Use of KAOS in Operational Digital Forensic Investigations

Benjamin Aziz

School of Computing
University of Portsmouth
Portsmouth, U.K.
benjamin.aziz@port.ac.uk

Clive Blackwell

Department of Computing and Communications Technologies
Oxford Brookes University
Oxford, U.K.
cblackwell@brookes.ac.uk

Abstract. This paper focuses on the operations involved in the digital forensic process using the requirements engineering framework KAOS. The idea is to enforce the claim that a requirements engineering approach to digital forensics produces reusable patterns for future incidents. Our patterns here will be operation-focused, rather than requirement-focused, which is simpler because the operations can potentially be exhaustively enumerated and evaluated. Thus, for example, given the complexity of the Ceglia versus Zuckerberg Facebook case involving alleged document forgery, we can show that one of the benefits coming out of the modelling exercise was the set of operations needed. This will give an estimate for the future of what kind of capabilities and resources are needed for other complex document-forgery cases involving computers. It may also help to plan investigations and prioritise the use of resources more widely within the case workload of investigators.

1 Introduction

In recent years, there has been a significant and increasing problem with management and control of digital forensic investigations, especially involving capability and resource planning. It is very difficult to allocate sufficient resources when planning new investigations leading to significant delays and inefficiencies that have become routine with many units having backlogs greater than 1 year. This is due to the complexity of such investigations, increasing volumes of stored and transmitted data, the diverse types and number of devices and network sources being examined, and the number of investigations with significant involvement using computers and the Internet.

adfa, p. 1, 2011.

We need to move away from the insistence on investigating all potential evidence, and therefore we need a convincing explanation of our search strategy to justify its sufficiency. For example, Paul Ceglia provided 1087 CDs to Zuckerberg's experts in his claim for 50% of Facebook [1]. This is the computer equivalent of supplying a truckload of boxes full of documents in legal discovery, where the relevant evidence is stapled between two unrelated pieces of evidence at the bottom of one of the boxes.

This *volume* and *complexity* challenge has been identified in numerous works in the literature (e.g. [2,3,4]). Therefore, providing an approach to discover the set of high-level operations that will be needed for a specific investigation, based on prior experiences with similar investigations, is becoming more and more urgent at both the planning and implementation stages.

In this paper, we propose the use of the operational model of a goal-driven requirements engineering methodology called KAOS [5] in capturing the operations and resources needed during the various stages of a digital forensic investigation. The KAOS operational model provides a methodological platform for describing how concrete requirements of any system or process can be operationalised, and therefore provide an abstract specification for the operations or tools underlying them. We show that such operationalisation can lead to the formulation of an operational pattern of an investigation for various types of cybercrimes.

The rest of the paper is organised as follows. In Section 2, we discuss related work. In Section 3, we describe how digital forensics processes used by forensic analysts and computer security personnel can be described as KAOS operations. We demonstrate informally also how this leads to the notion of an operational pattern underlying a specific type of crime. In Section 4, we show how this approach works in a real case study involving Facebook [6,7]. Finally, in Section 5, we conclude the paper and discuss directions for future work.

2 Related Work

There is a growing set of work on triage as a way of managing resources more effectively. The concept of triage is best known in medical emergencies, where triage is defined as: 'A process for sorting injured people into groups based on their need for or likely benefit from immediate medical treatment' [8]. Parsonage provides a good overview of the practical issues facing forensic investigators [9] when he explains how he successfully reduced the backlogs in his unit by several months. The Cyber Forensic Field Triage Process Model (CFFTPM) [10] proposes an approach for rapidly investigating digital evidence, which systematises their existing investigative methods into a defined forensic process.

These models attempt to cast the triage process within the wider investigative process involving all stages, not just the initial survey. These works, along with a few others

[11,12], help make the forensic process more efficient and effective by using techniques that attempt to measure and repeat the success of previous investigations.

