

2016



UNIVERSITÀ
degli STUDI
di CATANIA

Relazione Progettuale OpenData

REALIZZATO IN COLLABORAZIONE CON IL DOTT. CRISTIANO LONGO E
OPENDATAHACKLAB
GIORGIO OLIVA MATRICOLA W82/000085

Sommario

.....	1
Introduzione al progetto	3
Analisi di consistenza	4
Analisi dei dati	4
Analisi di completezza dei dati	5
Analisi di consistenza dei dati.....	6
Analisi Sintattica CSV.....	8
Scelta del file CSV	9
Caricamento del file	10
Classe ParserCSV	13
Classe Parser_Object_confiscated	15
Classe Generate_File_RDF_XML	16
Scrittura dei file.....	19
Interfaccia WEB_opendatahacklab-confiscated assets.....	21
Conclusioni.....	25

Introduzione al progetto

Il progetto relativo al completamento della materia magistrale Algoritmi e Complessità è stato concordato con il Dott.Cristiano Longo realizzato nell'ambito dei beni confiscati alla mafia a cura di opendatahacklab a cui il Dott.Longo fa parte vedi (<http://opendatahacklab.org>).

Il progetto consta essenzialmente di tre fasi:

1. Analizzare la sorgente dati relativa alle informazioni date dai comuni sui beni confiscati alla mafia, redigendo un report sulla consistenza e conformità di questi dati forniti dalla associazione Libera confrontandoli con le stesse informazioni ma date dai singoli comuni.
2. Progettazione ed elaborazione di un'analisi sintattica automatizzata di file CSV, con scopo ultimo quello di costruire 4 file di tipo RDF/XML con nome assets.owl, locations.owl, geometry.owl e per ultimo un unico file assetsAll.owl; il parserizzatore sintattico dovrà essere elaborato esclusivamente in linguaggio PHP.
3. Progettazione di una pagina Web con scopo illustrativo dei beni sequestrati dalla mafia, fornendo mappa geografica dei beni confiscati con relative informazioni degli stessi, esplicitandone la posizione.

Analisi di consistenza

Questo report di ricerca e di assistenza tecnica intende pianificare un intervento costruttivo sull'organizzazione dei dati pervenuti dalla associazione Libera nel report pubblicato nell'ottobre del 2015, dove palesa i punti essenziali da fornire e la mappatura dei beni confiscati locati nei comuni di Alcamo, Calatafimi, Campobello di Mazara, Castelvetrano, Partanna, Salemi e Vita al fine di renderli accessibili quindi pubblici, prendendo spunto dai dati consultabili riportati sui portali web comunali ufficiali.

Il lavoro che si intende realizzare e sottoscrivere è quello di creare un file Resource Description Framework (RDF) ovvero lo strumento base proposto da W3C per la codifica, lo scambio e il riutilizzo di metadati strutturati e consente l'interoperabilità tra applicazioni che condividono le informazioni sul Web; questo file si comporrà delle sole informazioni che identificano ogni bene confiscato; le informazioni a cui si fa riferimento sono raccolte nella tabella informativa in figura 1.

Questo documento verrà caricato in uno STORE RDF che verrà in un successivo momento interrogato per restituire al pubblico mediante una mappa tutte i beni confiscati con in allegato il dettaglio del bene.

Analisi dei dati

Prima di elaborare i dati bisogna fare apriori uno studio di consistenza di quest'ultimi ovvero verificare che i dati contenuti nei CSV forniti dall'associazione Libera sono significativi ed effettivamente utilizzabili per il nostro lavoro progettuale.

Le informazioni standardizzate da Libera sono esplicitate in figura.

<i>ID del bene</i>	<i>dati identificativi del concessionario</i>	<i>Estremi catastali</i>
<i>descrizione bene</i>	<i>ragione sociale</i>	<i>Foglio</i>
<i>metri quadri</i>	<i>telefono</i>	<i>Particella</i>
<i>estremi decreto di confisca</i>	<i>email</i>	<i>Subalterno</i>
<i>estremi provvedimento di sequestro</i>	<i>Indirizzo</i>	<i>Indirizzo fisico del bene</i>
<i>estremi provvedimento di destinazione</i>	<i>durata dell'atto di concessione</i>	<i>Via</i>
<i>estremi verbale di consegna</i>	<i>modalità di assegnazione</i>	<i>Numero Civico</i>
<i>numero del procedimento</i>	<i>descrizione del progetto di riutilizzo sociale</i>	<i>CAP</i>
<i>numero delle sentenza definitiva</i>	<i>allegato opzionale con descrizione del progetto (ad esempio il PDF che racconta il progetto)</i>	<i>Comune</i>
<i>assegnato</i>	<i>data compilazione scheda</i>	<i>Coordinate geografiche</i>
<i>condizioni attuali del bene</i>	<i>nome e ruolo di chi compila la scheda</i>	<i>latitudine</i>
<i>occupato</i>		<i>longitudine</i>
<i>accessibilità al bene</i>		<i>Note</i>
<i>valore del bene (euro)</i>		

figura 1

Analizzando i seguenti file reperiti dal report CSV forniti da Libera riguardo i comuni:

- Alcamo
- Calatafimi
- Campobello di Mazara
- Castelvetrano
- Partanna
- Salemi
- Vita.

Analisi di completezza dei dati

Sono state trovate incompletezze sostanziali relative ai dati forniti su tutti i comuni in elenco, riporto tali mancanze in seguito:

1. **Alcamo:** I dati relativi al nominativo del precedente possessore del bene a cui è stato confiscato non figura inoltre non figurano informazioni sugli estremi del provvedimento del sequestro, numero di provvedimento, numero di sentenza definitiva, il gestore a cui viene assegnato per la maggior parte dei beni non figura quindi informazioni incomplete come appunto le informazioni per il reperimento degli stessi come telefono, e-mail, indirizzo e durata dell'atto di concessione.
2. **C. Bello Mazara:** i dati che risultano essere incompleti sono i Mq del bene, Estremi provvedimento di sequestro, Numero della sentenza definitiva, Condizioni attuali del bene, Abusivo, Occupato, Accessibilità al bene, valore.
3. **Calatafimi-Segesta:** Mancanza ID_bene.
4. **Castelvetrano:** i dati che risultano essere incompleti sono, le informazioni relative a chi è stato confiscato, Mq del bene, Estremi provvedimento di sequestro, Numero del procedimento, Condizioni attuali del bene, abusivo, occupato, accessibilità.
5. **Partanna:** i dati che risultano essere incompleti sono, le informazioni relative a chi è stato confiscato, Estremi provvedimento di sequestro, Numero del procedimento, Numero della sentenza definitiva.
6. **Salemi:** i dati che risultano essere incompleti sono, , Estremi provvedimento di sequestro, Numero del procedimento, Numero delle sentenza definitiva.
7. **Vita:** ID_bene Mancante.

Analisi di consistenza dei dati

Importante ruolo ha l'analisi di consistenza evidenziando qualora ci fossero anomalie sull'emissione dell'informazione, analisi che ha riscontrato importanti anomalie di forma e contenuto nell'indicazione dell'informazione emessa.

Ecco le principali anomalie sugli attributi:

- **Abusivo:** Vengono riportate descrizioni che non fanno parte dello standard del dato.
- **ID_bene:** Vengono riportati diverse versioni dell'ID rispetto alla forma standard eccone un estratto, K-bene xxxx, k_bene xxxx, più ID per ogni bene, mancanza di ID.
- **Tipologia bene confiscato:** Categorizzazione troppo ristretta per la quantità di tipologie riportate.
- **Confiscato:** Vengono riportati date di nascita dei precedenti proprietari non conformi allo standard gg.mm.aaaa
- **Assegnato:** Vengono riportate descrizioni superflue.
Es. no-finalità
- **Occupato:** Viene riportato anche uno stato fittizio dell'occupazione
Es. Si-in corso.
- **Mq:** Viene riportata una descrizione delle metrature di diversi componenti del bene confiscato e non del complessivo.

Incongruenza generale notata è la non uniformità tra MAIUSCOLE e minuscole dei caratteri impiegati.

