Artificial Intelligence Applied to Computer Forensics

Bruno W. P. Hoelz
Brazilian Federal Police
National Institute of
Criminalistics
Brasília, DF, Brazil
werneck.bwph@dpf.gov.br

Célia Ghedini Ralha
Brasilia University
Computer Science
Department
Brasília, DF, Brazil
ghedini@cic.unb.br

Rajiv Geeverghese
Brasilia University
Computer Science
Department
Brasília, DF, Brazil
vect0rius@yahoo.com.br

ABSTRACT

To be able to examine large amounts of data in a timely manner in search of important evidence during crime investigations is essential to the success of computer forensic examinations. The limitations in time and resources, both computational and human, have a negative impact in the results obtained. Thus, better use of the resources available are necessary, beyond the capabilities of the currently used forensic tools. Herein, we describe the use of Artificial Intelligence in computer forensics through the development of a multiagent system and case-based reasoning. This system is composed of specialized intelligent agents that act based on the experts knowledge of the technical domain. Their goal is to analyze and correlate the data contained in the evidences of an investigation and based on its expertise, present the most interesting evidence to the human examiner, thus reducing the amount of data to be personally analyzed. The correlation feature helps to find links between evidences that can be easily overlooked by a human expert, specially due to the amount of data involved. This system has been tested using real data and the results were very positive when compared to those obtained by the human expert alone performing the same analysis.

Categories and Subject Descriptors

H.4 [**Information Systems Applications**]: Miscellaneous; I.2.11 [**Computing Methodologies**]: Distributed Artificial Intelligence—*Multiagent systems*

Keywords

computer forensics, artificial intelligence, multiagent systems, digital investigation, JADE

1. INTRODUCTION

The forensic examination of computer systems consists of several steps to preserve, collect and analyze evidences found in digital storage media, in such a way that they can be presented and used as evidence of unlawful actions involving those resources. As an example, we can cite web server invasions, whose possible outcomes could be the defacement of visible contents or the unauthorized access to private data, for instance. At a crime investigation, digital evidence can be of importance in a number of serious crimes such as child exploitation, forgery of documents, tax frauds and even terrorism.

The constant growth in the capacity of digital storage media and the widespread presence in everybody's daily life represents also a growth in the demand for those examinations and likewise in the volume of data to examine. In addition to this problem, the current set of forensic tools are not robust enough when it comes to analyzing a great number of evidences and correlate the findings. As a consequence, computer forensic experts' work excessively time consuming. The computational resources required to do such examinations are also a problem, since most of the forensic tools have no distributed processing capabilities.

Our goal is to present a tool to help experts during specialized forensic examinations in order to obtain significantly better results when compared to those obtained by the currently used tools considering three aspects: (i) reduction of routine and repetitive analysis while also reducing the amount of evidence that must be personally reviewed by the expert, (ii) correlation of evidences, (iii) distribution of processes. With this, human and computational resources can be applied more efficiently.

The rest of the article is divided as follows: Section 2 presents some characteristics related to forensic examinations of computational systems; Section 2.1 presents related work; the use of artificial intelligence to perform forensic examinations is presented in Section 3 and the experiments and its results are presented in Section 4. Finally, Section 5 presents some conclusions and points to future works.

2. FORENSIC EXAMINATIONS

Frequently, at real computer forensic examinations, experts can't determine beforehand which evidences will turn out to be the most relevant to the investigation of a crime. Consider the example of a *cybercafé* or any other scenario where several computers appear to share the same IP address. The traces will often lead to the *cybercafé* and not to any particular machine. The same difficulty can be faced while collecting evidences of fraud in companies with several machines and users. In these examples a pre-analysis of the suspect machines would limit the number of machines to collect, reducing the time required to complete the forensic

SAC'09 March 8-12, 2009, Honolulu, Hawaii, U.S.A.
Copyright 2009 ACM 978-1-60558-166-8/09/03 ...$5.00.

examinations. The problem is the lack of intelligent tools to help forensic experts with the pre-analysis phase, which results in the collection of a large number of machines to be examined, some of which will not contribute to the overall result of the investigation and will just increase the time needed to complete the examinations.