These papers outline many practical issues that we address more formally with operationalisation. The idea of using the full-blown capabilities of requirements engineering methodologies, and in particular KAOS, as a new approach in capturing and formalising the requirements underlying digital forensic investigations is relatively new [6,7,13]. In [13], KAOS was shown to be a suitable methodology for modelling the various processes involved in digital forensic investigations. In [6], the work was further extended and a framework was introduced to express the requirements of the context as well as the investigation. In addition, obstacles and anti-goals were used to capture the notion of impediments to investigations, such as anti-forensic activities that aim to hide illicit activities. Finally, in [7], the KAOS approach was applied in defining the requirements of a recent complex case of alleged document forgery involving Facebook.

3 Forensic Systems as KAOS Operations

KAOS is a generic methodology based on capturing, structuring and precisely formulating system goals [5]. KAOS is a goal-driven methodology consisting of several related models that aim to describe how and by whom the main set of goals underlying a system will be satisfied. In addition to its *goal model* that aims to formulate system goals, KAOS also incorporates models for the operationalisation of requirements (the *operation model*), the assignment of responsibility (the *agent model*) and the definition of risk (the *obstacle and anti-goal models*).

In [13], an approach was defined describing how digital forensic investigations can benefit from KAOS models. We used the goal model to formulate system goals and gradually decompose them into relevant activities for the investigative process. Think of goal models as a more systematic version of attack trees [14], where we progressively decompose the investigative goals into individual steps that can be implemented. The first stage is to decompose the identified goals into sub-goals and then secondly decompose the sub-goals into concrete requirements. Once the requirements have been discovered, it is possible to determine the operations that concretely implement an investigation and allocate these activities to responsible agents.

The implementation could include any of the operations at the planning, identification, search, seizure, analysis, documentation and court presentation stages of an investigation. Specifically, it would be relevant to the stages of an Equivocal Forensic Analysis (EFA) [15 Ch 7] representing the tools and techniques used to gather temporal, functional and relational data pertaining to the evidence.

A recent case involving alleged document forgery in Ceglia versus Zuckerberg and Facebook [7] was analysed, where the goals were systematically decomposed into a set of actionable requirements that could be executed by forensic experts. For example, with document forgery some requirements for the discovery of document tampering and fraudulent communication are shown in the blue trapeziums of Figure 1. These can be operationalised using a number of tools/software/operations such as a *document viewer*, a *metadata viewer*, a *system log viewer* and an *email viewer*, as illustrated by the operations in circles of Figure 1.

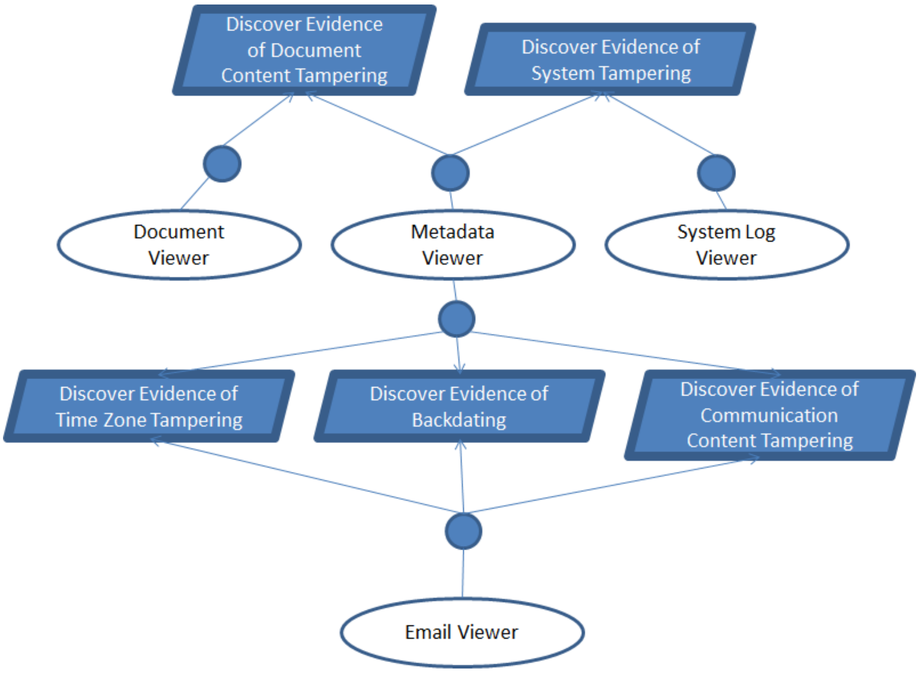
Discover Evidence of Document Content Tampering
Discover Evidence of System Tampering
Document Viewer
Metadata Viewer
System Log Viewer
Discover Evidence of Time Zone Tampering
Discover Evidence of Backdating
Discover Evidence of Communication Content Tampering
Email Viewer

