

De la scène de crime aux connaissances : représentation d'évènements et peuplement d'ontologie appliqués au domaine de la criminalistique informatique

Yoan Chabot*,** Aurélie Bertaux**
Tahar Kechadi*, Christophe Nicolle**

*School of Computer Science and Informatics, University College Dublin, Ireland

**Equipe CheckSem, Laboratoire Le2i, UMR CNRS 6306,

Faculté des sciences Mirande, 21078 Dijon, France

yoan.chabot@hotmail.fr

Résumé. Avec la démocratisation des technologies, les enquêtes de criminalistique informatique impliquent des volumes de données toujours plus grands et hétérogènes. Pour faciliter le travail des enquêteurs, nos travaux ont pour objectif de reconstruire automatiquement les évènements liés à un incident numérique, tout en respectant les exigences légales. Pour cela, il est nécessaire d'introduire un modèle de représentation de connaissances permettant de structurer les informations recueillies sur une scène de crime dans le but de faciliter l'utilisation de processus d'analyse automatisés. Ce papier propose un état de l'art des modèles de représentations d'évènements pour le domaine de la criminalistique informatique et introduit ensuite une nouvelle représentation basée sur une ontologie. Un processus de peuplement automatique est ensuite présenté afin d'instancier l'ontologie à partir de données collectées durant une enquête.

1 Introduction

Les nouvelles technologies occupant désormais une place prédominante dans nos vies quotidiennes, il est courant de trouver sur une scène de crime des objets numériques qui sont autant de sources d'informations possibles pour aider les enquêteurs dans la résolution d'une affaire. Le domaine de la criminalistique informatique propose des méthodes d'investigation numériques visant à fournir à la justice des pièces à conviction afin de déterminer la culpabilité ou l'innocence de suspects. Ce domaine de recherche s'intéresse à la résolution de crimes où les technologies sont une cible (e.g attaques par déni de service, utilisation frauduleuse de cartes bancaires), un vecteur principal (e.g. approche d'une victime par un pédophile via les réseaux sociaux) ou un vecteur secondaire (e.g. échange de SMS entre deux complices d'un braquage). Durant une enquête, il est nécessaire d'analyser des grands volumes de données hétérogènes de part la multiplication des objets numériques et l'augmentation de leur capacité de stockage. Par exemple, la boîte à outils *Plaso* (utilisée pour produire des chronologies à partir d'images disques) peut identifier plusieurs milliers d'évènements générés par des sources variées (historiques web, journaux d'évènements Windows, etc.) à partir d'une image disque de quelques

Représentation d'évènements et peuplement d'ontologie appliqués au domaine de la criminalistique informatique

giga-octets. Il est par conséquent nécessaire de développer des processus automatiques pour assister les enquêteurs dans le traitement et l'interprétation de ces données. Cependant, l'utilisation de représentation des données non structurées (format textuel Mactime (Farmer et Venema, 2004) par exemple, utilisé par un grand nombre d'outils) rend le développement de processus d'analyse complexe de part le manque d'informations sur la sémantique des données. Pour faire face à ces problèmes, le recours à une représentation ontologique permet d'une part de structurer les données et d'une autre part de standardiser la représentation des informations. Les objectifs d'une telle représentation sont la simplification du développement d'outils d'analyse ainsi que la mise à disposition des données, pour les investigateurs, sous une forme permettant une consultation intuitive de l'information. Dans nos travaux, nous introduisons une représentation des connaissances permettant de modéliser de manière précise un incident numérique et l'ensemble des étapes composant une enquête. Cette représentation est utilisée dans le cadre d'une architecture ayant pour objectif l'étude *a posteriori* de machines pour la construction et l'analyse de chronologies sémantiquement riches d'incidents numériques (Chabot et al., 2014b). L'utilisation d'une ontologie pour la reconstruction de scénarios d'incidents présentée dans ce papier est une approche novatrice permettant de combler les manques causés par l'utilisation de formats de données plus rudimentaires.

La Section 2 évalue les représentations d'évènements existantes au regard de quatre critères déterminants pour juger de la qualité d'un modèle. Une ontologie pour la représentation d'évènements composant des incidents numériques est ensuite présentée dans la Section 3. Pour conclure, la Section 4 introduit un processus d'extraction et de peuplement permettant d'instancier cette ontologie à partir de données extraites dans une scène de crime.

2 Étude des représentations d'évènements pour la criminalistique informatique

Cette section a pour objectif d'évaluer les solutions de représentation d'évènements au regard de quatre critères :