Inoltre durante l'analisi dei dati pervenuti, si nota una standardizzazione della tipologia di parole chiave e variabili inserite relative agli attributi identificativi dei file CSV:

- **Latitudine:** gradi
- **Longitudine:** in gradi
- **ID del bene:** indicato nella forma K_bene seguito da un numero intero
- **Accessibilità al bene:** buona, sufficiente, scarsa
- **Confiscato:** Nome, Cognome e Data di nascita (gg/mm/aaaa)
- **Mq:** Metratura, espressa in numeri interi, del bene confiscato
- **Assegnato:** Si/No

- **Tipologia bene confiscato** : appartamento, terreno agricolo, villa, altro
- **Condizioni attuali del bene**: cattive, normali, buone
- **Valore**: Quantità espressa in €.
- **Abusivo**: Si/No
- **Durata dell'atto di concessione**: durata espressa in giorni.
- **Occupato**: Si/No

ATTENZIONE

Viene utilizzata la stringa INFORMAZIONE NON RECUPERATA per indicare un elemento che non è stato recuperato ma presumo esista quindi risulta essere mancante, mentre viene indicato con INFORMAZIONE NON RECUPERABILE una informazione di cui non è praticamente possibile avere informazioni quindi fa parte della standardizzazione.

Analisi Sintattica CSV

Dopo un'accurata analisi gli scopi principali del lavoro sono essenzialmente:

- Estrapolare le informazioni salienti riguardanti i Beni confiscati alla mafia e dall'analisi effettuata precedentemente i principali sono:
 1. Latitudine: 1° coordinata GPS relativa al Bene Confiscato
 2. Longitudine: 2° coordinata GPS relativa al Bene Confiscato
 3. ID_Bene: Codice univoco che identifichi il Bene Confiscato
 4. Indirizzo: Posizione geografica del Bene Confiscato
 5. Assegnato : Risposta booleana se il Bene Confiscato è stato assegnato a una associazione amministratrice.
 6. Descrizione: Breve descrizione del bene, conformazione strutturale e stato al momento del sequestro dell'immobile.
- Fornire un set di classi PHP che analizzino il file caricato estraendone le informazioni discusse nel precedente punto producendo i file RDF/XML “**assets.owl**”, “**locations.owl**”, “**geometry.owl**” e per ultimo un unico file “**assetsAll.owl**” che li comprenda tutti.
- I file discussi nel precedente punto saranno oggetto di una pagina web che illustri geograficamente tutti i beni sequestrati alla mafia indicati nel file CSV del comune grazie all'apporto delle GoogleMApsAPIs e di SPARQL.

Oggetto del lavoro sono i file CSV relativi ai comuni dei comuni elencati precedentemente che come è visibile dalle figure comprendi molte informazioni relative ai beni sequestrati alla mafia da cui bisogna estrarre solo quei dati interessati per portare al completamento il lavoro.

latitude	longitude	ID del bene	Descrizione	Confiscato	Immagine	Mq	Estremi decr	Estremi prov	E
37,824261	12,87573		Terreno agricolo Musso Calogero (09/09/1994)	Due spezzoni	Decreto n. 3	Informazione F			
37,82466	12,875651		Appezzamento Luigi Cacioppo (25/10/1946)		6.360	Decreto n. 2	Informazione F		
37,943512	12,834838		Terreno agricolo Modesto Giuseppe (6/4/1994)	180190000 e	Decreto n. 4	Informazione F			
37,963032	12,889552	k-bene 1976	Fondo rustico Ambrogio Farina (24/01/1994)		79.520	Decreto n. 34	Informazione F		
37,893588	12,90467	k-bene 2753	Terreno agricolo Liborio Pirrone (03/05/1993)			Informazione F	Decreto n. 44	Informazione F	
37,892751	12,900363	K-bene 2753	Terreno agricolo Liborio Pirrone (03/05/1993)		710010	Decreto n. 44	Informazione F		
37,895277	12,900363	K-bene 1223	Terreni in pa Salvatore Crimi (19/11/1994)	#####		Decreto n. 9!	Informazione F		

Figura 2

Estremi prov	Estremi prov	Estremi verb	Numero del	Numero dell'	Assegnato	Tipologia ber	Condizioni at	Abusivo
Informazione	Prot. n. 3309	Consegnato f	Informazione	Informazione	Si	Terreno agricolo	Normali	No
Informazione	Prot. n. 4824	Consegnato f	Informazione	Informazione	No	Terreno agricolo	Normali	No
Informazione	Prot. n. 3477	consegnato f	Informazione	Informazione	No	Terreno agricolo	Normali	No
Informazione	Prot. n. 1591	Consegnato f	Informazione	Informazione	No	Terreno agricolo	Normali	No
Informazione	Prot.n. 1643;	Informazione n. 4/2011 RN	Informazione	Informazione	No	Terreno agricolo	Normali	No
Informazione	Prot.n. 1643;	Informazione n. 4/2011 RN	Informazione	Informazione	No	Terreno agricolo	Normali	No
Informazione	Prot. n. 3093	Consegnato f	Informazione	Informazione	No	Terreni adibiti	Normali	No

Figura 3

Occupato	Accessibilità	Valore	Eventuali alti	Gestore	Telefono	E-mail	Indirizzo Ges	Durata dell'a	Assegnazione
Sò	Buona	803.090,48	Nessuna	Fondazione	+39 0923189	fondazione@	Piazza della F	5110	Assegnazione
Si	Buona	1115000	CAMBIO DEL	Comune di C	Informazione	Informazione	Informazione	Informazione	Informazione
no	Buona	48567,2	cambio della	Comune di C	Informazione	Informazione	Informazione	Informazione	Informazione
No	Buona	Informazione	Agenzia del c	Comune di C	Informazione	Informazione	Informazione	Informazione	Informazione
No	Buona	Informazione	Destinato a f	Comune di C	Informazione	Informazione	Informazione	Informazione	Informazione
No	Buona	Informazione	Destinato a f	Comune di C	Informazione	Informazione	Informazione	Informazione	Informazione
Sò	Buona	Informazione	Cil bene	sta Ditta individu	Informazione	Informazione	Contrada Cal	Contratto sc	Concessione

Figura 4

Riutilizzo	Allegato opz	Data compil	Nome e ruoli	Foglio	Particella	Subalterno	Indirizzo	Note			
Recupero tos	Informazione non Recuperata			123	12, 13		Contrada Ro	Nella mappa	presso un punto indicativo		
Attività Istitu	Informazione non Recuperata			1	403,406,405, 1 e 3		Contrada Co	Nella mappa	presso un punto indicativo		
Attività Istitu	Informazione non Recuperata			117	77 (ex 51/b) e 76 (ex39/a)		Contrada La	Nella mappa	presso un punto indicativo		
Attività Istitu	Informazione non Recuperata			8	81,82,83		Contrada An	Nella mappa	presso un punto indicativo		
Attività Istitu	Informazione non Recuperata			84	131		Contrada Tu	Nella mappa	presso un punto indicativo		
Attività Istitu	Informazione non Recuperata			96	151		Contrada Na	Nella mappa	presso un punto indicativo		
Informazione	Informazione non Recuperata			42	14, 38,39,72,83,131,15,76		Contrada Ro	Nella mappa	presso un punto indicativo		

Figura 5

Scelta del file CSV

Primo passo per l'analisi del file **CSV** è stato quello di creare una pagina HTML di Choosing and Update file, tale pagina prende il nome di UploadFile.php, il contenuto di quest'ultima è visibile nella successiva figura.

```

UploadFile.php
<?php
/**
 * @author Giorgio Oliva
 */
?>
<!DOCTYPE html>
<html>
<head>
    <link href='https://fonts.googleapis.com/css?family=Source+Sans+Pro:600,400,300' rel='stylesheet' type='text/css'>
</head>
<body>
<div>
    <h1>Upload File csv</h1>
    <form action="Upload_Confiscated_Object.php" method="post" enctype="multipart/form-data">
        Seleziona Il file CSV del comune da Parserizzare
        <input type="file" name="fileToUpload" id="fileToUpload"><br/>
        <input type="submit" value="Carica CSV" name="submit">
    </form>
</div>
</body>
</html>

