Digital Investigation 22 (2017) 14–45

Contents lists available at ScienceDirect

Digital Investigation

journal homepage: www.elsevier.com/locate/diin

Advancing coordinated cyber-investigations and tool interoperability using a community developed specification language

Eoghan Casey [a, *], Sean Barnum [b], Ryan Griffith [c], Jonathan Snyder [c], Harm van Beek [d], Alex Nelson [e, 1]

[a] University of Lausanne, Switzerland
[b] MITRE, USA
[c] DoD Cyber Crime Center, USA
[d] Netherlands Forensic Institute, Netherlands
[e] National Institute of Standards and Technology, USA

ARTICLE INFO

Article history:
Received 3 March 2017
Received in revised form
4 August 2017
Accepted 5 August 2017
Available online 26 August 2017

Keywords:
Cyber-investigation
Digital forensics
Specification language
Standard representation
Unified cyber ontology
Information sharing
Digital evidence exchange
Evidence provenance
DFAX
DFXML
CybOX

ABSTRACT

Any investigation can have a digital dimension, often involving information from multiple data sources, organizations and jurisdictions. Existing approaches to representing and exchanging cyber-investigation information are inadequate, particularly when combining data sources from numerous organizations or dealing with large amounts of data from various tools. To conduct investigations effectively, there is a pressing need to harmonize how this information is represented and exchanged. This paper addresses this need for information exchange and tool interoperability with an open community-developed specification language called Cyber-investigation Analysis Standard Expression (CASE). To further promote a common structure, CASE aligns with and extends the Unified Cyber Ontology (UCO) construct, which provides a format for representing information in all cyber domains. This ontology abstracts objects and concepts that are not CASE-specific, so that they can be used across other cyber disciplines that may extend UCO. This work is a rational evolution of the Digital Forensic Analysis eXpression (DFAX) for representing digital forensic information and provenance. CASE is more flexible than DFAX and can be utilized in any context, including criminal, corporate and intelligence. CASE also builds on the Hansken data model developed and implemented by the Netherlands Forensic Institute (NFI). CASE enables the fusion of information from different organizations, data sources, and forensic tools to foster more comprehensive and cohesive analysis. This paper includes illustrative examples of how CASE can be implemented and used to capture information in a structured form to advance sharing, interoperability and analysis in cyber-investigations. In addition to capturing technical details and relationships between objects, CASE provides structure for representing and sharing details about how cyber-information was handled, transferred, processed, analyzed, and interpreted. CASE also supports data marking for sharing information at different levels of trust and classification, and for protecting sensitive and private information. Furthermore, CASE supports the sharing of knowledge related to cyber-investigations, including distinctive patterns of activity/behavior that are common across cases. This paper features a proof-of-concept Application Program Interface (API) to facilitate implementation of CASE in tools. Community members are encouraged to participate in the development and implementation of CASE and UCO.

Introduction

Any investigation can have a digital dimension, often involving information from multiple data sources, organizations, and jurisdictions. Whether in court, battlefield or boardroom, decision makers need to have confidence that the information provided to them is trustworthy. Cyber-investigations support this need and,

* Corresponding author.
E-mail address: eoghan.casey@unil.ch (E. Casey).

[1] Any mention of a vendor or product is not an endorsement or recommendation.

http://dx.doi.org/10.1016/j.diin.2017.08.002
1742-2876/

in that role, are integrated with other domains, including digital forensic science, incident response, counter-terrorism, criminal justice, forensic intelligence and situational awareness. Therefore, to be effective, cyber-investigation information needs to be represented and shared in a form that is usable in any of these contexts, and is flexible enough to accommodate evolving requirements.

This paper describes a community-developed specification language called Cyber-investigation Analysis Standard Expression (CASE), which is intended to serve these needs. The primary motivation for CASE is interoperability – to advance the exchange of cyber-investigation information between tools and organizations (Casey et al., 2017a,b). The power of such a standard is that it provides a common language and structure to support automated normalization, combination, correlation, and validation of information, which means less time extracting and combining data, and more time analyzing information. CASE also supports data marking for sharing information at different levels of trust and classification, and for protecting sensitive and private information (Casey et al., 2017a,b).

CASE is a rational progression from the foundational work on Digital Forensic Analysis eXpression (DFAX), which focused on digital forensic information (Casey et al., 2015).

> "*When investigating a single incident, being able to combine the results from multiple tools that are used to extract information from the digital evidence supports forensic reconstruction, including timeline creation and link analysis. In addition, being able to automate the comparison of similar results from multiple tools facilitates dual-tool verification. When crime spans borders, sharing of information between investigative agencies is crucial for a successful resolution. A fundamental requirement in digital forensics is to maintain information about evidence provenance as it is exchanged and processed, to help establish authenticity and trustworthiness. Furthermore, without a standardized approach to representing and sharing digital forensic information, investigators in different jurisdictions may never know that they are investigating crimes committed by the same criminal.*"
>
> (Casey et al., 2015)

DFAX was created to represent and exchange digital forensic information, using Cyber Observable eXpression (CybOX) to represent the purely technical information, such as digital traces. Although intended as a representation for cyber observables independent of any particular usage context, the initial development priority of CybOX focused on supporting cyber-attack pattern detection and cyber threat intelligence. Because of this, CybOX had limitations in terms of representing some technical content specifically relevant to digital forensic and cyber-investigation information. Since its transfer to the OASIS standards body, CybOX has become much more closely coupled with STIX (Barnum, 2014) reducing its utility and flexibility for information representations other than STIX. In 2016, the independent CybOX was replaced by STIX Observables as an integrated component of the STIX standard, which focuses on cyber threat intelligence (Barnum, 2014). STIX Observables focus on objects relevant to attacks on computer systems, including executable files, processes, Registry keys, email messages, IP addresses, domain names, and URLs. In addition, STIX Observables are embedded within and dependent on the cyber threat intelligence context-specific structure of the STIX schema, which does not cover related domains such as incident response and digital forensic science. In short, STIX does not provide a suitable foundation for representing various cyber-investigation use cases that require more comprehensive expressivity for a wider range of digital traces and their context (e.g., file systems and smartphone apps), and that are bolstered by an ontological approach.

CASE is being developed in unison with the Unified Cyber Ontology (UCO). Leveraging the lessons learned from CybOX and DFAX, UCO provides an improved data model and underlying ontology from which contextually specific cyber-related representations can be defined. Enhancements to UCO have been made to support information representation across multiple cyber domains (e.g., incident response, digital forensic science, counter-terrorism), and to facilitate cross-domain exchange of cyber forensic intelligence. CASE, as a specific profile of UCO, provides support for cyber-investigations in any context, including criminal, corporate and intelligence. CASE and relevant portions of UCO build on the Hansken data model developed and implemented by the Netherlands Forensic Institute (NFI). Building on the success of its precursor XIRAF, Hansken provides a robust platform that supports hundreds of investigations each year. The Hansken data model is a solid foundation for developing CASE, including most common traces that are encountered in cyber-investigations, and is flexible enough to add new types of traces (van Beek et al., 2015).

The novel contributions of this work include:

- Open community-developed specification language and ontology, with a proof-of-concept Application Program Interface (API) implementation, and examples of how to use CASE to support information exchange and tool interoperability;
- Alignment of ontology and data structures with existing forensic systems/tools to facilitate implementation and adoption by tool/system developers;
- Flexible data model (based on duck typing) that can be easily extended to represent any cyber-information and its properties;
- Formalized mechanisms to categorize and annotate *Traces* and *Actions*; including tracking forensic activities central to provenance in cyber-investigations;
- Use of JSON-LD as a default serialization to support full structural and semantic validation of all information in JSON serialized CASE content to the underlying ontological specification.

This paper starts with an overview of prior work and the evolution of CASE, and focuses on several use cases, encompassing the representation and exchange of extracted data and associated provenance details. An overview is provided of the kinds of information that can be represented by CASE, and the role of the underlying UCO is presented. Selection of JSON-LD as the initial serialization of CASE is explained.

The investigative scenario developed for this paper imagines The Oresteia by Aeschylus in the age of mobile devices. The purpose of this scenario is to show how CASE is used to capture information in cyber-investigations involving multiple related crimes to advance sharing, interoperability and analysis. A unifying CASE bundle representing this investigative scenario is provided in Appendix 2, and portions of the JSON are highlighted within the paper to illustrate specific aspects of CASE. The recommended identifier format is based on UUID, because the global uniqueness enables relationships to be defined across multiple cases and data sources. For readability, the examples for this paper use simplified labels instead of realistic UUIDs.

Example 1 shows the beginning of a CASE bundle containing multiple *Investigations*. Each *Investigation* contains a list of the associated elements that are defined in the remainder of the CASE bundle. To reduce repetitive examples in this paper, not every person in the scenario is explicitly represented using a complete *Identity* object. For illustrative purposes, each object that is referenced in this scenario uses the associated person's name in the simplified UUID (e.g. cassandra-device-uuid).

Example 1. Multiple related investigations wrapped in a CASE bundle utilizing JSON-LD serialization.


```
{
"@id": "bundle-3b13e958a-d975-41aa-b1bb-029d2b6707cd",
"@type": "Bundle",
"content": [
  {
    "@id": "investigation-4586742a-710a-454f-bcb8-b60e230ec1b2",
    "@type": "Investigation",
    "name": "Crime A",
    "focus": "Murder",
    "description": "In Mycenae, Atreus killed two sons of Thyestes,
cooked them (except for their hands and heads), fed them to Thyestes, and
then taunted Thyestes with his murdered sons' hands and heads.",
    "object": ["thyestes-uuid", "victim1-uuid", "role-relationship1-
uuid"]
  },
  {
    "@id": "investigation-b05226da-eaef-4bc5-a139-ca12c94dbdfd",
    "@type": "Investigation",
    "name": "Crime B",
    "focus": "Rape",
    "description": "In Mycenae, Thyestes raped his daughter Pelopia to
have a son (Aegisthus)",
    "object": ["thyestes-uuid", "offender1-uuid", "role-relationship2-
uuid", "cctv-recording-uuid", "provenance-record13-uuid"]
  },
  {
    "@id": "investigation-ac9fd560-261e-4cd6-af64-8b83d100b9a8",
    "@type": "Investigation",
    "name": "Crime C",
    "focus": "Murder",
    "description": "In Mycenae, Aegisthus killed Atreus (Agamemnon's
father)",
    "object": []
  },
  {
    "@id": "investigation-2545442b-321c-754d-bcb8-c40d321ce2c2",
    "@type": "Investigation",
    "name": "Crime D",
    "focus": "Murder",
    "description": "In Aulis, Agamemnon killed his daughter Iphigenia as
 a sacrifice to the gods",
    "object": []
  },
  {
    "@id": "investigation-952d677d-6b62-4e53-9bac-1b113d268ac5",
    "@type": "Investigation",
    "name": "Crime E",
    "focus": "Murder",
    "description": "In the Palace of Argos, Agamemnon and Cassandra were
 killed by Clytemnestra (accomplice Aegisthus)",
    "object": ["argos-palace-uuid", "cassandra-uuid", "victim5-uuid",
 "role-relationship5-uuid", "cassandra-device-uuid", "device-location-
 relationship1", "associated-device1-uuid", "clytemnestra-device-uuid",
 "forensic-action1-uuid", "annotation1-uuid", "provenance-record1-uuid",
 "forensic-action2-uuid", "annotation2-uuid", "provenance-record2-uuid",
 "cassandra-mobiledevice-forensicduplicate-uuid", "tool1-uuid",
 "provenance-record3-uuid", "cassandra-mobiledevice-mmssms-uuid", "trace-
 relationship3-uuid", "cassandra-image-partition6-uuid", "trace-
 relationship4-uuid", "tool2-uuid", "tool3-uuid", "forensic-action4-uuid",
 "forensic-action5-uuid", "sms-message1-uuid", "sms-message2-uuid"]
  },
  {
    "@id": "investigation-5aa33dc6-7a39-4731-a754-62a9c41e5220",
    "@type": "Investigation",
    "name": "Crime F",
    "focus": "Murder",
    "description": "In the Palace of Argos, Clytemnestra and Aegisthus
 were killed by Orestes (accomplice Electra)",
    "object": ["electra-uuid", "argos-palace-uuid", "electra-orestes-
 email-uuid", "orestes-facebookmsg-uuid"]
  },
```

This paper concludes with an overview of the ongoing efforts to develop and implement CASE further.

