A Methodology for File Relationship Discovery

Michal Ondrejcek, Jason Kastner, Rob Kooper, and Peter Bajcsy
National Center for Supercomputing Applications, University of Illinois, Urbana, USA
ondrejce@illinois.edu, jkastner@ncsa.uiuc.edu, kooper@ncsa.uiuc.edu, pbajcsy@ncsa.uiuc.edu

Abstract **- This paper addresses the problem of discovering temporal and contextual relationships across document, data, and software categories of electronic records. We designed a methodology to discover unknown relationships by conducting file system and file content analyses. The work also investigates automation of metadata extraction from engineering drawings and storage requirements for metadata extraction. The methodology has been applied to extracting information from a test collection of electronic records about the NAVY ship (TWR 841) archived by the US National Archive (NARA). This test collection represents a problem of unknown relationships among files that include 784 2D image drawings and 22 CAD models.**

Keywords **- Data processing, Data conversion, Optical character recognition**

I. INTRODUCTION

One of the major initiatives defined by the US National Archives is to preserve electronic records for the future [1]. Our work supports the initiative of the US National Archives and is part of a larger effort to automatically discover relationships and similarities among different versions of data, documents and software in order to support preservation decisions made by archivists. While we design a general methodology for the larger problem of discovering file relationships, the application of the methodology is to a specific collection of electronic records consisting of scanned 2D engineering drawings and 3D Computer Aided Design (CAD) models. Specifically, our approach leverages the semi-systematic representation of important information in normalized blocks, such as Title, References, and Drawing number blocks, of the engineering drawings also known as the blueprints, and the specifications of 3D CAD file formats. Thus, overlapping information about files could be obtained not only from the file system topology but also from the file content.

There are several challenges in automated processing of engineering drawings. For example, the drawings are scanned at low resolution (<300dpi) and hence character recognition is error-prone. The information blocks take on many different shapes or are only partially normalized leading to difficulties in template-based recognition. The lines of information blocks are frequently skewed and/or with gaps due to scanning. Finally, the text in information blocks is a combination of freehand lettering and pre-printed characters without a standard font (uppercase Gothic has been observed as the preferred font). These challenges have to be overcome during automated processing and the processing workflow often includes steps such as (a) information block detection in an engineering drawing, (b) recognition of information block type, (c) localization of sub-fields of an information block, (d) character recognition of the sub-field entries, (e) classification of extracted sub-field entries (taxonomy and ontology), (f) representation and storage of extracted entries referred to as metadata, and (g) visualization of sub-fields and extracted information for the purposes of recognition accuracy evaluation, correction and linking with other information.

Najman et al. used form-processing method for automating the indexing of title blocks [2]. There have been similar approaches reported in [3][4] about (a) indexing keyword fields extracted from the title blocks, and (b) automation of the detection of title blocks in the engineering drawing based on location hashing. Several software solutions have been developed in the past for metadata visualization [5][6] that use the Resource Description Framework (RDF) format for representation [7][8]. However, these solutions lack capabilities for comparing sub-field images and recognized characters or multiple entries extracted by character recognition. Knowing that the character recognition does not reach 100% accuracy, these solutions also do not support advanced editing, and filtering needed to perform corrections and to support semi-automated decisions about file relationships and document matching.

Our ultimate goal is to discover file relationships automatically given a set of files. The general methodology is illustrated in Figure 1. At the high level, the methodology consists of (1) file identification and content classification phase (the left

column of decisions about the file category), (2) extraction of file system based information (top box), (3) extraction of content based information (the middle box), (4) management of extracted information/ metadata (the boxes with red labels) and (5) services and exploratory interfaces to verification, validation, editing and file relationship discoveries (the bottom right boxes). This methodology feeds the application such as the archival appraisal and management of the files in this case (the bottom center box).

In our work, we demonstrate the general methodology by applying it to concrete data sets consisting of engineering drawings and 3D CAD models and matching metadata extracted from these file types. Thus, the prototype system for file relationship discovery includes specific components for (a) optical character recognition (OCR) of the information fields in 2D engineering drawings, (b) searching 3D file formats to find specific metadata about the content, and (c) file relationship discovery. The contribution of our work lies in the design of a general methodology for file discovery, and in the demonstration how the methodology can be applied to a specific collection of inter-related 2D and 3D data sets leading to an increased efficiency in management and archival of electronic records.

