An Ontology-Based Transformation Model for the Digital Forensics Domain

Sarunas Grigaliunas[1], Jevgenijus Toldinas[1], Algimantas Venckauskas[1]
[1]*Department of Computer Science, Kaunas University of Technology, Studentu St. 50–209, LT-51368 Kaunas, Lithuania*
eugenijus.toldinas@ktu.lt

***Abstract*—The creation of an ontology makes it possible to form common information structures, to reuse knowledge, to make assumptions within a domain and to analyse every piece of knowledge. In this paper, we aim to create an ontology-based transformation model and a framework to develop an ontology-based transformation system in the digital forensics domain. We describe the architecture of the ontology-based transformation system and its components for assisting computer forensics experts in the appropriate selection of tools for digital evidence investigation. We consider the use of the attributes of Extensible Markup Language document transformation to map the computer forensics ontology and we use the representations in the National Institute of Standards and Technology's "Computer Forensics Tool Catalog" for aligning one form with the other.**

***Index Terms*—Computers and information processing; computer-aided software engineering; digital forensics and software tools; XML document transformation.**

I. INTRODUCTION

In general, an ontology creates a common vocabulary to analyse the domain information within a certain area [1]. Therefore, by creating an ontology, it is possible to form common information structures, to reuse knowledge, to make assumptions within a domain and to analyse every piece of knowledge. This is important to the field of cyber forensics, because the knowledge that is shared between the domain of computer forensics experts and the specifications of tools that can be used for digital evidence investigation is still being developed.

The creation of an ontological model may allow these specific areas to be defined. However, if the development of digital evidence investigation tools continues to be random and disconnected from computer forensics expertise, it could be detrimental for computer forensics experts in the field. While researchers continue to develop other forms of forensic science in order to create their own models for their needs, they do not consider what experts will do at a higher level in the computer forensics investigation process [2].

We aim to create an ontology-based transformation model for the digital forensics domain and to develop a system for computer forensics experts in their respective domain that enables separate formulation and incorporation of domain-specific concepts as ontologies. To achieve this goal, we propose a set of transformation rules that maps those ontologies for each other. The next section reviews work in areas related to this.

Manuscript received 15 August, 2016; accepted 2 March, 2017.

II. RELATED WORK

There have been several schemas proposed in the past for representing digital forensic information, but these have not been widely adopted [3]–[5]. One schema that is in use is Digital Forensics XML [6]. This schema was primarily developed to represent the output from tools used to analyse storage media, including file system parsers, file carvers, and hash set generators.

Digital Forensics XML has been implemented in several digital forensic tools, including Fiwalk (based on the SleuthKit), and as a Python library with bundled programs that read and write Digital Forensics XML documents.

An ontology creates a common definition among particular domains in the field of science. Accordingly, common information structures and reusable knowledge can be formed. Moreover, assumptions in a domain can be made, and the most important is that items can be analysed in each section of a stage mentioned [7].

In the field of computer forensics, the concept of ontology plays a crucial role in describing and classifying specific stages in the process of investigation. A number of ontologies related to security and intrusion detection have been identified [8], [9].

Web ontology language (OWL) lineage can be drawn from the framework for representing knowledge introduced in 1975 by Minsky as the semi-structured data model. A frame language represents an object or concept, and attached to each frame are attributes that represent component parts of the concept or object. The underlying object-oriented paradigm may be seen as the application of frame-based theory to the structuring of software [10].

Documents containing OWL are intended to be easily published on the Web, where the language can be used by applications that process the content of information. Ontologies defined in OWL may import subsets of other ontologies. The language provides support for merging ontologies, encouraging separate ontology development, refinement and reuse.

Description logics are a subset of first-order logic (FOL) and are well-suited to expressing terminology and instance information, with efficient and decidable inference characteristics. The computational characteristics of FOL, on the other hand, are intractable [11].

III. PROBLEM DEFINITION AND ANALYSIS OF RELATED DOMAINS

Collection, examination, analysis and reporting are the main activities in the digital forensic evidence investigation process. Preparation will assist with the selection of an appropriate tool, fulfilling the necessary legal requirements, deciding on the level of management and arranging the necessary support [12]. The National Institute of Standards and Technology (NIST) realised the need for searching forensic tools by technical parameters based on specific forensic tool functionality and proposed the "Computer Forensics Tool Catalog" (CFTC) [13]. The primary goal of the CFTC is to provide an easily searchable catalogue of forensic tools for digital evidence investigation. In addition, NIST proposed a forensic tool taxonomy based on forensic tool functionalities.