To fill the need for intelligent tools and to better employ computational resources during forensic examinations is the main focus of this research. We consider impossible that forensic experts do proper content analysis of machines individually and also cross-analysis, considering the large amount of data collected and sent to a forensic laboratory for examination, given limited time and resources.

Considering the simplest case, experts have to analyze a stand-alone machine and are limited to the contents of that machine. But unfortunately, the simplest case is not the most common one, since computers are frequently connected to networks where they can exchange data. In addition, the ever increasing storage capacity of removable media complicates matters even further. These computers and removable media are examined individually because of the lack of tools to help with a concrete cross-analysis. As a result of the presented problems, a large number of potentially related evidences are lost during the forensic examinations. This is not only a problem to forensic computing, but also to network incident response.

2.1 Related Work

A number of different approaches have been proposed to deal with the three aspects presented before: (i) reduction in the amount of evidences to be examined, (ii) correlation of evidences and (iii) distribution of forensic examinations computational work.

In the various works of [18, 19, 20] we find the idea of the digital evidence bags (DEB). A DEB is a universal container for digital evidence from any source that allows the provenance to be recorded and continuity to be maintained throughout the life of the investigation. The author suggests the use of intelligent techniques to treat a selective information capture scenario, using a selective image approach with the DEB. There are important aspects of his work which considers the importance of files during the investigation task, like in our proposal: (i) how to capture and combine the experts knowledge of both domains technical and legal, and (ii) how to be certain that all the relevant information and evidences of other related crimes are captured in the DEB. The answer to this questions is very important in order to collect and examine only relevant evidences to the investigation, thus reducing the amount of time required to complete the examinations.

The work of [16] proposes the addition of a case-relevance indicator to the evidences. By their definition, case-relevance would be the *"property of any piece of information, which is used to measure its ability to answer the investigative 'who, what, where, when, why and how' questions in a criminal investigation."* The levels of relevance defined by their work go from "Absolutely Irrelevant" to "Probably Case-Relevant". The intelligent agents used in our tool do something similar, although they can sometimes diverge in the relevance given to a file. Such conflict is solved by another agent and ultimately reviewed by the human expert. By considering the case-relevance of an evidence, the expert can focus, for instance, on a subset of files found on a hard drive, temporarily ignoring those that are considered irrelevant.

[7] presents two approaches for analyzing large data sets of forensic data called Forensic Feature Extraction (FFE) and Cross-Drive Analysis (CDA). We consider CDA to be the most interesting, since it uses statistical techniques for correlating information within a single disk image and across multiple disk images. A recent work by [4] describes a tool called Forensics Automated Correlation Engine (FACE), whose objectives are similar to ours. They also present some scenarios where an increased level of correlation of disparate evidences was achieved.

The work of [15] makes the case for Distributed Digital Forensics and present some scenarios where the forensic work can't be performed anymore on a single workstation. They also propose a distributed framework and some performance results that show the advantages of the distributed approach, which would also enable more sophisticated analysis techniques. The tool presented in [4], unlike ours, does not employ the distribution of processes. In our case, our proposal benefit from the distributed nature of a multiagent system and we already observed the performance gains, which helped us obtain better computational resource usage and reduce the time required to perform the examination.

Finally, we also make use of the framework proposed by [1]. They propose a multi-tier, hierarchical framework to guide digital investigations. To us, the most important aspect of this framework is its objectives-based phases and sub-phases that are applicable to various layers of abstraction, and to which additional layers of detail can be added as needed. These objectives-based phases serve to us as a guide for the definition of the specialized intelligent agents employed in our tool. We also observed the case-based nature of the objectives, which means that different types of investigation require different sets of examinations. This gives us the opportunity to apply case-based reasoning (CBR) to the planning of our agents' actions.

3. AN INTELLIGENT TOOLKIT

In the literature we find many definitions for a multiagent system (MAS), but they all refer to a computational system composed by more than one agent [8, 22, 23, 6]. An intelligent software agent (ISA) uses Artificial Intelligence (AI) in the pursuit of goals [11, 17]. Thus, an ISA interacts with the environment, perceiving and acting autonomously over it to achieve defined goals. In an analogous way, a MAS is a system where many agents interact with the environment in a cooperative or competitive way to achieve individual or group objectives.

