TANK INFORMATION SYSTEM (TIS): A CASE STUDY IN MIGRATING WEB MAPPING APPLICATION FROM FLEX TO DOJO FOR ARCGIS SERVER AND THEN TO OPEN SOURCE

Bhaskar Reddy Pulsani [a]

[a] Telangana State Remote Sensing Applications Centre, 8th Floor, B Block, Swarnajayanti Complex, Ameerpet-38, India
pulsani@yahoo.co.in

KEY WORDS: Open Source, JavaScript, Flex, Dojo Toolkit, ArcGIS, PostgreSQL.

ABSTRACT:

Tank Information System is a web application which provides comprehensive information about minor irrigation tanks of Telangana State. As part of the program, a web mapping application using Flex and ArcGIS server was developed to make the data available to the public. In course of time as Flex be-came outdated, a migration of the client interface to the latest JavaScript based technologies was carried out. Initially, the Flex based application was migrated to ArcGIS JavaScript API using Dojo Toolkit. Both the client applications used published services from ArcGIS server. To check the migration pattern from proprietary to open source, the JavaScript based ArcGIS application was later migrated to OpenLayers and Dojo Toolkit which used published service from GeoServer. The migration pattern noticed in the study especially emphasizes upon the use of Dojo Toolkit and PostgreSQL database for ArcGIS server so that migration to open source could be performed effortlessly. The current ap-plication provides a case in study which could assist organizations in migrating their proprietary based ArcGIS web applications to open source. Furthermore, the study reveals cost benefits of adopting open source against commercial software's.

1. INTRODUCTION

Irrigation & CAD department of Telangana state is responsible for planning, Designing, Construction and maintenance of major, medium and minor dams across the state. Irrigation, carried out through provision of water is classified into three major categories namely major irrigation (cultivable command area above 10,000 ha.), medium irrigation (cultivable command area between 2000 ha to 10,000 ha) and minor irrigation (cultivable command area less than 2000 ha). Further, Erstwhile panchayat raj tanks i.e. tanks catering to cultivable command area less than 40 ha also come under the category of Minor Irrigation.

Minor irrigation plays an important role in the development of agricultural production and ensures food security in drought prone areas and also those outside the command area of major and medium projects. Therefore, to monitor minor irrigation tanks, a comprehensive list of tanks database along with delineation of minor irrigation tanks was undertaken by the erstwhile Andhra Pradesh State Remote Sensing Applications Centre (APSRAC) in collaboration with I & CAD department as part of Memorandum Of Understanding (MOU) signed between both the organizations. Furthermore, a website i.e., "Tank Information System" has been developed to display, query and analyze the collected information.

The ease of usability of a web application to the end user depends upon the choice of programming languages. Functioning of the web applications without any installation of plug-ins from the end user makes it all the more convenient. From this perspective, our goal is to assert the importance of Dojo Toolkit over Flex for ArcGIS Server and to emphasize upon its use for effortless migration of the proprietary based web application to open source. This is demonstrated by ponverting an already existing web mapping application built using Flex to JavaScript based Dojo Toolkit application. Further, the Dojo based JavaScript application is then migrated for some of the components using OpenLayers with GeoServer as WMS (Web Map Service)

provider. By doing so, we shall highlight upon the advantages of using opensource solutions over licensed software.

2. REVIEW OF ARCGIS BASED WEB APPLICATIONS

Initially the Application Development Framework (ADF) for developing web applications using ArcGIS Server existed in .Net and Java but later web mapping APIs for Flex and Silverlight have also been released to provide more flexibility to programmers. Both of these technologies provide APIs for developing rich and interactive application on the web (ESRI, n.d). Several publications exist where these programming languages and API's have been used. Yong et al, 2014 have built an application for monitoring soil quality and environmental pollution for farmland using Flex as client side interface. To improve and understand the traditional knowledge data in minority communities of china, Rong et al, 2012 have developed a web application using ArcGIS API for Flex to display, query and map this information. Flex viewer provides an easy way of building a web application through xml configuration files for those who do not have sufficient knowledge in coding. Using Flex viewer two different applications were built - GIS model of analysis to promoting tourism by Acevedo et al, 2014 and an irrigation management system used to improve the efficiency of irrigation department by Cheng-caia et al, 2012. Apart from Flex, several applications have also been built using ArcGIS API for Silverlight, one of which included development of a web based application for querying and accessing the coalfield spontaneous combustion of carbon emissions. This article by Zhu et al, 2014 mentions a number of references which refer to papers that use ArcGIS API for Silverlight for building web applications. While all the above mentioned applications used Flex and Silverlight for application development, none mention about the future of these technologies and the disadvantage of them being plug-in based for browsers.