Fig. 1. The Operationalization of Requirements for the Discovery of Document Authenticity and Tampering (top half) and Fraudulent Communications (bottom half) [6]

In addition, these activities needed to be allocated to responsible agents that are authorised and competent to carry out the operations. Any problem with operationalisation and assignment may indicate issues with the investigation that require further planning, or even abandonment if the problems cannot be adequately addressed.

Once a number of such operationalisation exercises have been carried out for a specific type of a crime, we can formulate an *operational pattern*, which means the set of all possible operations (systems, software, tools, services etc.) that can be involved within a digital forensic investigation into an instance of that crime. This could lead to more effective and efficient implementation of investigative techniques and limiting unpleasant surprises. There are also benefits for the investigative workload in general where new cases can be provisionally operationalised leading to decisions about case selection and resource provision early enough to address any issues.

3.1 From KAOS Requirements to Operations

We now provide a more precise definition of an operational pattern. We define the mapping from KAOS requirements to operations as a relation *f*:

$$f: R \rightarrow O \tag{1}$$

where *R* is the set of requirements models in KAOS and *O* is the set of KAOS operations, both of which are defined by users in some specific problem domain. For the case of criminal investigations, we redefine this function as follows:

$$f_c: R_c \rightarrow O_c \tag{2}$$

where *c* is an identifier indicating a specific type of crime; for example, this could be *identity fraud*, *cyber theft* or *document forgery*. It is important to note the *c* is used in a general way here; however, in real world criminal investigations, *c* would be chosen more in a more fine-grained manner to reflect specific type of crimes, such as *contract forgery* or *invoice forgery* as refinements of *document forgery*.

4 Facebook Case Study

We discuss the crime type of alleged *document forgery* in the Ceglia versus Zuckerberg and Facebook case. Here we focus on the operations (forensic tools, techniques, operations and capabilities) used, and we produce two definitions, one for $f_{Facebook}$ and one for the operations involved, $O_{Facebook}$, defined later.

Mr Ceglia alleged that he had agreed a contract with Mark Zuckerberg to give him 50% of Facebook, whereas Zuckerberg claimed the only contract between the pair did not mention Facebook and that Ceglia's contract was a manipulated version of the agreed one. See the expert report for Zuckerberg [1] and a comprehensive goal modelling analysis [7] for further details.

Here, we focus on the how we meet the requirements of `detect text changes`, `detect file metadata anomalies`, `detect system anomalies` and `search application data` with suitable operations to demonstrate the allegation of forgery. We discuss later how to operationalise the requirements in the physical forensic science domain to achieve the same goal.

$f_{Facebook}$ = {
(detect text changes → [perform formatting checks, writing style analysis, content analysis]),
(detect file metadata anomalies → [check file timestamps, search log records, check document usage]),
(detect system anomalies → [check clock tampering, investigate filesystem changes, examine registry keys]),

```
(find application evidence → [investigate email activity,
discover social networking use, analyse Internet usage, search
for hacking tools])
}
```

In the above example, the requirements are operationalised by different operations with different properties regarding their applicability and potential benefits. Typically, in requirements engineering, we choose one or possibly more ways to operationalise the requirements. Here, we do not know a priori which operations are possible and most likely to be effective, and so we need to consider the context of the case to determine the best method of operationalisation, so we suggest a number of operations to meet each requirement.

The requirement of `detect text changes` is operationalised in the Ceglia case by `perform formatting checks` to search for formatting discrepancies, `writing style analysis` to find inconsistencies, and `content analysis` to search for anomalies. There was no "golden copy" of the contract lodged in a secure location or with a trusted party to compare with the alleged contract, which would often be the case with official business contracts. The formatting checks found several discrepancies that Zuckerberg claimed were manipulated to squeeze in more text, but this is inconclusive as the contract was not a highly structured document with specified formatting. Next, the writing style can be internally inconsistent indicating multiple authors, but here there were only minor changes between the two contracts. Finally, the content can be ambiguous, conflicting or demonstrably bogus, but again this was not proven.

