Chapter 6

CASE-BASED REASONING IN LIVE FORENSICS

Bruno Hoelz, Celia Ralha and Frederico Mesquita

Abstract The traditional forensic search and seizure process employed by law enforcement is not always appropriate given large data volumes and the potential of hard drive encryption. This paper proposes a framework built on case-based reasoning to support a live forensic response during the search and seizure process. The framework assists a first responder by identifying the risks and the procedures to ensure the optimal collection of evidence based on prior cases. Test results demonstrate that the framework provides valuable assistance to first responders, reducing the time taken to complete a response and increasing the likelihood of a successful conclusion.

Keywords: Live forensics, case-based reasoning

1. Introduction

The use of strong cryptography in computing devices has altered the way first responders collect and secure digital evidence in computer crimes. First responders are increasingly using live forensic procedures, more so because the earlier method of turning off a computer by unplugging its power supply can lead to important evidence being lost. Increases in the quantity of digital evidence to be collected are also making live forensics the accepted norm [2].

However, the vast number of variables in a live forensic scenario complicates the search and seizure process. Developing a single process for the diversity of operating systems, installed applications and devices is an insurmountable task. Unfortunately, such a process is essential to maintaining data integrity and the chain of custody [8].

This paper presents a framework for live analysis that engages case-based reasoning to retrieve knowledge gained from previous cases and

reuse it in new cases [7]. Case-based reasoning also makes it possible to establish standard procedures for validating first responder actions during live analysis and minimizing possible errors. Tests conducted by the Brazilian Federal Police demonstrate that a decision support system relying on case-based reasoning can accurately identify similar cases and aid first responders in performing live forensics.

2. Live Forensics

Live forensics is conducted to address the issue of evidence volatility. A live response collects volatile evidence from a computer that is lost when the system is powered off. The volatile evidence includes information about the processes and services running on the computer, as well as the cryptographic key if hard drive encryption is used [11]. Live forensics is also used in enterprise environments when there is far too much media to collect in the time available, or the investigation is only concerned with a small amount of data.

When a law enforcement agent executing a search warrant encounters a computer that is powered off, there is nothing else to do but to seize the hard drive and hope that full-disk encryption is not being used. On the other hand, if the computer is powered on, the agent must answer three questions:

- Is live forensic analysis necessary in this case?
- If so, what data do I need to collect?
- How can I extract the data and ensure its integrity?

To answer the first question, it is important to understand why it is not always appropriate to perform a complete live analysis. A complete analysis includes data selection and extraction, keyword and registry searches, and analysis of user activity (recent files, open ports and running processes). Executing a search warrant is not a trivial task. Search warrants are often executed in potentially hostile locations, requiring agents to spend as little time as possible on the task.

Spending five minutes on a preliminary inquiry to determine if live forensic analysis is necessary can save time and effort, especially in a large operation with dozens of suspects. Likewise, it can facilitate triage, reducing the amount of data to be extracted and processed later.

Once the first responder decides to perform a live forensic analysis, there are two approaches to performing the analysis [10]. One is to conduct deeper live analysis at the location. The other is to extract all the relevant data and secure it for later analysis at a forensic laboratory.

The first approach requires the responder to execute various digital forensic procedures such as known file hash filtering, port scanning and keyword searches. These live analysis tasks can take a long time, and a rootkit can lead to false data being recovered [3].

In the second approach, the responder collects all the important data for processing at a forensic laboratory. Since volatile data can only be captured by a live analysis [10], there is no advantage to the first approach and the second approach is suitable in most cases.

The answer to the third question is the data extraction tools that must be used. Most of the tools employ on-the-fly hash computations that can be used to verify the integrity of the collected evidence. The primary issue is whether to extract the volatile memory or the hard drive data first. Since volatile memory is more prone to unintended modification, it must be acquired first in almost every case. All the actions and results during the extraction phase must be documented thoroughly because they cannot be repeated at a later time [4].