In this work, we present the latest results obtained by the use of the *MultiAgent Digital Investigation toolKit* (MADIK), a multiagent system to assist the computer forensics expert on its examinations. The system is composed of a set of ISAs that perform different analysis on the digital evidence related to a case on a distributed manner.

In MADIK, each ISA contains a set of rules and a knowledge base, both based on the experience of the expert on a certain kind of investigation. Since the examination of digital evidence in crime investigations share similarities, MADIK uses CBR to determine which agents are better employed in which kind of investigation. This also allows the agents to reason about the evidences in a way that is more adequate to the specific case in question. As an example, we can cite the use of hash sets in a child exploitation case. The ISA will use first the hash sets related to child exploitation, thus giving the examiner a quicker feedback on the presence of such files in a piece of evidence. At the moment, the MADIK has six specialized intelligent agents implemented:

- `HashSetAgent` calculates the MD5 hash from a file and compares it with its knowledge base, which contains sets of files known to be ignorable (e.g. system files) or important (e.g. contraband files like child exploitation imagery). We might cite that some of these hash sets contain more than 10 million hash values, from different softwares, as cited in [12]. Also we might note that the bigger the hash set the longer the comparison takes.

- `FilePathAgent` keeps on it's knowledge base a collection of folders which are commonly used by several application which may be of interest to the investigation like P2P (peer-to-peer), VoIP and instant messaging applications.

- `FileSignatureAgent` examines the file headers (the first 8 bytes of the file), to determine if they match the file extension. If someone changes the file extension in order to hide the true purpose of the file, this will be detected by this agent. It also keeps a list of commonly used prefixes and file names, such as the ones used by digital cameras.

- `TimelineAgent` examines dates of creation, access and modification to determine events like system and software installation, backups, web browser usage and other activities, some which can be relevant to the investigation.

- `WindowsRegistryAgent` examines files related to the windows registry and extracts valuable information such as system installation date, time zone configuration, removable media information and others.

- `KeywordAgent` searches for keywords and uses regular expressions to extract information from files such as credit card numbers, URLs or e-mail addresses.

The MADIK is not a replacement for commonly used forensic tools like AccessData Forensic ToolKit or Guidance EnCase. The proposed agents are a reduced set that allows for many rules to be conceived and many examinations to be carried over, as a proof of concept. New agents can be conceived by encapsulating the functionality of existing tools and scripts such as Foundstone's Galleta or Volatile Systems' Volatility. This works main contributions, in our view, are the definition of an automated architecture where specialized agents can analyze and correlate findings beyond the simple acquisition and extraction of data provided by the current tools, with the added benefit of seeking the better use of computational resources through distribution. The case-based approach also provides a way to improve the agents results over time, by learning from previous cases, as suggested in [3].

As another example of the case-based approach, we can also cite hash set comparisons in order to ignore unimportant files. If an unexperienced examiner tries to compare every hash set he has available against every single file, the process will take too long and the results will not be much better than those obtained by an experienced examiner who chooses the most likely hash sets so he can have quick but yet effective results.

To coordinate and organize the work of these specialists, we propose a four layer hierarchy, similar to human organizations, as used for example in the work of [14]. Figure 1 presents this hierarchy.

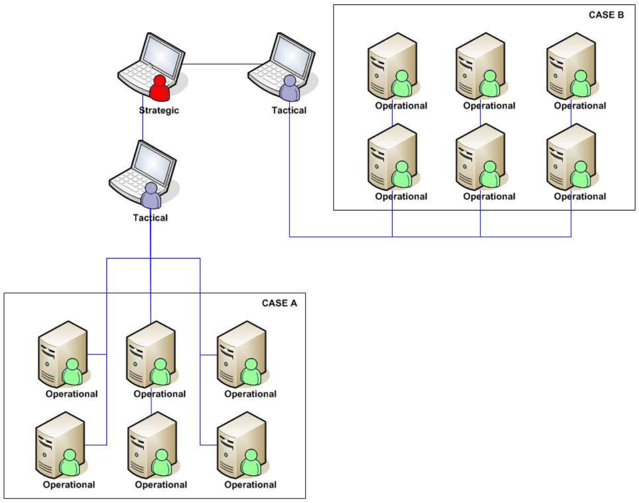


Figure 1: Agents hierarchy

The communication amongst levels is rigid. One agent communicates only with its immediate superior or with other agents in the same level. This simplifies the chain of control and coordination of the agents work. Agents can collaborate by observing and modifying one another's work through the use of a common base named blackboard [13, 10, 21, 5]. This gives the opportunity for agents to cooperate and reach good results.