Complétude du modèle : un modèle doit proposer un vocabulaire suffisamment complet pour représenter de manière précise les entités (événements, objets, processus, etc.) liées à un incident, leurs caractéristiques et les relations entre ces entités. Plusieurs formats de données (Bodyfile, Mactime (Farmer et Venema, 2004), TimeLiNe (Carvey, 2009)) existent pour représenter des chronologies d'évènements. Ces formats utilisant un faible nombre d'attributs, la représentation des évènements est imprécise. De plus, un autre inconvénient de ces formats est qu'ils ne permettent pas de représenter les relations entre entités. Des modèles de représentation plus évolués sont proposés tels que ECF (Chen et al., 2003) et FORE (Schatz et al., 2004). Ces modèles permettent de représenter des dimensions caractéristiques des évènements (temps, objets utilisés, participants impliqués dans un évènement, etc.). Toutefois et à l'instar des formats précédents, ils ne modélisent pas les relations entre les entités (e.g. il est possible de modéliser le fait qu'un évènement interagit avec un objet mais pas de spécifier la nature de cette interaction). (Mudholkar et Bharambe, 2013) proposent une ontologie incluant des dimensions également proposées dans notre modèle telles que le temps ou les protagonistes impliqués dans un incident. Cependant, l'ontologie proposée n'est pas suffisamment décrite

pour permettre son évaluation. Enfin, CybOX¹ est un ensemble de schémas XSD permettant la représentation d'entités (processus ou ressources) et d'évènements les affectant. L'une des spécificités de ce modèle est l'intégration de connaissances techniques à travers un ensemble d'objets (fichiers PDF, historiques Web, connexions réseau, etc.).

Traçabilité des informations : Le deuxième critère de cette étude est l'intégration de données dans le modèle assurant la traçabilité de l'information. Pour satisfaire les exigences légales, les pièces à conviction utilisées lors d'un procès doivent respecter plusieurs critères parmi lesquels la crédibilité des preuves et la reproductibilité de leur méthode de production (Baryamureeba et Tushabe, 2004). Un modèle doit donc permettre la modélisation de la provenance de chaque information produite durant une enquête, incluant des informations sur la nature de chaque tâche accomplie, sur les enquêteurs ayant contribué à chacune d'elles et sur les outils utilisés. Le modèle CybOX incorpore des éléments pour modéliser la provenance de l'information afin de mémoriser pour chaque entité la source d'information et les techniques utilisées ainsi que les contributeurs ayant participé à son identification. Un autre travail pertinent pour ce critère est la recommandation W3C PROV-O (Lebo et al., 2013) décrivant une ontologie composée de concepts et de relations permettant de définir une information ainsi que le processus utilisé pour la produire. Toutefois, cette ontologie n'étant pas appliquée à la criminalistique informatique, certaines caractéristiques spécifiques au domaine sont manquantes.

Automatisation des processus : Ce critère est lié au besoin de produire des outils automatisés capables de traiter de grands volumes de données. La conception de tels outils nécessite que les données soient représentées dans un format compréhensible par des machines. Le rôle de ce critère est d'évaluer le niveau de structuration des données ainsi que la mise à disposition de mécanismes pour faciliter l'utilisation des données par des processus d'analyse automatiques. L'approche ECF introduit une représentation à deux niveaux : un premier niveau contenant des informations génériques sur les évènements et un deuxième niveau contenant des informations spécifiques à chaque type d'évènement. La représentation canonique des évènements permet de les modéliser de manière uniforme indépendamment de la source à partir de laquelle ils sont extraits (facilitant l'analyse des informations) tout en préservant les spécificités de chaque évènement. Le recours à une ontologie, comme illustré dans l'approche FORE, facilite également l'automatisation de part la description formelle de la sémantique des entités permettant aux machines de comprendre la signification des données.

Utilisabilité du modèle : En complément de la mise à disposition des données pour des processus automatiques, les modèles doivent également permettre aux enquêteurs d'accéder à l'information et de comprendre les données (outils de recherche et de visualisation pour un accès intuitif et rapide aux données). Les formats textuels ne permettent pas aux enquêteurs de comprendre aisément les informations contenues dans une chronologie. Les ontologies sont plus à même de répondre à ce critère en proposant des visualisations sous forme de graphes permettant notamment d'identifier rapidement des connexions entre les entités.

En conclusion, aucun des modèles présents dans la littérature ne donne des réponses satisfaisantes à l'ensemble des critères énoncés. Dans la section suivante, nous introduisons un modèle de représentation des évènements apportant des réponses à ces quatre critères. L'ontologie proposée tire parti de l'ontologie PROV-O pour la représentation de la provenance des informations et du modèle CybOX pour garantir sa complétude.

1. <https://cybox.mitre.org/>

Représentation d'évènements et peuplement d'ontologie appliqués au domaine de la criminalistique informatique