3. Case-Based Reasoning

Case-based reasoning is a decision-aiding methodology that is based on human problem solving models [9]. It is founded on the assumption that similar problems have similar solutions, and that most types of problems tend to recur. The fundamental notion is a "case," a past experience composed of three elements: the initial state or problem description, a solution that presents the steps needed to solve the problem and the final state that is represented by a set of goals. The process of case-based reasoning matches and applies the solution of a prior case to each new case encountered.

Aamodt and Plaza [1] define the case-based reasoning cycle as a set of four consecutive steps (Figure 1). The first is the retrieve step, where given a problem, one or more previously successful cases are retrieved from the case repository. The second step is to reuse or adapt the retrieved case to solve the current problem. The third step is to revise the case based on the evaluation of results, review and adjustments by domain experts. The final step is to retain the case, expanding the case repository and knowledge database.

Case-based reasoning systems learn continuously from previous experience and have been used successfully in applications ranging from the explanation of anomalous events to automobile diagnosis [7]. The broad application of case-based reasoning makes it a good fit for the live forensics problem.

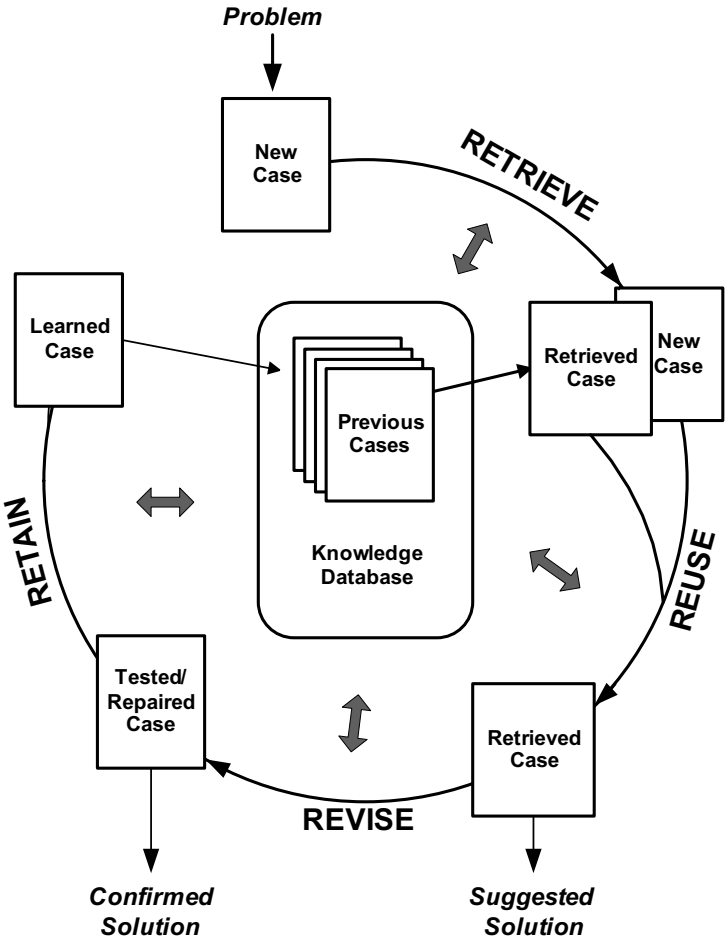

Figure 1. Case-based reasoning cycle (adapted from [11]).

4. Proposed Approach

Digital forensic experts are expected to have vast knowledge in several areas, but it is humanly impossible to have detailed knowledge about every system and application encountered in an investigation [5]. It is common for experts to be involved in as many as one thousand cases in a year. By acquiring knowledge from these cases and reusing the knowledge later, it is possible to mitigate the risks associated with full-disk encryption and other protection mechanisms, facilitate triage and the selective collection of data, and provide a decision support system for cases in which the first responder has no previous experience.