The purpose of this paper is to provide a foundation for broader community involvement in defining what to represent and how, including consumers and producers of cyber-investigation information in public and private sector institutions, experienced professionals and decision makers, tool developers, and several currently active information sharing groups, each with a diverse set of sharing models. Current community involvement includes government and industry, building consensus through collaboration and implementation. These activities include comparing and validating CASE and UCO against existing tools and systems to facilitate implementation and to identify gaps. The CASE repository provides information about these community activities and design decisions such as how file systems and accounts are represented (https://github.com/casework). The repository also contains more detailed examples of digital traces and cyber-investigation tool outputs represented using CASE.

Related work

Schemas proposed in the past focused on discrete subsets of digital traces and did not encompass the full scope of cyber-investigation information (Turner, 2005a, 2006; Eaglin and Craiger, 2005; Lee et al., 2008; Levine and Liberatore, 2009; Flaglien et al., 2011). Digital Forensics XML (DFXML) is a schema that is used by several tools to represent file system information (Garfinkel, 2009, 2012). DFXML primarily represents information on storage media, and does not cover the broader variety of digital traces in cyber-investigations. In addition, representation of provenance in DFXML is limited to execution of tools, and does not encompass the full scope of provenance in cyber-investigations.

The Advanced Forensic Format (AFF4) is used by some tools to store digital forensic information using the Resource Description Framework (Schatz, 1995; Cohen et al., 2009). The AFF4 data model is extremely flexible for storing raw data, and includes built-in compression and encryption. However, AFF4 does not encompass the full range of cyber-investigation information that is covered by CASE and UCO. CASE and AFF4 can be used in unison when data have been saved in an AFF4 file. For instance, an investigation represented using CASE can link to a forensic duplicate of storage media that was saved in an AFF4 file.

The XML Data Encoding Specification for Intelligence Document and Media Exploitation (DOMEX) was developed by the U.S. government to share certain types of information, including a limited set of mobile device details (ODNI, 2016). Although some elements in the DOMEX standard are used to keep track of provenance, the lack of supporting ontology, the very limited expressivity for characterizing cyber observables, and the inability to capture relationships limit the utility and flexibility of this standard.

For the representation of digital traces and their context, CASE incorporates lessons learned from prior schemas, and builds on the Hansken data model developed and implemented by the Netherlands Forensic Institute (NFI). The NFI developed a trace model to support a digital forensic platform called Hansken (van Beek et al., 2015). The Hansken data model improves upon an earlier version called XIRAF (Alink et al., 2006; Bhoedjang et al., 2012). A Hansken trace consists of a unique id, a (birth) name and a set of so-called types with properties. The Hansken data model uses duck typing which allows data to be defined by its inherent characteristics rather than enforcing strict data typing. A type in the Hansken trace model can be compared to a predefined Property Bundle in CASE as illustrated by examples throughout the remainder of this paper and Appendix 2. CASE objects can be assigned any rational combination of Property Bundles, such as a file that is an image and a thumbnail. When employing this approach, data types are evaluated with the duck test, which uses inference to the best explanation. Simply stated, if it walks like a duck, swims like a duck, quacks like a duck, and looks like a duck, then it probably is a duck. This flexible approach is favored over using the OWL concept of inheritance to define an object with various properties. Using inheritance requires permitted properties to be formally defined for each object type, which becomes unwieldy when unexpected combinations of objects are encountered, such as one type of data embedded within another type of data that was not imagined when the ontology was designed.

Examples of type in Hansken are ‘file’, ‘email’ and ‘contact’. A special type named ‘data’ in Hansken (called ‘ContentData’ in CASE) exists to define the properties of the data of the trace, like the entropy and hash values. Another special type is ‘tool’, which captures the ‘how’ portion of provenance for the trace. Each type in Hansken has an origin, defining where the type of the trace comes from. This origin can be ‘extracted’, ‘mined’, ‘processed’ or ‘user-added’. Extracted types are deterministic results of applying forensic tools to data, e.g. ‘file’ or ‘email’. Mined types such as ‘entity’ are the result of applying probabilistic algorithms. Mined types have a property to represent the confidence of the trace based on the probabilistic algorithm it originates from. Processed types describe the process and provide provenance details, such as the ‘tool’ type. Finally, user-added types describe metadata that is added by a user while analyzing the traces. CASE supports this full range of information.

CASE is developed in unison with the Unified Cyber Ontology (UCO) to represent in a consistent manner constructs that are common across a broad range of cyber related domains in order to support interoperability between these domains.

The Ontology for the Representation of Digital Incidents and Investigations (ORD2I) referenced UCO, and provided a proof-of-concept implementation for timeline reconstruction and analysis (Chabot et al., 2015). Both ORD2I and UCO define a separate layer for representing specialized domain knowledge as objects (Cyberitems in UCO, defined as *Traces* in CASE) that could be mapped to a standard representation for sharing and correlating between organizations and tools. A standardized *Traces* layer can be used to represent in-depth knowledge of specialized domains, and can be shared and maintained across related domains such as digital forensic science, intrusion investigation, incident response and cyber threat intelligence. Both UCO and ORD2I provide a generic way to represent activities involving object and entities, and also provide a generic way to represent case information and provenance (called traceability in ORD2I). ORD2I and UCO, and by extension CASE, represent actions performed by forensic examiners and investigators when processing the evidence (ord2i:*InvestigativeOperation* = case:*Forensic Action*), such as keyword searching and decryption, including the tools used (versions, arguments, etc.). To support provenance, CASE uses *Provenance Records* to further characterize *Traces* with information specific to the cyber-investigation domain, such as evidence number. CASE encompasses all aspects of provenance in cyber-investigation domains (e.g., collection at crime scene, photographing evidence, chain of custody documentation), whereas ORD2I concentrates on provenance in the context of data processing using forensic tools. The compatibility between UCO and ORD2I ontologies reflects growing community consensus that has strengthened the development of UCO and CASE.

There are similarities between UCO and the PROV ontology, which was developed to represent provenance of data (https://www.w3.org/TR/prov-overview). It is beneficial to use PROV as a sounding board while developing UCO. However, the PROV ontology focuses on producing data, and does not cover several

important cyber-investigation use cases. An *Activity* in PROV does not provide the needed functions of an *Action* in UCO/CASE, such as the ability to specify inputs, outputs and the instrument that was used. In addition, the result of an *Action* in CASE can be another *Action*, which is not covered by PROV. Furthermore, PROV does not have the same flexibility as CASE to represent links and associations between objects using *Relationship* objects. As UCO and CASE are developed, PROV will continue to be a valuable resource for reference.

Other ontologies and frameworks that have been developed to enable more sophisticated analysis can implement CASE to support standardization and interoperability. For instance, the Digital Evidence Semantic Ontology (DESO) can use CASE to represent known digital traces and to support triage searches of a digital crime scene for matching characteristics (Brady et al., 2015). The Digital Evidence Management Framework (DEMF) can use CASE to represent metadata and provenance information (Cosic and Baca, 2015). The ParFor project can use CASE to represent activities on computer systems (Turnbull and Randhawab, 2015).

The role of ontologies

> "*An ontology defines the basic terms and relations comprising the vocabulary of a topic area, as well as the rules for combining terms and relations to define extensions to the vocabulary.*"
>
> (Neches et al., 1991)

> "*An ontology is a formal, explicit specification of a shared conceptualization.*"
>
> (Studer et al., 1998)

The Unified Cyber Ontology (UCO) provides a rational latticework to buttress CASE, and to build specifications for other cyber-domains that follow an orderly and compatible blueprint. Using a simple analogy, UCO could be thought of as a collection of building blocks and parts, e.g., big blocks, little blocks, seats, tables, windows, wheels. CASE is a particular build that is comprised of various components made available by UCO, specifically suited to a particular kind of construction. Other domains can use many of the same building blocks and parts for their distinct purposes.

Information representations can be defined at various levels of formality, from ad-hoc serialization schemas to explicit models/ontologies. Serializations are necessary for concrete implementations of exchange. Basing these serializations on explicit ontology specifications offers significant advantages, including:

1) Minimized risk of ambiguity and misinterpretation (define semantics in addition to syntax);
2) Abstraction of concepts and structures for consistency and reuse;
3) Portability across serializations and technologies (not locked into a single approach);
4) Integrity of representation is more resilient to evolution and change.

Modelling the information for a specific domain at this level of abstraction and formality can yield clarity both within the domain itself, and for how the information concepts and structures for the domain fit within its broader context. For CASE, the specific domain of interest is cyber-investigation, and the broader context includes digital forensic science, computer/network defense, incident response, criminal justice, cyber/forensic intelligence, malware analysis, vulnerability research, and offensive/hack-back operations. The requirements of all these related domains intersect and overlap, which necessitates consistent, flexible and interoperable representations of this information across each of them. This means that some information concepts and structures necessary for CASE will also be necessary for other use cases within the broader ecosystem. For example, the ability to represent information such as a file, an email or an action is necessary not only for cyber-investigation (CASE) but also for other domains.

Formally modelling as an ontology also provides an explicit basis for semantic alignment with and mapping to other domain ontologies. This provides opportunities for automated translation of instantial content between ontologies as well as the deployment of such instantial content as linked data enabling querying and aggregation of distributed content seamlessly across domains regardless of their native ontology.

Unified cyber ontology

The blueprint provided by UCO defines component ontologies that lay out a proper foundation of fundamental concepts, and that build out various domain concepts in a way that is explicit while maintaining consistency and integrity. Foundational concepts are defined either in the 'uco-core' component or in other components focused on particular cross-cutting concepts such as *Action*. Domain concepts are defined in UCO components focused on the relevant domains such as *Investigation*. The current version of UCO has five component ontologies with four of them (uco-core, uco-action, uco-observable & uco-victim) focused on cross-cutting foundational concepts and one (uco-investigation) focused on domain concepts.

More specifically, UCO serves multiple domains by providing a consistent approach to specifying and extending objects, object identification (IDs), meaning and approach to relationships, approach to data markings, approach and format for expressing time, approach and structure for actions, approach and structure for cyber observable objects, as well as approach and structures for expressing various concepts, including identity, location, roles, tools, and annotations.

UCO specifies meaning and structure for all of these concepts once to avoid ambiguity and to minimize issues of conflicting duplication (typically between related domains or use cases), integrity failure, or inconsistency which can lead to decreased efficiency of automation and higher friction/effort in translation and integration of content. A unified representation of these common concepts facilitates sharing and automation across domain boundaries.

CASE consists of a selection of elements from UCO relevant for representing the information of cyber-investigations, including a universal base Object, *Relationship*, *Action*, *Investigation* (*Forensic Action*, *Provenance Record*), *Identities*, *Roles*, and a Property Bundle extension structure. This extension structure is used by CASE to represent a range of kinds of *Traces* and can be easily extended to cover any cyber item and its properties. Other domains such as malware analysis, cyber threat intelligence, vulnerability management, and security operations could similarly define standardized information representations as profiles of UCO by leveraging any appropriate elements such as UCO core concepts (UcoObject/Facet extension approach, *Relationship*, *Identity*, *Location*, etc.), and *Actions*, as well as defining domain-centric components extending UCO. These new domain-centric components of UCO could then

also be leveraged consistently by any use case for which they are relevant.

A non-comprehensive summary of UCO and its use within the current version of CASE is provided in Appendix 1. Fig. 1 provides a graphical overview of UCO and CASE elements and how they semantically relate to each other, with CASE items outlined in bold.

For ease of understanding and specification, UCO is specified as a UML conceptual model that is autoderived into a formal Resource Description Framework/Web Ontology Language RDF/OWL ontology specification (McGuinness and van Harmelen, 2004). RDF/OWL provides a formally explicit specification for the ontology as well as a rich and extensive ecosystem of technology support for serialization, transformation, semantic mapping and semantic querying. Various serialization bindings (JSON, XML, etc.) can then be specified against the RDF/OWL ontology specification. As discussed briefly below, the default serialization of JSON-LD provides explicit continuity and traceability all the way from the formal ontology to each property of the serialization. This integrated continuum of top-down and bottom-up specification allows implementers and developers to focus on the serialization relevant to them without having to think about the ontology, while domain experts and information architects focus on the overall ontology without worrying about any particular serialization or implementation.

Serialization

Once the overall specification and ontology have been created, there needs to be an implementation of the structure for practical applications.

These sorts of practical implementations of the information representation are serialization bindings of the model/ontology/specification to a particular concrete format such as XML, JSON, or protocol buffers. No single serialization format can be presumed to be the best answer for all situations. Different technology, environment, performance or policy requirements may require different serialization formats. It is important that CASE and UCO enables and supports the specification and use of different serialization formats.

