CSC541 Project
Data Structures and Algorithms for SOA-based enterprise configuration management

1 Project Summary

1.1 Brief description

Large SOA-based [1] enterprise applications are generally dynamic, complex, multi-tiered, distributed across multiple machines (e.g., a data center). Management of such applications is a challenge. XML-based [2] configuration files are widely used in the management of such applications. Examples of specific management tasks we will be investigating in this project include deployment of applications, selective restart of applications after a reboot and capacity planning. Such tasks require efficient querying, transformation and persistence of configuration data; they are among the main goals of any enterprise management tool.

1.2 Objectives

The main objective of this project is to give you hands-on experience with tree data structures (implemented as XML documents) and (in-memory) searching algorithms (as they are used in querying, transformation and persistence of XML documents). A secondary objective, regarding continuation of this project in a separate course will be further explained in class.

1.3 Background reading

An introduction to enterprise service applications and their management can be found in [3] and in [4]. This background will help you understand the context of the project but is not required. Since the configuration data is encoded in XML, an understanding of it is essential. For an XML primer, see [2].

2 Managing appliances in a multi-tiered data center

2.1 The actual data center environment

The data center environment employed at NCState is shown in Figure 1. It contains a Cisco firewall, 3 IBM Datapower appliances, a switch fabric consisting of two Cisco Catalyst switches and around 10 servers[1]. The datacenter environment configuration that you will be analyzing in this project, is a much larger enterprise but modeled similarly.

Figure 1: Our "data center" playground, Fall 2013.

2.1.1 Service endpoints, deployment policies, domains and managed sets

The following concepts will be useful in understanding the operation of the data center and the scope of the management tasks.

Service endpoints: Service end-points are software components of an application whose functionality is exposed via an open interface (typically XML serialized), over the network. Method calls for the application, from a remote client, running on a different platform is thus possible. In the context of modern enterprises, service-endpoints provide functionality for the enterprise traffic like authentication, validation, encryption, firewall protection and so on.

[1] OK, two switches do not make a realistic data center switch fabric and these are not real servers, they are PCs. But then, this is a poor professor's data center...

Deployment Policies: Typically, a business application may require/contain more than one service endpoints (e.g., authentication plus validation). From the management perspective we consider in this project, multiple service-endpoints are bundled into one or more deployment policies. Such a policy simplifies the management task; for example, it reduces the configuration workload.

Domains: In the current configuration, deployment policies are organized into domains in the pre-processing devices (DPDevices) to enable administrator compartmentalization. This is a logical isolation.

Managed sets: DPDevices catering to functionality required by multiple collaborating business partners in a datacenter and are organized into pools called managed sets.

2.1.2 Configuration policies

Configuration policies are specific to the administrator of the datacenter and are largely driven by business needs, traffic patterns, application entities and architectures. In the example provided, the datacenter is driven by the use of a large number of Datapower devices which provide a high-performance pre-processing tier. The ability to move deployment policies from one domain/device to another is driven by business needs (typically collaborating businesses), adaptability to traffic patterns that requires the servers to be started and shut dynamically and service migration across servers, resulting in the configuration data with the schema as provided.

2.2 The logical configuration

Our focus in this project will be on the following management tasks:

1. Deployment of a service endpoint.
2. Alteration of management parameters.
3. Enabling external applications with no XML support, access and query the fairly structured enterprise configuration data.

For the purposes of accomplishing these management tasks, the logical abstraction depicted in Figure 2 will be sufficient.

2.2.1 Specifying the logical configuration via XML

This logical configuration can described in XML as the following XML document exemplifies:

```
 <?xml version="1.0" encoding="UTF-8" ?>
<dat:DPManager xmi:version="2.0" VersionsStoredLimit="30"
         xmi:id="WAMTManager_0"
         xmlns:dat="http://www.obfuscated/datapower/lfs/schemas/7.0/
         datapowermgr.xmi" xmlns:xmi="http://www.omg.org/XMI">

  <devices xmi:id="DPDevice_0" deviceType="XI50" GUIPort="50080" HLMPort="5550"
         currentAMPVersion="1.0" quiesceTimeout="60"
```