There are, however, limitations in developing web applications using Silverlight and Flex. Like mentioned earlier, it is plug-in

based and requires installation of a plug-in engine in the browser. The other limitation is the diminishing popularity of flash compared to JavaScript, HTML5 and CSS which are being widely promoted as an alternative to flash. Due to various reasons asserted by Apple and Adobe, support for flash browser plug-in has been discontinued by Apple for all of their IOS Jobs, 2010. Therefore, restricting flash from apple's IOS and the popularity of HTML5 and JavaScript as a replacement for flash has drastically reduced the use of Adobe's flash player which in turn has raised many questions regarding the future of Silverlight with Silverlight 5 being the last release. Even though ArcGIS API for JavaScript was first introduced in 2008 as part of ARCGIS 9.3 release, It was not until the happening of these incidents that ESRI has started using the word Aggressive in their roadmap for programmers to start exploring web solutions based on JavaScript, Dojo Framework and HTML5 (Powell, 2014).

Dojo Toolkit is a JavaScript framework used for building Rich Internet Applications (RIA) that run natively on all the browsers. The applications built using this API are similar in functionality to the applications developed using non-JavaScript based Flex and Silverlight API's. Functioning of Dojo on all the browsers without having to install any plug-ins makes it an ideal choice for developing client based browser applications. This is a major reason for which it is preferred over other non-JavaScript frameworks.

JavaScript is a well established scripting language that runs on all the browsers. A number of frameworks have been developed around it for both web and mobile. When ArcGIS API for JavaScript was first released in 2008, it was in the nascent stages. Wang et al, 2010 have used ArcGIS server 9.3.1 and JavaScript API for building a web application to facilitating the Apollo Analyst's Notebook for replaying the Apollo missions. The same technologies have been used by Tang et al, 2010 for building water resource consulting system for various functionalities. Both the applications that have been mentioned have not used Dojo toolkit and instead used ArcGIS API for JavaScript which was not well established in those days. Even in the recent times, not many papers exist on Dojo Toolkit for GIS development. Based upon the capabilities that Dojo provides for developing client side interface, and it being open source, it is a sincere effort in this paper to highlight upon the many uses it offers for both proprietary and open source.

Hence, for all the history and reasons stated above, ESRI has now been fully concentrating its development efforts on building API's using JavaScript and Dojo Toolkit. These two API's are tightly coupled with each other so that application can run on different browsers.

3. DATASETS

Data collected since few year for the Tank Information System (TIS) project contained about 35000 tanks for the 10 districts of Telangana state. These tanks were identified by overlaying topographic maps (obtained from Survey of India) on cartosat satellite imagery. Out of all the tanks, 10000 have comprehensive information which includes detailed attribute information about location details, hydrological parameters, photos and videos. The other 25000 tanks lacked this information. For those that had information, the colour schema was put to dark blue colour and for those that didn't; light blue colour was given like displayed in Figure 2 and Figure 3. The developed applications were built to display and query this information

The tank data exists as polygons in shapefile format and the detailed attribute information exists in a separate excel table. A unique ID is used to pair each record of shapefile with excel table. Photos and videos of each tank are saved in folders named based on the unique ID.

4. DESIGN AND APPLICATION DEVELOPMENT

Architecture of Tank Information System (TIS) provided in Figure 1 consists of two components. One is the licensed component and the other is the open source component. Using the licensed component of ArcGIS Server, two client side interfaces have been developed using two different languages. They are ArcGIS API for Flex and ArcGIS API for JavaScript (along with Dojo Toolkit). Further, to test the capabilities of open source solutions, another application was developed with same functionalities using OpenLayers, Dojo Toolkit and GeoServer. Visual Studio was used as an Integrated Development Environment (IDE) with ASP.NET and C# as programming languages. The applications built using language specific API's have functionalities that mirror each other. All the three applications have:

1. typical functionalities such as zoom, pan, zoom to extent and identity
2. a window to display minimal tank information when a particular tank is clicked for its identity.
3. a provision to download full details of the tank as PDF.
4. buttons for downloading the excel files
5. dropdowns for querying and zooming into administrative levels such as district, mandal and village.
6. Another window to display dynamic statistics based on administrative level query.
7. Photo display for individual tank.