II. TEST DATA COLLECTION

We identified a collection of electronic records that correspond to engineering drawings (2D images) and 3D CAD models of NAVY ships. These electronic records come from the Records of the Office of the Chief of Naval Operations and from the General Records of the Department of the Navy. This test collection presents a problem of unknown relationships among files, which currently includes 784 engineering drawings for the Torpedo Weapon Retriever (TWR 841), and about 22 CAD models.

III. METHODOLOGY

The general methodology to solve the problem of file relationship discovery is shown in Fig.1 and it was described in the introduction section. The prototype system based on the general methodology consists of the following implementations. The file identification and metadata extraction about file formats uses DROID [9], the file format identification scheme that leverages the PRONOM [10] file format registry. We extended the DROID functionality to include 3D file format identification which is currently very sparsely supported by PRONOM. Our extension could be viewed as a plug-in called the NCSA 3D file registry [11][12]. The file system based extraction is implemented using the Aperture JAVA framework [13]. The optical character recognition (OCR) of information blocks in 2D engineering drawings is performed by a commercial solution developed by ABBYY. The automation of the OCR execution is achieved by an AutoHotKeys script. The metadata management coming from OCR is accomplished by a Java program with the pre-defined ontology for entries in information blocks. The Java program represents all metadata as RDF triples [7][8] and calls the Tupelo semantic content management system [14] to store the data in a context (e.g., mySQL database). Finally, the original files and the metadata extracted are visualized in a custom browser that supports verification, validation, metadata editing and viewing of images and 3D CAD Models. The browser is built on top of the Eclipse Rich Client Platform (RCP) and hence supports reconfigurable user interfaces for viewing and interactions. Details about individual components are provided next.

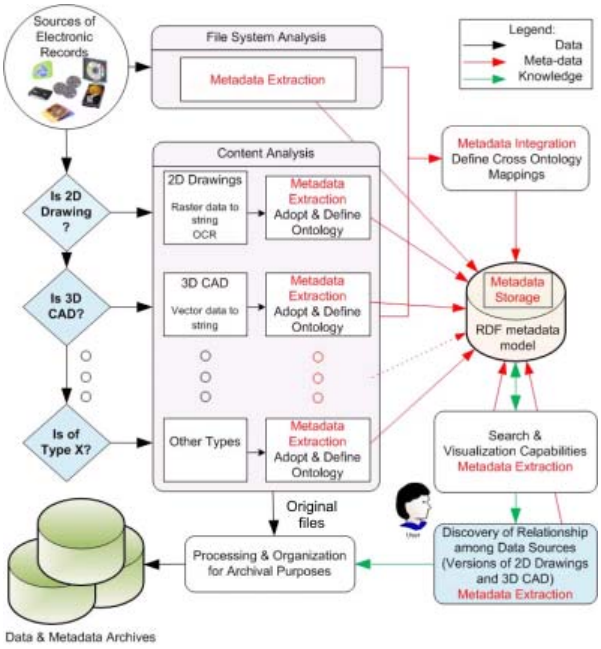


Figure 1. An overall design to discovering relationships among multiple sources of electronic records.

IV. PROTOTYPE COMPONENTS

A. *Metadata extraction from file systems*

We have conducted a survey of existing software packages that extract and save metadata from file systems. There are several existing utilities for extracting metadata about file systems specific to each

operating system (OS). For instance, the utilities on Mac OS include: Spotlight (since Mac OS 10.4). We have also found specialized software for this purpose following the RDF data model and decided to explore the Aperture Framework [13]. Aperture is a Java framework for extracting and querying full-text content and metadata from various information systems (e.g. file systems, web sites, mail boxes) and the file formats (e.g. documents, images) occurring in these systems. It saves the metadata following the Nepomuk ontologies [13].

In order to identify file formats and obtain additional metadata, we developed a tool that calls the DROID application programming interface (API). The result from DROID is metadata about each file including the registered PRONOM universal identification (ID). We convert the metadata into RDF triples and store them. If file formats are not supported by the current version of PRONOM (e.g., several 3D file formats) then DROID returns the unidentified file format flag. Those files are then checked against an internal 3D file registry called Polyglot [11][12]. The results are again converted into RDF triples and stored.

The collected metadata is then used to discover relationships among files. Each piece of information is represented as a RDF triple (subject-predicate-object) with a Universally Unique Identifier (UUID) for each file obtained from the PRONOM repository and predicates using custom vocabularies, or ontologies, to represent metadata from each of the extraction programs (Aperture, DROID, NCSA OCR of title blocks, NCSA 3D file analysis code). Table 1 shows the RDF triples generated for a file containing a 3D model in Autocad file format (dwg). The UUID is used as a key for storing a set of RDF triples about the same file. The specific PRONOM ID is highlighted.