The MADIK was implemented using the *Java Agent Development Framework - JADE*, which is fully developed with the Java language [9, 2]. JADE was used since it simplifies the implementation of multiagent systems, over a distributed platform. The system also relies on a PostgreSQL database and a rule engine (JBoss Drools) to implement the agents reasoning.

To better understand how the system works we will explain MADIK's operation processes. The strategic manager receives different cases to perform the forensic analysis. According to the organization's priorities, the strategic manager defines the order of execution and amount of resources (number of computers) for each case. A tactical manager is them assigned to one specific case which can contain several evidences, like a number of hard drives. The tactical manager defines the priority of its evidences and distributes them to the available operational managers, which are limited by the resources available to that case. The operational manager will employ the necessary specialized agents to perform the different tasks it deems important to examine a piece of evidence.

During the specialists' execution, they will insert their conclusions and remarks in the blackboard. Each entry contains the agent's recommendation, an user-friendly description and the time taken to examine the file, for benchmarking purposes. There are three distinct levels of recommendation: (i) `ignore` - the strongest recommendation to ignore a file, indicating its unimportant according to the agent, (ii) `alert` - strongly recommends the selection of a file, and (iii) `inform` - this recommendation is an intermediate value, which contains information to help the human reviewer to decide whether to select that file or not. There can be an additional sign (+ or -) representing an ignore or alert bias, respectively.

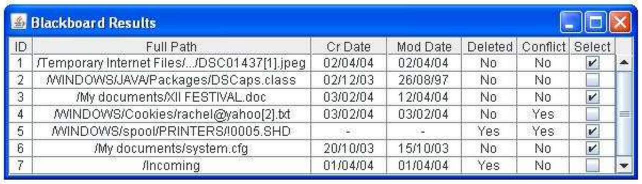
Table 1 presents an example of the blackboard results with the decision of some specialized agents - file names and attributes were omitted.

Table 1: Example of blackboard results

File	Decision	Conflict
1	FileSignatureAgent - Inform(+) (possible digital camera image) TimeLineAgent - Inform(+) (recently created)	NO
2	HashSetAgent - Ignore (zero-size file) FilePathAgent - Inform(+) (Windows spool file)	YES
3	FileSignatureAgent - Inform(+) (possible Yahoo! Login information) FilePathAgent - Inform(-) (cookie file)	YES
4	HashSetAgent - Alert (suspected child porn) FileSignatureAgent - Alert (bad extension: expecting a video file) FilePathAgent - Inform(+) (file in user folder)	NO

The different agents can diverge in their decisions, what causes a conflict in the blackboard that must be solved by the operational manager. The HashSetAgent for instance may find a file irrelevant to an investigation based on its hash sets, while the KeywordAgent may find the occurrence of a given keyword in that same file. The first will give an `ignore` recommendation while the last will give an `inform` with a + bias. The manager's decision depends on the kind of investigation and in the performance of these agents in the past investigations of the same kind. The better the agent performed in the past, the more trusted it is by the manager. This mechanism relies on a review interface used by the human examiner and is presented in Figure 2. Whenever a suggestion from the system is corrected by the user, by changing the `Select` field shown in the figure, the manager reduces the confidence on an agent's reasoning on that specific case. This reduces the agent's weight in a future conflict resolution in a similar scenario. If the user agrees with the agent, the manager confidence in the agent increases for that kind of investigation.

When the operational level finishes its work, the tactical level is able to correlate findings from different evidences by looking at the evidences' blackboards. Such knowledge discovery is not possible with the most commonly used tools. One simple example is the discovery of e-mail messages that originated in one computer and were received by another. Another one is indicating that a piece of removable media, such as an USB thumb drive, was inserted on a system or that one file found in one of the examined hard drives was remotely accessed from another seized hard drive.


