to move all game logic (such as validity of actions) into services, to prevent users from tampering with data. That would also create the possibility of other collaborators creating a frontend of their choosing for our service. The common goals of SOA, such as

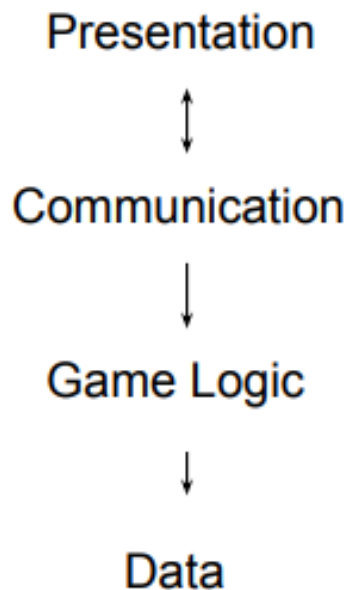


Figure 1.1: SOA for Augmented Chess

increased reusability and vendor diversification is not as important to us, as our "product" requires very specialised logic, not commonly found from vendors. Although interchangeability of the database might be useful.

1.2 Question 2

Select one of the services mentioned as answer to the previous question. Write a valid SOAP request for this service and a valid SOAP response.

A soap request to the **Game Logic** could look something like Listing 1.1, where a move of a piece is requested, and the board coordinates of the origin and destination of the move is passed as parameters.

Listing 1.1: "SOAP request"

```

<?xml version="1.0" ?>
<SOAP-ENV:Envelope
  SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
  xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/"
  xmlns:xsi="http://www.w3.org/1999/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/1999/XMLSchema">
  <SOAP-ENV:Body>
    <ns1:performMove
      xmlns:ns1="urn:GameLogic">

```

```

        <OrigCoordX xsi:type="xsd:int">1</OrigCoordX>
        <OrigCoordY xsi:type="xsd:int">1</OrigCoordY>
        <DestCoordX xsi:type="xsd:int">2</DestCoordX>
        <DestCoordY xsi:type="xsd:int">2</DestCoordY>
    </ns1:performMove>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>

```

While a response could be formed as in 1.2. The response then specifies whether or not the requested move was valid and therefore have been performed by the returned bool.

Listing 1.2: "SOAP response"

```

<?xml version="1.0" encoding="UTF-8" ?>
<SOAP-ENV:Envelope
  xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsi="http://www.w3.org/1999/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/1999/XMLSchema">
  <SOAP-ENV:Body>
    <ns1:performMoveResponse
      xmlns:ns1="urn:GameLogic"
      SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/
        soap/encoding/">
      <validMove xsi:type="xsd:bool">True</validMove>
    </ns1:performMoveResponse>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>

```

1.3 Question 3

Provide a WSDL description for one of the services mentioned in the answer to question 1.

A WSDL representation of the **Game Logic** service could be described as in listing 1.3.

Listing 1.3: "WSDL representation of Game Logic service"

```

<definitions name = "GameLogic"
  targetNamespace = "http://www.examples.com/wsdl/GameLogic.wsdl"
  xmlns = "http://schemas.xmlsoap.org/wsdl/"
  xmlns:soap = "http://schemas.xmlsoap.org/wsdl/soap/"
  xmlns:tns = "http://www.examples.com/wsdl/GameLogic.wsdl"
  xmlns:xsd = "http://www.w3.org/2001/XMLSchema">

  <message name = "PerformMoveRequest">
    <part name = "OrigCoordX" type = "xsd:int"/>
    <part name = "OrigCoordY" type = "xsd:int"/>
    <part name = "DestCoordX" type = "xsd:int"/>
    <part name = "DestCoordY" type = "xsd:int"/>
  </message>

```

```

</message>

<message name = "PerformMove">
  <part name = "return" type = "xsd:bool"/>
</message>

<portType name = "PerformMove_PortType">
  <operation name = "PerformMove">
    <input message = "tns:PerformMoveRequest"/>
    <output message = "tns:PerformMoveResponse"/>
  </operation>
</portType>

<binding name = "PerformMove_Binding" type = "tns:PerformMove_PortType">
  <soap:binding style = "rpc"
    transport = "http://schemas.xmlsoap.org/soap/http"/>
  <operation name = "PerformMove">
    <soap:operation soapAction = "PerformMove"/>
    <input>
      <soap:body
        encodingStyle = "http://schemas.xmlsoap.org/soap/encoding/"
        namespace = "urn:examples:GameLogic"
        use = "encoded"/>
    </input>

    <output>
      <soap:body
        encodingStyle = "http://schemas.xmlsoap.org/soap/encoding/"
        namespace = "urn:examples:GameLogic"
        use = "encoded"/>
    </output>
  </operation>
</binding>

<service name = "GameLogic">
  <documentation>WSDL File for GameLogic</documentation>
  <port binding = "tns:PerformMove_Binding" name = "PerformMove_Port">
    <soap:address
      location = "http://www.examples.com/PerformMove/" />
  </port>
</service>
</definitions>

```

The real **Game Logic** service should of course be able to do more than compute moves, but this serves as an adequate example.

1.4 Question 4

Write the request and response provided as answer to question 2 in the case of RESTful services.

The request written in REST would like in Listing 1.4. This would be a quite simple service because only one game could be played at a time

Listing 1.4: "REST representation of Game Logic request"

`https://www.augmentedchess.com/gamelogic/performmove?origcoor dx=1&origcoor dy=1&`

The response would be a simple JSON object containing one value, which could be true as in Listing 1.5.

Listing 1.5: "REST representation of Game Logic reponse"

```
{
  "validMove": true
}
```

1.5 Question 5

Evaluate the support provided by the framework you are using to develop your Web application. In case the support is enough, develop two simple Web services (one RESTful and one SOAP) using it. In case the support is not enough, consider using another framework (for example, <https://spring.io/> or <https://cxf.apache.org/>) to develop the Web services.

Our serverside framework is NodeJs. A REST web service implementation in Node would look like what is shown in Listing 1.6. The service can only return the array of users upon a GET request to `https://www.host.com/users`.

Listing 1.6: "Simple REST service in Node with the Express library"

```
var express = require('express');
var app = express();
var users = ["Thomas", "Jacob", "Martin"];

app.get('/users', function (req, res) {
  res.end(users);
})

var server = app.listen(80, function () {
  console.log("Example app listening on port 80")
})
```

A SOAP web service implementation in Node would look like this:

Listing 1.7: "SOAP service in Node with the Soap library"

```
var soap = require('soap');
var url = 'http://example.com/wsdl?wsdl';
var args = {name: 'value'};
soap.createClient(url, function(err, client) {
  client.MyFunction(args, function(err, result) {
    console.log(result);
  });
});
```

1.6 Question 8

Continue working in your Web application. Report the status of the development.

With the application it is currently possible to play a game of augmented chess with predefined armies on a single computer. Though it is possible to manage a users armies we still need a small implementation to retrieve said armies when starting a game.