TABLE 1. RDF TRIPLES (SUBJECT-PREDICATE-OBJECT) GENERATED FOR A FILE CONTAINING A 3D MODEL IN AUTOCAD FILE FORMAT.

Subject	Predicate	Object
<tag://'path',2009: file/'UUID1>	<http://'path'/2009/droid/ IdFile/hasFileName>	U2110_BHD12 _Autocad.dwg
<tag://'path',2009: file/'UUID1>	<http://'path'/2009/droid/ IdFile/hasIdentQuality>	Positive
<tag://'path',2009: file/'UUID1>	<http://'path'/2009/droid/ IdFile/FileFormatHit/hasF ormatName>	AutoCAD Drawing
<tag://'path',2009: file/'UUID1>	<http://'path'/2009/droid/ IdFile/FileFormatHit/hasF ormatVersion>	2004-2005
<tag://'path',2009: file/'UUID1>	<http://'path'/2009/droid/ IdFile/FileFormatHit/hasPr onomId>	http://www.nati onalarchives.go v.uk/pronom/f mt/36
<tag://'path',2009: file/'UUID1>	<http://'path'/2009/droid/ IdFile/FileFormatHit/hasM imeType>	image/vnd.dwg

B. *Metadata extraction from engineering drawings*

The title block is always located in the lower right corner of an engineering drawing. It should include sufficient information to identify the type of drawing, vendor name and information about the versions, authors and other creators. A title block is divided into several areas, fields, most importantly the drawing title and the drawing number. The drawing number may also contain information such as the sheet number, as part of a series. Other areas usually contain the signatures and approval dates. Most engineering drawings also have a reference block, which lists other drawings that are related to the system. Through the title and reference blocks cross-linking is possible. Other information blocks are list of materials, revisions, and so on.

Information blocks and especially the title block are standardized to some extent following the recommendations of American National Standard Engineering and Related Document Practices (ASME Y14/ANSI Y14) [15], International Standard Organization [16] (ISO 128 and 7200) rules or in this particular case DoD-STD-100C [17] for NAVY standards.

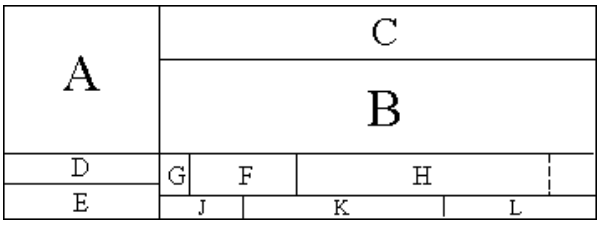
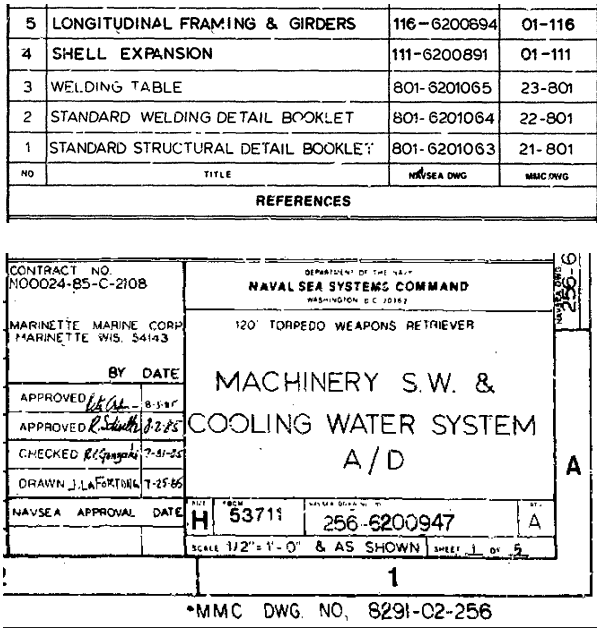


Figure 2. Title block template used in NAVY showing various information fields [17]: (A) vendor, record of preparation, (B) drawing title, (C) preparing activity, (D)(E) parts are not standardized, (F) Code identification number, (G) drawing size, (H) drawing number, (J) scale, (K) specification number and (L) sheet number.

Examples of a title block template with information fields is shown in Fig. 2, the actual title block and a revision block are presented in Fig. 3. By studying the (NAVY) standards for forming such blocks, we identified about five major templates; the main title block for various drawing sizes, continuation block of the subset pages, references block and we also extract the general drawing number (DWG NO string).

