Interoperability in eDiscovery Process: the critical challenges and the implemented solution based on the proposed XML schema

Mohammad R. Karim, Professor Oliver Popov

Abstract— Digital forensics is a very important area that carries special sensibility in analyzing and studying digital evidence not only in cyber crime cases, but also in house or localized digital crimes. eDiscovery makes it necessary to identify the evidence which is in the form of metadata. To discover and analyze the metadata in a proper manner for a litigation process in different environments, interoperability is one of the central problems. In order to attain interoperability, there is a standard termed as EDRM XML v1.1. However, this standard needs further modifications and enhancements to ensure real interoperability in an arbitrary eDiscovery process. A novel XML schema is proposed along with the necessary changes from the current EDRM XML v1.1. The assessment and the evaluation of the system demonstrated that XSD can ensure interoperability in eDiscovery process indeed.

Index Terms— Digital Forensic, eDiscovery, EDRM XML, XSD.

1 Introduction

INTEROPERABILITY is the ability of two or more systems/components to exchange information between them and to share the information that has been already exchanged. It is a central issue in an eDiscovery (Electronic Discovery) process. Moreover, it is also of a crucial importance in an organization that is involved in civil litigation because the lack of compatibility can bring to a halt all relevant activities within the organization. So, it appears that ensuring interoperability is still one of the major challenges in the Electronic Discovery Process.

1.1 Research Problem

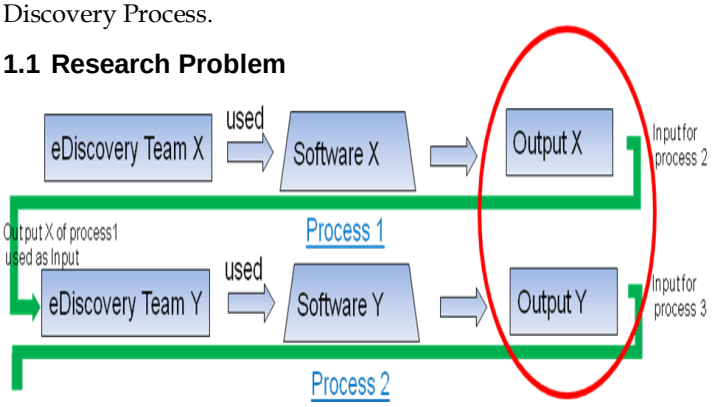

Fig 1: Problem in eDiscovery process

There are several problems to be tackled in eDiscovery process. Metadata processing is one of them. Let us assume that an eDiscovery team uses software X in one process and software Y in another process as depicted in fig 1. It is quite possible that while X has one type of data format, Y relies on another type of data format, which makes it is impossible to use the output of process 1 as an input for the process 2 because of the different data formats. So, there is a problem of interoperability. It may also happen whenever different organizations use different software.

1.2 The purpose and the goal of the research

The purpose of the research is to facilitate the movement of electronically stored information (ESI) from one step of the electronic discovery process to another step, across different platforms, overlapping environments, various organizational and institutional entities. Moreover, the intention is to assist civil litigation institutions in streamlining significantly the processes and enable the integration of multiple eDiscovery technologies within the eDiscovery process.

The goal of the research is to ensure interoperability in eDiscovery. In order to attain this, an appropriate data schema is designed for eDiscovery. The key idea is to use XML, which appears to be more efficient than some of the other existing schemas.

2 Interoperability by XML

XML and its rules are examined elaborately because its metatags are very important for this project. Moreover, the existing EDRM XML v1.1 metatags are considered in detail as well. The XML metatags by integrating the existing EDRM solution are proposed in order to make sure that the output of each eDiscovery process (for instance process 1) is interoperable with the input of another eDiscovery process (for example process 2).

- *Mohammad R. Karim is working as a consultant of Dan Soft Tech Aps, Copenhagen, Denmark, E-mail: mrka@kth.se*
- *Professor Oliver Popov , Head of Units, Systems Analysis and Security, Department of Computer and Systems Sciences, Stockholm University,Sweden, E-mail: popov@dsv.su.se*

2.1 XML: Extensible Markup Language

The Extensible Markup Language (XML) is a simple text-based language for representing structured information such as documents, data, books, configurations, invoices, or transactions. The language was derived from an older standard format known as SGML (ISO 8879) [1]. Since its introduction in 1998, XML has revolutionized how humans think about structuring, describing, and exchanging information. As a result, the XML usage in the software industry is of a large variety and rapidly growing. The importance of XML is of paramount significance web services, because almost all service technologies are based on it [2].