In [2], the cyber forensics ontology (CFO) was presented. The CFO comprises five layers of hierarchical structure with the resulting final layer being specified areas for certifying and specialising. Those layers belong to the technology and profession domains and are described as follows: hardware, software (technology domain), law, academia, military, private sector (profession domain).

We have analysed two domains: the CFO and the CFTC. The relationship between these domains is depicted in Fig. 1.

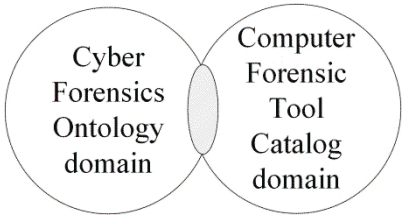


Fig. 1. Relationship between CFO domain and NIST "Computer Forensics Tool Catalog" domain.

As mentioned above, in the preparation stage, computer forensics experts need to make a significant decision with regard to the selection of an appropriate tool for digital evidence investigation. Although the NIST CFTC reveals many suitable tools classified by their functionality and expressed by appropriate artefacts, computer forensics experts use the CFO.

As shown in Fig. 1, only a small amount of artefacts, when expressed through ontologies for both domains, intersects.

To assist and facilitate computer forensics experts in selecting an appropriate tool for digital evidence investigation, we propose a computer forensics tool catalogue ontology (CFTCO) created from the NIST CFTC and an ontology-based transformation model (TM) for the digital forensics domain, shown in Fig. 2.

An ontology-based TM consists of two stages (Fig. 2). The first stage relies on an XML view creation for selected ontologies (CFO and CFTCO). In the second stage, the transformation process uses the XML view of the ontologies created in the first stage and maps their representations from one form to the other. The transformation process applies a set of transformation rules that will create a list of appropriate tools.

Formally, an ontology is a pair $0 = (D, R)$, where D is a domain and R is a set of relations defined in D.

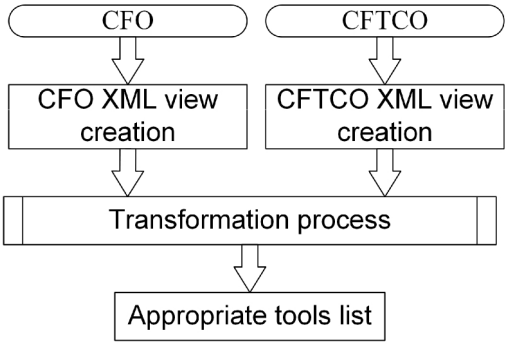


Fig. 2. Ontology-based transformation model.

To define the ontology-based TM, we formulated the following set A of ontology axioms:

Axiom A1: If O_{cf} is the computer forensics ontology and E_{xcf} is corresponding custom XML elements domain of CFO, then there exists a function $f_{cf} : O_{cf} \to E_{xcf}$.

Axiom A2: If O_{cftc} is the computer forensics tool catalogue ontology and E_{xcftc} is corresponding custom XML elements domain of CFTCO, then there exists a function $f_{cftc} : O_{cftc} \to E_{xcftc}$.

Axiom A3: If E_{xcf} is the custom XML elements domain of CFO and O_{cftc} is the CFTC domain ontology, then there exists a composition function $g_{cf} \circ f_{cf} : E_{xcf} \to O_{cftc}$.

Axiom A4: If E_{xcftc} is the custom XML elements domain of CFTCO and O_{cf} is the computer forensics domain ontology, then there exists a composition function $g_{cftc} \circ f_{cftc} : E_{xcftc} \to O_{cf}$.

We define the ontology-based transformation model (TM) as follows

$$TM(O) = A\left\{O_{cf}, f_{cf}, E_{xcf}, O_{cftc}, f_{cftc}, E_{xcftc}, g_{cf}, g_{gftc}\right\} \quad (1)$$

IV. FRAMEWORK TO DEVELOP AN ONTOLOGY-BASED TRANSFORMATION SYSTEM

For encoding documents in a format that is human- and machine-readable, Extensible Markup Language (XML) is widely used in computer systems. Examples of that are as follows.

For the Semantic Web with formally defined meaning, the Web Ontology Language (OWL 2) is used. Documents that are stored as Semantic Web contain classes, properties, individuals and data values provided by OWL 2 ontologies. [14].

For representing forensic processing results and forensic information, Digital Forensics XML (DFXML) was designed. When structured information sharing between independent tools and organisations is needed, DFXML allows defining the needs of forensic tools and analysts because of its suitability for abstractions [15].