Blackboard Results

ID	Full Path	Cr Date	Mod Date	Deleted	Conflict	Select
1	/Temporary Internet Files/.../DSC01437[1].jpeg	02/04/04	02/04/04	No	No	☑
2	/WINDOWS/JAVA/Packages/DSCaps.class	02/12/03	26/08/97	No	No	□
3	/My documents/XII FESTIVAL.doc	03/02/04	12/04/04	No	No	☑
4	/WINDOWS/Cookies/rachel@yahoo[2].txt	03/02/04	03/02/04	No	Yes	□
5	/WINDOWS/spool/PRINTERS/!0005.SHD	-	-	Yes	Yes	☑
6	/My documents/system.cfg	20/10/03	15/10/03	No	No	☑
7	/Incoming	01/04/04	01/04/04	Yes	No	□

Figure 2: Review interface

4. EXPERIMENTS AND RESULTS

In order to evaluate the MADIK we conducted two experiments using data from real investigations. In the first one we observed the reduction in the execution times of the HashSetAgent examinations with the distribution of its work. We wanted to determine the impact of distribution and the overhead caused by the agents communication and coordination protocols. We used from one to four computers with one to eight HashSetAgents in each machine. They divided the task of comparing the hashes of 500,000 files against a 26,000 files hash set. The results are presented in Figure 3.

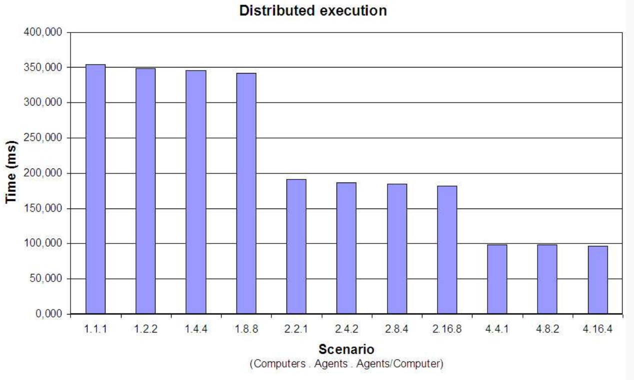


Figure 3: Execution times in distributed environments.

The results were positive with approximately 10% of the time lost due to the communication overhead between the agents. This indicates that the system can sustain more computers and agents before the communication effort required to coordinate the work increase beyond the distribution gains.

The second experiment presented in this paper is based on a retirement pension fraud investigation, involving workers in the public administration. The objective was to observe the blackboard and evaluate the levels of conflict between the agents, the percentage of reduction (files deemed irrelevant) and coverage (files examined by at least one agent). Four specialized agents were used: the HashSetAgent, FilePathAgent, FileSignatureAgent and TimelineAgent. Fourteen pieces of evidence, seized from the same location, were examined (10 hard drives and 4 removable media). The total number of files is 353,466 files for a total of 75.502 GB, including recovered files but excluding free space fragments.

The HashSetAgent found 246,941 duplicate files (44,362 distinct hashes), 3,025 zero-sized files and another 15,553 files that were ignored using the hash sets. Each of these results come from a different rule used by the agent. Considering some overlapping in these rules, the final reduction suggested by the HashSetAgent was 69.8%. In terms of coverage, it analyzed 81.3% of the case files where a hash calculation was applicable. Due to the nature of the case, only a reduced number of hash sets was used, which included the most commonly used software.

The FilePathAgent suggested the removal of 5,811 small GIF files (less than the average image size) located on the Temporary Internet File folder. It also suggested that 2,314 cookies be ignored. Other suggestions included to ignore 9,871 Java-related files and another 6,186 program documentation files. The total reduction suggested was of 6.8% of the files. The FilePathAgent identified the presence of OpenOffice and suggested the inclusion of 100 documents with the ODT format. It also suggested the inclusion of the 2,095 MS Office files that were located in user-related folders (e.g. My Documents) and alert to known systems that could have been used by the suspects in the fraud. The coverage of this agent in this case reached 9.8%.

The FileSignatureAgent suggested ignoring 200 digital photos, and inclusion of 275 cookies and 279 temporary internet files which could be related to webmail and banking sites. Some 200 files had an extension that did not match with the file header, thus receiving an `Alert` status. The agent suggested to ignore 67,621 files based on their types (executable files like DLL, EXE, VXD and Java class files which represented approximately 19% of the files). The coverage was of about 26%.