Optical character recognition (OCR) is a method for extracting the text information from images. We have first surveyed existing software packages and evaluated their cost and accuracy. The software included ABBYY FineReader [18], IRIS ReadIris [19],

low end TopOCR [20] and open source Gamera [21]. Based on OCR software performance especially on scanned templates with low resolution (< 300 dpi), we selected OCR software by ABBYY.



NO	TITLE	NAVSEA DWG	MMC DWG
5	LONGITUDINAL FRAMING & GIRDERS	116-6200694	01-116
4	SHELL EXPANSION	111-6200891	01-111
3	WELDING TABLE	801-6201065	23-801
2	STANDARD WELDING DETAIL BOOKLET	801-6201064	22-801
1	STANDARD STRUCTURAL DETAIL BOOKLET	801-6201063	21-801

REFERENCES

Figure 3. Examples of a reference (top) and a title block (bottom) in a 2D technical drawing file. Text extracted from various sub-areas (mainly the drawing numbers) were used for establishing relationships among the documents.

In order to discover relationships among files, the position of the blocks within each drawing has to be analyzed in addition to the file system information. The position of the blocks has been done manually by recording the coordinates of upper left and bottom right corners. The block sizes were run against the pre-defined templates and the OCR was applied. The content analysis has to be automated since the volume of engineering drawings is too large for manual transcription.

The problem has been decomposed into the following sub-tasks:

1. Develop templates and ontology for all fields in information blocks.
2. Detect the information block in an image and crop the area.
3. Identify the type of information block template and crop sub-areas for each information field.
4. Apply OCR software to each field and associate it with text strings resulting from OCR.
5. Perform cleansing of the text strings extracted by OCR according to the ontology
6. Represent text strings as metadata RDF triples and store them in a shared repository.

Multiple metadata from the drawings' blocks have been generated. The corresponding ontologies (Technical drawing/Blocks/NAVY) for the proper RDF implementation have not been found through open sources. The custom ontology for drawings (tdrw) has been developed by utilizing as much as possible the standards, such as the RDF vocabulary (rdf), Dublin Core Metadata Element Set (dc) and Friend of a Friend project (foaf) [22]. Fig. 4 shows snapshot of the metadata description from the title block, in a RDF/XML representation using tdrw ontology.

```
<rdf:Description rdf:about="uid:1249583951983/titleblock/1" rdf:type="tdrw:TitleBlock">
        <tdrw:rdfAuthor rdf:resource="'path'/AboutUs/People/contact.php?id=1016"/>
        <tdrw:recordOfPreparation>
                CONTRACT NO, N00024-85-C-2108,  MARINETTE MARINE CORP, MARINETTE WIS 54143
        </tdrw:recordOfPreparation>
        <tdrw:information>NAVSEA</tdrw:information>
        <tdrw:isPreparingActivity>
                DEPARTMENT OF THE NAVY, NAVAL SEA SYSTEMS COMMAND, WASHINGTON DC 20362
        </tdrw:isPreparingActivity>
        <tdrw:drawingTitle>
                120 TORPEDO WEAPONS RETRIEVER, TRANSVERSE BULKHEADS BELOW MAIN DOCK
        </tdrw:drawingTitle>
        <tdrw:drawingSize>H</tdrw:drawingSize>
        <tdrw:FSCMNumber>53711</tdrw:FSCMNumber>
        <tdrw:drawingNumber1>117-6200895</tdrw:drawingNumber1>
        <tdrw:drawingNumber2>A</tdrw:drawingNumber2>
        <tdrw:drawingScale>1/2'-1'-0" AND AS SHOWN</tdrw:drawingScale>
        <tdrw:sheetNumber>1 OF 3</tdrw:sheetNumber>

        <tdrw:hasMiniBox rdf:resource="uid:1249583951983/approved/1"/>
        <tdrw:hasMiniBox rdf:resource="uid:1249583951983/approved/2"/>
        <tdrw:hasMiniBox rdf:resource="uid:1249583951983/checked/1"/>
        <tdrw:hasMiniBox rdf:resource="uid:1249583951983/drawn/1"/>
</rdf:Description>
```

Figure 4. Part of the RDF/XML metadata description using technical drawing (tdrw) ontology.