4.1 Development of Flex Application for ArcGIS Server:

The lower component of Figure 1 shows the architectural implementation of TIS project built using ArcGIS technologies. Initially, the client side TIS application was developed using ArcGIS API for Flex in conjunction with ArcGIS Desktop, ArcGIS Server and ArcSDE. ArcGIS Desktop serves as a client for publishing resources on ArcGIS server and for importing/exporting data to/from ArcSDE (for PostgreSQL). PostgreSQL is an open source object relational database management system which allow for storing and retrieving data. ArcSDE is an enterprise database extender to PostgreSQL which adds geodatabase support for spatial management of data.

ArcCatalog is used to import tank shapefile and excel table into ArcSDE. These layers are loaded into ArcMap and necessary styling is applied. The project is saved to an MXD (a file extension) file with required layers and styles. This file is then published as rest service through ArcCatalog (used for accessing ArcGIS server). The published services are then utilized by Flex for building rich and interactive mapping applications. Apart from API specific capabilities, the current application has requirement to displaying statistics such as number of tanks in district/mandal/village for each dropdown query. This information existed in the database. For client/server database interactions, Flex supports several programming language options. In the current context, .NET web services with C# as programming language were used for this purpose.
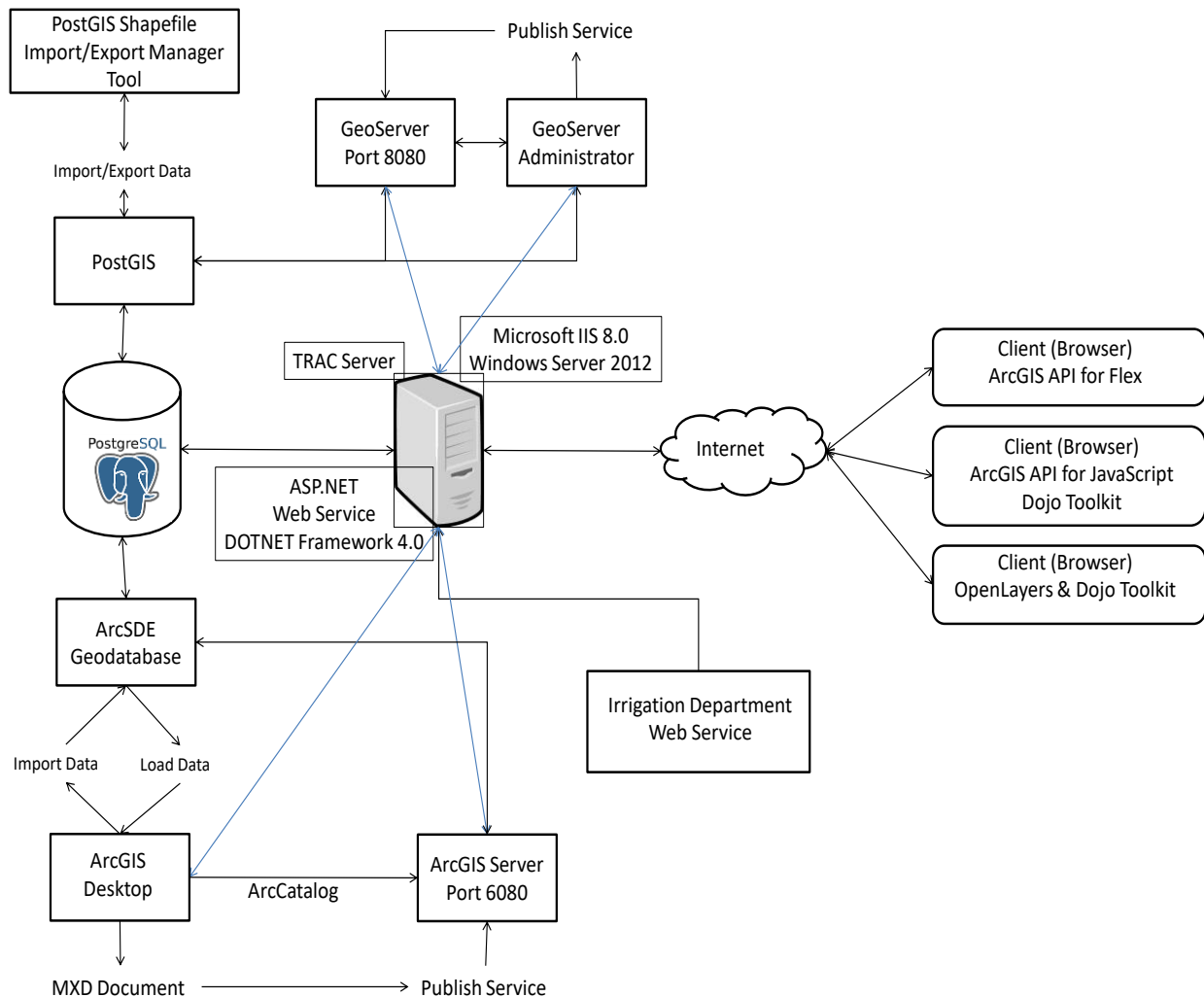
Figure 1 : Tank Information System Architecture