XML is the best way to set rules for encoding documents in an electronic manner because of its simplicity, generality and usability. It is defined in the specification termed as XML 1.0 and issued by the World Wide Web Consortium (W3C).

2.2 Role of XML in the field of interoperability

XML plays a vital role in the field of data interoperability because it has strong data binding ability. Some of its important features of XML are enumerated below.

A. XML is human readable; XML data can be easily understood from xml file from its tag and corresponding value.
B. XML is a metalanguage; W3C defines multiple languages that follow XML syntax rules.
C. XML creates own vocabulary; According to W3C specification for XML, it is opened to choose set of element type names.
D. XML Documents are well-formed & valid; hence they could be validated by the schema and W3C standards.

XML is a textual data format, with strong support via Unicode for various languages. There are a variety of programming interfaces which software developers may use to access XML data, and several schema systems are designed to aid in the definition of XML-based languages [1].

2.3 Addressing the interoperability problem in eDiscovery process via XML

The research tries to overcome the difficulties of interoperability in eDiscovery process. To enhance interoperability, XML is the best solution as it is a standard language for data description. In addition, XML is widely used for data exchange between different systems even for the incompatible programs, any computer networks, different data structures or various operating systems. XML provides syntactical and structural data interoperability because all the exchanged files contain the same XML mark-up, indexing, searching, combining, and also re-using text-based information.

2.4 Solving interoperability problem in eDiscovery process based on the existing EDRM solution

The current XML metatags for EDRM v1.1 do not include all the tags that are important. Namely, EDRM is trying to setup standard and current version is not so stable, it is evident that some meaningful metatags are missing. Hence, the tags that are missing and important are included within the existing solution. In addition, some tags are considered to be obsolete and these tags are excluded in the proposed XML metatags that is basis for the current solution provided by EDRM XML metatags v1.1. Moreover, some tags need to be modified since their attributes have to be more reliable. The proposed XML metatags are provided in Table 1.

Document Type	Field	Tag Name	Data Type	Description
All	Language	#Language	Text	The ISO 639-1 (two-character) primary language of the document. If omitted, EN can be assumed.
	StartPage	#StartPage	Text	Starting page number for this document (for Bates numbering)
	EndPage	#EndPage	Text	Ending page number for this document (for Bates numbering)
	ReviewComment	#ReviewComment	Text	Review comment
Message	From	#From	Text	The sender of the message. EDRM-recommended format is a semicolon-delimited list of RFC2821 formatted addresses (either "yourname@domain.com" or "Your Name <yourname@domain.xyz>"
	To	#To	Text	The recipient(s) of the message, same format as "From"
	CC	#CC	Text	The cc'ed recipient(s) of the message, same format as "From"
	BCC	#BCC	Text	The bcc'ed recipient(s) of the message, same format as "From"
	Subject	#Subject	Text	The subject of the message
	Header	#Header	Text	Message header of the message
	DateSent	#DateSent	DateTime	Date the message was sent
	DateReceived	#DateReceived	DateTime	Date the message was received
	HasAttachments	#HasAttachments	Boolean	Whether or not the email has attachments
	AttachmentCount	#AttachmentCount	Integer	The number of attachments the email has
	AttachmentNames	#AttachmentNames	Text	Concatenated list of attachment names separated by semicolons (path optional – for end-user searching)
	AttachmentType	#AttachmentType	Text	Type of attachment like jpeg,zip etc.
	ReadFlag	#ReadFlag	Boolean	Whether the email has been read or not
	ImportanceFlag	#ImportanceFlag	Boolean	Whether the email was sent with high importance
	MessageClass	#MessageClass	Text	(Outlook) message class
	FlagStatus	#FlagStatus	Text	(Outlook) flag status
File	FileName	#FileName	Text	The name of the original file
	FileExtension	#FileExtension	Text	The extension of the original file
	FileSize	#FileSize	Longinteger	The size of the original file in bytes
	Encrypted	#Encrypted	Boolean	Encryption was performed
	DateCreated	#DateCreated	DateTime	Date the file was created
	DateAccessed	#DateAccessed	DateTime	Date the file was last accessed
	DateModified	#DateModified	DateTime	Date the file was last modified
	DatePrinted	#DatePrinted	DateTime	Date the file was last printed
	Title	#Title	Text	(Office) Document title
	Subject	#Subject	Text	(Office) Document subject
	Author	#Author	Text	(Office) Document author
	Company	#Company	Text	(Office) Document company
	Category	#Category	Text	(Office) Document category
	Keywords	#Keywords	Text	(Office) Document keywords
	Comments	#Comments	Text	(Office) Document comments
The following data types are valid for tags:				
Type	Comment			
Text	Unlimited/unbounded			
LongInteger	64-bit integer	Change from 32 bit integer to Long		
DateTime	YYYY-MM-DDThh:mm:ss sTZD			
Decimal	Double	Change from float to double		
Boolean	(Yes/No), (True/False), (1/0)			
Function	Remove			

