UNIVERSITY OF AMSTERDAM

Gathering Evidence: Model-Driven Software Engineering in Automated Digital Forensics

J. van den Bos

Summary

Digital forensics concerns the acquisition, recovery and analysis of information stored on digital devices for the purpose of answering legal questions. Exponential increases in available storage capacity and network bandwidth, as well as growing device and service adoption by the public, have made manual inspection of all potentially relevant information infeasible in nearly all cases. A solution to this problem is *automated digital forensics*, which is the use of software to perform tasks in digital forensics automatically, reducing the time required to get results.

Many software engineering techniques exist that allow the construction of high performance and scalable solutions in the domain of digital forensics. Unfortunately, another major requirement complicates the application of standard techniques: handling the high variability in the shape of how investigated information is stored. The number of different devices, networks, platforms, and applications is huge and constantly changing. This leads to a constant stream of required changes to digital forensics software in order to recover as much information as possible.

Factoring out the commonality so that the constantly changing aspects of a solution can evolve separately from the stable aspects is a supposed strength of *model-driven software engineering* (MDSE). This separation of concerns is achieved through the use of a domain-specific language (DSL), which is a custom notation used to specify the changing parts of a solution. Changes expressed in this DSL are then automatically applied through the use of transformation tools such as code generators and interpreters, which handle fixed requirements such as high performance.

This thesis presents analyses and experiments that were performed in order to discover the benefits and costs of applying model-driven software engineering, specifically in the development and maintenance of solutions in the domain of automated digital forensics. The contributions are the following:

- A description of the results of domain analyses to establish initial requirements, including the domain of automated digital forensics in general, and data recovery and aspects of binary file formats in particular. Specific areas of interest are identified for the development of binary file format validators.

- Design and implementation of a DSL to describe binary file formats, applied in a forensic data recovery tool called a *file carver*. Experimental evaluation shows that the proposed model-driven approach has no negative effects on the runtime performance and data recovery qualities of the final solution, but does allow clear separation of concerns and requires fewer lines of code to maintain.

- Application of model transformation to let the user of the file carver trade accuracy of data recovery for runtime performance, without requiring changes by a software engineer. Experimental evaluation on a custom benchmark shows that runtime performance gains of up to a factor of three can be achieved, at the expense of up to 8% in precision and 5% in recall.

- Design of an experimental approach to observe the maintenance characteristics of a DSL, by generating realistic maintenance scenarios from a corpus of representative inputs. Application of the approach to the proposed DSL shows that it can accomodate all expected changes, and also identifies three language features to consider for further improvement.

- Design and implementation of an integrated development environment (IDE) that provides the DSL user with a fully synchronized view of all relevant information during development and maintenance. This includes syntax-colored views of the static file format description, the dynamic data recovery program state, as well as the input data.

Finally, the research presented in this thesis forms an extensive case study in the application of MDSE in the domain of automated digital forensics, using the Rascal metaprogramming language. It provides concrete evidence for the successful application of MDSE in the domain of automated digital forensics, and contributes to knowledge about the application of MDSE in general. The concise and versatile implementations provide a strong case for the usefulness and applicability of Rascal in DSL engineering.