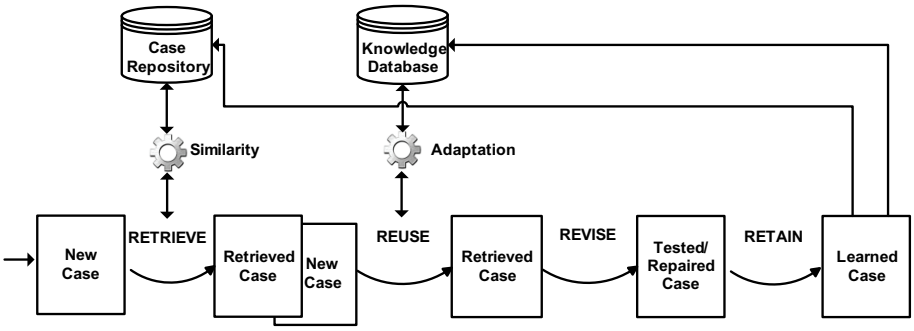

Figure 2. Proposed case-based reasoning workflow.

Case-based reasoning provides a means to collect and reuse previous solutions in new cases. Two databases help adapt the case-based reasoning cycle to digital forensics. The case repository contains data from previous cases while the knowledge database contains technical instructions and descriptions of procedures. Both databases are shared and updated by the participating experts. Figure 2 shows how the databases integrate into the proposed workflow.

In general, there are four components that must be tailored to the case-based reasoning application. The primary component is the case. The case definition facilitates the identification of the next two components, the similarities between cases and the ability to adapt cases. The fourth component is the case review method.

Table 1. Case attributes.

Suspect	**Crime**	**Computer Environment**	**Location**
Technical skills	Crime being investigated	Remote access	Ease of entrance
Positive identification	Role of the suspect	Risk of data loss	Nature of the location
	Arrest order	Specific systems	Location security

4.1 Case Attributes

A case consists of information regarding the suspect, the crime being investigated, the computer and network environment, and the location (Table 1). Regardless of the investigative procedures being used, some

information may not be available to the first responder. The missing information can be filled in by the first responder upon arriving at the scene or at a later point in time. Note that in the face of missing information the proposed framework would support less specific planning.

Much of the information is intertwined and can belong to more than one category. During the planning phase, it is necessary to determine the risk of data loss, the time limitations for live analysis, and the requirement of special equipment and software.

Information about the suspect includes whether he/she possesses the technical knowledge to employ full-disk encryption or to quickly destroy evidence. In a multi-user networked environment, it is important to know if the suspect has been precisely identified.

Other information relates to the crime being investigated, the role of the suspect in the crime, and if an arrest order exists. This information provides guidance on the most important data to be analyzed.

Most of the information related to the computer and network environment may only be determined at the scene. A key concern is whether or not the computer systems are powered on or off. Monitoring network traffic on the suspect's connection can help establish the most adequate time to perform the search. The risk of data loss due to remote access and the use of cryptography must also be determined.

Complex environments such as large enterprise networks and server farms provide unique challenges. The availability of trustworthy technical support at the location must be verified. Technical data about the network topology and operating system are also important in the planning phase.

Finally, key information regarding the search location includes the ease of access, nature of the location (e.g., home or office) and potential security issues. A heavily-guarded facility may be difficult to access and may present opportunities for the suspect to get rid of important evidence. A search warrant executed at a dangerous location may present security risks for the first responder and limit the time available to conduct live analysis.

4.2 Case Retrieval and Similarity Computations

Upon arriving at the scene, the first responder collects data about the case (Figure 3). This data, together with data collected during the planning phase, are input to the decision support system. The decision support system then retrieves previous cases that are similar, which it uses to provide recommendations to the first responder.

