We present a framework that integrates XML and Java RMI technologies to support the open architecture for Web-based Decision Support Systems (DSS), where multiple source data and heterogeneous decision models are independent from each other, and yet openly interchangeable in a distributed network.

From a system design point of view, Web-based DSS consists of three main components:

- **Data Management** – collects and pre-processes a large number of source data to provide historical and summarized information for analytical or decision making purpose.
- **Decision Models Management** – maintains and operates heterogeneous decision models residing locally and/or remotely in the network.
- **User Interface** – provides user-specific requirements for selecting appropriate decision models, and presents to the user the results from the decision making process.

Since our design handles multiple source data, in the implementation, we define a simple XML Document Type Definition (DTD) to specify the source data obtained from various Web sites. This file is referenced by the XML Converter whenever it needs to translate source data into XML format.

**A Sample XML DTD for Stocks:**

```xml
<ELEMENT SimpleStock(country,exchange,(stock=*))>
  <ELEMENT country CDATA>
  <ELEMENT exchange CDATA>
  <ELEMENT stock(date,price,name?,volume?,turnover?,dividend?)>
    <!ATTLIST stock symbol ID #REQUIRED>
  </ELEMENT date CDATA>
  <ELEMENT price CDATA>
  <ELEMENT name CDATA>
  <ELEMENT volume CDATA>
  <ELEMENT turnover CDATA>
  <ELEMENT dividend CDATA>
</ELEMENT>
```

**Data preparation for distribution:**

All input/output parameters are packed as simple vector of DataPacket class before distribution.

```java
public Class DataPacket implements Serializable {
  public String name;
  public String type;
  public Vector value_array;
}
```

**Java RMI Codes for Remote Model Execution:**

```java
public class RemoteModel implements RemoteModelInterface {
  public Vector execute(Vector input) throws RemoteException {
    // Remote model execution code...
    return outputVector;
  }
}
```

**Figure 1. Open Architecture for Web DSS**

**Figure 2. Two Approaches for Distributed Models**

In first approach, each model is treated as a single executable module. The second approach splits the models into distributed parts:

(a) Easy to code, simple to implement; Java only environment, platform-independent

(b) Platforms- and languages-independent; Need to deal with interoperability between two parties.