One more language that uses XML schemas for creating objects and other resources is the standardised language Cyber Observable eXpression (CybOX™) [16]. High-fidelity information about cyber observables with CybOX™ is encoded and used for communication.

In ASP.NET, applications save settings in the Web.config

file. When an application is deployed to different environments, the settings differ accordingly. For Web projects, Visual Studio (Integrated Development Environment) uses XML document transformation (XDT) to automate the process of changing the Web.config when applications are deployed to different destination environments [17].

A transform file is used for transformation purposes in which it is specified how the Web.config file should be changed when the application is deployed. To specify transformation actions, the XML-Document-Transform namespace with the XDT prefix is used. Two attributes are defined in the XML-Document-Transform namespace: "Locator" and "Transform". When an element in the Web.config needs to be changed, the "Locator" attribute specifies the element's name. A set of elements could also be specified using the "Locator" attribute. The "Transform" attribute specifies what to change in the elements that were specified using the "Locator" attribute [17]. "Locator" and "Transform" attributes and their meanings are shown in Table I.

TABLE I. XDT LOCATOR AND TRANSFORM ATTRIBUTES.

XDT attribute	Attribute parameters	Explanation
Locator	Condition	Specifies an XPath expression that is appended to the current element's XPath expression
	Match	Selects the element or elements that have a matching value for the specified attribute or attributes
	XPath	Specifies an absolute XPath expression that is applied to the development Web.config file
Transform	Replace	Replaces the selected element with the element that is specified in the transform file
	Insert	Adds the element that is defined in the transform file as a sibling to the selected element or elements
	InsertBefore	Inserts the element that is defined in the transform XML directly before the element that is selected by the specified XPath expression
	InsertAfter	Inserts the element that is defined in the transform XML directly after the element that is selected by the specified XPath expression
	Remove	Removes the selected element. If multiple elements are selected, removes the first element
	RemoveAll	Removes the selected element or elements
	RemoveAttributes	Removes specified attributes from the selected elements
	SetAttributes	Sets attributes for selected elements to the specified values

To realise our proposed two-stage ontology-based transformation model for the ontologies' transformations, we propose to use XDT transformation attributes "Locator" and "Transform".

V. ARCHITECTURE OF AN ONTOLOGY-BASED TRANSFORMATION SYSTEM

Here we describe the architecture of the ontology-based transformation system (OBTS) and its components for assisting computer forensics experts in the selection of appropriate tools for digital evidence investigation. The architecture of the OBTS is depicted in Fig. 3.

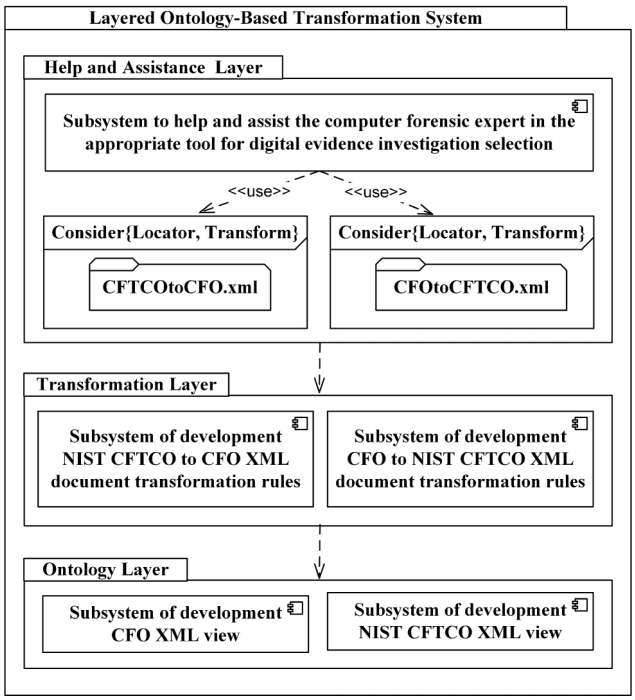


Fig. 3. Architecture of the ontology-based transformation system.

