Models of Models: Digital Forensics and Domain-Specific Languages
(Extended Abstract)

Daniel A. Ray* Phillip G. Bradford*

ABSTRACT

There are numerous and diverse digital forensics models for driving digital investigative processes. To encompass these diverse models we argue that there is need for two integrated Domain-Specific Languages (DSLs) [5]: a static one and a dynamic one. We attempt to motivate some research directions involving DSLs, digital forensics, and the creation of two descriptive languages for digital forensic models. We discuss both static and dynamic DSLs for static and dynamic forensic modeling, respectively.

1 Motivation

Good comprehensive models of digital investigation must provide a consistent and standardized framework that supports all stages of an investigation, both technical and non-technical, regardless of the specific type of crime. Many such models have been suggested, see for instance [9, 7, 4, 1].

The purpose of digital investigation models is to inform, shape, and standardize digital investigations. We are interested in how these models can be used in an integrated fashion to assist investigators. For example, a DSL may for digital forensics may have a library of the classical digital forensics models. The different models may be used and combined from this library through the DSL.

We differentiate two basic types of computer crimes: computer assisted crimes and computer focused crimes. In computer assisted crimes, the suspect uses computers for their basic functionality. In computer focused crimes, the suspect uses computers as the primary focus of their criminal activity. We argue that digital forensics for computer assisted crimes can often be dealt with by using static models. Where the more dynamic nature of computer focused crimes requires a Turing complete DSL. For example, an on-going investigation may have to dynamically model criminal behavior.

*Department of Computer Science, The University of Alabama, Box 870290, Tuscaloosa, AL 35487-0290. DanielRay@cs.ua.edu, pgb@cs.ua.edu
Such dynamic models may use static models intermittently. A possibility would be to integrate such a system into Proactive Forensics systems [3, 6, 2].

DSLs are, “...languages tailored to a specific application domain.” [5] DSLs provide a formal means for representing domain specific information that is not easily supported by General Purpose Languages such as C++, Java, Fortran, etcetera. DSLs are often developed because they can support domain-specific notation, constructs and abstractions [5].

DSLs can be Turing complete and have well-defined execution semantics (i.e. Excel macros, Prolog, XML and HTML), as an input language to an “application generator”, as a not primarily executable language but nevertheless suitable for input to a parser (such as BNF), or as a strictly declarative language [5]. We believe that the domain of digital forensics is one that would benefit from the creation of one or two DSLs at any of these levels, but particularly at the static as well as the Turing complete level.

2 A Selection of Previous Work on Models of Digital Investigation

A selection of models for digital forensics investigation follows.

2.1 The DFRWS Framework Meta-Model

The DFRWS Framework was developed between 2001 and 2003 at the Digital Forensics Research Workshop (DFRWS)[9]. The framework introduces “Digital Investigation Action Classes”. The classes that are defined by the framework serve to categorize the activities of an investigation into groups.

The framework does not dictate what particular actions must be pursued. Instead, it provides a list of candidate techniques, some of which are required. The specifics of the framework must be largely redefined for each particular investigation.

The framework is represented by a table including columns for each activity class. Each row contains candidate techniques. These candidate techniques may be performed in pursuit of the goals of the associated action class. In addition, the specific goals and purposes of each action class are described in [8].

2.2 The Reith, Carr, and Gunsch Model

The model presented by Reith, Carr, and Gunsch [7] is very similar to the DFRWS Framework [9]. The model includes Preservation, Collection, Examination, and Presentation classes similarly defined as those of the DFRWS.

The model also adds supports for tool preparation and the dynamic formulation of investigative approaches. This model also loosely supports iterations of individual activity classes.
2.3 The Ciardhuain Model

The model suggested by Ciardhuain [4] is based on previous models but exhibits an augmented waterfall architecture. The model’s activity classes are doubly linked so that the pursuit of work in one activity class can cause an iteration of some or all of the work in the previous activity classes.

The inclusion of structures known as information flows allows a deeper understanding of the source of evidentiary and other data. These flows must be defined on an organizational basis, but can apply to different investigations within the same organization.

2.4 The Beebe and Clark Model

The Beebe and Clark model [1] provides structure for activities through phases consisting of multiple sub-phases rather than through activity groupings. Sub-phases are objective based rather than strictly activity based. The objective based sub-phases each fall into a particular phase and consist of a hierarchy of particular activities that are subordinate to the particular objective.

In addition, the Beebe and Clark model [1] includes “digital investigation principles” which over arch all the phases and sub-phases and affect how they are performed.

The ultimate goals for each sub-phase are represented as objectives rather than specific tasks. This is an important and unique difference from the task-based models we have discussed thus far. Objectives are goals that can be expected of activities of a similar nature regardless of the specific case. Tasks are directly related to a specific case, type of crime, platform, etcetera.

2.5 Overview of Models

Each model’s distinct characteristics give it inherent advantages and disadvantages. The DFRWS model is rigid and linear but is particularly suitable where necessary investigative activities are well-understood. The RCG model is more dynamic and supports a limited form of iteration between activity classes. The Ciardhuain model supports iterations of activities between activity classes and provides support for tracking information sources. The objective-based structure of the Beebe and Clark model is most flexible and provides the best support when very little is known about what investigative techniques will be required.

Each model, despite their differences, has quite a lot in common with other models. It seems striking how similar these models are. The similarity suggests a descriptive language for use in describing the domain for investigative forensics.
3 Creating a Digital Investigation Domain-Specific Language

Domain characteristics will change from investigation to investigation. These characteristics influence which model, and which specific activities should be used. Questions must be answered such as; is reuse of investigative techniques important, will investigative results be highly scrutinized, will investigators be required to explicitly handle many flows of information both inside and outside the investigation, are the necessary activities clearly defined, etcetera.

According to Mernick, *et al.* [5], generating a DSL involves executing an analysis, design, and implementation phase. The problem domain is identified and domain knowledge is gained during the analysis phase. In the case of the digital forensics domain, much of this work has already been done. Indeed, we have already covered several of the most prominent domain models, and no doubt more exist. The single “domain model” that is needed for the purposes of designing the DSL will simply need to incorporate the approaches used by all the previous models.

Next, the design phase must be executed. Two considerations must be made during DSL design. Will the new DSL be related to any previous language or not and will the DSL design specification be formal [5]?

Lastly, the implementation phase must be executed in order to determine how the DSL will be implemented. Will the language be interpreted, compiled, pre-processed, etcetera [5].

4 Concluding Remarks and Further Directions

One possibility would be to base the creation of a descriptive level DSL on the Unified Modeling Language (UML). Action classes suggest classes or class hierarchies with individual activities as methods. The iterative nature of some of the models suggests the language must support iteration. The language must also support possible flows of work or computation. Also, the need to store data sources might be fulfilled by special language supported data types or meta-data. Obviously, these and other considerations will need to be addressed.

Lastly, the implementation phase must be executed in order to determine how the DSL will be implemented. Will the language be interpreted, compiled, pre-processed, etcetera [5].

In general, all of these considerations can be managed and a Domain-Specific Language can be generated that allows for easy description of forensics investigation domain-specific variables that will support future efforts for the development of automated software that assists digital investigators.
References