4.2 Development of JavaScript Application for ArcGIS Server:

Due to the limitation of flex being plug-in based and the withdrawal of support for flex on all browsers of ipad's and iphone's by Apple, the functionalities that existed on flex application were later migrated and reprogrammed completely using JavaScript technologies. Initially a design template was built using Dojo Toolkit with some basic interface components such as title area, map area, area to display statistics and toolbar area. This design template was later extended with mapping functionalities using ArcGIS API for JavaScript. While converting the application from Flex to JavaScript, Only the client side interface was reprogrammed completely. The server side components such as mapping services of ArcGIS server, web services of C# and ArcSDE database that existed for Flex application were retained without any changes.

The reprogrammed application built using Dojo was made to replicate all the mapping functionalities that existed in the previous Flex application. The User Interface components of Dojo are very much like Flex except that it does not have the

privilege of developing applications through an IDE (Integrated Development Environment).

4.3 Development of JavaScript Application using open source technologies

The upper component of Figure 1 shows the architectural part of the open source implementation for the current TIS project. The Dojo design template in the previous section formed the base for extending the application using OpenLayers (an open-source JavaScript Framework for displaying data to the user as interactive maps). This framework is similar to ArcGIS API for JavaScript and uses published services from GeoServer (an open-source server for publishing and processing geospatial data). There are many ways of publishing data through GeoServer. One is through file system where a shapefile could be read directly and the other is through PostGIS (an extension of PostgreSQL which adds support for spatial management of data). As the application had requirement of displaying dynamic statistics, data was imported to PostGIS for further access. The import was performed using an existing Shapefile Import/Export tool. GeoServer provides a web based login interface for administration of geospatial data through which data can be managed and published. Like how ArcCatalog or ArcGIS Server

Manager connects to ArcSDE of PostgreSQL to publish layers, the admin interface of GeoServer in a similar way connects to PostGIS of PostgreSQL to publish layers. The data of tanks published via this interface is consumed by OpenLayers as WMS service. For the part of server side processing, the current application also uses the same C# web services that were consumed by the previous two applications.

Three snapshots of the applications developed using language specific API's are shown in Figure 2, Figure 3 and Figure 4. All the three applications are built to show the functionalities that have been mentioned earlier in this section.

5. RESULTS AND DISCUSSION

Technologies used for developing the three applications have been explained in detail in section 4. In brief, there are four components to the development of all the three applications. They are User Layer, GeoService Layer (REST Services of ArcGIS Server or Web Map Services of GeoServer), WebService Layer (C# .NET Services) and Database Layer.