```

Figura 6

Caricamento del file

Una volta avvenuta la scelta del file avviene il caricamento del file tramite la classe “Upload_Confiscated_Object.php” che in prima istanza controlla che ci sia effettivamente il file caricato nell’apposito form e tiene a traccia del nome, della dimensione e del tipo di file che successivamente controlla sia conforme allo standard CSV se non è conforme il sullo schermo apparirà un messaggio d’errore come si può osservare in figura7.

```
if(isset($_FILES['fileToUpload'])){
    $errors= array();
    $file_name = $_FILES['fileToUpload']['name'];
    $file_size =$_FILES['fileToUpload']['size'];
    $file_tmp =$_FILES['fileToUpload']['tmp_name'];
    $file_type=$_FILES['fileToUpload']['type'];
    $num1;
    $file_name_array=explode('.',$file_name,-1);
    $convert_file_name=$file_name_array[0];
    set_include_path(get_include_path() . PATH_SEPARATOR . $path);
    $expensions= array('text/plain','text/csv','text/tsv','text/xls');
    if(in_array($file_type,$expensions)== false){
        $errors[]="Estensione non permessa, si prega di selezionare un file CSV corretto";
    }

    if($file_size > 2097152){
        $errors[]='File size must be exactly 2 MB';
    }

    if(empty($errors)==true){
        move_uploaded_file($file_tmp,"tmp/".$file_name);
    }
}
```

Figura 7

Funzione che detta l’inizio dell’analisi del file in questione è il metodo **parse_csv(\$file_name,\$del)** dove tra i parametri passati c’è appunto il file appena caricato; il metodo con estrema semplicità scansiona il CSV riga per riga interpretando la riga come un array quindi in ogni locazione troveremo una informazione semplice da consultare tramite l’indice **\$s**. Le informazioni salienti discusse precedentemente una volta estratte verranno inserite in un’istanza della classe **Parser_Object_confiscated** creata adHoc per raggruppare tutti i dati che descrivono un bene per agevolarne operazioni di consultazione e estrazione, nello specifico l’istanza è identificata da una variabile temporanea **\$tmp=new Parser_Object_confiscated;** che ad ogni iterazione del ciclo for aggiornerà i propri campi, tale procedimento è visibile nella successiva figura.

```

for ($c=0; $c < $num; $c++)
{
    $tmp -> setLatitude($data[0]);
    $tmp -> setLongitude($data[1]);
    $tmp -> setID_Bene($data[2]);
    $tmp -> setDescrizione($data[3]);
    $tmp -> setAssegnato($data[13]);
    $tmp -> setIndirizzo($data[34]);
    $tmp -> setName_file_csv($convert_file_name);
    if($data[$c]==(""))
    {
        break;
    }
    $arguments[$s]=$tmp;
}

```

Figura 8

Possiamo osservare l'intera funzione nella figura 9

```

<?php
function parse_csv($file_name,$del)
{
    $file_name_array=explode('.',$file_name,-1);
    $convert_file_name=$file_name_array[0];
    $arguments = array();
    $row = 1;
    $s=0;
    if (($handle = fopen($file_name,"r")) !== FALSE)
    {
        while (($data = fgetcsv($handle, 10000, $del)) !== FALSE)
        {
            $num = count($data); //Data è un array
            $row++;
            $tmp=new Parser_Object_confiscated;
            for ($c=0; $c < $num; $c++)
            {
                $tmp -> setLatitude($data[0]);
                $tmp -> setLongitude($data[1]);
                $tmp -> setID_Bene($data[2]);
                $tmp -> setDescrizione($data[3]);
                $tmp -> setAssegnato($data[13]);
                $tmp -> setIndirizzo($data[34]);
                $tmp -> setName_file_csv($convert_file_name);
                if($data[$c]==(""))
                {
                    break;
                }
                $arguments[$s]=$tmp;
            }
            $s=$s+1;
        }
        $Parser=new ParserCSV($arguments);
        $Parser -> Stampa();
        $Parser -> Carica_csv();
    }
}
fclose($handle);
$del=';';
print_r(parse_csv($file_name,$del));
?>

```

Figura 9

Al termine di tale procedura si riscontrano tutti i dati relativi a tutti i beni confiscati nel comune di Calatafimi Segesta catalogati per bene in oggetti di tipo **Parser_Object_confiscated**, e raggruppati nell'array **\$arguments**.

Questo array di "argomenti" sarà la principale informazione per la creazione dell'istanza **\$Parser** derivanete dalla classe **ParserCSV**, le informazioni prodotte sono visibili nelle figure 10.

Parserizzo il File CSV : Calatafimi-Segesta

Parserizzato il : 16/06/2016 - 14:39:13 Contenente i seguenti riferimenti

Bene N°1
Latitudine : 37.824261
Longitudine : 12.87573
ID_Bene :
Indirizzo : Contrada Rosignolo
Assegnato : Si
Descrizione : Terreno agricolo coltivato a vigneto

Calatafimi-Segesta
Bene N°2
Latitudine : 37.82466
Longitudine : 12.875651
ID_Bene :
Indirizzo : Contrada Coriolano
Assegnato : No
Descrizione : Appesamento di Terreno con insistente fabbricato costituito da un capannone (piano terra e scantinato) adiacante a due fabbricati a doppia elevazione, ad uffici e servizi igienici e un piccolo fabbricato ad una elevazione adibito ad alloggio caldaia e gruppo elettrogeno. completano l'impianto un edificio per alloggio custode e una cabina per l'alloggio di apparecchiature per l'impianto di depurazione. Il Comune di Calatafimi-Segesta con delibera della Giunta Municipale n. 221 del 22/12/2014 ha preso atto del provvedimento di concessione n. 40 del 17/12/2014 emesso dal Presidente del GAL Elimos con cui è stato concesso un contributo imponibile di euro 398.420,14 corrispondente al 100% della spesa ritenuta ammissibile oltre a euro 63.441,58 per iva e rimborso per la "realizzazione di piattaforma logistica a servizio delle imprese agricole, per lavori di manutenzione straordinaria

Calatafimi-Segesta
Bene N°3
Latitudine : 37.94352
Longitudine : 12.834838
ID_Bene :
Indirizzo : Contrada Lagani
Assegnato : No
Descrizione : Terreno agricolo con insistente fabbricato

Calatafimi-Segesta
Bene N°4
Latitudine : 37.963032
Longitudine : 12.889552
ID_Bene : k-bene 19769
Indirizzo : Contrada Angimb♦
Assegnato : No
Descrizione : Fondo rustico

Calatafimi-Segesta
Bene N°5
Latitudine : 37.893588
Longitudine : 12.90467
ID_Bene : k-bene 275353
Indirizzo : Contrada Tufforosso
Assegnato : No
Descrizione : Terreno agricolo

Calatafimi-Segesta
Bene N°6
Latitudine : 37.892751
Longitudine : 12.900363
ID_Bene : K-bene 275344
Indirizzo : Contrada Nanfo
Assegnato : No
Descrizione : Terreno agricolo

Calatafimi-Segesta
Bene N°7
Latitudine : 37.895277
Longitudine : 12.900363
ID_Bene : K-bene 122389, 122390, 122392
Indirizzo : Contrada Rocche
Assegnato : No
Descrizione : Terreni in parte adibiti a Cava di pietra calcarea e beni mobili tra cui: un autocarro Perlini T.15 telaio n.21879264, un autocarro Perlini T. 20 telaio n. 2052062, intero impianto di frantumazione. mezzi da rottamare: un autocarro Perlini T. 15 telaio n. 217s731780, un autocarro FIAT 662, Pala Gommata FIAT Allis telaio n. r20a2t250051, Pala Gommata Allis CHALMERS 745 h, Pala Cingolata FIAT Allis 14 c.

Calatafimi-Segesta
Scritti i File Sul disco!