Current implementation of the OCR framework uses the NCSA ImageToLearn [23] library written in Java for cropping, ABBYY software (a desktop application) for OCR, RDF parser library and NCSA Tupelo as a metadata repository. The automation is achieved by developing Java code that crops an information block area using Im2Learn library and creates sub-areas with title block fields, by calling an AutoHotKeys script to launch ABBYY OCR software and by creating RDF triple metadata representation and saving the metadata into a repository using custom Java code. We use ABBYY FineReader 9.0 OCR package for the Optical Character Recognition. The FineReader is a one computer standalone version of ABBYY OCR engine.

C. A framework supporting discovery of relationships

A custom tool called File To Learn has been specifically created for the comparison of two files in the metadata store, was then created to allow users to explore the metadata store. It consists of a visualization tool that displays RDF triples of two selected files to find intersecting values. There is a table with color-coded display to highlight shared

values. There are preview panes to show 3D models and also preview 2D engineering drawings to allow a user to confirm that a 3D model was sourced from a specific engineering draft.

The tool also gives the ability, after investigation using the previously mentioned functions to create new RDF triples for specific files in order to create logical links, or relationships, between files.

V. Experimental Results

A. Storage requirement for metadata extraction

In order to understand storage requirements for metadata extraction, experiments with diverse common file systems have been performed. The goal is to estimate the storage size dependencies on the number of files and folders in a file system, and the configuration of files and folders.

Test experiments have been conducted with synthetically created files systems and with real file systems. We hypothesize that there are two extreme distributions of files and folders in a file system: (1) a pyramid of evenly nested folders and equally distributed files in folders, and (2) one long nested sequence of folders with all files located at the deepest folder. Specifically, the former is a pyramid of evenly distributed folders with five text and five small image tif files in the deepest folder. The latter is one long nested sequence of folders with all files in the deepest folder. The file systems consisted of up to 100 000 files (total 11111 folders) with known file size, and a constant ratio 1:10 between the number of folder and the number of files. This ratio is an average computed over statistically significant number of computer systems in the Image Spatial Data Analysis group at NCSA. The Aperture framework was run on two systems with following hardware specifications: LINUX OS system denoted c1; 8 CPU Intel Xeon with 2.5GHz and 8GB RAM; and Windows XP OS (c2); 1 CPU 2GHz Intel and 2GB RAM.

As stated above the assumptions here are; approximately 1:10 folder to file ratio and only the MIME data file identification, in our case TIFF and text files. Also the Aperture generated triples have full name space and full ontology path.

Fig. 5 shows the experimental results for the synthetic case marked with the dotted blue line and for the real file systems (blue and red circles). As in the ideal case only the file-system data structure and type of the files (MIME extractor) have been analyzed. It is surprising that the real data follows the ideal case even with different folder to file ratios. In the real case the total size of the data folders is obviously much different than in the ideal case since it depends on the data themselves (large size images, audio, video etc.). In conclusion, each file generates about 1.8 kB of metadata (i.e. 10 000 files generates 18.2 MB) and approximately 10 RDF triples.

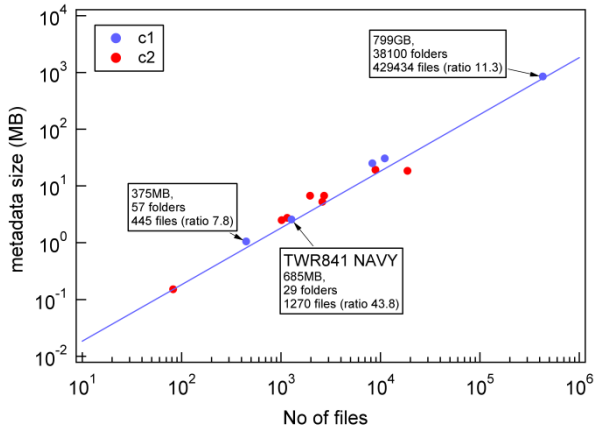


Figure 5. Metadata size as a function of number of files.

Additionally to the amount of triples extracted from the file system, 5 to 10 RDF triples is routinely extracted from PRONOM by DROID; and about 60 RDF triples per full title block and ~15 per continuation title and reference blocks extracted from 2D engineering drawings by OCR. Table 2 summarizes the number of RDF triples generated per file.

Table 2. Total number of RDF triples generated per file

Source of metadata	# RDF triples per file
File system	~10
DROID identification	5 - 10
Information blocks	15 - 60
3D files	5 - 10
Total	**35 - 90**

In a hypothetical case, for a medium size server with ~400 000 files of similar origin, drawings and 3D files this would lead to more than 25 million RDF triples for an average of 63 RDF triples per file. This empirical observation highlights the importance of scalable solutions in terms of storage representation and information retrieval.