5.1 Conversion of TIS from Flex to JavaScript Framework of ArcGIS

ESRI provides API's for developing web applications with different programming languages. The geoservices i.e. the REST services published using ArcGIS server are utilised across different API's such as Flex, JavaScript and Silverlight for developing web mapping applications. Since, the migration of application from Flex to Dojo required changes only at the client side, the other component such as GeoService, WebService and Database layers remained unaltered. Flex existed for a long time and the components of this framework are very matured compared to Dojo Toolkit. This is true for applications which are built using Flex for gaming, but when Flex is used for web application development, the User Interface (UI) components of Dojo Toolkit are equally matured compared to UI components of Flex. For this reason, Dojo Toolkit provides all the necessary functionality needed for developing web mapping applications. Even if we suppose a particular functionality does not exist in Dojo, there is always a choice for using another JavaScript framework such as jQuery in conjunction with Dojo for implementation of that functionality. Not just jQuery, but many other JavaScript Frameworks are also supported. Therefore, conversion of TIS application from Flex to Dojo required changes to be made only at the client side which required little effort.

5.2 Conversion of TIS to open source technologies

ArcGIS JavaScript API utilizes Dojo Toolkit functionalities for building some of its components like zoom in, zoom out and home widget buttons. Even though ArcGIS API for JavaScript and Dojo form an integral part of each other, Dojo is a separate JavaScript framework and independent of any of the JavaScript API's of ArcGIS. Initially, Dojo template formed the base for extending it with mapping functionality for both proprietary and open source application. Hence, for the part of building the client side application, half the work of building the user interface was already complete. The other half was only to integrate and extend the application using language specific mapping API's. By using Dojo Toolkit for both proprietary and open source, the burden of implementing the client interface with different programming framework has reduced the amount of time to build the template

and has also eased the development process with familiarity in the same language.

The database layer of TIS application is implemented using PostgreSQL database server. PostGIS for GeoServer and ArcSDE for ArcGIS server, are database extenders to PostgreSQL which provide spatial capabilities for managing and manipulating geographic data. PostgreSQL database provides an excellent architecture for installation of both these extensions. Both the softwares have well defined tools for importing shapefiles. For the current project, using PostgreSQL database server for PostGIS and ArcSDE the process of migration has lowered the burden of learning and managing another database provider. At corporate level there are many organization which use Micorsoft SQL Database Server. ArcSDE can also be installed on this server. If ArcSDE for MSSQL Server were to be used, then the task of migration would have become difficult for the database layer. A migration from MSSQL to PostgreSQL would require a developer to have administrative capabilities in both the databases and would also require him to have knowledge of the SQL commands which can differ slightly based on database software providers. Therefore, having two database server would certainly require more time and effort while migrating from proprietary to open source. Considering these factors, handling and managing the resources on one database server like PostgreSQL surely eased the migration of code.

The ASP.NET Web Service Layer was built to access the database for displaying statistics and to access photos and videos on the drive. These services are consumed by the language specific client side API's for proper display of task specific functionality. To access ASP.NET web service, the Dojo application makes an Ajax based xhrPost calls. As the purpose of web service is to support interoperability, they could be accessed by any programming language which have support to consume them. In the current context, Dojo was used to display statistics, photos and videos by consuming these web services. As the platform for the client side applications of both proprietary and open source is Dojo Toolkit, the code developed for any functionality using this library, could be directly used and shared across each other without any changes. Therefore, the application built using Dojo Toolkit for the open source required little effort and used the same code and web services developed for ArcGIS application.

 shows a list of technologies used for developing both the application. While converting ArcGIS based JavaScript application to open source, technologies such as Dojo Toolkit, PostgreSQL and ASP.NET C# Web Services have been shared by both applications. Like discussed earlier, having PostgreSQL as database server eases the migration by providing same SQL language and administration. Furthermore, familiarizing oneself with Open Layers libraries and little effort in learning to publish layers on GeoServer is all that is required while migrating the licensed application to the open source. The effortless migration has been made possible by making use of the following two major technologies - Dojo Toolkit on the client side and PostgreSQL database at the server side.

S.No	Layers	ArcGIS based JavaScript TIS Application	OpenLayers JavaScript TIS Application
1	User Layer	Dojo Toolkit and ArcGIS API for JavaScript	Dojo Toolkit and Open Layers JavaScript Libraries
2	Web Service Layer	ASP.NET and C# Web Services	ASP.NET and C# Web Services
3	GeoService Layer	ArcGIS Server	GeoServer
4	Database Layer	ArcSDE for PostgreSQL	PostGIS for PostgreSQL

Table 1 : Technologies used for developing commercial and open source TIS application