Figura 10

Classe ParserCSV

La classe ParserCSV si preoccupa fondamentalmente di creare i file “**assets.owl**”, “**locations.owl**”, “**geometry.owl**” e “**assetsAll.owl**”, aprirli e scrivere all’interno le giuste informazioni comprendenti i beni confiscati analizzate in precedenza, in seguito verranno esposti spezzoni di codici che realizzano quanto detto prima.

Nella figura 11 viene illustato come vengono creati i file con permesso di scrittura direttamente da codice PHP.

```
*****  
 * Create the handles  
 */  
 $assets=fopen('assets.owl','w+') or die('Unable to create file');  
 $locations=fopen('locations.owl','w+') or die('Unable to create file');  
 $coordinates=fopen('geometry.owl','w+') or die('Unable to create file');  
 $all=fopen('assetsAll.owl','w+') or die('Unable to create file');
```

Figura 11

In seguito questi file appena creati vengono aperti e al suo interno viene copiato un preambolo, ovvero codice RDF/XML molto statico che di fatto apre alcuni tag che verranno chiusi in seguito, i preamboli sono diversi e quindi ogni file da creare avrà un preambolo adHoc, insieme ai preamboli viene trascritto anche un time-stamp con la classi interessate nella creazione e la data e l’orario di creazione, tali operazioni sono visibili in figura.

```
*****  
 * Print headers.  
 */  
 fwrite($assets, file_get_contents('assets_preamble.owl.part'));  
 fwrite($locations, file_get_contents('locations_preamble.owl.part'));  
 fwrite($coordinates, file_get_contents('geometry_preamble.owl.part'));  
 fwrite($all, file_get_contents('assetsAll_preamble.owl.part'));  
 $timeString="Generated by ParserCSV.php & Generate_File_RDF_XML on ".date('d-m-Y-H-i');  
 fwrite($assets, "<!-- ".$timeString."-->\n");  
 fwrite($locations, "<!-- ".$timeString."-->\n");  
 fwrite($coordinates, "<!-- ".$timeString."-->\n");  
 fwrite($all, "<!-- ".$timeString."-->\n");
```

Figura 12

Passaggio saliente è la creazione dell’istanza della classe **Generate_File_RDF_XML** che prende il nome di **convert** di cui ne discuteremo in seguito le specifiche ma che in breve è la responsabile della creazione del contenuto dei quattro file.

Il contenuto dei file viene estrapolato dinamicamente per ogni bene confiscato dall’array **\$arguments** passato precedentemente e grazie ai metodi di **getLatitude(); getLongitude(); getID_Bene(); getIndirizzo(); getAssegnato() e getDescrizione();** si estrapolano le informazioni dall’oggetto **Parser_Object_confiscated**, che successivamente vengono

passati al metodo **convert** per il riempimento del corpo dei file come narra la figura 13.

```
$converter = new Generate_File_RDF_XML($assets, $locations, $coordinates,$all);
$num1 = count($this->Arguments_Confiscated);
for ($i = 1; $i <= $num1-1; $i++)
{
    $new_Latitudine=$this->Arguments_Confiscated[$i] -> getLatitude();      //Latitudine
    $new_Longitudine=$this->Arguments_Confiscated[$i] -> getLongitude();      //Longitudine
    $new_ID_Bene=$this->Arguments_Confiscated[$i] -> getID_Bene();           //ID_Bene
    $new_Indirizzo=$this->Arguments_Confiscated[$i] -> getIndirizzo();         //Indirizzo
    $new_Assegnato=$this->Arguments_Confiscated[$i] -> getAssegnato();          //Assegnato
    $new_Descrizione=$this->Arguments_Confiscated[$i] -> getDescrizione();     //Descrizione
    $name_city=$this->Arguments_Confiscated[$i] -> getName_file_csv();
    $countryCode='IT';
    // write contents file assets.owl, locations.owl,geometry.owl, assetsAll.owl
    $converter->convert($i, $new_Latitudine,$new_Longitudine, $new_ID_Bene,$new_Indirizzo,$new_Assegnato,$
    new_Descrizione,$name_city,$countryCode);
```

Figura 13

Al termine di queste operazioni verranno chiusi i tag RDF/XML lasciati aperti dai preamboli inseriti inizialmente e avverrà in fine la chiusura dei file.

```
*****  
* Print footers and close files.  
*/  
$footer=</rdf:RDF>\n\n";
fwrite($assets, $footer);
fwrite($locations, $footer);
fwrite($coordinates, $footer);
fwrite($all, $footer);
fclose($all);
fclose($assets);
fclose($locations);
fclose($coordinates);
```

Figura 14

Classe Parser_Object_confiscated

La classe in oggetto è stata creata adHoc per raggruppare le informazioni necessarie per la completa descrizione del bene confiscato, ma soprattutto per migliorare e agevolare l'estrazione e la manipolazione di queste informazioni durante le operazioni di parsing e trascrizione nel file sopra citati.

Nella Classe è visibile una struttura del tutto dinamica con due metodi, uno per l'estrazione e uno per la configurazione, per variabile che inevitabilmente identifica una informazione saliente del bene confiscato. Tale classe è possibile vederla nella sua completezza nella successive figure.

```
class Parser_Object_confiscated
{
    public $latitude;
    public $longitude;
    public $ID_Bene;
    public $Descrizione;
    public $Assegnato;
    public $Indirizzo;
    public $Name_file_csv;
    public function getName_file_csv()
    {
        echo $Name_file_csv;
        return $this->Name_file_csv;
    }
    public function getLatitude()
    {
        echo $latitude;
        return $this->latitude;
    }
    public function getIndirizzo()
    {
        echo $Indirizzo;
        return $this->Indirizzo;
    }
    public function getLongitude()
    {
        return $this->longitude;
    }
    public function getID_Bene()
    {
        return $this->ID_Bene;
    }
}
```

Figura 15

```
public function getAssegnato()
{
    return $this->Assegnato;
}
public function getDescrizione()
{
    return $this->Descrizione;
}
public function setLatitude($latitude)
{
    $this->latitude=$latitude;
    return $this;
}
public function setLongitude($longitude)
{
    $this->longitude=$longitude;
    return $this;
}
public function setID_Bene($ID_Bene)
{
    $this->ID_Bene=$ID_Bene;
    return $this;
}
public function setAssegnato($Assegnato)
{
    $this->Assegnato=$Assegnato;
    return $this;
}
```

Figura 16

```
public function setDescrizione($Descrizione)
{
    $this->Descrizione=$Descrizione;
    return $this;
}
public function setIndirizzo($Indirizzo)
{
    $this->Indirizzo=$Indirizzo;
    return $this;
}
public function setName_file_csv($Name_file_csv)
{
    $this->Name_file_csv=$Name_file_csv;
    return $this;
}
```

Figura 17

Classe Generate_File_RDF_XML

La classe in oggetto è la classe fondamentale dove avviene la trascrizione del corpo dei quattro file che ricordiamo sono “*assets.owl*”, “*locations.owl*”, “*geometry.owl*” e “*assetsAll.owl*”, non per caso il costruttore della classe ha bisogno, per essere istanziato questi quattro file, il costruttore della classe in questione è visibile in figura.

```
class Generate_File_RDF_XML{
    private $assetsHandle;
    private $locationsHandle;
    private $coordinatesHandle;
    private $allHandle;

    public function __construct($assetsHandle, $locationsHandle, $coordinatesHandle,$allHandle)
    {
        $this->assetsHandle=$assetsHandle;
        $this->locationsHandle=$locationsHandle;
        $this->coordinatesHandle=$coordinatesHandle;
        $this->allHandle=$allHandle;
    }
}
```

Figura 18

Ricordo che l’istanza della classe di cui si sta espletando la struttura la troviamo in figura 13

```
$converter = new Generate_File_RDF_XML($assets, $locations, $coordinates,$all);
```