B. Relationships discovery

Total amount of 784 2D drawings in TIFF file format have been identified which include 170 title blocks, 700 continuation title blocks, 150 reference blocks, a dozen of revision and list of material blocks and about 200 additional areas with the drawing numbers. The OCR accuracy drops down to 80% for

title blocks characters in certain drawings mainly due to poor original scans quality. The most difficult parts for OCR are the fields with handwritten information such as names and dates and fields with measuring units. The most important fields with drawing numbers and drawing references (digits) are much better resolved. The reported accuracy is comparable to the numbers reported in [2] (95% accuracy for the printed text and 68% for the hand-printed text).

Additionally to the drawing files 22 3D CAD models of the same system have been analyzed by extracting the metadata portion of the file specification.

All RDF triples are stored in metadata repository through the Tupelo metadata management system based on semantic web technologies [14]. The advantage of this solution is that Tupelo bridges various applications (desktop, web-based etc.) on one side and the underlying storage tools and technologies (MySQL, PostGIS databases for example) on the other side. A File To Learn framework accesses the triples again through Tupelo.

Following challenges in discovering relationships have been encountered so far.

First, there is a lack of evidence (or insufficient metadata information) extracted by the currently used metadata extraction software. This challenge has to be overcome by adding additional, primarily content-based, analytical tools for extracting more information for establishing file-to-file relationships. Table 3 shows an example of insufficient overlapping information in the case of the metadata gathered from 3D CAD files in STEP file format and from 2D engineering drawings in TIFF file format. Left column represents the STEP metadata specification, center and right columns show the same information extracted from the STEP 3D file and 2D title block of the same object. There are significant discrepancies of the same descriptor values. Although the 3D CAD files were created later in time from the engineering drawings (mainly manually), there is no overlapping information found in the 3D file and in the title blocks of engineering drawings. It is noted that this problem appeared mainly between 2D and 3D datasets. Relationships among 2D drawings themselves based on the matching drawing and project numbers in the title block and reference blocks have been established and visualized.

Second, there are multiple predicates referring to the same semantic meaning in ontologies used by individual metadata extraction programs (e.g., **creator** and **author**). The ontologies and namespaces used by the NCSA software have been carefully designed in order to follow the already existing standards such as, the Dublin Core metadata terms or the FOAF namespaces, and the Engineering Drawing Practices terminology. In addition, a new functionality has been added to the FRD framework that allows unifying different predicates having the same semantic meaning by having the user interface for creating new 'mapping' RDF triples or for consolidating predicates identified by an end user.

Third, there is a lack of understanding about how to find and how to rank overlapping information, as well as how to assign confidence in discovered relationships among files. A search engine have been built that would identify RDF triples linked to the same UUID (subject) and then find pairs of RDF triples for two UUIDs containing the same predicates and values (objects). For visual inspection based discovery of relationships, different colors are presented for the RDF triples without matched predicates, with the same predicates, and with the same predicates and values. A support for inserting new RDF triples manually has been added to the viewer that would be used by end users after discovering relationships (or for adding missing metadata). In this way it is possible to create new (more precise) relationships among the documents based on metadata, and stored this information as a new triple.

TABLE 3. EXTRACTED 3D STEP AND 2D TIFF METADATA

STEP metadata specification	3D CAD file metadata in STEP file format	Title block metadata of a 2D TIFF file format
FILE_ DESCRIPTION (/* description */ (''), implementation_ level */'2;1');	FILE_ DESCRIPTION (('),'2;1');	FILE_ DESCRIPTION (('),/* implementation_ level */ '2;1');
FILE_NAME (/* name */ '',	FILE_NAME ('D:\\NARA\\ Archieve_data_ samples\\BHD_FR 12\\U2110_BHD12 _2007_05_09.stp',	FILE_NAME ('120 TORPEDO WEAPONS RETRIEVER, MAIN DECK',
/*time_stamp */'',	'2007-05 10T13:45:37',	'04-10-86',
/* author */ (''),	('rakowpj'),	('LDOBSON'),
/* organization */ (''),	(''),	('NAVAL SEA COMMAND'),
/*preprocessor_ version */ ' ',	'Autodesk Inventor 11',	'',
/*originating_ system */ '',	'IDA-STEP',	'',
/*authorization */ '');	'');	'');

In order to test a solution to the third problem, the overlapping information has been manually inserted. The tag in the original 2D drawing, author (LDOBSON) has been changed in the metadata about the STEP file. Table 3 illustrates the problem with insufficient overlapping information for two files describing TWR841 ship deck. The first column shows the recommended metadata specifications of a STEP file format. The actual data of a 3D STEP and 2D image file in TIFF file format are shown in the second and the third columns.