The OBTS has the following layered subsystems to support two-stage transformations of ontologies:

1. Subsystem of development CFO view, intended for CFO XML view creation.
2. Subsystem of development NIST CFTCO view, intended for NIST CFTCO XML view creation.
3. Subsystem of XML transformation rules development from CFO to NIST CFTCO, intended for XDT rules generation using "Locator" and "Transform" attributes and saving them to the CFOtoCFTCO.xml file.
4. Subsystem of XML transformation rules development from NIST CFTCO to CFO, intended for XDT rules generation using "Locator" and "Transform" attributes and saving them to the CFTCOtoCFO.xml file.
5. Subsystem that uses CFTCOtoCFO.xml, CFOtoCFTCO.xml files and intended to help and assist computer forensics experts in the selection of an appropriate tool for digital evidence investigation.

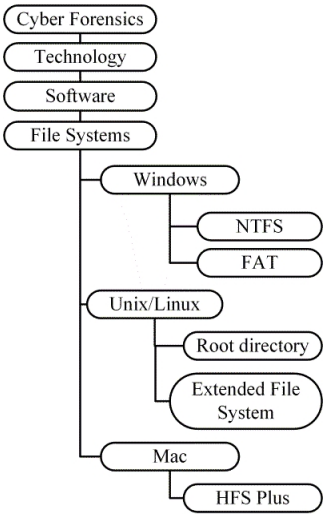
VI. CASE STUDY

As a case study of the proposed ontology-based transformation system, we present an example of XML document transformation from the CFO to the CFTCO.

At the ontology layer, XML views are developed. We use the fragment of the proposed in [2] ontology as an example of the CFO (Fig. 4(a)). The developed XML view is depicted in Fig. 4(b).

We create the CFTCO (Fig. 5(a)) from NIST's proposed forensic tool taxonomy by forensic tool functionalities. For the case study, we select the file carving subdomain [18]. The developed XML view is depicted in Fig. 5(b).

At the transformation layer, using XDT "Locator" and "Transform" attributes (see Table I), the XML document transformation rules are developed and then applied to the XML view. In our case study, an example of the developed transformation rule is depicted in Fig. 6.

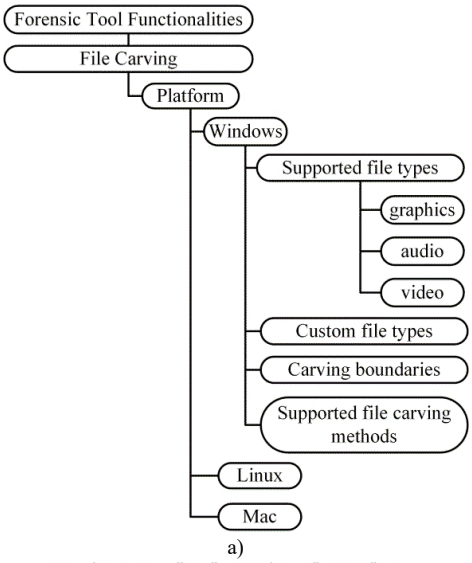


a)

```
<?xml version="1.0" encoding="UTF-8"?>
<technology>
  <software>
    <filesystem>
      <operatingsystem name="Windows">
        <filesystemtype>NTFS</filesystemtype>
        <filesystemtype>FAT</filesystemtype>
      </operatingsystem>
      <operatingsystem name="Unix_Linux">
        <filesystemtype>Root_directory</filesystemtype>
        <filesystemtype>Extended_File_System</filesystemtype>
      </operatingsystem>
      <operatingsystem name="Mac">
        <filesystemtype>HFS_Plus</filesystemtype>
      </operatingsystem>
    </filesystem>
  </software>
</technology>
```

b)

Fig. 4. CFO example (a) and its representation in XML view (b).



a)

```
<?xml version="1.0" encoding="UTF-8"?>
<filecarving>
  <platform name="Windows">
    <supportedfiletypes name="graphics">
      <fileformat>jpg</fileformat>
      <fileformat>png</fileformat>
      <fileformat>bmp</fileformat>
      <fileformat>gif</fileformat>
    </supportedfiletypes>
    <supportedfiletypes name="audio">
    <supportedfiletypes name="video">
    <customfiletypes>
    <carvingboundaries>
    <supportedfilecarvingmethods>
  </platform>
  <platform name="Linux">
  <platform name="Mac">
</filecarving>
```

b)