newly added tags modified tags deleted tags

Table 1: XML metatags for integration within the existing EDRM solution

E-mail attachments are quite common the various messaging systems that belong to larger information or communication platforms. Consequently, they are important factors in an eDiscovery process because of storing and manipulation of files like .zip, .exe and .jpeg etc. It is significant when a database file is transferred from one platform to another platform. Without knowing the file type, the problem becomes more complex and even the interoperability may become impossible to achieve. So there is a clear need to include the type of the attachment in the XSD schema.

Security is an important issue and encryption is a part of it. The best way to know the file or attachment or folder is encrypted or not when these attachments are transferring one platform to other if there is any specific indicator is present. If those files are encrypted, decryption method must be applied to read those files. To ensure interoperability, it should be well known in advance whether or not the file is encrypted. So, encryption tags should be included with the already present XML metatags.

The limits of long integer should be increased from 32 bit to 64 bit due to large storage devices. Now-a-days, terabyte hard disks may be quite common in personal computer systems. In addition, 64 bit operating systems are available on the market for different platforms. All these makes the arguments for increasing the decimal value to double credible.

Since text tags are used and they are essential unbounded, a long text is not necessary and it should be removed from the XML metatags.

3 Interoperability with XML Schema

XSD is analyzed thoroughly in order to identify the essential primitive elements and attributes. It is designed to ensure interoperability through the proposed XSD schema for EDRM v1.1.

3.1 XSD: XML Schema Definition

XSD is actually a description of XML document. It describes the syntax and defines sets of grammatical rules according to the order of elements and attributes that can be represented in a XML document. It also ensures that each XML document must be in specific formats and specific data types. Moreover, to consider the document as a 'valid' XML document, the document must follow a set of rules from schema. There are two ways a XML schema can be expressed. One standard from W3C and another is RELAX NG. In this research work, W3C standard is followed because it is more widely used than RELAX NG. For better performance, strict grammatical rules and high rang of acceptable level; W3C standard is used most of the web services.

3.2 Importance of XSD on data interoperability

To ensure interoperability, XML file must be validated with corresponding XSD. It is a very powerful tool for data binding of XML document. To understand the importance of XSD, a simple method is discussed below.

An agreement about the structure of the XML document has been done when data is completely managed and fully exchanged in XML format between two platforms. For the data exchange to occur, the values of the elements and the attributes must be within the required range as well as in the desired format. For this reason, an XML Schema is needed. A Schema defines the structure of the XML document along with rules to validate the values of the attributes and the elements as well as their corresponding formats. Without XSD, data binding in XML would be extremely difficult. It should be clear from the above discussion that XSD is very important factor for data interoperability in XML document.

4 The work and the operation of the system

The implemented system has a user interface where one can issue commands to store evidence (input) in the system and then get an xml as the output from the system. The system takes xml or any other type of file (also attribute of file) as an input but produces only xml output. When it takes xml as an input, it must validate the XML document with the XSD. The output, which is an XML document, is also validated with the XSD because of ensuring interoperability.