5.3 Free nature of open source technologies

Web application could be developed with not just the technologies that have been used in the current study but a wide choice exists. On the proprietary front, ESRI technologies are used on a large scale both on desktop and web platform. The licensing and cost of software packages of ESRI are very complex and vary depending upon the functionality an organization requires. Opting for ESRI technologies incurs heavy costs. A full license of ArcGIS Server would cost around 65,00,000 Lakh rupees. The current study provides a case where all of the functionalities that existed on the proprietary based TIS application was migrated and reprogrammed using open source technologies. The open source based TIS application requires nil amount compared to the one developed using proprietary based ArcGIS technologies. The current study should encourage many to opt for open source technologies which are giving cost effective solutions without having to pay for the software technologies used. The level of functionalities provided by ESRI technologies are also being implemented on the open source, examples of which are OpenLayers in combination with GeoServer. Therefore, choice of open source technologies offers more benefits in terms of cost, independence to tweak source code and redistribution of developed system

6. CONCLUSIONS

The conversion of application from Flex to Dojo Toolkit has been successfully carried out by reprogramming all the functionalities that existed for Flex to Dojo. Dojo framework has all the required components for building a complete web mapping application without having the need for looking back at using Flex framework. From the discussions we had from section 2, it is but obvious to say that plug-in based technologies like Flex and Silverlight are outdated. JavaScript Frameworks which run natively on all the browsers is where the future is heading and therefore, ArcGIS API for JavaScript integrated with Dojo Toolkit is the right choice for better user experience. ESRI in this regard has taken the right decision to concentrate its future development efforts on JavaScript.

The current study shows that migrating and reprogramming of the commercial application to open source could be achieved effortlessly if Dojo Toolkit and PostgreSQL are used. ESRI's decision to develop its JavaScript Framework in tandem with Dojo Toolkit has encouraged ArcGIS Server developers to shift their programming skills to Dojo Toolkit. This is surely a healthy sign for the open source community since it becomes all the more easy for commercial developers to later shift to the open source technologies for those who are using JavaScript API's for developing web mapping applications.

REFERENCES

Acevedo, Y.V.N., Villalobos, O.A.R., Ramirez, K.J.G, 09/2014, Gis model of analysis to promote tourism through the use of web application, TECCIENCIA

Cheng-caia, Z., Mao, Z., Xi-mei, S., Henan Zhaokou Irrigation Management System Design Based on Flex Viewer, Procedia Engineering (2012), ELSEVIER

ESRI, Creating ArcGIS Server Solutions, Roadmap for Developers, n.d., URL: http://resources.esri.com/help/9.3/arcgisserver/adf/dotnet/roadm ap_developers.htm

Powell, J., Esri's Roadmap for Web Developers, February 2014, URL: http://blogs.esri.com/esri/arcgis/2014/02/21/esris-roadmap-for-web-developers/

Rong, D., Dayuan, X., Luo, G., Faming, S., December 2012, Research on Constructing a Traditional Knowledge Database Information System Based on ArcGIS Server, Intelligent Networks and Intelligent Systems (ICINIS), IEEE

Steve Jobs, April 2010, " Thoughts on Flash" an Open Letter, URL: http://www.apple.com/hotnews/thoughts-on-flash/

Tang, Q., Tongji, Z., Ren, G., 2010, Design and implementation of water resources consulting system based on ArcGIS Server, Geoscience and Remote Sensing (IITA-GRS), IEEE

Wang, J., Heet, T., Scholes, D.M., 2010, A WebGIS for Apollo Analyst's Notebook, Advanced Geographic Information Systems, Applications and Services (GEOPROCESSING), IEEE

Yong, M., Zhang, M., Wang, S., Liua, G., January 2014, A Flex and ArcGIS Server based system for farmland environmental quality assessment and prediction in an agricultural producing area, Computers and Electronics in Agriculture, ELSEVIER.