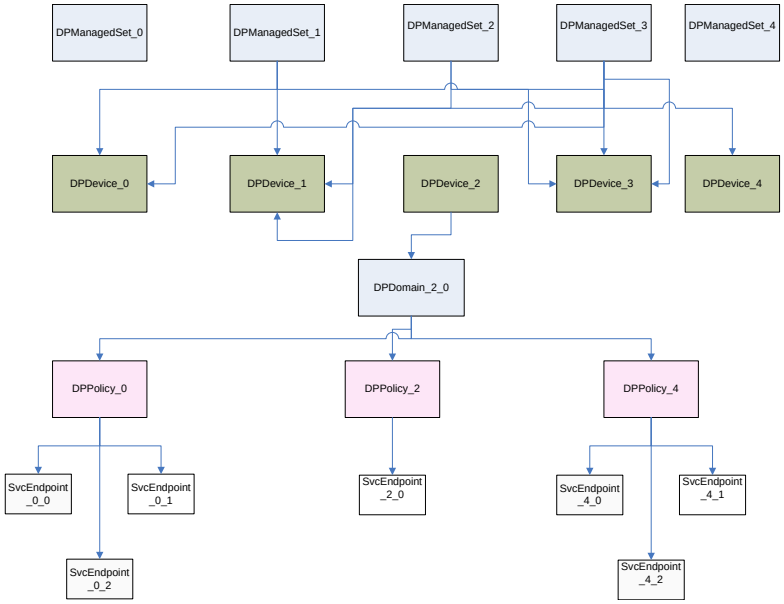

Figure 2: Service domains and policies.

```
        featureLicenses="MQ;TAM;DataGlue;JAXP-API;PKCS7-SMIME;SQL-ODBC;Tibco-EMS;WebSphere-JMS;" />

  <devices xmi:id="DPDevice_1" deviceType="XC10" GUIPort="50080" HLMPort="5550"
        currentAMPVersion="1.0" quiesceTimeout="60"
        featureLicenses="MQ;TAM;DataGlue;JAXP-API;PKCS7-SMIME;SQL-ODBC;Tibco-EMS;WebSphere-JMS;" />

  <devices xmi:id="DPDevice_2" deviceType="XI50" GUIPort="50080" HLMPort="5550"
        currentAMPVersion="1.0" quiesceTimeout="60"
        featureLicenses="MQ;TAM;DataGlue;JAXP-API;PKCS7-SMIME;SQL-ODBC;Tibco-EMS;WebSphere-JMS;">

<domains xmi:id="DPDomain_ 2_0" highestVersion="0" SynchDate="0"
        OutOfSynch="true" quiesceTimeout="60" SyncMode="MANUAL">

<deploymentPolicy xmi:id="DPPolicy_2" highestVersion="0" SynchDate="0" policyType="NONE">
  <serviceend-point xmi:id="SvcEndpoint_2_0" type="webserviceproxy" operation="none"
        port="30000" targetserver="anonymizedendserver_2578" />
</deploymentPolicy>

<deploymentPolicy xmi:id="DPPolicy_4" highestVersion="0" SynchDate="0" policyType="NONE">
  <serviceend-point xmi:id="SvcEndpoint_4_0" type="webserviceproxy" operation="none"
        port="30000" targetserver="anonymizedendserver_978" />
  <serviceend-point xmi:id="SvcEndpoint_4_1" type="xml-firewall" operation="none"
        port="30001" targetserver="anonymizedendserver_8300" />
  <serviceend-point xmi:id="SvcEndpoint_4_2" type="webapplicationfirewall" operation="none"
        port="30002" targetserver="anonymizedendserver_3573" />
</deploymentPolicy>

<deploymentPolicy xmi:id="DPPolicy_0" highestVersion="0" SynchDate="0" policyType="NONE">
  <serviceend-point xmi:id="SvcEndpoint_0_0" type="webserviceproxy" operation="none"
        port="30000" targetserver="anonymizedendserver_7121" />
  <serviceend-point xmi:id="SvcEndpoint_0_1" type="webapplicationfirewall" operation="none"
        port="30001" targetserver="anonymizedendserver_6939" />
```

```
  <serviceend-point xmi:id="SvcEndpoint_0_2" type="multiprotocolgateway" operation="transformation"
         port="30002" targetserver="anonymizedendserver_8663" />
</deploymentPolicy>

<deploymentPolicy xmi:id="DPPolicy_2" highestVersion="0" SynchDate="0" policyType="NONE">
  <serviceend-point xmi:id="SvcEndpoint_2_0" type="webserviceproxy" operation="none"
         port="30000" targetserver="anonymizedendserver_2578" />
  </deploymentPolicy>
  </domains>
  </devices>

  <devices xmi:id="DPDevice_3" deviceType="XI50" GUIPort="50080" HLMPort="5550"
      currentAMPVersion="1.0" quiesceTimeout="60"
      featureLicenses="MQ;TAM;DataGlue;JAXP-API;PKCS7-SMIME;SQL-ODBC;Tibco-EMS;WebSphere-JMS;" />
  <devices xmi:id="DPDevice_4" deviceType="XC10" GUIPort="50080" HLMPort="5550"
      currentAMPVersion="1.0" quiesceTimeout="60"
      featureLicenses="MQ;TAM;DataGlue;JAXP-API;PKCS7-SMIME;SQL-ODBC;Tibco-EMS;WebSphere-JMS;" />
  <managedSets xmi:id="DPManagedSet_0" devicemembers="" />
  <managedSets xmi:id="DPManagedSet_1" devicemembers="device_1,device_0,device_3," />
  <managedSets xmi:id="DPManagedSet_2" devicemembers="device_1,device_3," />
  <managedSets xmi:id="DPManagedSet_3" devicemembers="device_3,device_4,device_0,device_1," />
  <managedSets xmi:id="DPManagedSet_4" devicemembers="" />
  </dat:DPManager>
```

2.3 Deliverables on the required background

The two deliverables listed in this section give you credit for the effort you will put in understanding the environment and the management application. The specific questions will be determined once we get feedback from the class on your XML expertise. The credit per deliverable will be announced on the course web site.