The TimelineAgent detected several points of software installation or system updates. It suggested ignoring the executable files and documents created on these events. Approximately 63,000 files were ignored. It alerted about the modification of 376 files on weekends, since such event was not expected for this kind of case and evidence and 39 documents modified in the previous two weeks, which was of interest to the investigation. The agent reached a reduction factor of 17.8% and 48% of coverage.

Regarding the reduction factor (the number of files which the agents suggested we could ignore), the final percentage obtained was of 73%. The total coverage of the analysis was of about 85%, meaning that 85% of the files were examined by at least one agent.

The conflict set was smaller than 1% of the files. One conflict, for instance, occurred between the FilePathAgent and the FileSignatureAgent. The first one suggested to ignore all the cookie files, while the second suggested the examination of some of those cookies. The conflict set should increase with the addition of new agents and rules.

In terms of correlation, we search the blackboard for executable files and user created documents in two or more evidences. The results indicated the presence of seven custom made systems installed on every machine. The presence of 52 documents in the removable media that were also in one of the examined hard drives and another 428 files that were on more than one evidence. As a future work, we wish to improve correlation by adding specialized agents dedicated to this task.

The examination of these evidences by the system took about two hours using one machine with four cores and four gigabytes of RAM. The MD5 hashes were already calculated and were not added to the total time. The time required by two human examiners to perform the same examination was of about 24 hours. This time does not take into account the time required by the forensic tools to recover files, generate thumbnails or calculate hashes. We considered only the time needed to examine the file, do keyword searches the examiner deems necessary and select the most interesting files.

Although not evaluated numerically, we noticed that the results obtained by the MAS were similar, but not as complete as the expert's analysis. We observed that a lot of time is spent by the human examiner ignoring files and inspecting fragments of temporary internet files. The use of the MADIK can already be helpful in this case, although the reduction factor still need to be improved.

We also wish to increase the coverage of the agents' analysis by introducing new rules in their reasoning process and also new agents, so we can examine as much files in the evidence as possible with at least one agent. We observed interesting results in terms of coverage. In some forensic tools, files such as messaging applications' logs and OpenOffice documents are typically hidden in more generic categories that can be easily overlooked. In this experiment, these kinds of files were examined and presented to the user by at least one agent.

5. CONCLUSIONS

This paper described an application of AI in computer forensics and the latest results obtained with the use of the MADIK, a MAS to assist the experts during computer forensic examinations. The system has been tested using real data with four specialized agents. Its results show that the application of a MAS in the forensic examination of computers is an interesting approach to improve the usage of computational resources available and reduce the time required to conduct the examination through the reduction in the volume of evidence to be examined. The intelligent and autonomous agents that compose the system seek to incorporate the experts knowledge to perform the analysis of a great volume of data, giving the expert a subset that is more likely an important evidence to the investigation.

The MADIK is not proposed as a replacement to the currently used, commercial or not, such as AccessData FTK (Forensic ToolKit) and Guidance EnCase, but as a complement. One of our future works will focus on the integration of `MADIK` with one of these tools. Such integration will greatly increase the possibilities of analysis and automation of routine tasks. The system's architecture makes it easy to included new agents. In the future, we intend to expand our set of agents by also wrapping the functionality of several existing command-line tools around an intelligent specialized agent.

The distributed platform presented promising results, which indicate that the system can sustain more machines before the communication and coordination costs interrupt the distribution gains. This distributed nature allows us to conceive new analysis that were very expensive in terms of computer resources and could not be carried by a single workstation in a reasonable time. These can even be executed in the idle time of the computer available in the forensic laboratory. We look forward to implement agents that perform time-consuming analysis such as face detection and recognition or keyword search in a distributed manner, as it is currently with password breaking software.

The correlation features, although limited, already show the possibilities of discovering important evidences that are easily overlooked when data are examined separately without a cross-analysis phase. The correlation features must also be extended to cover a wider range of situations. This will be improved with the addition of new agents. As new experiments are conducted, we expect to evaluate the CBR features and learning mechanisms.

The combination of the reduction in the volume of evidences to be examined by the expert and the reduction in the execution times obtained with the distributed processing of the evidences already show the potential of the tool and the productivity gains it can offer to computer forensic experts and to investigators that face an ever increasing volume of digital evidence.