To enable community development and vetting of CASE and UCO, JSON-LD has been selected as the initial default serialization binding (Lanthaler and Gütl, 2012). JSON-LD is 100% valid JSON with some specific JSON structures defined which allow full structural and semantic validation of each object, array and field in the JSON content to a relevant ontological specification for that

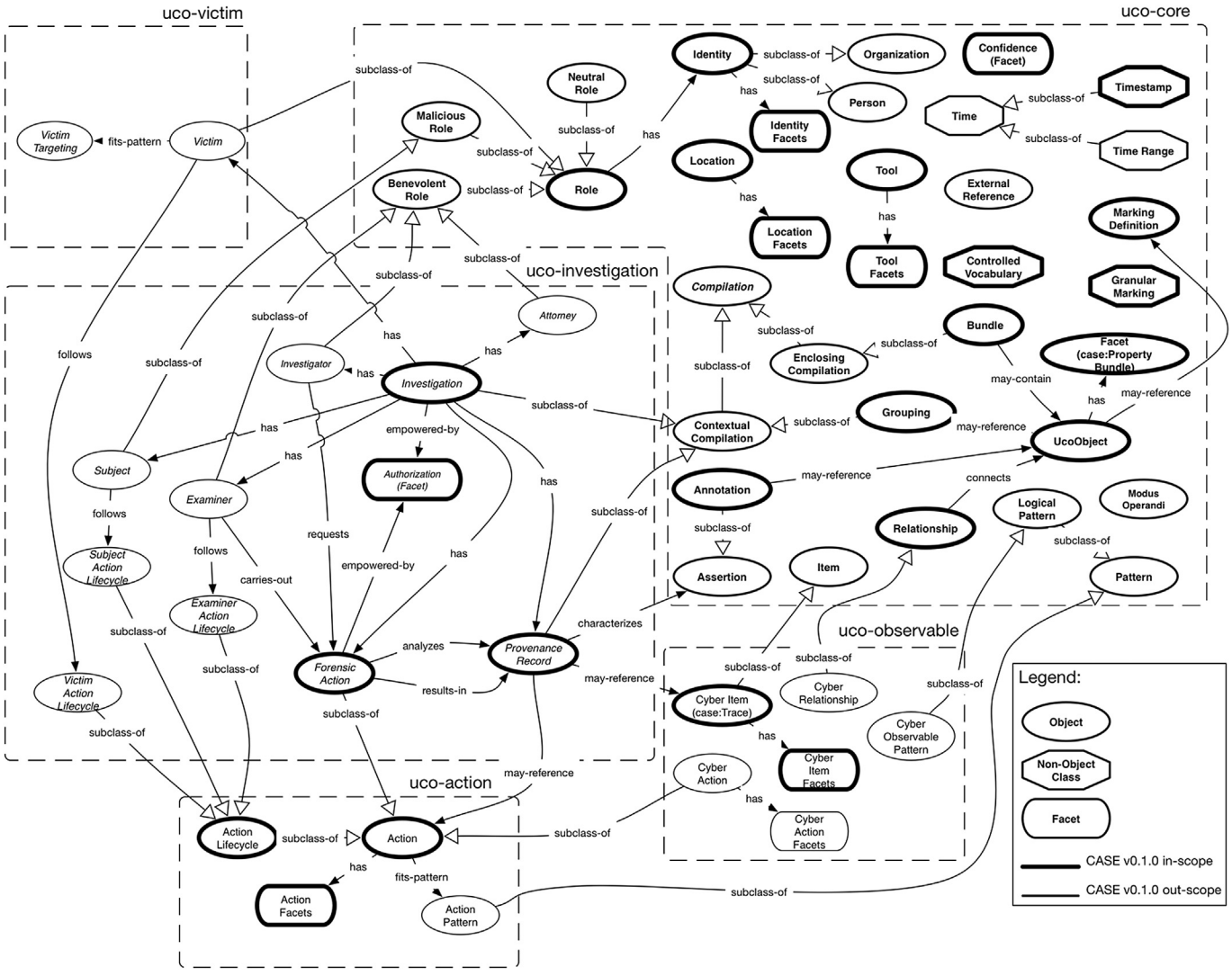

Fig. 1. UCO v0.1.0/CASE v0.1.0 Overview. All items shown are elements of UCO v0.1.0. Items with bold outline are elements of CASE v0.1.0.

element. This explicit validation yields assured integrity between the ontology and the serialization, and offers significant automation advantages including built-in API support for a range of languages (python, ruby, PHP, Go, C#, java, etc.) and for lossless transformation between several serialization formats (JSON-LD, RDF-XML, Turtle-RDF, etc.).

It is important to understand that this common format is for the purpose of **expressing and exchanging** cyber-investigation information. The common format is not intended to define the data model that individual organizations use to develop their databases or applications. As such, developers of systems and applications can translate the common format to their internal implementations. Furthermore, the JSON serialization is only one form of serialization, and the common format could be represented in XML, Turtle (RDF), protocol buffers, or other serializations.

CASE Traces and Property Bundles

In the context of cyber-investigations, traces are the fundamental objects of study. A trace is a vestige, left from a past event or activity, criminal or not. To represent cyber-investigations, it is necessary to capture details about specific traces and their context such as manufacturers and serial numbers of storage media, network connection details, and names of files stored on a removable USB device with associated date-time stamps and cryptographic hash values. To represent this variety of information, as well as other non-trace cyber-investigation information (identities, locations, tools, etc.), CASE defines Objects and potentially associated Property Bundles containing details about the object. CASE leverages the base UcoObject type, derived Object sub-types, and Property Bundles that are defined by UCO. Within CASE, a *Trace* is a special type of object which captures information commonly encountered in cyber investigations. *Traces* include a mobile device, a file extracted from a device, an email address extracted from a file, a location extracted from EXIF metadata.

Some properties are native to the Object itself, and properties that represent contextually related characteristics of the Object are represented using Property Bundles. A *Trace* can have multiple Property Bundles (*File*, *Picture*, *Thumbnail*), each with its own set of properties. A simple example of a *Trace* with the Property Bundles that defines it as an Android device is provided in Example 2. The JSON in this example is JSON-LD, which uses strict @type values to specify the type for all JSON objects, enabling their explicit traceability back to the specifications for these types in UCO. The @type specifies the properties that it expects to find using JSON-LD validation. The *Trace* @type is a sub-class of Object that is used to characterize digital "things" such as devices, files, URLs, and network connections.

Example 2. Properties of an Android device represented using CASE.

```
{
"@id": "cassandra-device-uuid",
"@type": "Trace",
"propertyBundle": [
  {
    "@type": "Device",
    "manufacturer": "Samsung",
    "model": "SM-G925F Galaxy S6 Edge",
    "serialNumber": "FDG344657"
  },
  {
    "@type": "MobileDevice",
    "keypadUnlockCode": "1234",
    "IMEI": "359305065690067",
    "clockSetting": "2017-06-22T07:36:24.35Z",
    "timezoneSetting": "UTC+01:01 (Europe/Rome)",
    "storageCapacity": "11 GB"
  },
  {
    "@type": "MobileAccount",
    "MSISDN": "1239275339"
  }
]
},
```

Each Object is assigned an identifier (@id) that can be used to refer to the Object, as discussed in the next section (CASE references).

Example 3. Properties of a File represented using CASE.

```
{
 "@type": "Trace",
 "@id": "cassandra-mobiledevice-mmssms-uuid",
 "propertyBundle": [
   {
     "@type": "File",
     "createdTime": "2017-06-22T08:12:19.32Z",
     "fileSystemType": "EXT3",
     "extension": "db",
     "fileName":"/data/data/com.android.providers.telephony/mmssms.db",
     "isDirectory": false,
     "sizeInBytes": 122925
   },
   {
     "@type": "ContentData",
     "sizeInBytes": 122925,
     "magicNumber": "U1FMaXRlIGZvcm1hdCAzAA==",
     "hash": [
       {
        "@type": "Hash",
        "hashMethod": "SHA256",
        "hashValue": "a13225720074371d56a4f4d5117fbb4953c5b1d316b31f21edcb7ed8fdf66c6e"
       }
     ]
   }
 ]
}
```

Property Bundles can be defined and added within CASE and UCO as needed to represent specialized cyber-investigation information.

It is worth noting that, during the development of CASE, an unsuccessful attempt was made to represent everything using a single type of Object (an Item). The idea was to use Property Bundles for all properties and to let each Item be defined by the Property Bundles that were assigned to it, strictly following the duck typing model implemented in the Hansken system (van Beek et al., 2015). Although this approach works well for things like *Traces* that are specific to a cyber-investigation, many other things are conceptually distinct, can be shared across domains, and have value in independent semantic definitions. It was determined that different types of Objects were needed, including *Annotations*, *Identities*, *Locations*, *Relationships*, *Roles*, and *Tools* as detailed in the **UCO Core Entities** section below.

CASE references

In general terms, a "reference" is a property of an Object that cannot be changed (a.k.a. immutable) that points to another Object, representing a relationship to that other Object. In CASE, such references are represented using an embedded property that specifies the @id of another Object. Example 4 shows SMS messages with reference to accounts that contain phone numbers.

Example 4. SMS message represented using CASE with references to *Account Traces*.

```
{
   "@id": "sms-message1-uuid",
   "@type": "Trace",
   "propertyBundle": [
    {
      "@type": "Message",
      "application": "sms-application1",
      "messageText": "A wedded wife, she slays her lord,
Helped by another hand!",
      "from": "cassandra-mobileacct-uuid",
      "to": [
        "argive-elder1-phoneaccnt-uuid",
        "argive-elder2-phoneaccnt-uuid",
        "argive-elder3-phoneaccnt-uuid"
      ],
      "sentTime": "2017-06-20T09:34:42.12Z"
    }
   ]
 },
 {
  "@id": "argive-elder1-phoneaccnt-uuid",
  "@type": "Trace",
  "propertyBundle": [
    {
      "@type": "PhoneAccount",
      "phoneNumber": "1237771337",
    }
  ]
 }
```

The *Account* Property Bundle is used to represent properties of any type of account. Properties of specialized types of accounts such as a phone number or an email address, are represented with separate Property Bundles focused on the specific account type (e.g. *PhoneAccount*, *EmailAccount*). Additional examples of accounts represented using CASE are provided in the "accounts.json" file on GitHub.[2]

Another illustrative example of references in an Object being represented with the *Message* and *Attachment* Property Bundles is provided in the CASE repository within "message.json."[3] In that example, *Message* properties 'from' and 'to', and *Relationship* properties 'target' and 'source' all reference other Objects. The fact that these properties are references to other objects is defined in the underlying ontology. Some properties, if defined in the ontology, can enforce ordering. CASE utilizes the Ordered List Ontology as a specialized implementation for ordered arrays within the current version of UCO. This ordering is demonstrated using the same example ("message.json" on GitHub) – each referenced Object ('message1', 'message2', 'message3') within the 'messages' property is listed in a particular order.

UCO core entities

CASE implements UCO to represent certain types of information that transverse the cyber domain as core entities, including *Annotations*, *Identities*, *Locations*, *Relationships*, *Roles*, and *Tools*. These core entities are sub-classes of Object, which is an intentional deviation from the duck test to avoid ambiguity resulting from the implicit rather than explicit identification of what concept is being expressed. Explicit characterization of these core entities and objects (based on the concept being represented) provides semantic clarity, consistency of use across domains and use cases, and facilitates alignment of CASE and UCO with external ontologies and representations.

Example 5. A crime scene Location represented using CASE.

```
{
 "@id": "argos-palace-uuid",
 "@type": "Location",
 "propertyBundle": [
   {
     "@type": "SimpleAddress",
     "locality": "Argos",
     "region": "Greece",
     "postalCode": "98052",
     "street": "Palace Blvd"
   },
   {
     "@type": "LatLongCoordinates",
     "latitude": 48.860346,
     "longitude": 2.331199
   }
 ]
}
```

Example 6. A person Identity represented using CASE.

```
{
  "@id": "cassandra-uuid",
  "@type": "Identity",
  "propertyBundle": [
    {
      "@type": "SimpleName",
      "givenName": "Cassandra",
      "familyName": "Troy"
    },
    {
      "@type": "BirthInformation",
      "birthdate": "1968-09-25T17:59:43.25Z"
    }
  ]
}
```

CASE also leverages cyber-investigation focused Object types from UCO, including *Investigation*, *Forensic Action*, *Provenance*

[2] https://github.com/casework/case/blob/master/examples/accounts.json.
[3] https://github.com/casework/case/blob/master/examples/message.json.