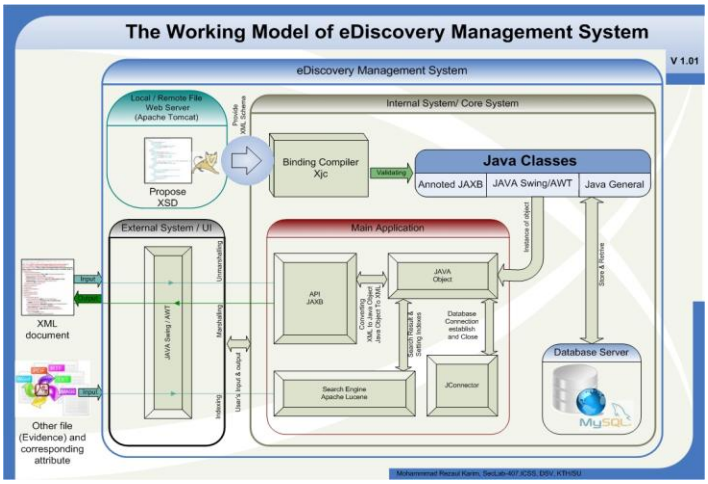

Fig 2: The working model of eDiscovery Management System

The actual system from an operational point of view could be divided in two parts, namely an internal system and an external system, as shown on fig 2.

4.1 Internal or core system

The internal system is the core part of eDiscovery Management System, which comprises of several components. One can say that the main application (MA) is the driver that runs the whole system, where as Java classes are the engine of this system. Furthermore, the main application part is made up of several components as well.

The JAXB API is responsible for marshalling Java objects to XML document and unmarshalling the XML document to Java objects. The other component that is a Java object is an instance of Java classes. Yet, another component of the MA, Apache Lucene, is a search engine that provides searching services by indexing the contents of the system. The final component of the MA is the JConnector that connects the storage system with the core system.

The MA has the ability to show all the information stored in database and can perform edit and add actions. The other part of the internal system is MySQL database server which is the storage system. The schema compiler (xjc) is the indicator of validation.

4.2 External system or user interface

The external system is the user interface where user input and system output is handled with Java Swing and AWT. It has a two-way direct communications with the MA part of the internal system.

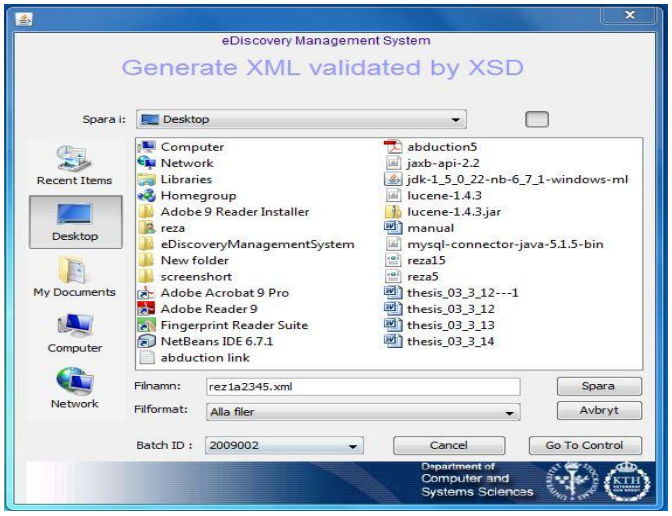

Fig 3: Generate XML with validation by XSD

The fig 3 provides the proper interface to take the user input as an XML file or other file format (such as jpeg, doc, pdf, zip, binary). It produces XML output only.

4.3 Local files system/Remote files system/Web Server

The XSD is stored in either the local files system or the remote files system or web server. The schema is used by the corresponding schema compiler in the internal system to validate the XML document.

5 TESTING AND EVALUATION

eDiscovery Management System is tested to verify and validate the intended functionality of the system, namely that the expected output is equal to the real output. The test suite is designed which contains the expected values. Furthermore, the system is evaluated by NetBeans profiler as well as TPTP (Test & Performance Tools Platform). The automatic evaluation tools provide information about the system status such as the memory heap, dead lock situation, active and sleeping threads, hot spots, and the time of method invocation time.

5.1 Evaluation

The system evaluation was done automatically with NetBeans profiler, which is a powerful tool that provides important information about the runtime behavior of an application. The NetBeans Profiler keeps tracks of thread state, memory usage, and CPU performance. It uses innovative technology to allow the system developer to tightly control exactly which parts of an application are profiled, resulting in reduced overhead and easier to interpret output results. The profiled application may run locally or on a distributed remote system. By being tightly integrated into the NetBeans IDE workflow, the NetBeans Profiler makes it easy to identify performance problems and memory leaks [4].