3 Une ontologie pour la représentation d'évènements liés à des incidents numériques

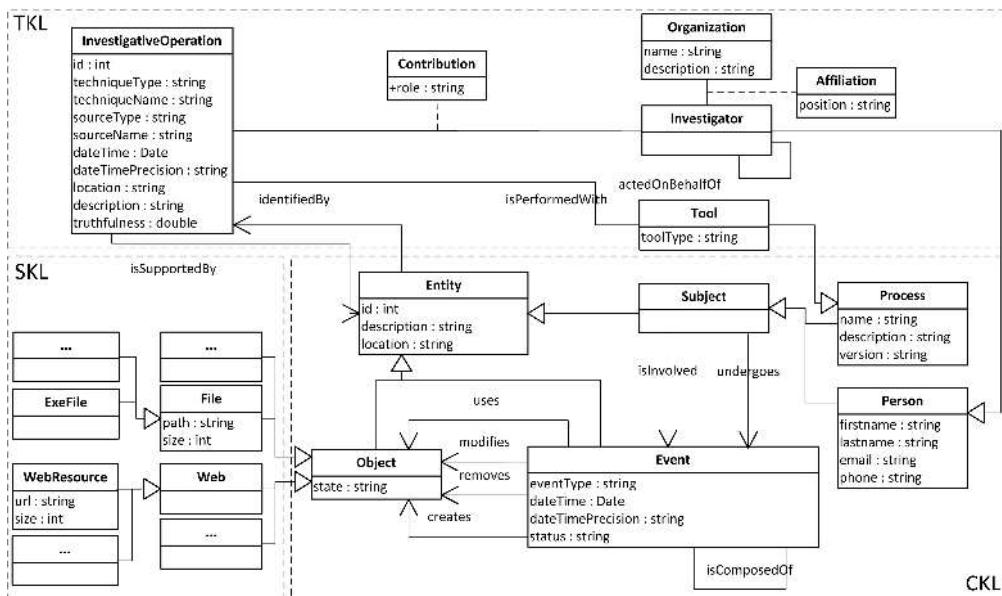


FIG. 1 – Ontologie pour la représentation d'incidents numériques

Pour satisfaire les critères énoncés dans la section précédente, une ontologie implémentée à l'aide du langage *OWL 2 RL* est utilisée. Une ontologie permet de représenter les connaissances d'un domaine donné en structurant ces informations sous forme d'entités, de relations et de contraintes logiques sur ces entités et relations. Les ontologies sont ainsi capables de représenter formellement les connaissances générées durant une enquête (connaissances sur les événements, les processus et les personnes, etc.). Contrairement à des formats de données plus rudimentaires, elles permettent de représenter des relations entre entités ainsi que la logique sous-jacente aux données. La nature explicite et formelle des ontologies permet de faciliter la conception et l'emploi d'outils d'interprétation et d'analyse (déduction de nouvelles informations, vérification de la cohérence des connaissances, etc.) en complément des enquêteurs. Les ontologies sont également une structure facilement manipulable grâce à des outils tels que SPARQL, un langage d'interrogation conçu pour travailler sur des graphes de connaissances. Enfin, la structuration en triplets rend possible la visualisation sous forme de graphes, une représentation claire et intuitive pour les enquêteurs.

L'ontologie proposée dans ce papier est implémentée à l'aide du profil *OWL 2 RL* (sous ensemble de *OWL 2 DL*, un langage basé sur les logiques de descriptions *SHROIQ(D)*). Le choix de ce langage est motivé par plusieurs raisons, dont la mise à disposition d'une expressivité suffisante pour modéliser le domaine nous intéressant. *OWL 2 DL* permet notamment de définir des hiérarchies de classes et de propriétés, des restrictions ou encore d'établir des faits sur les individus tels que l'égalité d'instances. L'utilisation du profil *OWL 2 RL* permet de contraindre

sur certains aspects (expressions de classes notamment) le langage *OWL 2 DL* afin de garantir la décidabilité et la rapidité (complexité polynomiale) des raisonnements à base de règles (Motik et al., 2009). La nécessité d'opérer sur de grands volumes de données et la volonté de proposer aux enquêteurs des outils d'inférence et d'analyse puissants rendent le langage *OWL 2 RL* pertinent pour implémenter une ontologie pour la représentation de chronologies d'incidents. Pour répondre aux besoins de complétude et de traçabilité de l'information, l'ontologie proposée est divisée en trois couches ("Common Knowledge Layer", "Specialized Knowledge Layer" et "Traceability Knowledge Layer") illustrées dans la Figure 1. Afin de garantir la lisibilité, seules les classes, propriétés et attributs nécessaires à la compréhension du papier sont représentés. La classe centrale de l'ontologie est la classe *Entity*, notion abstraite subsumant les classes principales de l'ontologie. Chaque instance de *Entity* est définie par un identifiant unique, une description courte et une localisation. *Entity* est directement spécialisée par les classes *Event*, *Object* et *Subject* et indirectement spécialisée par les classes *Investigator*, *Tool* (TKL) et *Process*, *Person* (CKL).

3.1 Traçabilité des informations et reproductibilité des processus

La couche TKL, inspirée de l'ontologie PROV-O (Lebo et al., 2013), stocke des informations sur la manière dont l'enquête est menée (e.g. participants, étapes de l'enquête, informations en entrée/sortie de chaque étape, etc.). L'objectif de cette couche est de satisfaire les exigences légales en assurant d'une part la reproductibilité des résultats via la mémorisation de chaque action et d'une autre part la crédibilité des résultats en conservant le cheminement et les données utilisées pour produire les résultats. Chaque tâche (*InvestigativeOperation*) est caractérisée par un ensemble d'attributs permettant notamment de définir : le type de techniques utilisées (extraction à partir d'une source d'informations, déduction de nouvelles connaissances, corrélation d'évènements, etc.), les sources d'informations utilisées (archives de conversations, registre Windows, etc.), la date et le lieu où la tâche a été effectuée ou encore une valeur numérique quantifiant le degré de confiance du résultat (peu élevé par exemple, dans le cas d'une tâche utilisant des informations potentiellement corrompues par des assaillants). Les instances de *InvestigativeOperation* sont liées aux outils (*Tool*) utilisés et aux personnels (*Investigator*) impliqués en utilisant respectivement les propriétés d'objets *isPerformedWith* et *Contribution*. Chaque instance de *InvestigativeOperation* est utilisée pour augmenter la connaissance des enquêteurs sur les évènements survenus durant l'incident. Ainsi, chaque tâche de l'enquête est liée aux évènements ainsi qu'aux sujets et objets qu'elle a permis d'identifier. La propriété d'objet *identifiedBy* modélise le fait que toute entité est identifiée à l'aide d'une instance de *InvestigativeOperation* (e.g. une tâche d'extraction d'information à partir d'un historique web peut engendrer l'identification d'un évènement représentant la visite d'une page web). Pour certaines tâches, les enquêteurs doivent raisonner sur des informations déjà existantes pour produire de nouvelles connaissances. La propriété d'objet *isSupportedBy* modélise ce principe en liant les instances de *InvestigativeOperation* aux informations utilisées par celles-ci.