Example 7. A Tool represented using CASE.

```
{
 "@id": "tool1-uuid",
 "@type": "Tool",
 "name": "MobileExtractor",
 "toolType": "Extraction",
 "creator": "Zeus",
 "version": "5.3"
 "propertyBundle": [
    {
        "@type": "ToolConfiguration",
        "configurationSetting": [
            {
                "@type": "ConfigurationSetting",
                "itemName": "extraction_method",
                "itemValue": "omnipotent"
            }
       ]
    }
},
```

Record, *Action*, and *Action Lifecycle* which are described in the following sections.

An open point for community discussion is which existing standards to utilize for representing core entities such as Identity and Location. This discussion needs to address what information is treated as part of *Identity* versus a *Trace*, such as an email address. Account information extracted from a computer or mobile device can be linked to an identity, which may be an actual person or fictitious entity. However, it important to realize that identity information describes a specific entity in a given context, during a certain time, with some level of confidence (Casey and Jaquet-Chiffelle, 2017). CASE can capture these subtleties of identity information using *Relationships* and *Confidence* as detailed in the following sections.

Relationships

Capturing the relationships between Objects is important in cyber-investigations. As a general rule, in CASE, a link between two Objects should be represented as an independent *Relationship* Object specifying the type of connection (a.k.a. external relationship). Otherwise, if a related Object represents an inherent and indelible property within an Object, it should be represented as an identifier reference within the Object (a.k.a. embedded reference). For example, when location information is not extracted from a *Trace* or is added later, it needs to be represented using a separate *Relationship* object that links the *Trace* with an associated *Location* object as shown in Example 8. However, when a *Trace* contains location information such as longitude and latitude coordinates, these can be represented using a property with a reference to the associated *Location* Object.

Example 8. Example of a *Relationship* represented using CASE, referring to the Android device in Example 2 above located at the crime scene represented in Example 5.

```
{
  "@id": "device-location-relationship1",
  "@type": "Relationship",
  "source": "cassandra-device-uuid",
  "target": ["argos-palace-uuid"],
  "kindOfRelationship": "located-at",
  "startTime": "2017-06-19T13:59:43.25Z",
  "endTime": "2017-06-22T15:59:43.25Z",
  "isDirectional": true,
{
```

As shown in Example 9, *Relationships* in CASE can also be used to link an individual to a specific computer or mobile device, and can represent what role the person had in the investigation (e.g., victim, offender, and investigator).

Example 9. *Relationships* defined in CASE to represent the *Role* that a particular person had in the *Investigation* (Cassandra was a Victim) and her associated mobile device (represented in Example 2).

```
{
  "@id": "victim5-uuid",
  "@type": "Role",
  "name": "Victim"
},
{
  "@id": "role-relationship5-uuid",
  "@type": "Relationship",
  "source": "cassandra-uuid",
  "target": ["victim5-uuid"],
  "kindOfRelationship": "has-role",
  "isDirectional": true
},
{
  "@id": "associated-device1-uuid",
  "@type": "Relationship",
  "source": "victim5-uuid",
  "target": ["cassandra-device-uuid"],
  "kindOfRelationship": "has-device",
  "isDirectional": true
},
```

Considerable attention was given to how CASE represents file systems, files, and their contents because these are fundamental objects in most cyber-investigations and supporting tools. Separate Property Bundles were defined for *FileSystem*, *File* and *ContentData*. The link between a *FileSystem* and *File* is represented using a *PathRelation* relationship, with properties on a "*contained-within*" *Relationship* characterizing where the file *Trace* (*Relationship.Source*) is located within the enclosing container (*Relationship.Target*) as shown in Fig. 2 and represented in Examples 10 and 11.

Example 10. Example of a *Relationship* represented using CASE, showing the file system location of the file represented in Example 3.

```
{
  "@id": "trace-relationship3-uuid",
  "@type": "Relationship",
  "source": "sqlite_database",
  "target": "cassandra-image-partition6-uuid",
  "kindOfRelationship": "contained-within",
  "isDirectional": true,
  "propertyBundle": [
   {
    "@type": "PathRelation",
    "path": "/data/data/com.android.providers.telephony/mmssms.db"
   }
  ]
},
```

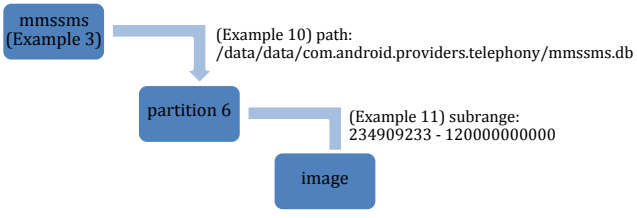

Fig. 2. Schematic example of multiple *Relationships* in CASE, depicting the sample JSON in Examples 10 and 11.

Example 11. Example of using *Relationship* in CASE to represent an EXT3 file system in a partition within a forensic duplicate.

```
{
"@id": "cassandra-image-partition6-uuid",
"@type": "Trace",
"propertyBundle": [
 {
      "@type": "DiskPartition",
      "diskPartitionType": "MSDOS"
      "partitionID": "06"
      "partitionOffset": "63"
      "partitionLength": "245235063"
 },
 {
      "@type": "FileSystem",
      "fileSystemType": "EXT3"
 },
 {
      "@type": "ContentData",
       "sizeInBytes": 245235000,
       "hash": [
         {
           "@type": "Hash",
           "hashMethod": "SHA256",
           "hashValue":
 "0611ea093d19b1c73a5285ff43741dd77f2a8d983c1c71044eb072e44f5dcb0a"
         }
       ]
     }
   ]
 },
 {
 "@id": "trace-relationship4-uuid",
 "@type": "Relationship",
 "source": "cassandra-image-partition6-uuid",
 "target": "cassandra-mobiledevice-forensicduplicate-uuid",
 "kindOfRelationship": "contained-within",
 "isDirectional": true,
 "propertyBundle": [
  {
       "@type": "DataRange",
       "rangeOffset": 234909233
       "rangeSize": 120000000000
   }
   ]
 }
```

This approach is flexible enough to represent files within other data structures, not only file systems. For instance, CASE can represent files within other files, such as files in an ISO disk image file that is stored on a computer. More extensive documentation and an illustrative example can be found in the “files.json” example on GitHub.[4]

Another example of representing files was generated in the “bulk_extractor_forensic_path.json” on GitHub to show how CASE can be used to represent a forensic_path created by the Bulk Extractor tool based on an example in Garfinkel, 2013.[5]

Confidence

UCO, and by extension CASE, includes confidence as a property of any Object including Relationships. Example 12 shows how the link between an *Identity* (e.g., a person) and an *Account* can be represented using CASE with the associated confidence.

[4] https://github.com/casework/case/blob/master/examples/file.json.

[5] https://github.com/casework/case/blob/master/examples/bulk_extractor_forensic_path.json.

Example 12. An *Identity* linked to a Facebook account represented using CASE.

```
{
  "@id": "orestes-uuid",
  "@type": "Identity",
  "propertyBundle": [
    {
      "@type": "SimpleName",
      "givenName": "Orestes",
      "familyName": "Argos"
    },

     {
       "@type": "BirthInformation",
       "birthdate": "1999-12-25T01:59:59.01Z"
     }
   ]
 },
{
  "@id": "associated-facebookaccount-uuid",
  "@type": "Relationship",
  "source": "orestes-uuid",
  "target": ["orestes-facebookaccount-uuid"],
  "kindOfRelationship": "has-account",
  "isDirectional": true
  "propertyBundle": [
    {
      "@type": "Confidence",
      "confidence": "Confirmed by other sources"
    }
  ]
}
```

The confidence property of the *Confidence* Property Bundle is of *Controlled Vocabulary* type, meaning that it can be set to a string from a defined list. Using this approach, confidence can be set for any link or inference, including timestamp accuracy, location, account ownership, investigative assertion, and expert opinion. CASE supports user-defined representations of confidence. An open question for community discussion is what vocabulary to use by default for representing confidence. One approach is to use values between 0 and 1 to represent confidence, but this does not cover situations in which the confidence is unknown or the information is known to be incorrect. Another approach is to use the Admiralty Code Credibility Scale, with the following values: 1 = Confirmed by other sources, 2 = Probably True, 3 = Possibly True, 4 = Doubtful, 5 = Improbable, 6 = Truth cannot be judged. Whatever approaches are used to represent confidence, there will be situations in which the confidence is unknown or known to be unreliable.

Annotations

A flexible *Annotation* Object in CASE can be associated with any Object or set of Objects. An *Annotation* can consist of free text and/or keywords as shown in Example 13. This provides a mechanism for labeling and grouping, and also for adding general notes and comments to Objects or groups of Objects (e.g., bookmarks).

Example 13. Example of an *Annotation* represented using CASE.

```
{
  "@id": "annotation1",
  "@type": "Annotation",
  "tag": ["selfie", "picture"],
  "description": "Digital photograph of corpses taken
at crime scene by killer",
  "object": [
      "orestes-selfie-photograph-uuid"
 ]
}
```

Annotations are intended to be flexible enough to represent bookmarks that encapsulate multiple Objects, and to associate notes with specific things. An *Annotation* can be added by a process or a person, including the categorization of a *Trace* as extracted, mined, processed or user-added.

Future development of CASE and UCO will determine whether *Annotations* can be used effectively to represent assertions made by either a person or tool, or whether a separate object is necessary. Related to this, ORD2I uses the "isSupportedBy" property to model the link between an outcome and its input. For instance, isSupportedBy can be used to show that new information was deduced from a specific *Trace*, effectively representing how investigators reached a given conclusion.

Provenance

In any cyber-investigation, it is necessary to capture information about the origin of a *Trace* and how it was handled after it was found, generally referred to as provenance (Turner, 2005a,b; Levine and Liberatore, 2009; Casey et al., 2015).

In a legal context, the evidence authentication process uses information about provenance, including evidence collection documentation, continuity of possession forms (chain of custody), audit logs from forensic acquisition tools, and integrity records, which all help establish the trustworthiness of digital traces.

In the context of forensic examination, provenance refers to the source and extraction method of specific *Traces* such as e-mail messages, attachments, and their associated metadata being extracted from a Microsoft Outlook PST file using a specific software application. Analyzing the provenance of a Trace can also be used to ascertain whether it is forged or the genuine object.

The CASE standard uses *Provenance Records* to capture contextual and descriptive information about Objects that is specified by cyber-investigation/forensic personnel or tools. A simple example of an Object with multiple *Provenance Records* is a single device that is initially labeled and described by one agency, and later labeled and described differently by another agency. CASE represents these labeling occurrences as two forensic actions and associated *Provenance Records* on the same *Device* Object. Example 14 shows two *Provenance Records* for the same Android smartphone handled and labeled by two different police departments (see the next section for discussion of Forensic Actions).

Example 14. Example of a CASE *Provenance Records* for the Android device represented in Example 2.

```
{
  "@id": "provenance-record1-uuid",
  "@type": "ProvenanceRecord",
  "description": "Mobile device used by murder victim Cassandra",
  "exhibitNumber": "ArgosPD-20170622-001A",
  "object": [ "cassandra-device-uuid" ]
},
{
  "@id": "provenance_record2-uuid",
  "@type": "ProvenanceRecord",
  "description": "Android smartphone seized by Argos PD",
  "exhibitNumber": "AthensPD-2017220601",
  "object": [ "cassandra-device-uuid" ]
}
```

One of the strengths in the Hansken data model is that it differentiates between properties that are extracted, mined, processed, and annotated. This explicit categorization makes the meaning and use of *Trace* properties clearer. To capture this benefit, CASE uses the Action Lifecycle structure discussed below.

Another benefit of the Hansken data model is that provenance is baked into the data model using a tree structure. However, the

simple parent child relationship structure in the Hansken model does not capture the specific nature of relationships between digital objects, which limits the flexibility for representing cyber-investigation information. The CASE standard uses Forensic Actions to provide additional flexibility for representing provenance as described in the next section.

Actions

In addition to the activities to process available data sources, cyber-investigations involve analysis of offender activities that are represented within digital data. Any of these activities can be represented as an *Action*, which is a core entity sub-class of Object in CASE. An *Action* can be used to represent any activity associated with Objects and other *Actions*, capturing higher-level, human understandable portrayals of patterns (a.k.a. artifacts) that enable more efficient forensic analysis (Hargreaves and Patterson, 2012; Casey et al., 2015; Brady et al., 2015). Example 15 shows an *Action* that represents *Traces* of file wiping found on the suspect's computer.