Zhu, Z., J. Bi, X Wang and W Zhu, 2014, The research and implementation of coalfield spontaneous combustion of carbon emission WebGIS based on Silverlight and ArcGIS server, Earth and environmental Science, IOPScience
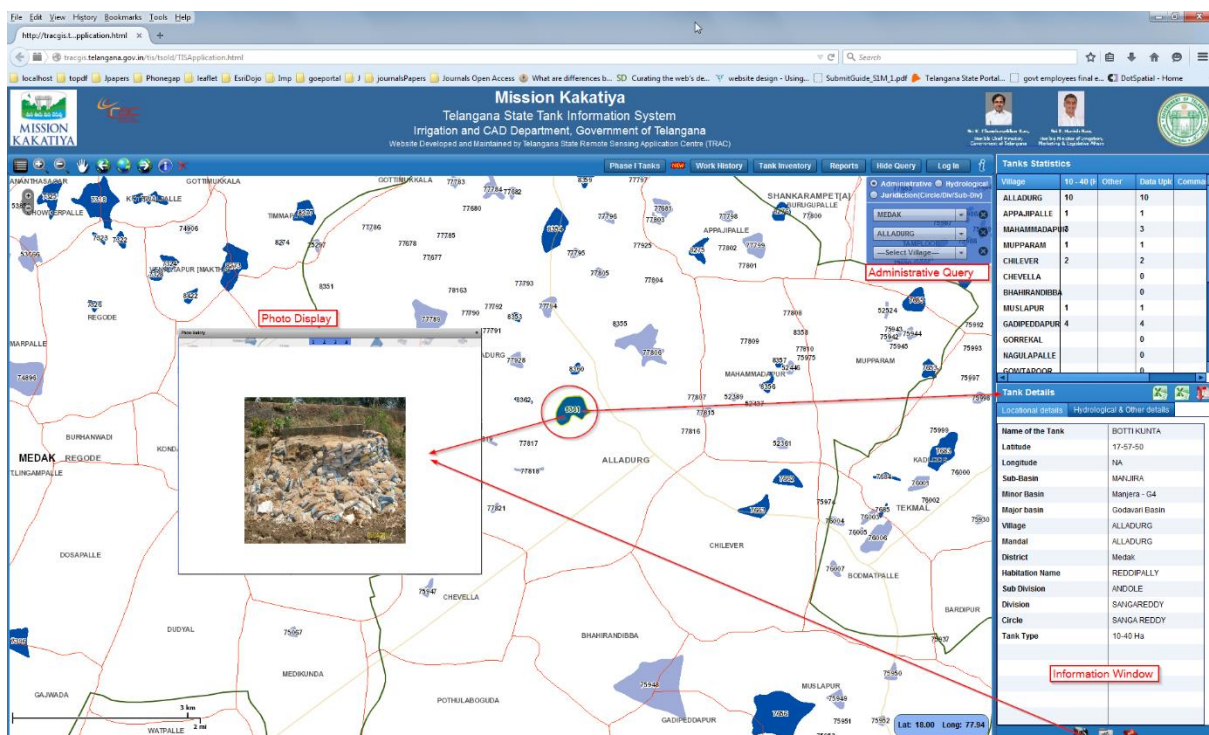
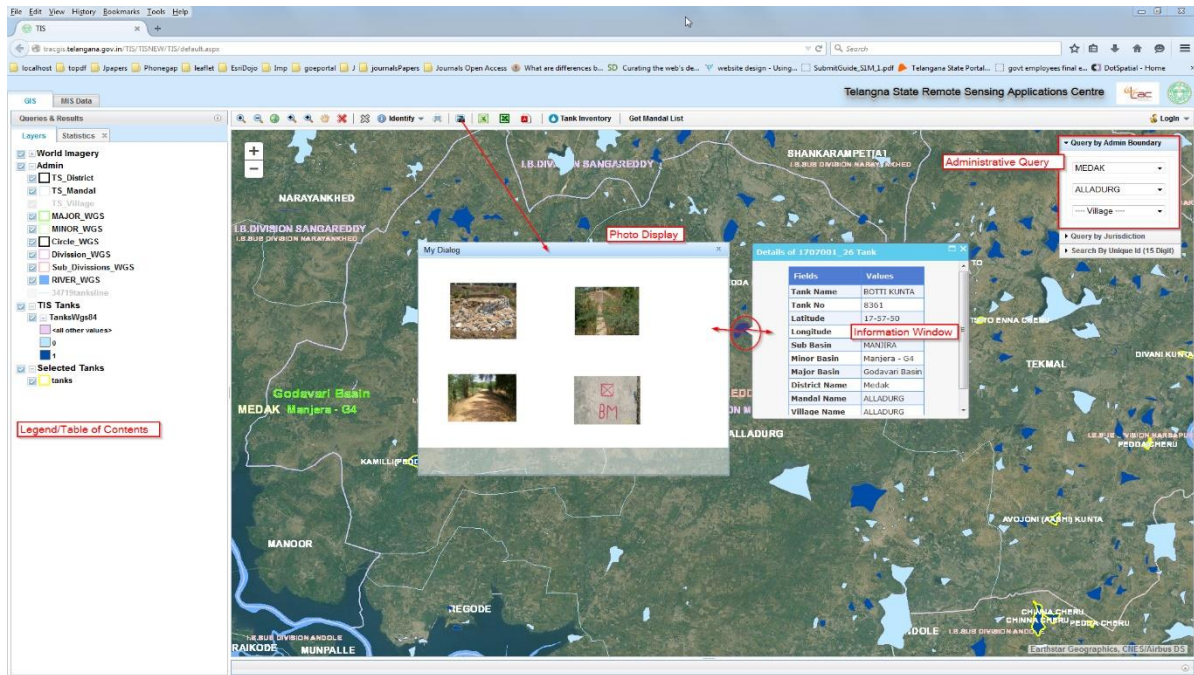Figure 2 : TIS Application developed using ArcGIS API for Flex