3.2 Connaissances génériques sur l'incident

La couche CKL, dérivée du modèle formel introduit dans (Chabot et al., 2014a), est utilisée pour stocker des connaissances génériques sur les évènements. Elle modélise notamment des connaissances temporelles, des informations sur les objets utilisés par chaque évènement et les

Représentation d'évènements et peuplement d'ontologie appliqués au domaine de la criminalistique informatique

sujets participant à chacun d'eux. Son objectif est d'obtenir une représentation uniforme des évènements composant un incident afin de simplifier les tâches d'analyse en aval. La classe *Event* permet de modéliser tout évènement numérique survenant sur une machine. Chaque instance de *Event* est définie par un type (e.g. copie d'un fichier, suppression d'une clé de registre, etc.), un intervalle de temps représentant la durée ainsi qu'un statut (succès, échec, en cours, inconnu). La propriété *isComposedOf* est utilisée pour lier un évènement à un évènement le composant. Les classes *Subject*, *Process* et *Person* sont utilisées pour modéliser les protagonistes impliqués dans les évènements. Un sujet peut participer (*isInvolved*) à un évènement ou subir (*undergoes*) ce dernier. La classe *Object* représente les ressources utilisées (*uses*), modifiées (*modifies*), supprimées (*removes*) ou créées (*creates*) par les évènements.

3.3 Représentation de connaissances métiers

La couche SKL est utilisée pour stocker des connaissances spécialisées sur les évènements, et notamment les objets utilisés par ces derniers. Elle permet de modéliser des connaissances techniques sur tout objet numérique pouvant être identifié dans une scène de crime numérique. Les informations techniques sur les évènements (adresses IP, chemin et métadonnées de fichiers, etc.) stockées dans cette couche sont des informations de valeur durant la phase d'analyse. La couche SKL propose un panel important de classes permettant de représenter un grand nombre d'objets numériques. Cette couche inclut notamment des objets permettant de représenter :

- Des fichiers (*File*) : *OLECF*, *Link*, *ArchiveFile*, *ImageFile*, *PDFFile*, *ExeFile*.
- Des comptes d'utilisateurs (*Account*) : *UnixUserAccount*, *WinUserAccount*, *ComAccount*.
- Des objets spécifiques au Web et à son utilisation (*Web*) : *Webpage*, *WebResource*, *EmailMessage*, *Bookmark*, *Cookie*, etc.
- Des objets relatifs aux communications (*Communication*) : *MMS*, *SMS*, *Chat*, *Call*.
- Des clés de registre (*RegisterKey*).

4 Peuplement de l'ontologie à partir de traces extraites dans une scène de crime

Cette section a pour objectif d'illustrer le peuplement de l'ontologie à partir de données extraites dans une scène de crime. L'introduction de techniques de peuplement automatisées est primordiale pour permettre le traitement des grands volumes de données extraits lors d'une enquête. La méthode de peuplement utilisée dans notre approche est un processus séquentiel, illustré dans la Figure 2, débutant par la collecte des traces numériques trouvées sur une machine et se terminant par l'instanciation des concepts et des propriétés de l'ontologie.

4.1 Utilisation de la boîte à outils Plaso pour l'extraction d'information à partir de traces numériques

La première étape consiste à extraire l'ensemble des informations contenues dans les différentes sources d'évènements présentes dans l'image disque analysée (image disque des vo-