5.2 Profiling

NetBeans profiling tool was used to profile the implemented system. Profiling tools show the actual situation of every thread in the system.

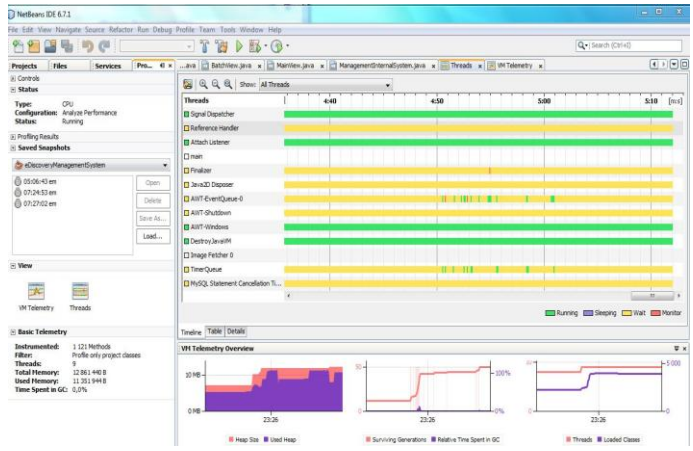

Fig 4: The output result of profiler for eDiscovery Management System (General View)

The VM overview is shown in the bottom of fig 4. The system memory heap used around 11MB of memory. Moreover, the system has near 10 threads running when the loaded classes are below 10 at a specific instance of time.

5.3 Testing in a different environment

The purpose of testing in a different environment is to confirm the system interoperability. It is very important to know whether the proposed XSD makes the system interoperable or not. For this, the system was tested in different environments and with different software. Moreover, it was tested with the software that was built on .NET technology of Microsoft Visual Studio 2010 Beta. The system was tested via web services and on different platforms such as Windows (XP, Vista, Windows 7), Linux (Ubuntu 10.0) and Sun OS (Open Solaris 0.5).

5.4 Discussion

A data schema is a data structure described in formal language. A strong data binding means that a schema follows specific rules and its structure is tightly coupled with those rules. W3C provides a standard for data schema such as the XSD schema. Therefore, it is potentially widely acceptable solution for interoperability. The XSD for eDiscovery process is designed in such a way that it can handle all the possible cases of electronic discovery contents. Thus, the XSD schema maintains interoperability and it does not depend on any platform or vendor specific products.

6 CONCLUSIONS

The research goal of the project was to provide a possible solution for ensuring interoperability in eDiscovery process because current EDRM v1.1 has few drawbacks.

The research finds XML schema (XSD) driven XML document is the best solution for interoperability problem. That is the reason why the current XML metatags for EDRM v1.1 must be a subject to some modifications. Hence, the research recommends the necessary changes in the indicated metatags to ensure interoperability in the process of eDiscovery . The generated XML file must be validated by the modified XSD to ensure interoperability.

The research of this project also recommended some security features to be added in metatags as well as in XSD and in the application level security. Hybrid encryption is recommended for application level security and SASL for communication level security. Moreover, WS-Security policy should be applied to the whole system.

The proposed system along with the associated recommendations is implemented with the JAXB, Apache Lucene, Apache Tomcat, MySQL and Java EE 6 technologies. The Java code is portable and platform independent and JAXB serves as glue between the portable data XML and the portable code Java. The system from a software engineering point of view is tested with JUnit and NetBeans Profiling tool, while from the user point of view by particular group of people. In both cases, the system produced positive results.

REFERENCES

[1] World Wide Web Consortium Home Page. (2009). [On-line]. Accessed on June 5, 2009 at URL: http ://www.w3.org /standards /xml core

[2]Edition, Steve Graham, Doug Davis, & Simeon Simeonov (2005). *Building Web Services with Java, Making sense of XML, Soap, WSDL, and UDDI* (2nd Ed.).New York, USA: Sams Publishing.

[3]Wikipedia Home Page. (2009). [On-line]. Accessed on October 21, 2009 at URL: http://en.wikipedia.org/wiki/XML

[4]Java Performance Testing Tool Home Page. (2009). [On-line]. Accessed on December 21, 2009 at URL: http: //www. javaperformancetuning.com /tools/netbeansprofiler