

Flexible Damage Assessment Framework

Table of Contents

- Table of Contents 1
- Terminology 2
- History 3
- Effort..... 4
 - Multiple damage reports 4
 - Environment..... 4
 - Flexibility 4
 - Performance..... 4
 - Reliability 4
 - Availability..... 4
- Concerns 5
 - Security 5
- Architecture 6
 - Additional data 6

Terminology

HighLevelArchitectureDiagram
DamageAssessment-High-Level.pdf
OutageDBDiagram
Outage-DatabaseDiagram.pdf
OutageXmlDictionary
OutageXmlDictionary.xls
SecurityLayer
A webservice through which the internet will access our services.
STARDBDiagram
STAR-DatabaseDiagram.pdf
STARDBDiagramInput
STAR-DatabaseDiagram-Input.pdf
XmlDictionary
XmlDictionary.xls
WebServiceDetail
DA_Interface.doc

History

The STAR Web Workspace application of Con Edison is a windows application to enter Damage Assessments. The current workflow to enter the Damage Assessments to the STAR system is as following:

(Make a diagram)

A damage assessment crew is assigned to a particular job and sent to the field. He is given a paper form so that he can fill the details of the damage assessment. At the damage site, the crew assesses the damage and fills the form appropriately. On reporting back at the Con Edison location, another employee collects the paper Damage Report and enters the corresponding fields in the Web Workspace application.

Effort

The current effort for creating the flexible damage assessment framework addresses many shortcomings in this process.

Multiple damage reports

One of the functional requirements in the enhanced version of the Web Workspace windows application is to allow multiple damage reports for one job. This is useful since a crew or multiple might have to work on a single job. On top of that, even a single crew might have to visit a few sites and enter multiple reports. This multiplicity is allowed by FlexDAF and has been tested successfully.

Environment

Instead of using the windows application environment which is always-connected and is attached to a single-datasource, FlexDAF provides a disconnected mechanism to enter damage assessments.

Flexibility

Since FlexDAF is a framework, it provides us with the flexibility to upload damage assessments regardless of the front-end, hardware and middleware platform.

Performance

It is expected that the software clients of FlexDAF will not need to instantiate the executables, as is done in the current environment. This will reduce the load on the client machine, keeping it responsive.

Reliability

The framework components are first-class software elements. These are: a web server for the web services and an Oracle database server to retrieve and upload data. These servers are a part of the Con Edison software infrastructure and carry the highest support level of the IT group.

Availability

For the same reasons as reliability, the framework enjoys a high level of availability from the intranet.

Additionally, we have created and tested an iPad application which is a client of this framework. We have a relatively good response on 4G and an under-2-second response on Wi-Fi.

These tests make us comfortable that the framework can be used in the field and thereby bring all the damage assessments to the view of engineers and management as soon as possible. Real-time field tests are yet to be conducted and without the results from such tests, we cannot be too sure about the availability from the internet.

Concerns

Security

Currently, a web service provides a security framework to connect to our own web service. Although this gives us feeling of comfort about security, we need to increase our knowledge base of this security framework to be confident.

Architecture

The architecture of the FlexDAF can be best understood from the Detail Diagram in High Level Architecture Diagram (refer Terminology).

We are omitting SecurityLayer (refer Terminology) in this explanation as it is still in the fluid state.

The details of the elements shown in the STAR DB section of the diagram can be seen in the STARDBDiagram and STARDBDiagramInput (refer Terminology)

Let's start from the Input tables and the Input Stored Procedure elements in the STAR database. The input stored procedure gets the data from the Input Tables, which is basically data for all the open jobs alongwith the detail of the customers affected and the crew assigned.

The client of the FlexDAF (GUI) will call this stored procedure via the web service DamageReport's GetAsgnDmaCrews web method.

Note: Currently, this SP has not been created and we are only sending an ad-hoc query.

The GUI is expected to allow the user to choose the job assigned to him. The user should then proceed to enter one more damage assessments, along with the sites, reports and pictures. The GUI should also provide the user a functionality to upload the data that he/she has entered.

When the user does upload the data, the data flows into the "Final STAR tables" in the STAR database via the webservice EMA's UploadData_STAR method. This web method calls the upload Stored Procedure which automatically decides if an upload is a new entry or an update to an old one and acts accordingly. An intermediate table has been introduced after discussion with STAR team. STAR team checks this table to retrieve new data and uses it to update workagenda, highlighting the job having a damage assessment report associated to it.

The data that expected by the web method and its flow through all the elements is detailed in the XmlDictionary (refer Terminology).

Additional data

The details of the elements shown in the Obvient DB section of the diagram can be seen in the STARDBDiagram and STARDBDiagramInput (refer Terminology)

In the process of developing a GUI prototype of the Damage Assessment, a subject matter expert identified many data items which are not currently required by STAR but will be of interest to the engineers. These are mainly free-flow descriptions and pictures. The STAR data along with these additional data items are currently being uploaded to the Obvient database. The methodology followed is similar to the one followed for the STAR upload process. At the time of the upload, the data is sent to the web service EMA's UploadData web method which calls the Upload Stored Procedure in the Obvient and fills those tables.

The documentation related to Obvient will be initiated once the documentation related to STAR is finished.

Detail about the web services and their methods can be seen in the [WebServiceDetail](#) (refer [Terminology](#)).