Figura 19

Le operazioni di scrittura dei file sono state concepite secondo una programmazione dinamica e secondo le regole che l’ingegneria del software ci suggerisce, ovvero metodi piccoli e di semplice interpretazione, quindi vengono riportati sotto i metodi che semplicemente scrivono sui file precedentemente aperti le informazioni sui beni.

```
 /**
 * Write a string to the assets handle
 *
 * @param String $str
 */
private function writeToAssets($str){
    fwrite($this->assetsHandle, $str);
    fwrite($this->allHandle, $str);
}

 /**
 * Write a string to the locations handle
 *
 * @param String $str
 */
private function writeToLocations($str){
    fwrite($this->locationsHandle, $str);
    fwrite($this->allHandle, $str);
}

 /**
 * Write a string to the coordinates handle
 *
 * @param String $str
 */
private function writeToCoordinates($str)
{
    fwrite($this->coordinatesHandle, $str);
    fwrite($this->allHandle, $str);
}
```

Figura 20

Come è visibile dalla figura le funzioni **writeToAssets(\$str);** **writeToLocations(\$str);** **writeToCoordinates(\$str);** scrivono sul rispettivo file Assets.owl, locations.owl e geometry.owl, il contenuto che si trova nella variabile **\$str.**

Il metodo incaricato all'iterativa scrittura delle varie parti dei vari file è la funzione convert che come mostrato in figura:

```
public function convert($i, $latitude, $longitude, $ID_Bene, $Indirizzo, $Assegnato, $Descrizione,$municipality,$countryCode)
```

Figura 21

Prende come parametri un contatore **\$i** per identificare il bene e quindi la locazione dell'array **\$args** in cui è stato memorizzato, **\$latitude** e **\$longitude** che identificano geograficamente insieme all'indirizzo, che si troverà nella variabile **\$Indirizzo**, del bene confiscato, poi troviamo la variabile **\$Assegnato** che come sopracitato informa se il bene è stato assegnato a un'associazione amministratrice o meno, essa è una delle tante modifiche fatte con questa nuova versione rispetto alla precedente, una breve descrizione che troveremo sulla variabile **\$Descrizione** che descrive brevemente qualora ci fosse la struttura e lo stato del bene, **\$ID_Bene** invece riporta in codice univoco che identifica il bene nel comune e infine **\$municipality**, **\$countryCode** che riportano il codice del paese e la municipalità.

In seguito troveremo le figure che esplicitano il codice che si preoccupa di comporre la stringa di caratteri che poi tramite l'apposito metodo descritto prima, in figura 23 è possibile vedere la trascrizione del body del file **Assets.owl**, in tale file si trova la descrizione all'interno dei **rdfs:label** e una **locationURI**, un oggetto complesso formato sostanzialmente da tre informazioni come da esempio es. **Indirizzo_Comune_CountryCode**. Al contempo come Ontologia narra, al momento della trascrizione del bene si deve indicare se il bene è assegnato o meno quindi in base all'informazione riportata dalla variabile **\$Assegnato**, cambia la classe di riferimento e quindi cambiano i tag in figura 22 è possibile notare il risultato di tale operazione con un particolare esempio di bene non assegnato.

```
<ca:UnassignedConfiscatedAsset rdf:about="4">
    <rdfs:label>Fondo rustico</rdfs:label>
    <locn:location rdf:resource="contrada_angimb%8F__calatafimi-segesta_it" />
</ca:UnassignedConfiscatedAsset>
```

Figura 22

```

//***** write the Asset
if(strcmp($Assegnato,"Si"))
{
    $this->writeToAssets("<ca:UnassignedConfiscatedAsset rdf:about=\"$i\"\n");
    $this->writeToAssets("\t<rdfs:label>$Descrizione</rdfs:label>\n");
    $this->writeToAssets("\t<locn:location rdf:resource=\"$locationUri\" /\>\n");
    $this->writeToAssets("</ca:UnassignedConfiscatedAsset>\n");
}
else
{
    $this->writeToAssets("<ca:AssignedConfiscatedAsset rdf:about=\"$i\"\n");
    $this->writeToAssets("\t<rdfs:label>$Descrizione</rdfs:label>\n");
    $this->writeToAssets("\t<locn:location rdf:resource=\"$locationUri\" /\>\n");
    $this->writeToAssets("</ca:AssignedConfiscatedAsset>\n");
}

//put an asset in a location, check whether the location already exists in the ontologies,
//create it if not.
if (array_key_exists($locationUri, $this->locations))
    return;
$locations[$locationUri]=true;

```

Figura 23

Sotto troviamo il codice relativo alla trascrizione del file locations.owl, in esso troviamo come mostra la figura 24 con il risultato di tale operazione, sono visibili tutti i tag interessati nel body e i relativi contenuti.

```

<locn:Location rdf:about="contrada_coriolano__calatafimi-segesta_it">
    <locn:address>
        <locn:Address rdf:about="contrada_coriolano__calatafimi-segesta_it_address">
            <rdfs:label>Contrada Coriolano, Calatafimi-Segesta, IT</rdfs:label>
            <locn:fullAddress>Contrada Coriolano, Calatafimi-Segesta, IT</locn:fullAddress>
            <locn:postName>Calatafimi-Segesta</locn:postName>
            <locn:adminUnitL1>IT</locn:adminUnitL1>
        </locn:Address>
    </locn:address>
    <locn:geometry rdf:resource="contrada_coriolano__calatafimi-segesta_it_geometry" />
</locn:Location>

```

Figura 24

Generato dal seguente codice, in questa sezione troviamo l'elemento \$PointUri che non è nient'altro che il precedente \$locationUri con la concatenazione della stringa "_geometry".

```

//***** write the Locations
$this->writeToLocations("<locn:Location rdf:about=\"$locationUri\"\n");
//at first the address
$pointUri=$locationUri.'_geometry';
$this->writeToLocations("\t<locn:address>\n");
$this->writeToLocations($location->getLOCNAddress($locationUri.'_address'));
$this->writeToLocations("\t</locn:address>\n");
$this->writeToLocations("\t<locn:geometry rdf:resource=\"$pointUri\" /\>\n");
$this->writeToLocations("</locn:Location>\n");
//***** End write the Locations

```

Figura 25

In ultima istanza abbiamo la trascrizione del file `geometry.owl` anch'esso scritto dal sopradetto metodo abilitato a fare ciò, nel file troviamo l'elenco di tutti i beni compresi fra tag :

`<geo:Point rdf:about="Indirizzo __comune_municipalità_geometry">` , di tutte le coordinate GPS dei beni confiscati il risultato di tale operazione è visibile nella figura 26, e generati dal codice visibile in figura 27.

```
<geo:Point rdf:about="contrada_rosignolo__calatafimi-segesta_it_geometry">
    <geo:lat>37.824261</geo:lat>
    <geo:long>12.87573</geo:long>
</geo:Point>
```

Figura 26

```
//***** Write the Geopoints //$/locationUri
$latitude=$this->changeDot($latitude);
$longitude=$this->changeDot($longitude);
    $this->writeToCoordinates("<geo:Point rdf:about=\"$pointUri\\n\"");
    $this->writeToCoordinates("\t<geo:lat>$latitude</geo:lat>\\n");
    $this->writeToCoordinates("\t<geo:long>$longitude</geo:long>\\n");
    $this->writeToCoordinates("</geo:Point>\\n");
//***** End Write the Geopoints
```

Figura 27

Per motivi di standardizzazione la virgola che si trova sistematicamente nelle coordinate GPS, quindi in latitudine e longitudine viene sostituita dal punto, questa operazione è effettuata dal metodo: `changeDot($change_string)`.

```
/** 
 * Change the formatting of longitude and latitude changing the comma with the point
 *
 * @param String $str
 */
private function changeDot($change_string)
{
    $array=explode(',',$change_string);
    $change_string_dot=$array[0].".".$array[1];
    return $change_string_dot;
}
```

Figura28

Scrittura dei file

Una volta terminata l'operazione in cui il parser è in pieno regime di funzionamento, producendo nella cartella relativa al comune per cui si è fatta partire la procedura i file RDF/XML “`assets.owl`”, “`locations.owl`”, “`geometry.owl`” e per ultimo un unico file “`assetsAll.owl`” che li comprenda tutti e che servirà come parte fondamentale per il successivo passo descritto nella sezione successiva.