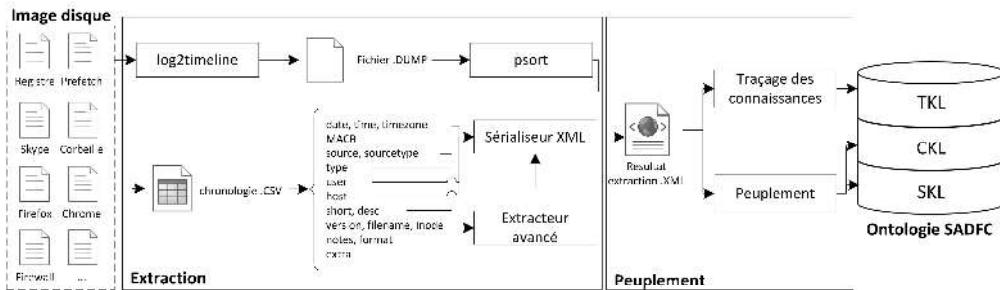


FIG. 2 – Chaîne d'extraction et de peuplement

lumes de la machine étudiée). Durant une investigation, de nombreuses sources peuvent être utilisées afin d'obtenir des informations sur les activités de l'utilisateur. Pour gérer l'ensemble de ces sources, l'outil *log2timeline* (Gudhjonsson, 2010), proposé dans la boîte à outils *Plaso*, est utilisé. Ce dernier collecte des informations à partir de nombreuses sources d'informations, parmi lesquelles : les sources inhérentes au système d'exploitation (e.g. base de registre, système de fichier, corbeille, journaux d'événements) ; les historiques, cookies et fichiers de cache des navigateurs Web ; les fichiers et journaux inhérents à des logiciels divers tels que Skype, Google Drive, etc. Le résultat produit lors de cette étape est un fichier *.dump* contenant l'ensemble des informations extraites de l'image disque. Une transformation du résultat est ensuite nécessaire pour rendre les données utilisables par les processus en aval. Pour cela, l'outil *pso* de la boîte à outils *Plaso* est utilisé. Cet outil permet de sérialiser les données produites par *log2timeline* dans de nombreux formats parmi lesquels le format CSV. Un fragment d'exemple de résultat obtenu en sortie de l'outil *pso* est donné dans la figure 3. Chacune des lignes du fichier illustré dans l'exemple est une entrée décrivant une action survenue sur la machine étudiée. Le premier évènement extrait représente le téléchargement d'un fichier *.exe* à l'aide de Google Chrome. La deuxième entrée décrit l'exécution de ce même fichier *.exe* qui a pu être identifiée via les informations contenues dans le dossier Windows Prefetch. Enfin, la troisième entrée représente la suppression du fichier téléchargé (envoi dans la corbeille).

4.2 Extraction avancée et sérialisation des informations

La sérialisation CSV est structurée en dix-sept attributs donnant des informations temporelles (*date*, *time* et *timezone*), une description de la source d'informations via les champs *source* et *sourcetype* (e.g. historiques de Chrome, Windows Prefetch, corbeille, etc.), une description de l'évènement et de ses conséquences (*MACB*, *type*, *short*, *desc* et *extra*), des informations sur les outils utilisés pour l'extraction et les fichiers utilisés comme source d'informations (*version*, *filename*, *inode*, *notes* et *format*). Certains champs (*date*, *time*, etc.), ne nécessitent aucun traitement particulier et l'extraction des connaissances est aisée. D'autres champs tels que le champ *desc* nécessitent davantage de traitements car leur contenu dépend fortement du type et du déroulement de l'évènement. Ces champs présentent les mêmes inconvénients que les formats textuels présentés dans l'état de l'art et la variabilité de leur contenu rend leur manipulation complexe. Un deuxième problème posé par la chronologie produite par *log2timeline* et *pso* est celui du volume de données. En effet, une chronologie produite à l'aide de ces

Représentation d'évènements et peuplement d'ontologie appliqués au domaine de la criminalistique informatique

```

date , time , timezone , MACB , source , sourcetype , type , user , host , short , desc , version ,
filename , inode , notes , format , extra
11/24/2014,11:50:24 ,UTC ,... B,WEBHIST,Chrome History ,File Downloaded,-,WIN-I51P7DIKOO0,C:\Users\User1\Downloads\Firefox Setup Stub 33.1.1.exe downloaded (244336 bytes),
https://download-installer.cdn.mozilla.net/pub/firefox/releases/33.1.1/ Firefox%20Setup%20Stub%2033.1.1.exe. Received: 244336 bytes out of: 244336 bytes.,2,TSK:/Users/User1/AppData/Local/Google/Chrome/User Data/Default/History,.43770,-,sqlite ,plugin:chrome_history
11/24/2014,11:51:19 ,UTC,,A..,LOG,WinPrefetch ,Last Time Executed,-,WIN-I51P7DIKOO0,FIREFOX .EXE was run 4 time(s),Prefetch [FIREFOX.EXE] was executed - run count 4 path: \USERS\USER1\DESKTOP\SOFTWARE\FIREFOX.EXE hash: 0x9336A096 volume: 1 [serial number: 0x724766A7 device path: \DEVICE\HARDDISKVOLUME1],2,TSK:/Windows/Prefetch/FIREFOX .EXE-9336A096.pf,43408,-, prefetch ,number_of_volumes: 1 volume_device_paths: [u'\DEVICE\HARDDISKVOLUME1'] volume_serial_numbers: [1917281959L] version: 23 prefetch_hash: 2469830806
11/24/2014,11:51:31 ,UTC.M... ,RECBIN,Recycle Bin ,Content Deletion Time,-,WIN-I51P7DIKOO0 ,Deleted file: C:\Users\User1\Downloads\Firefox Setup Stub 33.1.1.exe,C:\Users\User1\Downloads\Firefox Setup Stub 33.1.1.exe,2,TSK:$Recycle.Bin/S-1-5-21-2714290424-3384145025-262107571-1000/ $IOP4Y0X.exe,50944,-,recycle_bin ,file_size: 244336