Fig. 5. CFTCO file carving subdomain (a) and its representation in XML view (b).

```
<?xml version="1.0" encoding="UTF-8"?>
<technology xmlns:xdt="http://schemas.microsoft.com/XML-Document-Transform">
  <software>
    <filesystem>
      <operatingsystem xdt:Locator="Match(name)" xdt:Transform="Replace" name="Windows">
        <tool name="BlackLight">
        <tool name="Data Recovery System(DRS)">
        <tool name="DFF">
          <version>1.3</version>
          <tool_release_date>February 2013</tool_release_date>
          <available_test_reports/>
          <vendor>ArxSys</vendor>
          <homepages xmlns:xlink="http://www.w3.org/1999/xlink">
        </tool>
        <tool name="Magnet AXIOM">
        <tool name="OSForensics">
        <tool name="PhotoRec">
      </operatingsystem>
      <operatingsystem xdt:Locator="Match(name)" xdt:Transform="Replace" name="Unix_Linux">
        <tool name="DFF">
        <tool name="Magnet AXIOM">
        <tool name="PhotoRec">
      </operatingsystem>
      <operatingsystem xdt:Locator="Match(name)" xdt:Transform="Replace" name="Mac">
        <tool name="DFF">
        <tool name="Magnet AXIOM">
        <tool name="PhotoRec">
      </operatingsystem>
    </filesystem>
  </software>
</technology>
```

Fig. 6. XML document transformation rule from CFOtoCFTCO.xml.

The application of the transformation rules produces a list of appropriate tools in XML form, which is depicted in Fig. 7.

```
<?xml version="1.0" encoding="UTF-8"?>
<filecarving>
  <operatingsystem name="Windows">
    <tool name="BlackLight">
      <version>2015R3.1</version>
      <tool_release_date>October 2015</tool_release_date>
      <available_test_reports> </available_test_reports>
      <vendor>BlackBag Technologies</vendor>
      <homepages xmlns:xlink="http://www.w3.org/1999/xlink">
    </tool>
    <tool name="Data Recovery System(DRS)">
    <tool name="DFF">
    <tool name="Magnet AXIOM">
    <tool name="OSForensics">
    <tool name="PhotoRec">
  </operatingsystem>
  <operatingsystem name="Linux">
  <operatingsystem name="Mac">
</filecarving>
```

Fig. 7. XML document of the CFO fragment.

After XML document transformation rules are applied and the transformed XML document is ready to use, the subsystem will help and assist computer forensics experts in selecting an appropriate tool for digital evidence investigation, showing the names of suitable tools. OPML [19] uses the XML-based format that allows exchange of the outline-structured information between applications running on different operating systems and environments. Portable digital format (PDF) readers can open an OPML file. In the proposed OBTS, we use OPML to encode transformed XML files for further use with a PDF reader (see Fig. 8).

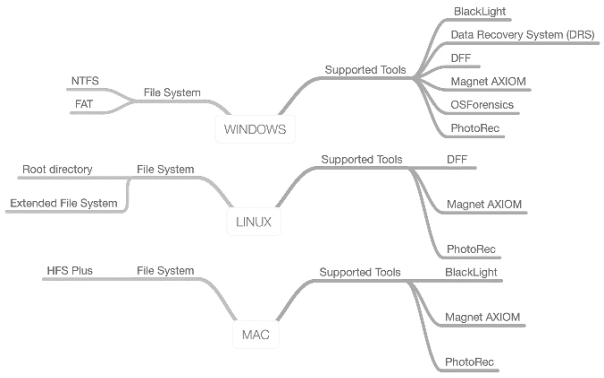


Fig. 8. OPML file opened with a PDF reader.

The subsystem also holds the home page URL for each tool selected from the list.

VII. CONCLUSIONS

The CFO and the CFTCO have created common

definitions in the digital forensics domain. While both belong to the same digital forensics domain, they are very different and only a small amount of artefacts, when expressed through ontologies, intersects. Typically, computer forensics experts operate in terms of the CFO, but the NIST taxonomy of forensic tools for digital evidence investigation is given in CFTC terms. In this paper, we consider three challenging tasks: (1) to propose a two-stage model for transformations of ontologies from the CFO to the CFTCO and vice versa; (2) to suggest XML document transformations (XDT) to map CFO and CFTCO representations from one to the other; and (3) to develop a multi-layered architecture and ontology-based transformation system (OBTS) in which the proposed model and XDT are realised. In the case study, we create a set of transformation rules and show that OBTS transforms the CFO to the NIST tool list and is able to assist computer forensics experts in selecting an appropriate tool for further digital evidence investigation. Future work will focus on the extension of our OBTS with an intelligent agent that will search the NIST CFTC on the Web and generate an XML view of the CFTCO.

ACKNOWLEDGMENT

The authors wish to thank prof. V. Stuikys for his valuable efforts in reviewing the paper.