A prototype of a FRD framework for visual inspection based discovery of relationships has been developed. A user can select two files, the search engine will find all connecting links and the overlapping information is presented to the end user together with a small preview of each file (we added support for STEP and TIFF files for the preview). A user can determine if a 3D model was created from a 2D engineering drawing and then create a symbolic link between the two files for future queries. The discovery of such relationships can be achieved by examining different pieces of metadata, such as, a creator of an engineering drawing and a creator of a 3D model or file locations of the two files (for example, both files may reside in the same directory or the 3D files reside in a 3D model child directory). The File To Learn framework allows a user to view the metadata represented as RDF triples in a graph viewer as shown in Fig. 6. The tool shows the pair of 3D file in STEP file format and 2D image file in TIFF file format with three connecting links denoted by a grey ellipse based on metadata extracted from a file system (files belong to folders that are topologically related, e.g., 3D file and 2D file share parent folder).

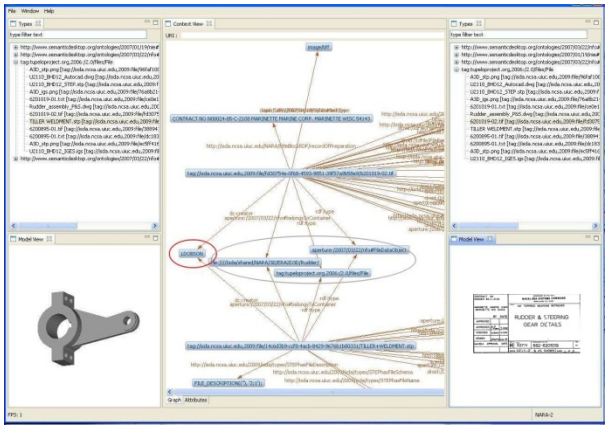


Figure 6. A File To Learn tool provides support for visual inspection based discovery of file relationships.

The red ellipse corresponds to the overlapping information about authors (creators) extracted from the content of the files. The previews of file contents in separate panes have been added to support visual confirmation of file relationships. Fig. 7 presents the same information in a tabular view with color coded entries. The colors correspond to file descriptors that are equal (blue ~ the same predicate and value), are not equal (red - the same predicate but different value) and are different (green - predicate with a value occurs only in one file as a descriptor).

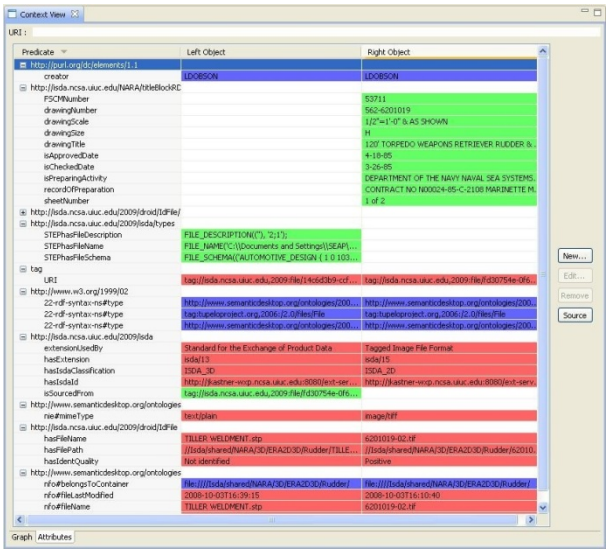


Figure 7. A tabular presentation of the over-lapping information in the FRD tool.

VI. CONCLUSIONS

We presented a methodology and a prototype for file relationship discoveries based on file system and content based metadata extraction. The prototype gathers metadata describing each file from four sources: (1) file system which reflects the directory of the file, (2) file type using DROID and PRONOM, (3) images demonstrated by extracting information from the engineering drawings' title and other blocks and (4) ASCII characters in original electronic files (e.g., in the STEP 3D file format).

The framework has been tested with the data collection of the engineering drawings and 3D CAD Models for the NAVY ship, model TWR 841. The conservative estimate of the RDF triples generated by the developed system is about 63 per file.

