**Document Based Applications**

**C.D. jones**

The goal of document based applications, Doc-Apps, is to be able to generate an entire application by creating one schema. The process would theoretically take a requirements document, convert it to the Doc-Appp structure, which would then retrieve the necessary application submodules and configure them in the proper way to have a fully functional application, cutting down development time, and lowering costs tremendously.

The submodules would be retrieved from a sort of app-store type system and put together very similar to how a puzzle pieces form a puzzle. Doc-Apps are a part of something I call the Puzzle-Piece OS, an OS generated in the Doc-App manner and easily customizable by using the switchable puzzle pieces (submodules/modules).

Document Based Applications are dynamic. Even for an operating system they remain dynamic. Acting like a descriptor, the document(s) describing the application is updated and maintained in real-time without recompilation. The way it performs this task is by using a technology similar to web services, allowing foreign applications to communicate with each other. In this case, the foreign applications are the modules as dictated by the Doc App.

Module

From www.module.com

Module

My Schema Doc-App

Module

Module from Cloud

Foo.exe

Modules can be registered and can be pulled from the internet or the cloud via external links.

**Very Simple Example of Doc-Apps**

**//Data Types**

//Default Built In Data Types

<connectionstream datatype\_module URL=”home://Integer.xsd”/>

<connectionstream datatype\_module URL =”home://String.xsd”/>

//A custom data type (i.e. a Java Class)

<connectionstream customdatatype\_module URL=”home://MyClass.xsd”/>

//A custom module (i.e. a local custom Doc-App to connect to )

//given any name (hence randommodulename)

<localconnection randommodulename FILE=”EncryptionClass.xsd”/>

//A custom module (not in the default package)– like above but remote, not local

<connectionstream somemodulename URL=”docapp://www.docappstandards/List.xsd”/>

**Algorithms**

//load binary search tree algorithm from home

<connnectionstream algorithm\_module URL=”home://displayBST.xsd/>

**Custom Code**

//this is embedded in the doc app

<algorithm newdata name=”new\_alg” >

for (int i=0;i<100;i++)

{

array[i] = 10;

}

</algorithm>