Per completezza sono stati oggetto del lavoro tutti i file dati da Libera quindi sono stati elaborati tutti e sette i comuni in oggetto, consegnando al Dott.Longo una directory per ogni comune comprensiva di file generati, e una directory in cui sono reperibili le classi per il corretto funzionamento del parserizzatore; tale materiale è reperibile su come è visibile dalla figura 29 (<https://github.com/opendatahacklab/mafiaconfiscatedassets>)

This screenshot shows the GitHub repository page for `opendatahacklab/mafiaconfiscatedassets`. The repository has 63 commits, 3 branches, 0 releases, and 2 contributors. The latest commit was made 23 hours ago. The commits list includes changes for various municipalities: Alcamo, C.Bello Mazzara, Calatafimi_Segesta, Castelvetrano, Class_Parser, Partanna, Salemi, and Vita.

Author	Commit Message	Time Ago
cristianolongo	Merge branch 'master' of https://github.com/opendatahacklab/mafiaconfiscatedassets...	Latest commit 1766836 23 hours ago
Alcamo	Rename Home_Alcamo.html to index.html	23 hours ago
C.Bello Mazzara	Rename Home_C-Bello-Mazzara.html to index.html	23 hours ago
Calatafimi_Segesta	Rename Home_Calatafimi-Segesta.html to index.html	23 hours ago
Castelvetrano	Rename Home_Castelvetrano.html to index.html	23 hours ago
Class_Parser	#25 [Calatafimi] Invalid character in the generated file - added	23 hours ago
Partanna	Rename Home_Partanna.html to index.html	23 hours ago
Salemi	Rename Home_Salemi.html to index.html	23 hours ago
Vita	Rename Home_Vita.html to index.html	23 hours ago

Figura 29

Interfaccia WEB.opendatahacklab-confiscated assets

Ultimo passo per il completamento dell'intero progetto è finalizzare le informazioni raccolte e astrarle all'utenza per renderle fruibili, quindi si intende progettare una pagina Web, fornendo mappa geografica dei beni confiscati con relative informazioni degli stessi, esplicitandone la posizione.

Per fare ciò elemento cardine di questa pagina sono i file che sono stati estratti in precedenza e che verranno palesati all'interno della pagina come editabili tramite un semplice download, così da potere fruire da parte dell'utenza delle informazioni ottenute dai comuni anche in formato RDF/XML, in seguito viene riportata la struttura di uno dei sette comuni in oggetto così da poterne discuterne la struttura.

```
Home_Alcamo.html *
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <title>opendatahacklab - confiscated assets - Alcamo</title>
    <link rel="stylesheet" type="text/css" href="commons/css/odhl.css" />
    <script type="text/javascript" src="sparql_suite/sparql_processor.js"></script>
    <script type="text/javascript" src="sparql_suite/locn_sparql_processor.js"></script>
    <script src="https://maps.googleapis.com/maps/api/js?v=3.exp"></script>
  </head>
  <body>
    <header class="main-header">
      
      <h1>opendatahacklab - confiscated assets - Alcamo</h1>
      <p class="subtitle">THE OPEN DATA HACKING LABORATORY - Powered by
        <a href="http://dydra.com"></a>
        <a class="wt-tech" target="_blank" href="http://wt-tech.it">WT-TECH</a>
      </p>
      <nav>
        <ol class="breadcrumb">
          <li><a href="http://opendatahacklab.org/site/index.html">home</a></li>
          <li><a href="http://opendatahacklab.org/site/projects.html">projects</a></li>
          <li><a href="../index.html">confiscatedassets</a></li>
          <li>Alcamo</li>
        </ol>
        <a href="https://github.com/opendatahacklab/mafiaconfiscatedassets" alt="Source Code" title="GitHub repository">
          </a>
        <a href="data.html"></a>
      </nav>
    </header>
    <p>The <em>ontology of confiscated assets in Alcamo</em> has been produced from
    the <a href="http://www.comune.alcamo.tp.it/in-comune/trasparenza,-valutazione-e-merito-antecedente-il-2013/249-beni-confiscati/945-beni-confiscati-al-19-settembre-2013.html">
    Elenco Beni Confiscati</a> dataset released by the municipality of Alcamo. The file is in PDF format. Thus we had to extract the table,
    converting them in CSV, by the extraction tool <a href="http://tabula.technology/">Tabula</a>.
    Then, this file has been manually updated to solve some issues (for example, keep separate address and civic number).
    Finally, it has been converted in RDF by the <code>Generate_File_RDF_XML.php</code>
    script and inferences have been performed using the reasoner Pellet.
    The ontology is available as OWL downloadable files:</p>
    <ul>
      <li><a href="assets.owl" download><code>assets.owl</code></a></li>
      <li><a href="locations.owl" download><code>locations.owl</code></a></li>
      <li><a href="geometry.owl" download><code>geometry.owl</code></a></li>
    </ul>
  </body>
</html>
```

Figura 30

```

<p>In addition, it can be accessed via its own
<a href="http://dydra.com/cristianolongo/confiscated-assets-alcamo/sparql">SPARQL Endpoint</a>.</p>
<p><em>USAGE NOTE</em>: coordinates reported in <code>geometry.owl</code> and accessible via
the SPARQL Endpoint have been obtained by the google geocoder. As consequence, this information
is not freely reusable, but it is subject to the conditions imposed by google.</p>
<div id="map" class="map" />
<script>
//Create Map
var canvas=document.getElementById("map");
var map= new google.maps.Map(canvas, {
    zoom: 11,
    center: new google.maps.LatLng(37.8689888,12.8121137)
});
//create the processor instance
var processor = new GoogleMapsLOCNProcessor(null, null, map);
//finally run the query fa comparire i tags
sparql_query("http://dydra.com/cristianolongo/confiscated-assets-alcamo/sparql", processor);
</script>
</body>
</html>

```

Figura 31

Nella figura 30 figurano tutti i collegamenti che servono per realizzare il servizio come i collegamenti alla SPARQL SUITE ovvero una suite che descrivono Il protocollo SPARQL (vedi <http://www.w3.org/TR/sparql11-protocol/>) è basato sul protocollo http, le richieste SPARQL vengono inviate agli endpoint come richieste GET o POST e l'endpoint risponde con un esito.

Le richieste si suddividono in richieste di query o update. In caso di richieste di tipo query effettuate con successo, la risposta alla chiamata GET o POST conterrà anche tutte le soluzioni dell'interrogazione in uno dei formati XML, JSON o CSV (il formato di risposta va specificato nella richiesta).

Il protocollo è usato per richiedere tutte le posizioni GPS dei vari beni confiscati, questi vengono reperiti dal file **assetsAll.owl** preventivamente caricato nel server così da modificare dinamicamente, qualora venissero modificati, i beni di quel dato comune.

Si notano tutti i riferimenti a cui le informazioni relative ai beni confiscati sono state reperite quindi è visibile il link al comune di riferimento, sono riportati soprattutto i file assets.owl, location.owl e geometry.owl, che sono presentati in una doppia modalità o fruibili direttamente sul browser come mostrano le immagini in seguito.