The requirement of checking the contract for evidence of textual changes could be met by a combination of the various operations, none of which is sufficient alone. Invalidating a requirement is also a crucial investigative goal to avoid wasting resources on fruitless investigation allowing attention to be focused elsewhere, or making spurious claims that may fatally weaken a case. Refutation can occur by demonstrating a fact that is inconsistent with a requirement, or by showing that no feasible operations can satisfy a requirement.

In the Ceglia case, the results of checking the contract text and computer systems for anomalies using the various operations were suggestive of forgery but inconclusive. However, other requirements were operationalised more convincingly. The electronic contract had undergone multiple irregular alterations meeting the requirement of `detect file metadata anomalies` by the operation `check file timestamps` that discovered highly anomalous timestamps in draft contracts found on Ceglia hard disk. In addition, the physical forensic checks of the paper contract persuasively showed deceptive manipulation (see later).

Some operations do not satisfy primary requirements by themselves but suggest new requirements and possible operations. For example, an application level search operation `investigate email activity` found undisclosed email accounts used by Ceglia that eventually led to the contract he sent to his lawyer in 2004 agreeing with Zuckerberg’s version.

We obtain a second instance of type *document forgery* from a synthetic case of a fake invoice by a financial controller that altered a legitimate invoice by increasing the invoice amount and changing the payee to their name. This could involve the following, slightly different, definition of the function $\mathtt{f}_{\mathtt{DocumentForgery}}$:

```
$f_{InvoiceForgery}$ = {
(detect text changes → [hash comparison, good copy comparison,
perform formatting checks, find goods substitution, find
financial irregularities]),
(detect file metadata anomalies → [check file timestamps, search
log records, check document usage]),
(detect system changes → [check clock tampering, investigate
filesystem changes, examine registry keys]),
(find application evidence → [check invoice number, check
invoice database, find unauthorised applications])
}
```

In the faked invoice example, the requirement of `detect text changes` is operationalised using comparison with a known authentic document by `good copy comparison` or `hash comparison`. In this case, a securely stored trusted copy of the invoice can be checked manually against the fake, or the invoice may be hashed with the hash securely stored for later comparison, neither of which was possible with the alleged Facebook contract.

.

Next, invoice formatting is highly structured with each section possibly having a particular design, configuration and size, and stylised with the organisational heading and logo amongst other things. Any significant departure would be indicative of forgery, but here a legitimate invoice was altered by an insider and so appeared genuine.

The operations can be decomposed by progressive refinement into more specific activities that can be carried out by suitable agents. Content analysis here has been decomposed into the more granular `find goods substitution` and `find financial irregularities` that are common operations to detect fraudulent invoices. For example, for financial irregularities, invoice items may be inconsistent with the price or the total may exceed the allowed authorisation limit. Enforcing database constraints and double entry bookkeeping are typical operations to help defeat or discover fraud.

The file metadata and system checks are the same as for the contract, as low-level checks are fairly stereotypical. Whereas the system checks are similar to the Facebook case, the application checks are different to reflect the different deceptive activities performed.

We can also decompose the investigation into various types of examination in different forensic domains. In the Ceglia case, we consider the physical forensic analysis separate from the logical examination, as different operations are performed by different parties, although they are attempting to satisfy similar requirements.

```
f_FacebookPhysical = {
(detect text changes → [check content alterations, good  copy
comparison, check handwriting consistency, check signature,
check figures)],
(detect paper anomalies → [check paper age, check paper type,
check paper thickness, check paper whiteness)],
(detect mark anomalies → [check ink properties, compare with
invoice template, check watermark, check letterhead, check for
fingerprints)],
(detect printer properties → check toner anomalies, check
printer tracking data)]
}
```

As can be observed, there are both similar and different requirements for physical investigation compared with digital analysis. Physical checks can be operationalised with many different physical or chemical checks, so the given operations are only indicative of the checks that can be performed.