Example 15. Example of the *Action* of a wiping tool used on the suspect's computer to overwrite a file. The original file name is unknown (represented as ????.???) and the resulting filename is aaaaaaaa.aaa.

```
    {
      "@id": "file1",
      "@type": "Trace",
      "propertyBundle": [
        {
          "@type": "File",
          "filePath": "\username\secretfiles",
          "fileName": "????.???"
        }
      ]
    },
    {
      "@id": "file2",
      "@type": "Trace",
      "propertyBundle": [
        {
          "@type": "File",
          "filePath": "\username\secretfiles",
          "fileName": "aaaaaaaa.aaa"
        }
      ]
    },
    {
      "@id": " action13",
      "@type": "Action",
      "name": "wiped",
      "startTime": "2017-01-15T21:49:42.22Z",
      "propertyBundle": [
    {
      "@type": "ActionReferences",
      "instrument": "wiping_tool1",
      "environment": "suspect_computer1",
      "performer": "suspect1",
      "object": [
        "file1"
      ],
      "result": [
        "file2"
      ]
    }
  ]
},
{
  "@id": "annotation13",
  "@type": "Annotation",
  "tag": ["suspect"],
  "description": "Traces of file wiping using tool found on suspect's computer.",
  "object": [
    "action13"
  ]
}
```

In addition to representing extracted timestamps as properties, any activities with an associated timestamp could also be represented as Actions, which is an interpretation with some associated confidence, depending on the origin and/or trust in the clock. There are multiple advantages of representing this information as Actions rather than extracted data. Although this representation of the data may not be the same as how it is displayed within an application, the information is not changed and can be mapped between the JSON serialization and application specific representation.

When an Action contains indelible location or performer information, this can be represented using a property that references the associated Object. However, when the location or performer is not indelible within a *Trace* or might need to be updated later, this information must be represented using a separate Relationship object. For instance, when location details are part of extracted data, they can be represented via reference. The indelible quality of a reference is not asserting that the information is true, just that it is what was represented in the extracted data. For any situation requiring interpretation or potential future revision of location or performer, the information should be represented with a Relationship, with some level of confidence.

Forensic Actions

Cyber-investigations require traceability and chain of evidence. *Forensic Actions* provide the backbone for provenance information, or audit trail, maintaining a chain of all evidence handling and processing activities. To maintain information about how evidence is exchanged and processed, each action that is performed on an Object is represented using a *Forensic Action* as shown in Example 16, including information about who (performer or tool) did what (a verb such as seized, imaged, executed), when, and how. This illustrative example captures activities to preserve a CCTV recording of a rape, generating an associated *Provenance Record*. Further processing of the CCTV recording such as enhancement or facial comparison can be represented by *Forensic Actions*, using the *Provenance Record* as the input object in order to keep track of each step in the provenance.

Authorizations for cyber-investigations such as legal authorizations (search warrant, etc.) are represented in CASE as *Traces*, which are referenced within *Forensic Actions* as shown in bold in Example 16 above. This approach allows a single authorization to be referenced from multiple *Forensic Actions* without duplicating the information.

The environment in which the *Forensic Action* occurred can be represented, whether it be a computer system (forensic tool running on a laptop running Chrome), or a physical location (photographing evidence in a bathroom at the crime scene).

The role of the *performer* can be specified within each *Forensic Action*, allowing one person to have multiple roles throughout the forensic lifecycle: a first responder during the preservation phase, a forensic examiner during the examination phase, and an expert witness during the presentation phase.

As a general rule, *Forensic Actions* do not reference Objects directly, but rather reference *Provenance Records* when available in order to maintain provenance information. The exception to this general rule is the special case of a *Forensic Action* on the original Object being handled for the first time, such as when it is seized at a crime scene as shown in Example 16 above. In addition, a *Forensic Action* can output other *Forensic Actions*, such as when an automated tool launches modules to process Objects as shown in Example 17.

Example 16. Example of a *Forensic Action* and *Provenance Record*.

```
{
  "@id": "forensic_action13",
  "@type": "ForensicAction",
  "name": "preserved",
  "startTime": "2017-01-16T06:34:22.56Z",
  "propertyBundle”: [
    {
      "@type": "ActionReferences",
      "instrument": ["warrant1"],
      "location": " mycenae-palace-uuid",
      "performer": "mypd-investigator2",
      "object": [
        "cctv-recording-uuid"
      ],
      "result": [
        "provenance_record13"
      ]
    }
  ]
},
{
  "@id": "annotation14",
  "@type": "Annotation",
  "tag": ["forensic"],
  "description": "Forensic preservation of CCTV recording showing
    Thyestes sexually assaulting Pelopia.",
  "object": [
    "forensic_action13"
  ]
},
{
  "@id": "provenance_record13",
  "@type": "ProvenanceRecord",
  "description": "CCTV Recording",
  "exhibitNumber": "MYPD-2017011601",
  "object": ["cctv-recording-uuid"]
}
```

Example 17. Example of one forensic tool spawning a subprocess, represented using CASE as a *Forensic Action* that results in another *Forensic Action*.


```
{
"@id": "forensic-action4-uuid",
"@type": "ForensicAction",
"name": "extracted",
"startTime": "2017-06-22T09:57:23.64Z",
"endTime": "2017-06-22T10:31:19.24Z",
"propertyBundle": [
 {
   "@type": "ActionReferences",
   "location": "argos-palace-uuid",
   "performer": "aeschylus-uuid",
   "instrument": "tool2-uuid",
   "environment": "forensic-computer1-uuid",
   "object": [
     "cassandra-mobiledevice-forensicduplicate-uuid",
     "provenance_record2-uuid"
   ],
   "result": [
     "forensic-action5-uuid",
     "provenance-record3-uuid",
     "cassandra-mobiledevice-mmssms-uuid"
   ]
 }
]
},
{
"@id": "forensic-action5-uuid",
"@type": "ForensicAction",
"name": "parsed",
"startTime": "2017-06-22T09:57:23.64Z",
"endTime": "2017-06-22T10:31:19.24Z",
"propertyBundle": [
 {
   "@type": "ActionReferences",
   "location": "argos-palace-uuid",
   "performer": "forensic-action4-uuid",
   "instrument": "tool3-uuid",
   "environment": "forensic-computer1-uuid",
   "object": [
     "cassandra-mobiledevice-mmssms-uuid"
   ],
   "result": [
     "sms-message1-uuid",
     "sms-message2-uuid",
     "argive-elder1-phoneaccnt-uuid",
     "argive-elder2-phoneaccnt-uuid",
     "argive-elder3-phoneaccnt-uuid"
   ]
 }
]
},
```

Action Lifecycles

All activities that can be represented using *Actions* can be categorized within any predefined *Action Lifecycle*, whether the lifecycle relates to activities performed by offenders, victims, or cyber-investigators such as chain of custody, data extraction, correlation to generate a timeline, semantic reasoning, etc. (Casey et al., 2015).

The *Action Lifecycle* construct in CASE can be adapted to distinguish between *Forensic Actions* performed during various phases of a cyber-investigation. Each *Forensic Action* can be categorized according to the phase(s) of a *Forensic Action Lifecycle* that represents steps in the forensic process in order to provide context for each action as shown in the "forensic_lifecycle.json" example on GitHub. For instance, one organization could describe the steps of the forensic process as documentation, preservation, examination, analysis, and presentation. Another organization could describe the steps of the forensic process differently, but still use the *Forensic Action Lifecycle* structure and categorize each *Forensic Action* accordingly. In ORD2I, the Extraction, Settlement, Enhancement, and Analysis phases of Semantic Analysis of Digital Forensic Cases (SADFC) can be represented in CASE as an *Action Lifecycle*. In addition to providing context for each *Forensic Action*, this categorization can be useful for gathering insight into what tools were used or results were produced in different phases of the forensic process to determine preference and trends.

This type of information can be used to address various questions such as how much time was taken by each phase of an investigation, determining which tools are most useful for a given phase, and isolating which results were generated at different phases.

A significant difference between CASE/UCO and ORD2I is the distinction between actions performed by victims or offenders, such as evidence destruction or concealment. Extending UCO core objects, CASE captures actions carried out by offenders, victims, or other people involved in a cyber-investigation. This approach supports analysis from a cyber-investigation perspective (who did what in a crime or cyberattack) such as whether a "Webpage Visit" event/action was performed by the victim or offender.

The *Action Lifecycle* can also be used to categorize criminal activities, such as a terrorist planning an attack, a sexual predator's grooming of victims, or a network intruder's method of operation, e.g., kill chain phases. This generalized approach can be used to classify each action in a case, which provides context to understand cyber-investigation activities, including studying specific categories of activities, or develop statistics about what tools are used in different phases of a cyber-investigation. *Action Lifecycle* and *Action Patterns* in CASE can support future work in pattern matching and higher level event composition (Casey et al., 2015).

Proof-of concept API

Members of the community are developing an Application Program Interface (API) to facilitate implementation of CASE in tools. A proof-of-concept API that exports some objects to CASE (https://github.com/casework/case-api-python). This proof-of-concept was written in python and uses the library rdflib as a backend for constructing the graph. This simple API demonstrates how to use an existing RDF library and a small wrapper to serialize data in CASE, producing output in JSON and XML.

The API allows developers to quickly populate and serialize an RDF graph following the CASE structure. To use this API, implementers must understand which structures to create on the basis of the CASE rule set, such as creating Relationships to link related files. It is also necessary for implementers to know which properties are allowed within each UcoObject and PropertyBundle type.

To demonstrate its use, the API was implemented using plaso and includes JSON and XML output (see examples https://github.com/casework/case-implementation-plaso). The plaso to CASE exporter was implemented using the proof-of-concept API to map event objects and path specifications created by plaso's log2timeline tool into one or more CASE UcoObjects. Example 18 shows how the API can be used to generate a simple representation of a blob extracted from a SQLite database.

Example 18. Using the proof-of-concept API to serialize in JSON or XML the CASE representation of a blob extracted from a SQLite database.

```
import rdflib
import case

document = case.Document()

sqlite_database = document.create_uco_object('Trace')
sqlite_database.create_property_bundle(
  'File',
  fileSystemType=case.CASE.EXT4,
  isDirectory=False,
  filePath='/data/data/com.whatsapp/cache/messages.db',
  fileName='messages.db',
  extension='db',
  modifiedTime=rdflib.Literal('2010-01-15T17:59:43.25Z', datatype=rdflib.XSD.dateTime),
  accessedTime=rdflib.Literal('2010-01-15T17:59:43.25Z', datatype=rdflib.XSD.dateTime),
  createdTime=rdflib.Literal('2010-01-15T17:59:43.25Z', datatype=rdflib.XSD.dateTime))

sqlite_blob = document.create_uco_object('Trace')
hash = document.create_hash(
  hashMethod=case.CASE.SHA256,
  hashValue='5994471abb01112afcc18159f6cc74b4f511b99806da59b3caf5a9c173cacfc5')
```

```
sqlite_blob.create_property_bundle(
  ‘ContentData’,
  sizeInBytes=4513,
  hash=hash)

relationship = document.create_uco_object(
  'Relationship',
  source=sqlite_blob,
  target=sqlite_database,
  kindOfRelationship='contained-within',
  isDirectional=True)

relationship.create_property_bundle(
  'SQLiteBlob',
  tableName='AttachmentTable',
  columnName='data',
  rowCondition='pk_id == 5')

document.serialize(format='json-ld', destination='output.json')
```

Future development of the API will validate a CASE bundle by checking it against the current OWL ontology and generating a warning if any discrepancies are detected. In addition, a higher-level API could be created to help enforce the CASE rule set, properties, and expected data types without the need for validation.

Conclusions

The community-developed specification language (CASE), and the underlying ontology (UCO) described in this paper, support standardized representation and exchange of cyber-investigation information between tools, organizations, and jurisdictions. In addition to advancing interoperability and interconnectivity, CASE and UCO provide a common language and structure that can support automated normalization, combination correlation and validation of information, which means less time extracting and combining data, and more time analyzing information.

Codifying and sharing information in a standardized form enables digital investigators to search for similar patterns in their cases. Finding similar patterns between cases can support reuse of previously effective solutions, such as forensic analysis methods for proving that wiping occurred and possibly recovering remnants of overwritten files, thus reducing duplication of effort and increasing consistency of forensic analysis (Casey, 2013). Furthermore, searching for specific patterns across cases can potentially reveal links between related crimes.