6. REFERENCES

[1] Nicole Beebe and Jan Guynes Clark. A hierarchical, objectives-based framework for the digital investigations process. *Digital Investigation*, 2(2):147–167, 2005.

[2] Fabio Luigi Bellifemine, Giovanni Caire, and Dominic Greenwood. *Developing Multi-Agent Systems with JADE*. Wiley Series in Agent Technology, Sussex, England, 2007. ISBN 978-0-470-05747-6.

[3] D. Bruschi and M. Monga. How to reuse knowledge about forensic investigations, 2004.

[4] Andrew Case, Andrew Cristina, Lodovico Marziale, Golden G. Richard, and Vassil Roussev. Face: Automated digital evidence discovery and correlation. *Digital Investigation*, 5(Supplement 1):S65–S75, September 2008.

[5] Daniel D Corkill. Collaborating Software: Blackboard and Multi-Agent Systems & the Future. In *Proceedings of the International Lisp Conference*, New York, USA, October 2003.

[6] Mark d'Inverno and Michael Luck. *Understanding Agent Systems*. Springer Series in Agent Technology, Berlin, Germany, 2^{nd} revised and extended edition, 2004. ISBN 3-540-40700-6.

[7] Simson L. Garfinkel. Forensic feature extraction and cross-drive analysis. *Digital Investigation*, 3S:S71–S81, 2006.

[8] Michael N. Huhns and Munindar P. Singh, editors. *Readings in Agents*. Morgan Kaufmann, San Francico, USA, 1998. ISBN 1-55860-495-2.

[9] Telecom Italia Lab (TILAB). Java Agent DEvelopment framework - JADE. Online. `http://jade.tilab.com`.

[10] V. Jagannathan, R. Dodhiawala, and L.S. Baum, editors. *Blackboard Architectures and Applications*. Academic Press, Orlando, FL, USA, 1989.

[11] George F. Luger. *Artificial Intelligence: Structures and Strategies for Complex Problem Solving.* Addison-Wesley, USA, 4^{th} edition, 2002. ISBN 0-201-64866-0.

[12] Steve Mead. Unique file identification in the national software reference library. *Digital Investigation*, 3(3):138–150, 2006.

[13] H. Penny Nii. Blackboard systems, part one: The blackboard model of problem solving and the evolution of blackboard architectures. *AI Magazine*, 7(2):38–53, 1986.

[14] S. Pinson and P. Moraïtis. An intelligent distributed system for strategic decision making. *Group Decision and Negotiation*, 6:77–108, 1996.

[15] Vassil Roussev and Golden G. Richard III. Breaking the performance wall: The case for distributed digital forensics. In *Digital Forensic Research Workshop - DFRWS*, 2004.

[16] Gong Ruibin and Mathias Gaertner. Case-relevance information investigation: Binding computer intelligence to the current computer forensic framework. *International Journal of Digital Evidence*, 4(1), 2005.

[17] Stuart J. Russell and Peter Norvig. *Artificial Intelligence: A Modern Approach.* Prentice-Hall, USA, 2^{nd} edition, 2002. ISBN 0-13-790395-2.

[18] Philip Turner. Unification of digital evidence from disparate sources (digital evidence bags). In *Digital Forensic Research Workshop - DFRWS*, 2005.

[19] Philip Turner. Selective and intelligent imaging using digital evidence bags. *Digital Investigation*, 3(Supplement-1):59–64, 2006.

[20] Philip Turner. Applying a forensic approach to incident response, network investigation and system administration using digital evidence bags. *Digital Investigation*, 4(1):30–35, 2007.

[21] H. Velthuijsen, editor. *The Nature and Applicability of the Blackboard Architecture.* PTT-Research, Maastricht, 1992.

[22] Gerhard Weiß, editor. *Multiagent Systems: a Modern Approach to Distributed Artificial Intelligence.* The MIT Press, Cambridge, USA, 2^{nd} edition, 2000. ISBN 0-262-23203-0.

[23] Michael Wooldridge. *An Introduction to MultiAgent Systems.* John Wiley & Sons, Ltd., Sussex, England, 2002. ISBN 0-471-49691-X.