Il file XML specificato apparentemente non ha un foglio di stile associato. L'albero del documento è mostrato di seguito.

```
<rdf:RDF xml:base="http://www.comune.alcamo.ip.it/assets/">
  <owl:Ontology rdf:about="http://www.comune.alcamo.ip.it/assets/">
    <rdfs:label>Confiscated assets in Calatafimi-Segesta</rdfs:label>
    <owl:imports rdf:resource="http://www.w3.org/ns/locn"/>
    <owl:imports rdf:resource="http://opendatahacklab.github.io/mafiaconfiscatedassets/confiscatedassets_vocab.owl"/>
  </owl:Ontology>
  <!-- Generated by ParserCSV.php & Generate_File_RDF_XML on 18-06-2016-11-18 -->
  <ca:AssignedConfiscatedAsset rdf:about="#1">
    <rdfs:label>Appartamento</rdfs:label>
    <locn:location rdf:resource="via_del_carroccio_n_17-17_scala_b_interno_4_piano_2%2C%A1__alcamo_it"/>
  </ca:AssignedConfiscatedAsset>
  <ca:AssignedConfiscatedAsset rdf:about="#2">
    <rdfs:label>Appartamento</rdfs:label>
    <locn:location rdf:resource="via_alessandro_volta_n_18_piano_j%2C%A1__alcamo_it"/>
  </ca:AssignedConfiscatedAsset>
  <ca:AssignedConfiscatedAsset rdf:about="#3">
    <rdfs:label>
      Appartamento di cui il piano terra è costituito da 2 vani e da accessori per mq 72, mentre il piano ammezzato con vano e camerino occupa una superficie di mq 39
    </rdfs:label>
    <locn:location rdf:resource="via_porta_stella_n_41-43_piano_terra_e_piano_1%C2%A1__alcamo_it"/>
  </ca:AssignedConfiscatedAsset>
  <ca:AssignedConfiscatedAsset rdf:about="#4">
    <rdfs:label>
      Appartamento che occupa una superficie di mq 23 a piano terra oltre mq 14 a piano ammezzato
    </rdfs:label>
    <locn:location rdf:resource="via_xi_febbraio_n_5_piano_terra__alcamo_it"/>
  </ca:AssignedConfiscatedAsset>
  <ca:AssignedConfiscatedAsset rdf:about="#5">
    <rdfs:label>Appartamento a quattro elevazione fuori terra</rdfs:label>
    <locn:location rdf:resource="via_xi_febbraio_n_14%2C16%2C18__alcamo_it"/>
  </ca:AssignedConfiscatedAsset>
  <ca:UnassignedConfiscatedAsset rdf:about="#6">
    <rdfs:label>
      Lotti di terreni destinati alla pubblica viabilità e parcheggio pubblico (Via Girolamo Giardina)
    </rdfs:label>
    <locn:location rdf:resource="via_girolamo_giardina_gi%C2%88_c_da_stazzone_s_ippolito__alcamo_it"/>
  </ca:UnassignedConfiscatedAsset>
  <ca:AssignedConfiscatedAsset rdf:about="#7">
    <rdfs:label>
      Fabbricato composto da quattro piani fuori terra sito nella via Silvio Tornamira n. 27, 27/A e 29 consistente in un appartamento a piano terra, con accesso anche dalla via Gibilrossa N. 28, di mq 64 circa con magazzino di mq 13 circa, in un altro
    </rdfs:label>
```

Figura 32

Il file XML specificato apparentemente non ha un foglio di stile associato. L'albero del documento è mostrato di seguito.

```
<rdf:RDF xml:base="http://www.comune.alcamo.ip.it/assets/">
  <owl:Ontology rdf:about="http://www.comune.alcamo.ip.it/assets/">
    <rdfs:label>Locations of confiscated assets in Calatafimi-Segesta
    </rdfs:label>
    <owl:imports rdf:resource="http://www.w3.org/ns/locn"/>
    <owl:imports rdf:resource="http://opendatahacklab.github.io/mafiaconfiscatedassets/confiscatedassets_vocab.owl"/>
  </owl:Ontology>
  <!-- Generated by ParserCSV.php & Generate_File_RDF_XML on 18-06-2016-11-18 -->
  <locn:Location rdf:about="via_del_carroccio_n_17-17_scala_b_interno_4_piano_2%2C%A1__alcamo_it">
    <locn:address>
      <locn:Address rdf:about="via_del_carroccio_n_17-17_scala_b_interno_4_piano_2%2C%A1__alcamo_it_address">
        <rdfs:label>
          Via del Carroccio n. 17-17 Scala B Interno 4 Piano 2i, Alcamo, IT
        </rdfs:label>
        <locn:fullAddress>
          Via del Carroccio n. 17-17 Scala B Interno 4 Piano 2i, Alcamo, IT
        </locn:fullAddress>
        <locn:postName>Alcamo</locn:postName>
        <locn:adminUnit>IT</locn:adminUnit>
        <locn:Address>
        </locn:address>
        <locn:geometry rdf:resource="via_del_carroccio_n_17-17_scala_b_interno_4_piano_2%2C%A1__alcamo_it_geometry"/>
    </locn:Location>
    <locn:Location rdf:about="via_alessandro_volta_n_18_piano_j%2C%A1__alcamo_it">
      <locn:address>
        <locn:Address rdf:about="via_alessandro_volta_n_18_piano_j%2C%A1__alcamo_it_address">
          <rdfs:label>Via Alessandro Volta n. 18 Piano 1i, Alcamo, IT</rdfs:label>
          <locn:fullAddress>Via Alessandro Volta n. 18 Piano 1i, Alcamo, IT</locn:fullAddress>
          <locn:postName>Alcamo</locn:postName>
          <locn:adminUnit>IT</locn:adminUnit>
          <locn:Address>
          </locn:address>
          <locn:geometry rdf:resource="via_alessandro_volta_n_18_piano_j%2C%A1__alcamo_it_geometry"/>
        </locn:Location>
        <locn:Location rdf:about="via_porta_stella_n_41-43_piano_terra_e_piano_1%C2%A1__alcamo_it">
          <locn:address>
            <locn:Address rdf:about="via_porta_stella_n_41-43_piano_terra_e_piano_1%C2%A1__alcamo_it_address">
              <rdfs:label>
                Via Porta Stella n. 41-43 Piano terra e piano 1i; Alcamo, IT
              </rdfs:label>
```

Figura 33

Il file XML specificato apparentemente non ha un foglio di stile associato. L'albero del documento è mostrato di seguito.

```

<- <rdf:RDF xml:base="http://www.comune.alcamo.tp.it/assets/">
  <- <owl:Ontology rdf:about="http://www.comune.alcamo.tp.it/assets/">
    <- <rdfs:label>
      Coordinates of confiscated assets in Calatafimi-Segesta
    </rdfs:label>
    <owl:imports rdf:resource="http://www.w3.org/2003/01/geo/wgs84_pos#"/>
    <owl:imports rdf:resource="http://opendatahacklab.github.io/maffaconfiscatedassets/confiscateddatasets_vocab.owl"/>
  </owl:Ontology>
<- <!-- Generated by ParserCSV.php & Generate_File_RDF_XML on 18-06-2016-11-18
-->
<- <geo:Point rdf:about="via_del_carroccio_n_17-17_scala_b_interno_4_piano_2%2C%2A1__alcamo_it_geometry">
  <geo:lat>37.979139</geo:lat>
  <geo:long>12.954397</geo:long>
</geo:Point>
<- <geo:Point rdf:about="via_alessandro_volta_n_18_piano_i%C2%A1__alcamo_it_geometry">
  <geo:lat>37.980461</geo:lat>
  <geo:long>12.965357</geo:long>
</geo:Point>
<- <geo:Point rdf:about="via_porta_stella_n_41-43_piano_terra_e_piano_1%C2%A1__alcamo_it_geometry">
  <geo:lat>37.980903</geo:lat>
  <geo:long>12.965481</geo:long>
</geo:Point>
<- <geo:Point rdf:about="via_xi_febbraio_n_5_piano_terra__alcamo_it_geometry">
  <geo:lat>37.980463</geo:lat>
  <geo:long>12.965767</geo:long>
</geo:Point>
<- <geo:Point rdf:about="via_girolamo_giardina_gi%C2%88_c_da_stazzone_s_ippolito__alcamo_it_geometry">
  <geo:lat>37.979539</geo:lat>
  <geo:long>12.948659</geo:long>
</geo:Point>
<- <geo:Point rdf:about="via_silvio_tormamira_n_27%2C_27%2Fa_e_29%2C_accessible_anche_dalla_via_gilbirossa_n_28__alcamo_it_geometry">
  <geo:lat>37.976741</geo:lat>
  <geo:long>12.962941</geo:long>
</geo:Point>
<- <geo:Point rdf:about="via_benedetto_croce_angolo_via_virgilio__alcamo_it_geometry">
  <geo:lat>37.972638</geo:lat>
  <geo:long>12.961248</geo:long>

```

Figura 34

Oppure viene data la possibilità di essere scaricati nel proprio dispositivo come mostrano le figure.