The physical tests demonstrated beyond reasonable doubt that Ceglia's contract was created using a fake page 2 attached to the legitimate page 1 from the original contract. This was demonstrated in multiple ways by physical forensic experts, especially LaPorte [16] who showed that different toners and inks were used on the two pages.

4.1 Operational Patterns

We define here an *operational pattern* to be the set of all KAOS operations corresponding to a specific type of crime, as follows:

$$O_c = \bigcup_{r \in dom(f_c)} f_c(r) \qquad (3)$$

This definition states that the set of operations, O_c, corresponding to a particular crime type, *c*, is the union of all the operations that can be obtained from every instance of a requirements model for that type of crime. Intuitively, these are the operations resulting from all the models corresponding to a particular crime type. We call the set O_c the *operational pattern corresponding to the crime type c*.

Hence, for the example of document forgery above, we can identify the general operational pattern as:

$f_{DocumentForgery}$ = {
(detect text changes → [perform formatting checks, writing style analysis, content analysis, good copy comparison]),
(detect file metadata anomalies → [check file timestamps, search log records, check document usage]),
(detect system anomalies → [check clock tampering, investigate filesystem changes, examine registry keys]),
(find application evidence → [investigate email activity, discover social networking use, analyse Internet usage, search for unauthorised applications, perform specific application checks])
}

This operational pattern includes all the operations involved in a forensic investigation of document forgery from all prior cases (based here only on two examples). This allows us to plan for the operations that may be successful in future similar cases. We do not incorporate the physical forensic checks discussed above, as these are outside the digital forensic boundary and are irrelevant to the digital forensics investigator.

There is an issue where the operations are at different levels of abstraction. We can reduce an operational pattern to a more abstract form by combining several operations into one, more abstract operation, which could represent a higher-level system-based view of the operations. For example, in the Facebook case, it may be possible to group `detect text changes physically`, and `detect text changes logically` into one abstract called `detect text changes`. Here, we have defined the abstract form of the same operation split across forensic domains.

We can also group operations that perform the same function in a different way. `perform specific application checks` is an abstraction of `check invoice number` and `check invoice database` needed in the invoice forgery case. `find goods substitution`, and `find financial irregularities` are subsumed by content analysis. `hash comparison`, and `good copy comparison` are subsumed by `check document integrity`. We see that it is possible to move up and down this abstraction hierarchy depending on the specific case requirements.

Finally, we propose a table showing all these operations and their relationships. We can determine the different types of operations, their pre and postconditions, and their costs and benefits. In adopting such an approach, we can provide an idea of what operations are needed in similar cases in the future along with potential issues. The ‘costs’ include wasted investigative effort and the benefits include suggesting alternative operations that may satisfy the same requirement. Preconditions include the resources and skills needed to carry out the operation and postconditions include the results needed to satisfy the requirement.

5 Conclusion

This work presented an approach demonstrating how the mapping from requirements underlying a digital forensic investigation to possible operations can lead to the notion of an operational pattern in specific types of crime. Operationalisation leads to the creation of evidence (or its absence) that may lead to requirements satisfaction (or failure) eventually closing the loop to goal fulfilment (or not).

Operationalisation helps us investigate the adequacy of different operations and potential technical and legal issues that may arise. One of the main advantages is that it allows for the possibility of determining *in advance*, because of previous experience with similar crimes, the set of high value operations that might be involved in an investigation. This could be used to aid triage in the selection and prioritisation of tasks, and the determination whether to discontinue the case if some of the key tasks cannot be executed adequately or at all.

We should consider the continuous evolution of technology, which inevitably will introduce new criminal tools and techniques that will undermine existing operational patterns. This implies constant revision of such patterns, but having a high level of abstraction may help to understand the investigation conceptually and aid the discovery of solutions tot these impediments.

The examples demonstrated the practical use of operationalisation in a limited sense. We have some different operations for the contract and invoice forgery cases, but we need to study many more cases of document forgery to discover additional operations and establish well-defined criteria for their classification and determination of the abstraction and refinement hierarchy.