Standardized representation of *Traces* can also be useful for application footprinting by recording all traces of a given *Action* (e.g., install, execute, uninstall). For example, the NIST Diskprint project is expanding the National Software Reference Library (NSRL) metadata reference set by recording changes made to a system by an application over its lifecycle (http://www.nsrl.nist.gov/diskprint/). As a way of communicating these changes, NIST outputs the file metadata in multiple serialized formats. Sharing this kind of software Diskprint information is a powerful means of facilitating digital forensic analysis and tool development.

The current version of CASE has been released for broader community use and development (https://github.com/casework) along with the supporting UCO. The ongoing development work includes validating CASE and UCO by implementing it in existing tools and systems used to support cyber-investigations, and adding new properties as needed.

All members of the community are invited to be involved in further enhancements and applications of CASE, including public and private sector institutions, experienced professionals, and tool developers. Being involved at this early stage of design and development is an opportunity to make sure that the standard can support specific use cases. Open questions include the decision of what existing standard to use for representing Locations (e.g., KML) and representing Identities (e.g., CIQ, NIEM). The community is currently refining how CASE represents combinations of entities and traces, such as contact details stored in an address book on a computer or mobile device. The community is also considering a consistent way to represent lists stored in different applications, such as Todo lists, Note lists, Reminders lists, Task lists, and possibly Calendar entries. Related to this, the community needs to decide the preferred way to represent an ordered list that will be compatible with different formats (Drummond et al., 2006). Future work can extend CASE to represent similarity digests, language translation, and behavior patterns.

Acknowledgements

This work has been encouraged and supported by Steven Shirley and William Eber at DoD Cyber Crime Center, Barbara Guttman and Mary Laamanen at the National Institute of Standards and Technology, Erwin van Eijk and Ruud van Baar at the Netherlands Forensic Institute, and Greg Back, Eric Katz and Justin Grover at MITRE, and Simson Garfinkel.

Appendix 1. UCO v0.1.0/CASE v0.1.0 class summary.

Each of the classes identified in the table specifies appropriate properties for its characterization. For the sake of brevity, these properties are not presented here, but are documented on the UCO GitHub repository.

UCO v0.1.0	Description	CASE v0.1.0
uco-core		
UcoObject	A cyber-relevant concept. This is the base object defining the minimal core set of properties to act as a consistent, unifying and interoperable foundation for all explicit and interrelated content objects within the Unified Cyber Ontology (UCO).	N/A (leveraged indirectly through subclasses)
Facet	A grouping of properties characterizing a particular aspect/facet of an object.	PropertyBundle (simple rename)
Relationship	An association or link between two uco objects.	Relationship
Identity	Characterization of the identifying properties of an individual or organization. Numerous facets defined.	Identity
Role	Usual or customary function based on contextual perspective (e.g., Investigator, Attorney, Offender). Multiple general sub-roles also defined, including Victim.	Role
Location	A geophysical place, site or position. Numerous facets defined.	Location
Tool	Characteristics of a tool used in a cyber context potentially including its usage environment and configuration characteristics. Multiple general sub-classes of tool also defined.	Tool
Compilation	A grouping of things.	N/A
Bundle (Compilation)	A contained compilation of UCO content with no presumption of shared context.	Bundle
Grouping (Compilation)	A compilation of referenced UCO content with a shared context.	Grouping
MarkingDefinition	Represents a specific marking that may be applied to UCO data.	MarkingDefinition
GranularMarking	Marking definitions to be applied to particular portions of a particular UCO object.	GranularMarking
Time	Consistent formats for conveying time and date.	Time
Assertion	A statement asserted to be true.	N/A
Annotation	A statement asserted to be true in relation to one or more other objects.	Annotation
ExternalReference	Characteristics of a reference to a resource outside of UCO.	N/A
uco-action	**Description**	**CASE v0.1.0**
Action	Something that may be done or performed. Numerous facets defined.	Action
ActionLifecycle	An action pattern consisting of an ordered set of multiple actions or sub action-lifecycles.	ActionLifecycle
ActionPattern	A logical pattern of characteristic action property values.	N/A
uco-observable		
CyberAction	Something that may be done or performed within the digital domain.	N/A (leverages the general Action object)
CyberItem	A distinct article or unit within the digital domain. Numerous facets defined for various cyber item types.	Trace (simple rename)
CyberRelationship	An association or link between two cyber observable objects.	N/A (leverages the general Relationship object)
CyberObservablePattern	A logical pattern composed of cyberitem and cyberaction properties.	N/A
uco-investigation	**Description**	
Investigation	An exploration of the facts involved in a cyber-relevant set of suspicious activity.	Investigation
ForensicAction	An action taken as part of forensic handling or processing to support a cyber investigation.	ForensicAction
ProvenanceRecord	A provenantial connection between a forensic action and a set of observations (items and/or actions) or interpretations that result from it.	ProvenanceRecord
Authorization (Facet)	Identifies some form of authorization for investigatory action, such as a court order.	Authorization