The ontology of confiscated assets in Alcamo has been produced from the [Elenco Beni Confiscati](#) dataset released by the municipality of Alcamo. The file is in PDF format. Thus we had to extract the table, converting them in CSV, by the extraction tool [Tabula](#). Then, this file has been manually updated to solve some issues (for example, keep separate address and civic number). Finally, it has been converted in RDF by the [Generate_File_RDF_XML.php](#) script and inferences have been performed using the reasoner Pellet. The ontolo

Apertura di assets.owl

È stato scelto di aprire:

- assets.owl
- locations.owl
- geometry.owl

tip: Extensible Markup Language (7,4 kB)

da: /Applications/MAMP/htdocs/Alcamo

Che cosa deve fare Firefox con questo file?

Aprirlo con Scegli...

Salva file

Da ora in avanti esegui questa azione per tutti i file di questo tipo.

consequence, this information is not freely reusable, but it is subject to the



The ontology of confiscated assets in Alcamo has been produced from the [Elenco Beni Confiscati](#) dataset released by the municipality of Alcamo. The file is in PDF format. Thus we had to extract the table, converting them in CSV, by the extraction tool [Tabula](#). Then, this file has been manually updated to solve some issues (for example, keep separate address and civic number). Finally, it has been converted in RDF by the [Generate_File_RDF_XML.php](#) script and inferences have been performed using the reasoner Pellet. The ontolo

Apertura di locations.owl

È stato scelto di aprire:

- assets.owl
- locations.owl
- geometry.owl

tip: Extensible Markup Language (15,3 kB)

da: /Applications/MAMP/htdocs/Alcamo

Che cosa deve fare Firefox con questo file?

Aprirlo con Scegli...

Salva file

Da ora in avanti esegui questa azione per tutti i file di questo tipo.

consequence, this information is not freely reusable, but it is subject to the



In addition, it can be accessed via its own [SPARQL Endpoint](#).

USAGE NOTE: coordinates reported in [geometry.owl](#) and accessible via the SPARQL endpoint.

conditions imposed by google.

USAGE NOTE: coordinates reported in [geometry.owl](#) and accessible via the SPARQL endpoint.

conditions imposed by google.

In addition, it can be accessed via its own [SPARQL Endpoint](#).

USAGE NOTE: coordinates reported in [geometry.owl](#) and accessible via the SPARQL endpoint.

conditions imposed by google.

USAGE NOTE: coordinates reported in [geometry.owl](#) and accessible via the SPARQL endpoint.

conditions imposed by google.

In addition, it can be accessed via its own [SPARQL Endpoint](#).

USAGE NOTE: coordinates reported in [geometry.owl](#) and accessible via the SPARQL endpoint.

conditions imposed by google.

USAGE NOTE: coordinates reported in [geometry.owl](#) and accessible via the SPARQL endpoint.

conditions imposed by google.

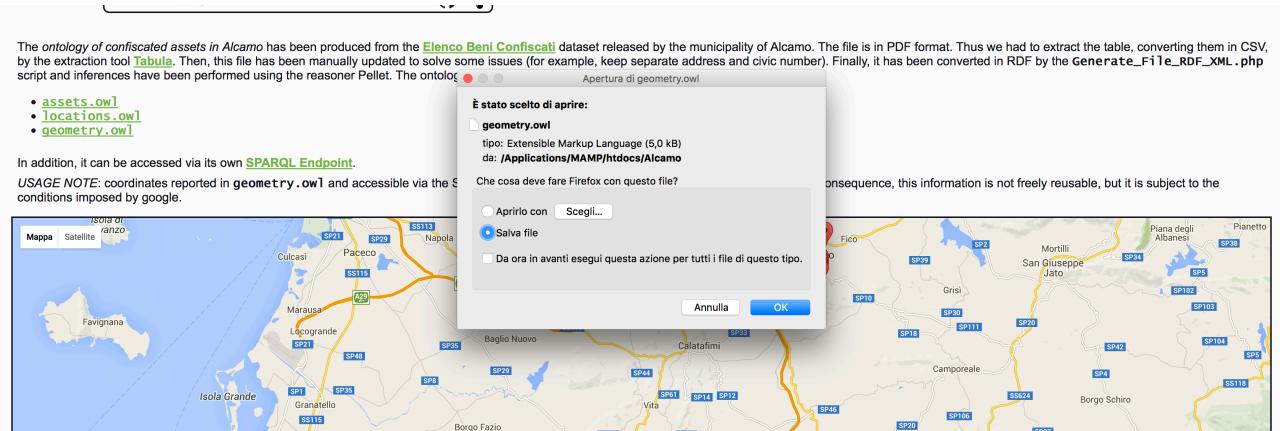


Figura 34

La figura 31, mostra invece lo script responsabile della visualizzazione della mappa del comune in oggetto che per motivi puramente dimostrativi è stato preso in esempio il comune di Alcamo; lo script consiste nello stampare a video la mappa geografica del comune tramite le coordinate GPS e uno zoom dimensionale così da poter tarare la risoluzione di quanto vedere del comune, successivamente viene istanziato il **processor** di tipo **GoogleMapsLOCNProcessor** ovvero l'oggetto responsabile della chiamata **SPARQL** che verrà successivamente effettuata con la successiva istruzione: `sparql_query("http://dydra.com/cristianolongo/confiscated-assets-alcamo/sparql", processor);`

Conclusioni

Concludendo, il progetto è stato realizzato in tutte le sue sfaccettature eseguendo con completezza tutti i punti ovvero:

1. Analizzare la sorgente dati relativa alle informazioni date dai comuni sui beni confiscati alla mafia, redigendo un report sulla consistenza e conformità.
2. Progettazione ed elaborazione di un'analisi sintattica automatizzata di file CSV, con scopo ultimo quello di costruire 4 file di tipo RDF/XML.
3. Progettazione di una pagina Web con scopo illustrativo dei beni sequestrati dalla mafia, fornendo mappa geografica dei beni confiscati.

L'ultima immagine mostra il risultato del progetto relativi a due dei sette comuni relazionati.



opendatahacklab - confiscated assets - Alcamo

THE OPEN DATA HACKING LABORATORY - Powered by WT-TECH

>> home >> projects >> confiscatedassets >> Alcamo

The ontology of confiscated assets in Alcamo has been produced from the [Elenco Beni Confiscati](#) dataset released by the municipality of Alcamo. The file is in PDF format. Thus we had to extract the table, converting them in CSV, by the extraction tool [Tabula](#). Then, this file has been manually updated to solve some issues (for example, keep separate address and civic number). Finally, it has been converted in RDF by the [Generate_File_RDF_XML.php](#) script and inferences have been performed using the reasoner Pellet. The ontology is available as OWL downloadable files:

- [assets.owl](#)
- [locations.owl](#)
- [geometry.owl](#)

In addition, it can be accessed via its own [SPARQL Endpoint](#).

USAGE NOTE: coordinates reported in [geometry.owl](#) and accessible via the SPARQL Endpoint have been obtained by the google geocoder. As consequence, this information is not freely reusable, but it is subject to the conditions imposed by google.



opendatahacklab - confiscated assets - Calatafimi-Segesta

THE OPEN DATA HACKING LABORATORY - Powered by WT-TECH

>> home >> projects >> confiscatedassets >> Calatafimi-Segesta

The ontology of confiscated assets in Calatafimi-Segesta has been produced from the [Elenco Beni Confiscati](#) dataset released by the municipality of Calatafimi-Segesta. The file is in PDF format. Thus we had to extract the table, converting them in CSV, by the extraction tool [Tabula](#). Then, this file has been manually updated to solve some issues (for example, keep separate address and civic number). Finally, it has been converted in RDF by the [Generate_File_RDF_XML.php](#) script and inferences have been performed using the reasoner Pellet. The ontology is available as OWL downloadable files:

- [assets.owl](#)
- [locations.owl](#)
- [geometry.owl](#)

In addition, it can be accessed via its own [SPARQL Endpoint](#).

USAGE NOTE: coordinates reported in [geometry.owl](#) and accessible via the SPARQL Endpoint have been obtained by the google geocoder. As consequence, this information is not freely reusable, but it is subject to the conditions imposed by google.