Deliverable 1. Understanding XML, the language for defining the enterprise configuration.
A configuration file edgeconfig.xml will be assigned to each team (different for each team and accessed via google sites, provided once you have chosen your project and have been divided into teams). For the edgeconfig.xml file, the following questions will need to be answered:

1. A question on the concept of XML namespaces and how they are relevant to the management application.
2. A question on the concept of XML attributes and how they are relevant to the management application .
3. A question on defining a service domain, with different configuration parameters, for a different management task (e.g., long-term capacity planning).

Specific questions are not given as they will be different for each team.

Deliverable 2. Understanding the enterprise configuration.
Consider the configuration file edgeconfig.xml (assigned to each team). Also assigned is a sample configuration and the graphical representation of the enterprise according to the sample configuration file for reference.

1. How many services are deployed?

2. How many appliances are there?

3. How many policies are there?

4. Provide a graphical representation of the components of the enterprise as described in the file, similar to the one depicted in Figure 2.

5. The above XML document is a tree data structure; what is the depth of that tree? how many leaf nodes are there?

3 Analysis and Implementation

In answering the following, no graphical interface is required. You may use any language you are comfortable with. We strongly recommend that you use Jazzhub [5] as your programming development environment. Email kboloor@ncsu.edu for support.

The configuration files (e.g., edgeconfig2.xml) will be different for different teams and randomly generated. The questions for each deliverable will be related to the edgeconfig2.xml file given to each team.

Deliverable 3. Study the configuration defined in the file egdeconfig2.xml assigned to you. References [6] and [7] provide an introduction to XML querying in java and .net platforms respectively.

1. Write a program to analyze the file and answer the questions assigned in file questionsdeliverable3 (also different for each team).

Deliverable 4. Data persistence in relational form. The focus here is on querying persisted relational data. References [8] and [9] are some examples.

1. For the file edgeconfig2.xml assigned to you write a program to represent the configuration in a relational database of your choice.
2. Write a program to Query the data in the database and answer the questions assigned in file questionsdeliverable4 whenever an xml file (with the same schema as edgeconfig2.xml) is input.

Some example questions that can be included in the questionsdeliverable3 and questionsdeliverable4 file assigned to you are:

1. How many DPDevices are present in the schema ?
2. How many unique DPPolicies are applied in all to the schema?
3. How many Service Endpoints are present in DPPolicy_0?
4. Which DPDevice has service endpoints with target servers anonymizedendserver_2578?
5. Which DPPolicy has been deployed in most DPDevices?
6. Which DPDevice has the highest number of DPPolicies applied?
7. Which domain in DPDevice_3 has service-endpoints targeting server anonymizedendserver_7121?
8. Which DPDevice has the maximum number of multiprotocolgateway with operation transformation applied?

Deliverable 5. Searches in and transformations of the configuration file. Reference [10] is an example.

Example transformations to be performed are as follows:

1. For the file edgeconfig2.xml provided to you, write a program to transform the file according to the transformation.txt provided.

2. Remove DPDomain_2_0 and the containing deployment policies from DPDevice_2.
3. DPDevice_3 is down, remove it from the config file.
4. Remove all serviceend-point with type "webapplicationfirewall" from the config file.
5. Change port for all service-endpoints with targetserver=anonymizedendserver_8663 to 30020.
6. Change all devices with type XC10 to XI50
7. All service-endpoints that are of type=xmlfirewall to type=webapplicationfirewall

References

[1] http://www.developer.com/services/article.php/1010451/Service-Oriented-Architecture-Introduction-Part-1.htm

[2] http://www.ibm.com/developerworks/xml/tutorials/xmlintro/

[3] http://msdn.microsoft.com/en-us/library/bb841464.aspx

[4] http://www-01.ibm.com/software/integration/wamc/

[5] http://storage.networksasia.net/content/ibm-offers-jazz-hub-students-free

[6] http://www.ibiblio.org/xml/books/xmljava/chapters/ch05.html

[7] http://msdn.microsoft.com/en-us/magazine/cc302158.aspx

[8] http://javaboutique.internet.com/tutorials/mapping/

[9] http://javaboutique.internet.com/tutorials/mapping/

[10] http://saxon.sourceforge.net/