Appendix 2. Investigative scenario in JSON: The Oresteia by Aeschylus in the age of mobile devices


```
{
"@id": "bundle-3b13e958a-d975-41aa-b1bb-029d2b6707cd",
"@type": "Bundle",
"description": [
  "This illustrative scenario imagines The Oresteia in the age mobile devices for the
purpose of demonstrating use of CASE to represent digital investigations into multiple
related crimes.",
  "To reduce repetitive examples in this illustrative scenario, not all Identity objects are
explicitly represented here. Instead, each object that is referenced in this scenario uses
the associated person's name in the simplified UUID.",
  "Thyestes is the victim in Crime A, and the offender in Crime B",
  "Clock on Clytemnestra's device is one day and one hour slow (offet -25 hours)",
  "There will be an action for each successful parsing of a file and file objects for each
collected file."
],
"content": [
  {
    "@id": "investigation-4586742a-710a-454f-bcb8-b60e230ec1b2",
    "@type": "Investigation",
    "name": "Crime A",
    "focus": "Murder",
    "description": "In Mycenae, Atreus killed two sons of Thyestes, cooked them (except for
their hands and heads), fed them to Thyestes, and then taunted Thyestes with his murdered
sons' hands and heads.",
    "object": ["thyestes-uuid", "victim1-uuid", "role-relationship1-uuid"]
  },
  {
    "@id": "investigation-b05226da-eaef-4bc5-a139-ca12c94dbdfd",
    "@type": "Investigation",
    "name": "Crime B",
    "focus": "Rape",
    "description": "In Mycenae, Thyestes raped his daughter Pelopia to have a son
(Aegisthus)",
    "object": ["thyestes-uuid", "offender1-uuid", "role-relationship2-uuid", "cctv-
recording-uuid", "provenance-record13-uuid"]
  },
  {
    "@id": "investigation-ac9fd560-261e-4cd6-af64-8b83d100b9a8",
    "@type": "Investigation",
    "name": "Crime C",
    "focus": "Murder",
    "description": "In Mycenae, Aegisthus killed Atreus (Agamemnon's father)",
    "object": []
  },
  {
    "@id": "investigation-2545442b-321c-754d-bcb8-c40d321ce2c2",
    "@type": "Investigation",
    "name": "Crime D",
    "focus": "Murder",
    "description": "In Aulis, Agamemnon killed his daughter Iphigenia as a sacrifice to the
gods",
    "object": []
  },
  {
    "@id": "investigation-952d677d-6b62-4e53-9bac-1b113d268ac5",
    "@type": "Investigation",
    "name": "Crime E",
    "focus": "Murder",
    "description": "In the Palace of Argos, Agamemnon and Cassandra were killed by
Clytemnestra (accomplice Aegisthus)",
    "object": ["argos-palace-uuid", "cassandra-uuid", "victim5-uuid", "role-relationship5-
uuid", "cassandra-device-uuid", "associated-device1-uuid", "device-location-relationship1",
"clytemnestra-device-uuid", "forensic-action1-uuid", "annotation1-uuid", "provenance-
record1-uuid", "forensic-action2-uuid", "annotation2-uuid", "provenance-record2-uuid",
"cassandra-mobiledevice-forensicduplicate-uuid", "tool1-uuid", "provenance-record3-uuid",
"cassandra-mobiledevice-mmssms-uuid", "trace-relationship3-uuid", "cassandra-image-
partition6-uuid", "trace-relationship4-uuid", "tool2-uuid", "tool3-uuid", "forensic-action4-
uuid", "forensic-action5-uuid", "sms-message1-uuid", "sms-message2-uuid"]
  },
```

```
  {
    "@id": "investigation-5aa33dc6-7a39-4731-a754-62a9c41e5220",
    "@type": "Investigation",
    "name": "Crime F",
    "focus": "Murder",
    "description": "In the Palace of Argos, Clytemnestra and Aegisthus were killed by
Orestes (accomplice Electra)",
    "object": ["electra-uuid", "argos-palace-uuid", "electra-orestes-email-uuid", "orestes-
facebookmsg-uuid"]
  },
  {
   "@id": "argos-palace-uuid",
   "@type": "Location",
   "propertyBundle": [
     {
       "@type": "SimpleAddress",
       "locality": "Argos",
       "region": "Greece",
       "postalCode": "98052",
       "street": "Palace Blvd"
     },
     {
       "@type": "LatLongCoordinates",
       "latitude": 48.860346,
       "longitude": 2.331199
     }
   ]
 },
 {
  "@id": "cassandra-uuid",
  "@type": "Identity",
  "propertyBundle": [
    {
      "@type": "SimpleName",
      "givenName": "Cassandra",
      "familyName": "Troy"
    },
    {
      "@type": "BirthInformation",
      "birthdate": "1968-09-25T17:59:43.25Z"
    }
  ]
  },
  {
    "@id": "victim5-uuid",
    "@type": "Role",
    "name": "Victim"
  },
  {
    "@id": "role-relationship5-uuid",
    "@type": "Relationship",
    "source": "cassandra-uuid",
    "target": ["victim5-uuid"],
    "kindOfRelationship": "has-role",
    "isDirectional": true
  },
  {
    "@id": "associated-device1-uuid",
    "@type": "Relationship",
    "source": "victim5-uuid",
```

```
  "target": ["cassandra-device-uuid"],
  "kindOfRelationship": "has-device",
  "isDirectional": true
},
{
"@id": "cassandra-device-uuid",
"@type": "Trace",
"propertyBundle": [
  {
    "@type": "Device",
    "manufacturer": "Samsung",
    "model": "SM-G925F Galaxy S6 Edge",
    "serialNumber": "FDG344657"
  },
  {
    "@type": "MobileDevice",
    "keypadUnlockCode": "1234",
    "IMEI": "359305065690067",
    "clockSetting": "2017-06-22T07:36:24.35Z",
    "timezoneSetting": "UTC+01:01 (Europe/Rome)",
    "storageCapacity": "11 GB"
  },
  {
    "@type": "MobileAccount",
    "MSISDN": "1239275339"
  }
]
},
{
"@id": "device-location-relationship1",
"@type": "Relationship",
"source": "cassandra-device-uuid",
"target": ["argos-palace-uuid"],
"kindOfRelationship": "located-at",
"startTime": "2017-06-19T13:59:43.25Z",
"endTime": "2017-06-22T15:59:43.25Z",
"isDirectional": true,
"propertyBundle": [
  {
    "@type": "Confidence",
    "confidence": "Probably True"
  }
]
},
```

```
{
"@id": "thyestes-uuid",
"@type": "Identity",
"propertyBundle": [
  {
    "@type": "SimpleName",
    "givenName": "Thyestes",
    "familyName": "Mycenae"
  },
  {
    "@type": "BirthInformation",
    "birthdate": "1964-10-03T14:39:23.15Z"
  }
]
},
{
  "@id": "victim1-uuid",
  "@type": "Role",
  "name": "Victim"
},
{
  "@id": "role-relationship1-uuid",
  "@type": "Relationship",
  "source": "thyestes-uuid",
  "target": ["victim1-uuid"],
  "kindOfRelationship": "has-role",
  "isDirectional": true
},
{
  "@id": "offender1-uuid",
  "@type": "Role",
  "name": "Offender"
},
{
  "@id": "role-relationship2-uuid",
  "@type": "Relationship",
  "source": "thyestes-uuid",
  "target": ["offender1-uuid"],
  "kindOfRelationship": "has-role",
  "isDirectional": true
},
{
"@id": "electra-uuid",
"@type": "Identity",
"propertyBundle": [
  {
    "@type": "SimpleName",
    "givenName": "Electra",
    "familyName": "Argos"
  },
```

```
  {
    "@type": "BirthInformation",
    "birthdate": "1998-03-02T14:23:42.23Z"
  }
]
},
{
"@id": "associated-emailaccount1-uuid",
"@type": "Relationship",
"source": "electra-uuid",
"target": ["electra-emailaccount-uuid"],
"kindOfRelationship": "has-account",
"isDirectional": true
},
{
"@id": "clytemnestra-device-uuid",
"@type": "Trace",
"propertyBundle": [
  {
    "@type": "Device",
    "manufacturer": "iPhone",
    "model": "MG552",
    "serialNumber": "F18Q4LGRG5MD"
  },
  {
    "@type": "MobileDevice",
    "keypadUnlockCode": "123789",
    "IMEI": "359305065690067",
    "clockSetting": "2017-06-21T06:36:24.35Z",
    "localeLanguage": "en_GR",
    "phoneActivationTime": "2017-05-09T07:36:24.35Z",
    "storageCapacity": "11 GB"
  },
  {
    "@type": "iPhoneDevice",
    "uniqueID": "B3858A69A29375E6C706226B3633A3A11EB2A774",
    "ownerName": "Clytemnestras iPhone"
  },

  {
    "@type": "OperatingSystem",
    "name": "iOS",
    "manufacturer": "Apple",
    "version": "10.3"
  },
  {
    "@type": "MobileAccount",
    "MSISDN": "1237471334"
  },
  {
    "@type": "WiFiAddress",
    "value": "d0:33:11:13:e7:a1"
  },
  {
    "@type": "BluetoothAddress",
    "value": "d0:33:11:13:e7:a2"
  }
]
},
```

```
{
"@id": "forensic-action1-uuid",
"@type": "ForensicAction",
"name": "preserved",
"startTime": "2017-06-21T22:36:24.35Z",
"propertyBundle": [
  {
    "@type": "ActionReferences",
    "instrument": "athens-warrant1-uuid",
    "location": "argos-palace-uuid",
    "performer": "euripides-uuid",
    "object": [
      "cassandra-device-uuid"
    ],
    "result": [
      "provenance_record1-uuid"
    ]
  }
]
},
{
"@id": "annotation1-uuid",
"@type": "Annotation",
"tag": ["forensic"],
"description": "Forensic preservation of Cassandra mobile device.",
"object": [
  "forensic_action1-uuid"
]
},
{
"@id": "forensic-action10-uuid",
"@type": "ForensicAction",
"name": "transferred",
"startTime": "2017-06-22T08:01:23.14Z",
"propertyBundle": [
  {
    "@type": "ActionReferences",
    "instrument": "athens-warrant1-uuid",
    "location": "athenspd-evidenceroom-uuid",
    "performer": "aeschylus-uuid",
    "object": [
      "cassandra-device-uuid"
    ],
    "result": [
      "provenance_record1-uuid"
    ]
```

```
  }
]
},
{
"@id": "provenance-record1-uuid",
"@type": "ProvenanceRecord",
"description": "Mobile device used by murder victim Cassandra",
"exhibitNumber": "ArgosPD-20170622-001A",
"object": ["cassandra-device-uuid"]
},
{
"@id": "provenance_record2-uuid",
"@type": "ProvenanceRecord",
"description": "Android smartphone seized by Argos PD",
"exhibitNumber": "AthensPD-2017220601",
"object": ["cassandra-device-uuid"]
},

{
"@id": "forensic-action2-uuid",
"@type": "ForensicAction",
"name": "extracted",
"startTime": "2017-06-22T08:12:19.32Z",
"endTime": "2017-06-22T08:39:19.24Z",
"propertyBundle": [
  {
    "@type": "ActionReferences",
    "location": "argos-palace-uuid",
    "performer": "aeschylus-uuid",
    "instrument": "tool1-uuid",
    "environment": "forensic-computer1-uuid",
    "object": [
      "provenance-record1-uuid"
    ],
    "result": [
      "cassandra-mobiledevice-forensicduplicate-uuid",
      "provenance_record2-uuid"
    ]
  },
  {
     "@type": "MobileExtractor:ToolArguments",
     "aquisitionType": "Physical Extraction",
     "method": "Boot Loader"
   }
]
},
{
"@id": "annotation2-uuid",
"@type": "Annotation",
"tag": ["forensic"],
"description": "Forensic extraction of data from Cassandra mobile device.",
"object": [
  "forensic_action2-uuid"
]
},
{
"@id": "provenance-record2-uuid",
"@type": "ProvenanceRecord",
"description": "Mobile device used by murder victim Cassandra",
"exhibitNumber": "AthensPD-2017220601-02",
"object": ["cassandra-mobiledevice-forensicduplicate-uuid"]
},
{
    "@type": "Trace",
    "@id": "cassandra-mobiledevice-forensicduplicate-uuid",
    "propertyBundle": [
```

```
        {
          "@type": "File",
          "createdTime": "2017-06-22T08:12:19.32Z",
          "extension": "dd",
          "fileName": "AthensPD-2017220601-01.dd",
          "fileSystemType": "NTFS",
          "filePath": "C:/evidence/AthensPD-2017220601-01.dd",
          "isDirectory": false,
          "sizeInBytes": 90080500
        },
        {
          "@type": "ContentData",
          "hash": [
            {
              "@type": "Hash",
              "hashMethod": "SHA256",
              "hashValue":
"7ea081166336119da78ee4bbdbd06840b94efe28988a2bdb0bcf2387a481e283"
            }
          ],
          "sizeInBytes": 9080500
        }
      ]
  },
  {
   "@id": "tool1-uuid",
   "@type": "Tool",
   "name": "MobileExtractor",
   "toolType": "Extraction",
   "creator": "Zeus",
   "version": "5.3"
   },
   {
   "@id": "provenance-record3-uuid",
   "@type": "ProvenanceRecord",
   "description": "SMS SQLite database on mobile device used by murder victim Cassandra",
   "exhibitNumber": "AthensPD-2017220601-02-03",
   "object": ["cassandra-mobiledevice-mmssms-uuid"]
   },
   {
      "@type": "Trace",
      "@id": "cassandra-mobiledevice-mmssms-uuid",
      "propertyBundle": [
        {
          "@type": "File",
          "createdTime": "2017-06-22T08:12:19.32Z",
          "fileSystemType": "EXT3",
          "extension": "db",
          "fileName":"/data/data/com.android.providers.telephony/mmssms.db",
          "isDirectory": false,
          "sizeInBytes": 122925
        },
        {
          "@type": "ContentData",
          "sizeInBytes": 122925,
          "magicNumber": "U1FMaXRlIGZvcm1hdCAzAA==",
          "hash": [
            {
              "@type": "Hash",
              "hashMethod": "SHA256",
              "hashValue":
"a13225720074371d56a4f4d5117fbb4953c5b1d316b31f21edcb7ed8fdf66c6e"
            }
          ]
        }
      ]
```

```
},
{
"@id": "trace-relationship3-uuid",
"@type": "Relationship",
"source": "sqlite_database",
"target": ["cassandra-image-partition6-uuid"],
"kindOfRelationship": "contained-within",
"isDirectional": true,
"propertyBundle": [
 {
   "@type": "PathRelation",
   "path": "/data/data/com.android.providers.telephony/mmssms.db"
 }
]
},
{
"@id": "cassandra-image-partition6-uuid",
"@type": "Trace",
"propertyBundle": [
 {
      "@type": "DiskPartition",
      "diskPartitionType": "MSDOS"
      "partitionID": "06"
      "partitionOffset": "63"
      "partitionLength": "245235063"
 },
    {
      "@type": "FileSystem",
      "diskPartitionType": "EXT3"
 },
 {
      "@type": "ContentData",
      "sizeInBytes": 245235000,
      "hash": [
        {
          "@type": "Hash",
          "hashMethod": "SHA256",
          "hashValue": "0611ea093d19b1c73a5285ff43741dd77f2a8d983c1c71044eb072e44f5dcb0a"
        }
      ]
    }
  ]
},
{
 "@id": "trace-relationship4-uuid",
 "@type": "Relationship",
 "source": "cassandra-image-partition6-uuid",
 "target": ["cassandra-mobiledevice-forensicduplicate-uuid"],
 "kindOfRelationship": "contained-within",
 "isDirectional": true,
 "propertyBundle": [

       {
      "@type": "DataRange",
      "rangeOffset": 234909233
      "rangeSize": 120000000000
            }
  ]
 },
 {
 "@id": "tool2-uuid",
 "@type": "Tool",
 "name": "Plaso",
 "toolType": "Extraction",
 "creator": "Joachim Metz",
 "version": "1.5.2_201701013",
```

```
   "propertyBundle": [
      {
          "@type": "ToolConfiguration",
          "configurationSetting": [
              {
                  "@type": "ConfigurationSetting",
                  "itemName": "identifier",
                  "itemValue": "624f2636e65e451e8dd7cb044ec44b69"
              },
              {
                  "@type": "ConfigurationSetting",
                  "itemName": "filter_file",
                  "itemValue": ""
              },
              {
                  "@type": "ConfigurationSetting",
                  "itemName": "filter_expression",
                  "itemValue": ""
              },
              {
                  "@type": "ConfigurationSetting",
                  "itemName": "preferred_encoding",
                  "itemValue": "cp1252"
              },
              {
                  "@type": "ConfigurationSetting",
                  "itemName": "parser_filter_expression",
                  "itemValue": "sqlite"
              },
              {
                  "@type": "ConfigurationSetting",
                  "itemName": "preferred_year",
                  "itemValue": ""
              },
              {
                  "@type": "ConfigurationSetting",
                  "itemName": "enabled_parser_names",
                  "itemValue": "sqlite, sqlite/twitter_ios, sqlite/kik_messenger,
sqlite/android_sms, sqlite/android_gmail, sqlite/android_facebook"
              },
              {
                  "@type": "ConfigurationSetting",
                  "itemName": "debug_mode",
                  "itemValue": "False"
              },
              {
                  "@type": "ConfigurationSetting",
                  "itemName": "command_line_arguments",
                  "itemValue": "C:/Python27/Scripts/log2timeline.py C:/evidence/AthensPD-
2017220601-01.dd.plaso C:/evidence/AthensPD-2017220601-01.dd --no-dependencies-check --
parsers sqlite"
              }
          ]
      }
   ]
   },
   {
   "@id": "tool3-uuid",
   "@type": "Tool",
   "name": "sqlite/android_sms",
   "toolType": "Parser",
   "creator": "Joachim Metz",
   "propertyBundle": [
      {
          "@type": "ToolConfiguration",
          "configurationSetting": [
```

```
            {
                "@type": "ConfigurationSetting",
                "itemName": "query",
                "itemValue": "SELECT _id AS id, address, date, read, type, body FROM sms"
            },
            {
                "@type": "ConfigurationSetting",
                "itemName": "schema_match",
                "itemValue": "True"
            }
        ]
    }
 ]
 },
 {
 "@id": "forensic-action4-uuid",
 "@type": "ForensicAction",
 "name": "extracted",
 "startTime": "2017-06-22T09:57:23.64Z",
 "endTime": "2017-06-22T10:31:19.24Z",
 "propertyBundle": [
  {
    "@type": "ActionReferences",
    "location": "argos-palace-uuid",
    "performer": "aeschylus-uuid",
    "instrument": "tool2-uuid",
    "environment": "forensic-computer1-uuid",
    "object": [
      "cassandra-mobiledevice-forensicduplicate-uuid",
      "provenance_record2-uuid"
    ],
    "result": [
      "forensic-action5-uuid",
      "provenance-record3-uuid",
      "cassandra-mobiledevice-mmssms-uuid"
    ]
  }
]
 },
 {
 "@id": "forensic-action5-uuid",
 "@type": "ForensicAction",
 "name": "parsed",
 "startTime": "2017-06-22T09:57:23.64Z",
 "endTime": "2017-06-22T10:31:19.24Z",
 "propertyBundle": [
  {
    "@type": "ActionReferences",
    "location": "argos-palace-uuid",
    "performer": "forensic-action4-uuid",
    "instrument": "tool3-uuid",
    "environment": "forensic-computer1-uuid",
    "object": [
      "cassandra-mobiledevice-mmssms-uuid"
    ],
    "result": [
      "sms-message1-uuid",
      "sms-message2-uuid",
      "argive-elder1-phoneaccnt-uuid",
      "argive-elder2-phoneaccnt-uuid",
      "argive-elder3-phoneaccnt-uuid"
    ]
  }
 ]
 },
 {
```

```
            {
                "@type": "ConfigurationSetting",
                "itemName": "query",
                "itemValue": "SELECT _id AS id, address, date, read, type, body FROM sms"
            },
            {
                "@type": "ConfigurationSetting",
                "itemName": "schema_match",
                "itemValue": "True"
            }
        ]
    }
]
},
{
"@id": "forensic-action4-uuid",
"@type": "ForensicAction",
"name": "extracted",
"startTime": "2017-06-22T09:57:23.64Z",
"endTime": "2017-06-22T10:31:19.24Z",
"propertyBundle": [
 {
   "@type": "ActionReferences",
   "location": "argos-palace-uuid",
   "performer": "aeschylus-uuid",
   "instrument": "tool2-uuid",
   "environment": "forensic-computer1-uuid",
   "object": [
     "cassandra-mobiledevice-forensicduplicate-uuid",
     "provenance_record2-uuid"
   ],
   "result": [
     "forensic-action5-uuid",
     "provenance-record3-uuid",
     "cassandra-mobiledevice-mmssms-uuid"
   ]
 }
]
},
{
"@id": "forensic-action5-uuid",
"@type": "ForensicAction",
"name": "parsed",
"startTime": "2017-06-22T09:57:23.64Z",
"endTime": "2017-06-22T10:31:19.24Z",
"propertyBundle": [
 {
   "@type": "ActionReferences",
   "location": "argos-palace-uuid",
   "performer": "forensic-action4-uuid",
   "instrument": "tool3-uuid",
   "environment": "forensic-computer1-uuid",
   "object": [
     "cassandra-mobiledevice-mmssms-uuid"
   ],
   "result": [
     "sms-message1-uuid",
     "sms-message2-uuid",
     "argive-elder1-phoneaccnt-uuid",
     "argive-elder2-phoneaccnt-uuid",
     "argive-elder3-phoneaccnt-uuid"
   ]
 }
]
},
{
```

```
   "@id": "sms-message1-uuid",
   "@type": "Trace",
   "propertyBundle": [
    {
      "@type": "Message",
      "application": "sms-application1",
      "messageText": "A wedded wife, she slays her lord, Helped by another hand!",
      "from": "cassandra-mobileacct-uuid",
      "to": [
        "argive-elder1-phoneaccnt-uuid",
        "argive-elder2-phoneaccnt-uuid",
        "argive-elder3-phoneaccnt-uuid"
      ],
      "sentTime": "2017-06-20T09:34:42.12Z"
    }
   ]
   },
   {
  "@id": "sms-message2-uuid",
  "@type": "Trace",
  "propertyBundle": [
    {
      "@type": "Message",
      "application": "sms-application1",
      "messageText": "Low lie the shattered towers whereas they fell, and I--ah burning
heart!--shall soon lie low as well.",
      "from": "cassandra-mobileacct-uuid",
      "to": [
        "argive-elder1-phoneaccnt-uuid",
        "argive-elder2-phoneaccnt-uuid",
        "argive-elder3-phoneaccnt-uuid"
      ],
      "sentTime": "2017-06-20T09:37:35.13Z"
    }
   ]
   },
   {
  "@id": "argive-elder1-phoneaccnt-uuid",
  "@type": "Trace",
  "propertyBundle": [
    {
      "@type": "PhoneAccount",
      "phoneNumber": "1237771337",
    }
   ]
   },
   {
   "@id": "electra-orestes-email-uuid",
   "@type": "Trace",
   "propertyBundle": [
    {
      "@type": "EmailMessage",
      "to": ["orestes-emailaccount-uuid"],
      "from": "electra-emailaccount-uuid",
      "subject": "Revenge our father",
      "body": "To me, too, grant this boon-dark death to deal unto Aegisthus, and to 'scape
my doom.",
      "receivedTime": "2017-06-21T13:44:23.40Z",
      "sentTime": "2017-06-21T13:44:22.19Z",
      "messageID": "CAKBqNfyKo+ZXtkz6DUjWpvHy6O82jTbkNA@mail.gmail.com"
    }
   ]
   },
   {
  "@id": "annotation1",
  "@type": "Annotation",
```

```
  "tag": ["selfie", "picture"],
  "description": "Digital photograph of corpses taken at crime scene by killer",
  "object": [
      "orestes-selfie-photograph-uuid"
   ]
   },
   {
   "@id": "orestes-facebookmsg-uuid",
   "@type": "Trace",
   "propertyBundle": [
    {
      "@type": "FacebookMessage",
      "from": ["orestes-facebookaccount-uuid"],
      "to": ["friends"],
      "body": "There lies our country's twofold tyranny, My father's slayers, spoilers of my
home.",
      "sentTime": "2017-06-21T14:44:54.19Z"
    },
    {
      "@id": "attach_relationship1",
      "@type": "Relationship",
      "source": "location1",
      "target": "orestes-facebookmsg-uuid",
      "kindOfRealtionship": "attachment-of",
      "isDirectional": true,
      "propertyBundle": [
        {
          "@type": "Attachment",
          "url": "http://www.facebook.com/corpses.jpg"
        }
      ]
    }
   ]
   }
]
}
```


