김일성종합대학학보 JOURNAL OF KIM IL SUNG UNIVERSITY
(자연과학) (NATURAL SCIENCE)
주체103(2014)년 제60권 제6호 Vol. 60 No. 6 JUCHE103(2014).

A Method of Implementing SWF Player Web Plugin using XML

Kim Yong Sim, Yun Hui Gwang

The great leader Kim Jong Il said:

"What is essential in this regard is to develop programs of our own style."

The SWF (Shock Wave Flash) file format delivers vector graphics, text, video, and sound over the Internet. Designed to be an efficient delivery format, the SWF file format requires the implementation of SWF Player plugin to play SWF files on Mozilla Firefox.

Mozilla provides plugin interfaces to help the operation of programmes with the web browser.

Our SWF Player plugin uses Mozilla plugin API(NPAPI) so as to enable Mozilla web browser to play SWF files. For SWF player to run on web browser, it is necessary to implement the plugin modules which communicate with the player.

In this paper we proposed a method of implementing plugin modules for the communication of web browser and the player using XML .

1. Implementing Web Plugin using XML

The plugin API functions implemented in the SWF player are listed below.

NS_PluginInitialize: This function gets called when the plugin is loaded.

NS_NewPluginInstance: This instantiates a new object for the browser. Returning a pointer to the C++ plugin object is what ties the C++ and C emphasis parts of the API together.

NS_DestroyPluginInstance: This destroys our instantiated object when the browser is off.

NS_PluginShutdown: This is called when a plugin shuts down.

NPP_GetMIMEDescription: This is called to get the MIME types supported by the plugin.

NS_PluginGetValue : This is used by Firefox to query information from the plugin, like the supported MIME type, the version number, and a description.

SWF Player plugin algorithm on the web browser is shown below.

① When plugin is loaded, NS_PluginInitialize function is called and it carries out initialization process.

② NS_NewPluginInstance function is called when plugin object is created, and at this time xid(X Window ID) value is sent through the parameter.

③ The plugin sends xid value to the SWF player.

④ The player checks whether it is running in standalone mode or plugin mode by checking xid value.

⑤ If xid value is set(nonzero), it is supposed that the player is started by the web plugin. We start the player by subclassing QX11EmbedWidget with the xid instead of QMainWidget.

⑥ The QX11EmbedWidget object is embedded into web browser and plays swf files. Specifying xid value into the parameter of embedInto function in QX11EmbedWidget class, SWF player is embedded into web browser and plays.

2. The Definition of XML Data Structure

The events occurring in the web browser should be sent to the player. We used socket for inter-process communication between the plugin and the player, and XML as the communication data.

The advantage of using XML is that XML can store and organize just about any kind of information in a form that is tailored to your needs. [2] For our future extension of communication data structure and flexibility, we choose XML as communication data.

Data communication structure is as follows.

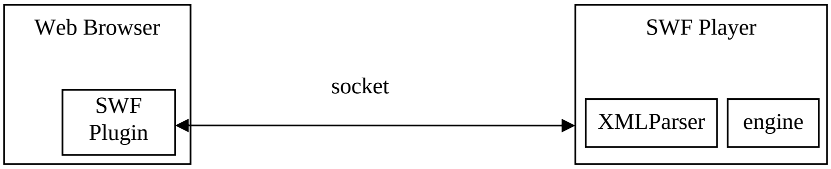

Fig. 1. Communication structure between SWF player and web plugin

In this paper, we defined XML of action codes like below.

```
<pre>
  <invoke name=" function name"   returntype=" return value type">
    <arguments>
       [<type > variable name </ type>
       [<type > variable name </ type >, …]]
    </arguments>
  </invoke>
</pre>
```

Conclusion

In this paper, we proposed a method of building web plugin module and communication structure between web plugin and the SWF player using XML.

References

[1] John Shapley; Inter-Process Communication in Linux: The Nook & Crannies, Prentic-Hall, 160～174, 2008.

[3] Alexander Stoyen et al.; Using XML for Interprocess Communication in a Space Situational Awareness and Control Application, Focal, 80～96, 2003.