The File To Learn framework has been developed for finding document relationships based on the extracted metadata triples. It provides unique capabilities to search and query the RDF repository, visualization of relationships as graphs, and tabular comparison and editing of exiting metadata to verify and establish new relationships based on tacit knowledge.

ACKNOWLEDGMENT

This research was partially supported by a National Archive and Records Administration (NARA) supplement to NSF PACI cooperative agreement CA #SCI-9619019. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the National Science Foundation, the National Archive and Records Administration, or the U.S. government.

REFERENCES

[1] The Strategic Plan of the National Archives and Records Administration (NARA) 2006-2016, "Preserving Past to Protect Future", 2006, URL http://www.archives.gov/about/plans-reports/strategic-plan/

[2] T.F. Syeda-Mahmood, "Extracting Indexing Keywords from Image Structures in Engineering Drawings," Proc. 5th International Conference on Document Analysis and Recognition (ICDAR'99), 1999, pp.471-474

[3] T.F. Syeda-Mahmood, "Locating Indexing Structures in Engineering Drawing Databases Using Location Hashing," Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'99), 1999, vol. 1, pp.1049

[4] Laurent Najman, Olivier Gibot, Stéphane Berche, "Indexing Technical Drawings Using Title Block Structure Recognition," Proc. 6th International Conference on Document Analysis and Recognition (ICDAR'01), 2001, pp.0587

[5] RDF Gravity, 2004, URL, http://semweb.salzburgresearch.at/apps/rdf-gravity/index.html

[6] Welkin - graph-based RDF visualize, 2008, URL http://simile.mit.edu/welkin/

[7] RDF, 2004, URL, http://www.w3.org/TR/rdf-mt/

[8] D. Alemang, and J. Hendler, "Semantic Web for the Working Ontologist: Effective Modeling in RDFS and OWL", Ed. Morgan Kaufmann, San Francisco, 2009.

[9] DROID is a software tool to perform automated batch identification of file formats. 2009, URL http://sourceforge.net/projects/droid/

[10] PRONOM is a resource registry (information) about the file formats, software products and other technical components. 2006, URL http://www.nationalarchives.gov.uk/aboutapps/pronom/puid.htm

[11] K. McHenry and P. Bajcsy, "Key Aspects in 3D File Format Conversions," Proc. 3rd Society of American Archivists and the Council of State Archivists (SAA Research Forum), 2009

[12] K. McHenry and P. Bajcsy, "3D Data Analysis," WVU/NETL/ERA Workshop on Digital Preservation of Complex Engineering Data, 2009

[13] Aperture, 2008, URL, http://sourceforge.net/project/showfiles.php?group_id=150969; Nepomuk Ontologies, URL, http://www.semanticdesktop.org/ontologies/

[14] J. Futrelle, "Harvesting RDF Triples", Proc. International Provenance and Annotation Workshop (IPAW'06) 2006, pp. 64-72; Tupelo, 2009, http://tupeloproject.ncsa.uiuc.edu/

[15] ASME Y14/ANSI Y14, "American National Standard Engineering and Related Document Practices", 2000, URL, http://webstore.ansi.org/

[16] ISO 128, "Technical drawings - General principles of presentation," 2003; ISO 7200, "Technical product documentation -- Data fields in title blocks and document headers," 2004, International Standard Organization (ISO), URL http://www.iso.org/iso/home.htm

[17] DoD-STD-100C, "Engineering Drawing Practices," MIL-HDBK 1006/1, "Policy and Procedures for Project Drawing and Specification Preparation," 1978, replaced by DOD-STD-00100D, URL http://www.everyspec.com/DoD/DoD-STD/; see also URL, http://www.tpub.com/engbas/3-12.htm

[18] ABBYY FineReader 9.0, 2008, URL http://finereader.abbyy.com/

[19] IRIS ReadIris Pro 11, 2008, http://www.irislink.com/

[20] TopOCR, OCR Software for your Camera, 2008, URL, http://www.topocr.com/

[21] Gamera, "Gamera project, a framework for building document analysis applications," 2008, http://gamera.informatik.hsnr.de/

[22] "RDF vocabulary, rdf:," URL, http://www.w3.org/1999/02/22-rdf-syntax-ns#,

[23] "Dublin Core Metadata Element Set, dc:," URL, http://xmlns.com/foaf/0.1/,

[24] "Friend of a Friend project, foaf:" URL, http://purl.org/dc/elements/1.1/

[25] R. Kooper, D. Clutter, S.-C. Lee, P. Bajcsy; "ImageToLearn", 2006, URL http://isda.ncsa.uiuc.edu/Im2Learn/