Appendix A. Supplementary data

Supplementary data related to this article can be found at http://dx.doi.org/10.1016/j.diin.2017.08.002.

References

Alink, W., Bhoedjang, R., Boncz, P., de Vries, A., 2006. Xiraf—xml-based indexing and querying for digital forensics. In: Suppl. 1, Proceedings of the 6th Annual DFRWS Conference, Digital Investigation, vol. 3. Elsevier.

Barnum, S., 2014. "Whitepaper: Standardizing Cyber Threat Intelligence Information with the Structured Threat Information EXpression (STIX)" February 20, 2014, Version 1.1, Revision 1. http://stixproject.github.io/getting-started/whitepaper.

van Beek, H.M.A., van Eijk, E.J., van Baar, R.B., Ugen, M., Bodde, J.N.C., Siemelink, A.J., 2015. Digital forensics as a service: game on. Digital Investigation Special Issue Big Data Intelligent Data Analysis 15, 20—38. Elsevier.

Bhoedjang, R.A.F., van Ballegooij, A.R., van Beek, H.M.A., van Schie, J.C., Dillema, F.W., van Baar, R.B., et al., 2012. Engineering an online computer forensic service. Digit. Investig. 9 (2). Elsevier.

Brady, O., Overill, R., Keppens, J., 2015. DESO: addressing volume and variety in large-scale criminal cases. Digit. Investig. 15, 72—82. Elsevier.

Casey, E., 2013. Reinforcing the Scientific Method in Digital Investigations using a Case-Based Reasoning (CBR) System. PhD Dissertation. University College Dublin.

Casey, E., Back, G., Barnum, S., 2015. Leveraging CybOX to standardize representation and exchange of digital forensic information. In: Suppl. 1. Proceedings of the 2nd Annual DFRWS EU Conference, Digital Investigation, vol. 12. Elsevier.

Casey, E., Barnum, S., Griffith, R., Snyder, J., van Beek, H., Nelson, A., 2017a. The evolution of expressing and exchanging cyber-investigation information in a standardized form. In: Biasiotti, Bonnici, Mifsud, Cannataci, Ruchi (Eds.), Handling and Exchanging Electronic Evidence across Europe. EU EVIDENCE Project, Springer, Berlin in press.

Casey, E., Biasiotti, M.A., Turchi, F., 2017b. Using Standardization and Ontology to Enhance Data Protection and Intelligent Analysis of Electronic Evidence. Discovery of Electronically Stored Information Workshop (DESI VII), ICAIL 2017. https://www.umiacs.umd.edu/~oard/desi7/.

Casey, E., Jaquet-Chiffelle, D.-O., 2017. Do Identities Matter?, Policing: a Journal of Policy and Practice. Special Issue. Oxford University Press. Available at https://dx.doi.org/10.1093/police/pax034.

Chabot, Y., Bertaux, A., Nicolle, C., Kechadi, T., December 2015. An ontology-based approach for the reconstruction and analysis of digital incidents timelines. Digit. Investig. 15, 83—100. http://dx.doi.org/10.1016/j.diin.2015.07.005. Elsevier: London.

Cohen, M., Schatz, B., Garfinkel, S., September 2009. "Extending the advanced forensic format to accommodate multiple data sources, logical evidence, arbitrary information and forensic workflow. Proceedings of DFRWS2009 Digit. Investig. 6 (Suppl.), S57—S68. Elsevier.

Cosic, J., Baca, M., 2015. Leveraging DEMF to ensure and represent 5ws&1h in digital forensic domain. Int. J. Comput. Sci. 13 (2).

Drummond, N., Rector, A., Stevens, R., Moulton, G., Horridge, M., Wang, H.H., Seidenberg, J., 2006. Putting OWL in order: patterns for sequences in OWL. In: 2nd OWL Experiences and Directions Workshop, Athens, GA. http://www.cs.man.ac.uk/~drummond/publications/OWLListsPaper/owl-lists-iswc.pdf.

Eaglin, R., Craiger, J.P., 2005. Data sharing and the digital evidence markup language. In: Presented at 1st Annual GJXDM Users Conference, Atlanta.

Flaglien, A.O., Mallasvik, A., Mustorp, M., Arnes, A., November 2011. Storage and exchange formats for digital evidence. Digit. Investig. 8 (2), 122—128.

Garfinkel, Simson, 2009. Automating Disk Forensic Processing with SleuthKit, XML and Python. Systematic Approaches to Digital Forensics Engineering (IEEE/SADFE 2009). California, Oakland.

Garfinkel, S., 2012. Digital forensics XML and the DFXML toolset. Digit. Investig. 8, 161—174. Elsevier.

Garfinkel, S., 2013. Digital media triage with bulk data analysis and bulk_extractor. Comput. Secur. 32, 56—72. Elsevier.

Hargreaves, C., Patterson, J., 2012. An automated timeline reconstruction approach for digital forensic investigations. Digit. Investig. 9 (Suppl.) (DFRWS2012 Proceedings).

Lanthaler, M., Gütl, C., 2012. On using JSON-LD to create evolvable RESTful services. In: Proceedings of the 3rd International Workshop on RESTful Design (WS-rest 2012) at WWW2012. ACM Press, Lyon, France, pp. 25—32. http://json-ld.org/.

Lee, S., Park, T., Shin, S., Un, S., Hong, D., 2008. A new forensic image format for high capacity disk storage. Information Security and Assurance, 2008. ISA 2008. In: International Conference on Information Security and Assurance. IEEE Computer Society, 24—26 April.

Levine, B.N., Liberatore, M., September 2009. DEX: digital evidence provenance supporting reproducibility and comparison. Digit. Investig. 6 (Suppl.), S48—S56. Elsevier.

McGuinness, D.L., van Harmelen, F., February 2004. "OWL Web Ontology Language Overview" W3C Recommendation. https://www.w3.org/TR/owl-features/.

Neches, R., Fikes, R., Finin, T., Gruber, T., Patil, R., Senator, T., Swartout, W.R., 1991.

Enabling technology for knowledge sharing. AI Mag. Winter 36–56.
Office of the Director of National Intelligence, 2016. XML Data Encoding Specification for Intelligence Document and Media Exploitation. https://www.dni.gov/index.php/about/organization/chief-information-officer/information-security-marking-access?id=1204.
Schatz, B., 1995. Digital Evidence: Representation and Assurance. PhD Dissertation. Queensland University of Technology. http://eprints.qut.edu.au/16507/1/Bradley_Schatz_Thesis.pdf.
Studer, Benjamins, Fensel, 1998. Knowledge engineering: principles and methods. Data Knowl. Eng. 25, 161–197.
Turnbull, B., Randhawab, S., June 2015. Automated event and social network extraction from digital evidence sources with ontological mapping. Digit. Investig. 13, 94–106.
Turner, P., September 2005a. "Unification of digital evidence from disparate sources (digital evidence bags)" proceedings of DFRWS2005. Digit. Investig. 2 (3), 223–228. Elsevier.
Turner, P., 2005b. Digital provenance – interpretation, verification and corroboration. Digit. Investig. 2 (1), 45–49. Elsevier.
Turner, P., 2006. Selective and intelligent imaging using digital evidence bags proceedings of DFRWS2006. Digit. Investig. 3 (Suppl.), 59–64. Elsevier.