REFERENCES

[1] N. F. Noy, D. L. McGuinness, *Ontology Development 101: A Guide to Creating Your First Ontology*. Stanford University, Stanford, CA. [Online]. Available: http://protege.stanford.edu/publications/ontology_development/ontology101-noy-mcguinness.html

[2] A. Brinson, A. Robinson, M. Rogers, "A cyber forensics ontology: Creating a new approach to studying cyber forensics", *Digital Investigation*, vol. 3, pp. 37–43, 2006. [Online]. Available: http://dx.doi.org/10.1016/j.diin.2006.06.008

[3] P. Turner, "Digital provenance - interpretation, verification and corroboration", *Digital Investigation: The International Journal of Digital Forensics & Incident Response*, vol. 2, no. 1, pp. 45–49, 2005. [Online]. Available: http://dx.doi.org/10.1016/j.diin.2005.01.002

[4] P. Turner, "Unification of digital evidence from disparate sources (Digital Evidence Bags)", *Digital Investigation*, vol. 2, no. 3, pp. 223–228, 2005. [Online]. Available: http://dx.doi.org/10.1016/j.diin.2005.07.001

[5] K. Lim, S. Lee, "A methodology for forensic analysis of embedded systems", in Future *Generation Communication and Networking*, *(FGCN 2008)*, 2008. [Online]. Available: http://dx.doi.org/10.1109/FGCN.2008.225

[6] S. Garfinkel, "Digital forensics XML and the DFXML toolset", *Digital Investigation*, vol. 8, no. 3–4, pp. 161–174, 2012. [Online]. Available: http://dx.doi.org/10.1016/j.diin.2011.11.002

[7] A. Luthfi, "The use of ontology framework for automation digital forensics investigation", *International Journal of Computer*, *Electrical, Automation, Control and Information Engineering*, vol. 8, no. 3, pp. 454–456, 2014. [Online]. Available: scholar.waset.org/1999.4/9997754

[8] H. Chen, T. Finin, A. Joshi, "An ontology for context-aware pervasive computing environments", *The Knowledge Engineering Review*, vol. 18, no. 3, pp. 197–207, 2003. [Online]. Available: http://dx.doi.org/ 10.1017/S0269888904000025

[9] A. Razzaq, Z. Anwar, H. F. Ahmad, K. Latif, F. Munir, "Ontology for attack detection: An intelligent approach to web application security", *Computers & Security*, vol. 45, pp. 124–146, 2014. [Online]. Available: http://dx.doi.org/10.1016/j.cose.2014.05.005

[10] O. Lassila, D. McGuinness, "The role of frame-based representation on the semantic web", *Linkoping Electronic Articles in Computer and Information Science*, vol. 06, no. 5, 2014. [Online]. Available: http://www.ep.liu.se/ea/cis/2001/005/

[11] B. N. Grosof, I. Horrocks, R. Volz, S. Decker, "Description logic programs: combining logic programs with description logic", in *Proc. 12th Int. Conf. World Wide Web (WWW 2003),* Budapest, Hungary, 2003, pp. 48–57. [Online]. Available: http://dx.doi.org/10.1145/775152.775160

[12] S. Saleem, O. Popov, I. Bagilli, "Extended abstract digital forensics model with preservation and protection as umbrella principles", *Procedia Computer Science,* vol. 35, pp. 812–821, 2014. [Online]. Available: http://dx.doi.org/10.1016/j.procs.2014.08.246

[13] Computer Forensics Tool Catalog. [Online]. Available: http://toolcatalog.nist.gov/index.php

[14] Web Ontology Language. [Online]. Available: http://www.w3.org/TR/2012/REC-owl2-overview-20121211/

[15] S. Garfinkel, "Digital forensics XML and the DFXML toolset", *Digital Investigation*, vol. 8, no. 3–4, pp. 161–174, 2012. [Online]. Available: http://dx.doi.org/10.1016/j.diin.2011.11.002

[16] Cyber Observable eXpression (CybOX™). [Online]. Available: http://cyboxproject.github.io/about/

[17] Web.config Transformation Syntax for Web Project Deployment Using Visual Studio. [Online]. Available: https://msdn.microsoft.com/en-us/library/dd46532

[18] Computer Forensics Tool Catalog. Forensic Tool Taxonomy [Online]. Available: http://toolcatalog.nist.gov/taxonomy/index.php?ff_id=5

[19] Outline Processor Markup Language. [Online]. Available: http://dev.opml.org/