```

FIG. 3 – Données produites par log2timeline et formatées à l'aide de psort

outils, à partir de l'image disque d'une machine ayant fonctionné environ trente minutes avec une utilisation standard, est composée d'environ 300 000 entrées. L'étude du fichier est réalisée manuellement (e.g. grep, recherche par dates, etc.) par les enquêteurs et l'interprétation de la chronologie est par conséquent particulièrement laborieuse. Cet état de fait valide le choix de l'utilisation de modèles de représentation des données plus avancés tels que les ontologies. Dans l'objectif de faciliter le peuplement de l'ontologie à partir des données produites par *Plaso*, une étape intermédiaire d'extraction de l'information et de sérialisation au format XML est introduite dans notre processus. Pour cela, les informations structurées telles que les informations temporelles (*date*, *time* et *timezone*), les informations sur l'utilisateur (*user*), les informations sur l'hôte (*host*) ainsi que des informations sur le type d'évènements (*source* et *sourcetype*) sont tout d'abord extraites. Les informations contenues dans les champs plus faiblement structurés, tel que le champ *desc*, sont ensuite extraites. Le contenu du champ *desc* dépendant de la source d'informations et du type d'évènements, un ensemble de motifs est défini afin d'extraire correctement les informations. Dans le cas de la première entrée de la Figure 3 correspondant au téléchargement d'un fichier à l'aide de Google Chrome, un motif est utilisé pour extraire dans le champ *desc* l'URL du fichier téléchargé et le chemin local utilisé pour son stockage ainsi que la taille du fichier. L'étape suivante consiste à filtrer les données collectées afin de conserver uniquement les données pertinentes pour alimenter notre modèle dans le but de réduire la quantité de données, améliorer la lisibilité du résultat final et optimiser les temps de traitements. Après le filtrage, les données sont ensuite sérialisées au format XML. La sérialisation de la première entrée de la Figure 3 est donnée dans la Figure 4. Cette figure montre notamment la décomposition à l'aide des motifs des informations contenues dans le champ *desc* au sein de l'élément XML *description*.

```

1 <footprint id="1">
2   <datetime>11/24/2014 11:50:24 UTC</datetime>
3   <type>Chrome History</type>
4   <subtype>Download of a file</subtype>
5   <location>WIN-151P7DIKOO0</location>
6   <user></user>
7   <process>Google Chrome</process>
8   <description>
9     <url>https://download-installer.cdn.mozilla.net/pub/firefox/releases/33.1.1/
10    win32/fr/Firefox%20Setup%20Stub%2033.1.1.exe</url>
11   <localPath>C:\Users\User1\Downloads\Firefox Setup Stub 33.1.1.exe</localPath>
12   <receivedBytes>244336</receivedBytes>
13   <sizeFile>244336</sizeFile>
14   </description>
15   <extra>https://download-installer.cdn.mozilla.net/pub/firefox/releases/33.1.1/
      Firefox%20Setup%20Stub%2033.1.1.exe. Received: 244336 bytes out of: 244336
      bytes.</extra>
16 </footprint>

```

FIG. 4 – Données XML en sortie du processus d'extraction

4.3 Peuplement de l'ontologie et traitements sur les connaissances

La dernière étape consiste à peupler l'ontologie à partir du résultat de l'extraction. Pour chaque élément *footprint* composant le fichier XML, les couches CKL et SKL sont peuplées en créant des instances d'évènements, d'objets et de sujets et des liens entre les individus conformément aux propriétés formelles définies dans l'ontologie (Chabot et al., 2014a). Les relations liant un évènement à un objet ou à un sujet sont déduites en fonction du type de l'évènement (e.g. dans le cas du déplacement d'un fichier vers la corbeille, l'évènement est lié au fichier via la propriété :*removes*). La Figure 5 présente une sérialisation en Turtle des connaissances relatives à l'évènement de téléchargement du fichier à l'aide de Google Chrome utilisé tout au long du document. L'instance :*event1* (lignes 9 à 19) représente cet évènement et est liée aux instances :*webResource1* et :*exeFile1*. L'instance :*webResource1* (lignes 20 à 24) de la classe :*WebResource* représente la ressource distante téléchargée par l'utilisateur. Cette instance est liée par une relation d'utilisation :*uses* à l'évènement car le téléchargement est réalisé à partir de cette ressource distante. :*exeFile1* (lignes 25 à 29) est une instance de la classe :*ExeFile* représentant le fichier .exe local, téléchargé par l'utilisateur. Cette instance est reliée à l'évènement par une relation de création :*creates*. Le processus Google Chrome, utilisé pour mener à bien l'évènement, est représenté par l'instance :*googleChrome* (lignes 30 à 34) et lié à l'évènement via la propriété d'objet :*isInvolved*. Des connaissances décrivant la manière dont les connaissances précédentes ont été extraites sont ensuite ajoutées dans l'ontologie dans la couche TKL. L'instance :*investigativeOperation1* (lignes 42 à 51) représente la tâche d'extraction de l'information réalisée à l'aide de l'outil Plaso (lignes 52 à 57).