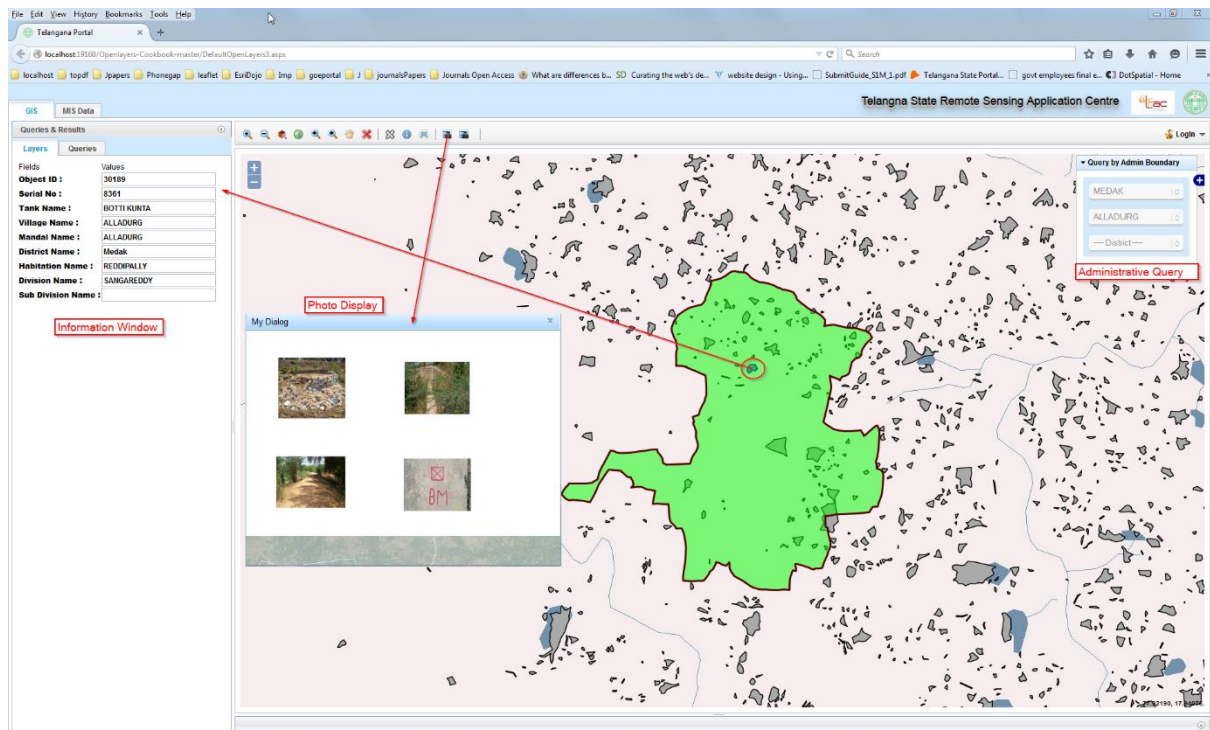Figure 3 : TIS Application developed using ArcGIS API for JavaScript and Dojo Toolkit

Figure 4 : TIS Application developed using Openlayers and Dojo Toolkit