It is clearly useful to determine the set of operations that may be available before investigation, but we may also need other features of KAOS such as behavioural and agent models for a more complete and constructive analysis. Further work will investigate the context where some operations are to be preferred over others depending upon the goals, resources, initial evidence and surrounding circumstances.

Finally, formalisation is possible in simple scenarios using the LTL semantics underlying KAOS. However, we may have to focus on specific aspects of forgery to obtain the detailed deterministic operations needed for logical proof.

References

1. Stroz Friedberg. Report of Digital Forensic Analysis in: Paul D. Ceglia v. Mark Elliot Zuckerberg, Individually, and Facebook, Inc. Civil Action No: 1:10-cv-00569-RJA, 26 March 2012. Available from http://www.wired.com/images_blogs/threatlevel/2012/03/celiginvestigation.pdf.

2. T. Lindsey. Challenges in Digital Forensics. 2006 (Last viewed 20 May 2013). Available from http://www.dfrws.org/2006/proceedings/Lindsey-pres.pdf.

3. G. Mohay. Technical Challenges and Directions for Digital Forensics. In *1st International Workshop on Systematic Approaches to Digital Forensic Engineering*, 2005.

4. Bradley Schatz. Digital Evidence: Representation and Assurance. Ph.D. thesis, Queensland University of Technology, 2007.

5. Axel van Lamsweerde. *Requirements Engineering: From System Goals to UML Models to Software Specifications*. Wiley, 2009.

6. Benjamin Aziz, Clive Blackwell and Shareeful Islam. A Framework for Digital Forensics and Investigations: The Goal-Driven Approach. In *International Journal of Digital Crime and Forensics*, IGI-Global, USA, January 2014 (to appear).

7. Clive Blackwell, Shareeful Islam and Benjamin Aziz. Implementation of Digital Forensics Investigations Using a Goal-Driven Approach for a Questioned Contract. In *Proceedings of the 9th Annual IFIP WG 11.9 International Conference on Digital Forensics*, Orlando, USA, Springer, January 2013 (to appear in post-proceedings).

8. Dictionary.com, "triage," in *The American Heritage® Stedman's Medical Dictionary*, Houghton Mifflin Company, 2000, at http://dictionary.reference.com/browse/triage.

9. H Parsonage, *Computer Forensics Case Assessment and Triage – some ideas for discussion*, 2009, at http://computerforensics.parsonage.co.uk/triage/ComputerForensicsCaseAssessmentAndTriageDiscussionPaper.pdf.

10. MK. Rogers, J Goldman, R Mislan, T Wedge and S Debrota, Computer Forensics Field Triage Process Model, *Journal of Digital Forensics, Security and Law*, Vol 1 no 2, 2006, at www.jdfsl.org/subscriptions/JDFSL-V1N2-CFFTPM.pdf.

11. Overill, R.E., Silomon, J.A.M., Digital Meta-Forensics: Quantifying the Investigation, Proc. 4th International Conference on Cybercrime Forensics Education & Training, 2010.

12. R. Overill, M. Kwan, K. Chow, P. Lai, and F. Law, "A Cost-Effective Forensic Investigation Model", IFIP WG 11.9, 5th International Conference on Digital Forensics, 2009.

13. Benjamin Aziz. Towards Goal-Driven Digital Forensics Investigations. In *Proceedings of the 2nd International Conference on Cybercrime, Security and Digital Forensics* (Cyfor-12), London, UK, 2012.

14. Schneier, Bruce (December 1999). "Attack Trees". Dr Dobb's Journal, v.24, n.12. at http://www.schneier.com/paper-attacktrees-ddj-ft.html.

15. Brent E. Turvey, Crime Scene Analysis, In Criminal Profiling: An Introduction to Behavioral Evidence Analysis, Brent E. Turvey (ed), Academic Press; 4 edition (2011)

16. G. LaPorte. Paul D. Ceglia v. Mark Elliot Zuckerberg and Facebook, Inc. United States District Court Western District of New York, Civil Action No. 1:10-cv-00569-RJA, Document 326, Case Riley Welch LaPorte & Associates Forensic Laboratories (RWL), filed 03/26/12.