Au terme du peuplement de l'ontologie, chaque élément *footprint* en entrée est représenté par un graphe ontologique. Les étapes suivantes consistent à consolider les connaissances présentes dans l'ontologie ainsi qu'à traiter et analyser ces dernières. Ces étapes sont hors de la portée de cette publication mais sont toutefois décrites succinctement ici. La consolidation des connaissances est une étape permettant l'identification des connexions entre les différents graphes de connaissances dans le cas où les évènements interagissent avec des objets ou des

Représentation d'évènements et peuplement d'ontologie appliqués au domaine de la criminalistique informatique

sujets identiques. La phase de consolidation comprend également une étape d'inférence de nouvelles connaissances afin de compléter les connaissances des enquêteurs sur l'incident. Après l'étape de consolidation, un graphe de connaissances de grande taille représentant les informations contenues dans le résultat produit par *Plaso* est obtenu. La structure de ce graphe est dictée par le schéma de notre ontologie. Ce graphe présente de nombreux avantages car il structure l'information et ainsi facilite sa compréhension par les enquêteurs et la mise en place de processus automatiques d'analyse. Le premier outil d'analyse proposé dans notre approche est un outil de corrélation d'évènements permettant de détecter des couples d'évènements liés (Chabot et al., 2014a). L'identification de tels couples est réalisée à l'aide de quatre critères : l'interaction des deux évènements avec des objets communs ou des sujets communs, la proximité temporelle et la validation ou non de règles métiers définies par les spécialistes. Par exemple, soit un évènement A représentant la création d'un marque page pour une page donnée et l'évènement B représentant la visite de cette même page, la valeur du score de corrélation entre les évènements A et B est augmentée par l'utilisation d'un objet commun (la page Web) et l'interaction avec un même processus (le navigateur Web). Le deuxième outil proposé est un algorithme de recherche de motifs permettant de détecter des actions illicites en identifiant des séquences d'évènements particulières (une action illicite peut être composée de plusieurs évènements autorisés d'où la nécessité d'utiliser un système à base de motifs pour détecter correctement les actions délictueuses).

5 Conclusion et travaux futurs

Durant une enquête de criminalistique informatique, les enquêteurs doivent faire face à plusieurs problèmes parmi lesquels le volume de données à traiter, l'hétérogénéité de ces données et les exigences légales. Pour servir de support au développement d'outils d'aide à la décision pour les enquêteurs, il est nécessaire d'introduire un modèle de représentation des informations permettant une modélisation précise des connaissances, l'intégration de données sur la traçabilité, l'automatisation des tâches en aval et une restitution des données intuitive et rapide. Afin de répondre à ces besoins, nous proposons un nouveau modèle de représentation des évènements basé sur une ontologie *OWL 2 RL* et couplé à un processus de peuplement automatisé. Cette ontologie, grâce à son expressivité, permet de répondre au besoin de complétude et à la nécessité de représenter des informations sur la provenance des résultats. De plus, la possibilité d'associer à cette ontologie des outils d'interrogation et de visualisation permet un accès intuitif et efficace aux connaissances et facilite la compréhension des données.

Les travaux futurs se concentreront tout d'abord sur l'extension du processus de peuplement à de nouvelles sources d'informations, afin de s'approcher d'une vision complète des évènements survenus durant un incident. Un autre objectif important est la validation de l'ontologie par des experts du domaine de la criminalistique informatique puis l'obtention d'un consensus au sein de la communauté pour l'adoption d'un modèle de représentation commun afin de faciliter l'interopérabilité des outils utilisés dans le domaine. Enfin, ces travaux soulèvent également des questions éthiques. Bien que notre approche s'applique au domaine de la criminalistique informatique, elle peut également être utilisée par des sociétés tiers (e.g. un fournisseur de services peut utiliser un modèle similaire pour profiler ses utilisateurs) au risque de porter atteinte à la vie privée.

Références

- Baryamureeba, V. et F. Tushabe (2004). The enhanced digital investigation process model. In *Proceedings of the Fourth Digital Forensic Research Workshop*. Citeseer.
- Carvey, H. (2009). Timeline analysis, pt iii, <http://windowsir.blogspot.fr/2009/02/timeline-analysis-pt-iii.html>.
- Chabot, Y., A. Bertaux, C. Nicolle, et T. Kechadi (2014a). A Complete Formalized Knowledge Representation Model for Advanced Digital Forensics Timeline Analysis. *Digital Investigation* 11(2), S95–S105.
- Chabot, Y., A. Bertaux, C. Nicolle, et T. Kechadi (2014b). Automatic Timeline Construction and Analysis For Computer Forensics Purposes. In *IEEE Joint Intelligence & Security Informatics Conference 2014 (IEEE JISIC2014)*, La Haye, Netherlands, pp. 4.
- Chen, K., A. Clark, O. De Vel, et G. Mohay (2003). Ecf-event correlation for forensics. In *First Australian Computer Network and Information Forensics Conference*, Perth, Australia, pp. 1–10. Edith Cowan University.
- Farmer, D. et W. Venema (2004). The coroner's toolkit (tct), <http://www.porcupine.org/forensics/tct.html>.
- Gudhjonsson, K. (2010). Mastering the super timeline with log2timeline. *SANS Reading Room*.
- Lebo, T., S. Sahoo, D. McGuinness, K. Belhajjame, J. Cheney, D. Corsar, D. Garijo, S. Soiland-Reyes, S. Zednik, et J. Zhao (2013). Prov-o : The prov ontology. *W3C Recommendation, 30th April*.
- Motik, B., B. C. Grau, I. Horrocks, Z. Wu, A. Fokoue, et C. Lutz (2009). Owl 2 web ontology language : Profiles. *W3C recommendation* 27, 61.
- Mudholkar, M. et U. Bharambe (2013). A study on significance of event ontology approach in web crime mining. *International Journal of Latest Trends in Engineering and Technology* 2.
- Schatz, B., G. Mohay, et A. Clark (2004). Rich event representation for computer forensics'. *Proceedings of the Fifth Asia-Pacific Industrial Engineering and Management Systems Conference (APIEMS 2004)* 2(12), 1–16.

Summary

Due to the democratization of technologies, computer forensics investigators have to deal with volumes of data increasingly large and heterogeneous. To facilitate the work of investigators, our work aims at reconstructing automatically the events related to a digital incident, while respecting legal requirements. To reach this goal, it is necessary to introduce a knowledge representation model allowing to structure the information collected from a crime scene in order to facilitate the use of analysis processes. This paper first gives a comprehensive state of the art of event representation models for digital forensics and then proposes a new ontology. In addition, an automatic settlement process is then presented to instantiate the ontology using data collected on a machine seized during an investigation.

Représentation d'évènements et peuplement d'ontologie appliqués au domaine de la criminalistique informatique

```

1 @prefix : <http://www.w3.org/2002/07/owl#> .
2 @prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
3 @prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
4 @base <http://www.semanticweb.org/sadfc> .
5 <http://www.semanticweb.org/sadfc> rdf:type :Ontology ;
6   :versionIRI <http://www.semanticweb.org/sadfc/1.0.0> ;
7   :imports <http://www.w3.org/2006/time> .
8 :event1 rdf:type :Event ,
9   :NamedIndividual ;
10  :hasID 1 ;
11  :uses :webResource1 ;
12  :creates :exeFile1 ;
13  :hasEventType "Chrome History"^^xsd:string ;
14  :hasEventSubtype "Download of a file"^^xsd:string ;
15  :isIdentifiedBy :investigativeOperation1 ;
16  :hasLocation "WIN-I51P7DIKOO0"^^xsd:string ;
17  :hasDateTime :interval1 ;
18  :hasDateTimePrecision "sec"^^xsd:string .
19 :webResource1 rdf:type :WebResource ,
20   :NamedIndividual ;
21   :hasID 2 ;
22   :hasSize 244336 ;
23   :hasURL "https://download-installer.cdn.mozilla.net/pub/firefox/
releases/33.1.1/Firefox%20Setup%20Stub%2033.1.1.exe"^^xsd:string .
24 :exeFile1 rdf:type :ExeFile ,
25   :NamedIndividual ;
26   :hasID 3 ;
27   :hasSize 244336 ;
28   :hasPath "C:\\Users\\User1\\Downloads\\Firefox Setup Stub
33.1.1.exe"^^xsd:string .
29 :googleChrome rdf:type :Process ,
30   :NamedIndividual ;
31   :hasID 4 ;
32   :hasName "Google Chrome"^^xsd:string ;
33   :isInvolved :event1 .
34 :interval1 rdf:type :NamedIndividual ,
35   <http://www.w3.org/2006/time#Interval> ;
36   <http://www.w3.org/2006/time#hasEnd> :instant1 ;
37   <http://www.w3.org/2006/time#hasBeginning> :instant1 .
38 :instant1 rdf:type :NamedIndividual ,
39   <http://www.w3.org/2006/time#Instant> ;
40   <http://www.w3.org/2006/time#inXSDDateTime> "2014-11-24T11:50:24"^^xsd:string .
41 :investigativeOperation1 rdf:type :InvestigativeOperation ,
42   :NamedIndividual ;
43   :hasID 5 ;
44   :hasTruthfulness "100.0"^^xsd:double ;
45   :hasTechniqueType "Information Source"^^xsd:string ;
46   :hasTechniqueName "Extraction using Plaso"^^xsd:string ;
47   :hasSourceName "Google Chrome History"^^xsd:string ;
48   :isPerformedWith :plaso .
49   :hasDateTime :interval2 .
50   :hasDateTimePrecision "sec"^^xsd:string ;
51 :plaso rdf:type :Tool ,
52   :NamedIndividual ;
53   :hasID 6 ;
54   :hasVersion "1.1.0"^^xsd:string ;
55   :hasToolType "Digital Forensics"^^xsd:string ;
56   :hasName "Plaso"^^xsd:string .
57 :interval2 rdf:type :NamedIndividual ,
58   <http://www.w3.org/2006/time#Interval> ;
59   <http://www.w3.org/2006/time#hasEnd> :instant2 ;
60   <http://www.w3.org/2006/time#hasBeginning> :instant2 .
61 :instant2 rdf:type :NamedIndividual .
62   <http://www.w3.org/2006/time#Instant> ;
63   <http://www.w3.org/2006/time#inXSDDateTime> "2014-11-25T14:53:10"^^xsd:string .

```

FIG. 5 – Sérialisation Turtle du résultat à l'issu du peuplement de l'ontologie