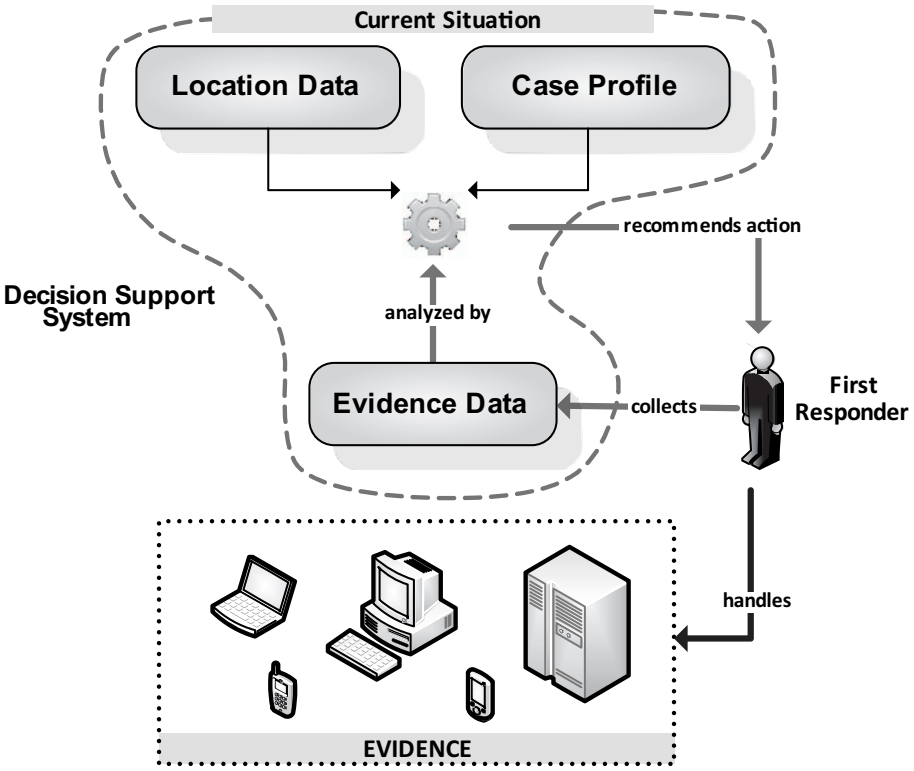

Figure 3. User and system interaction.

Case similarity matching uses a self-organizing map [6]. The vector containing the current case attributes is compared with the vectors in each cell of the self-organizing map. The most similar self-organizing map vector corresponds to an abstract case that generalizes several similar prior cases.

Figure 4 shows a self-organizing map that was constructed in our experiments. Cases with similar forensic procedures are located near each other in the figure. For example, a phishing scam and a child abuse case share certain characteristics such as the high use of webmail and interactions with online communities. As such, they also share a set of common live forensic procedures.

4.3 Case Adaptation and Reuse

After similar cases are retrieved, a solution must be crafted for the current situation. The retrieved cases provide a set of abstract forensic procedures; these procedures must be concretized according to the data provided by the first responder.

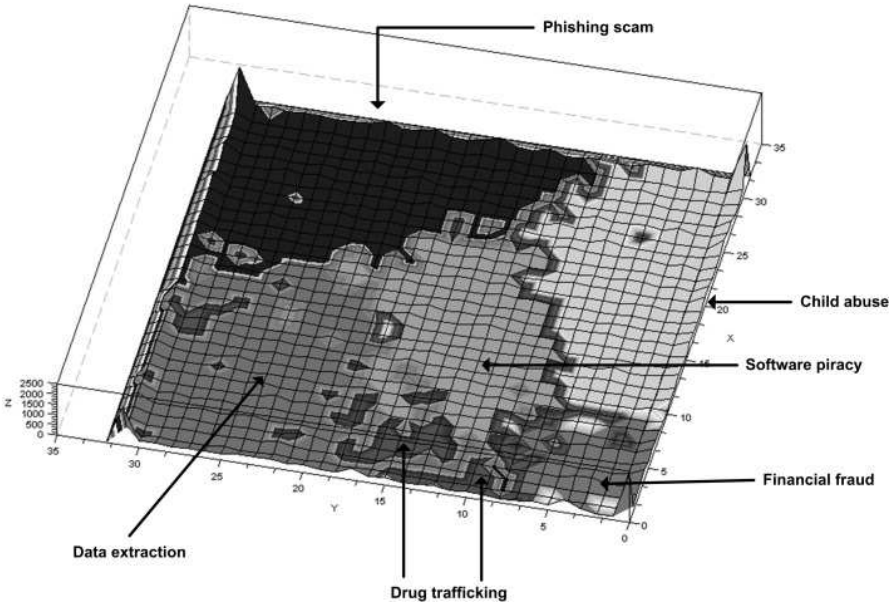

Figure 4. Self-organizing map of previous cases.

For example, an abstract forensic procedure could be to verify the existence of full-disk encryption. Based on the data provided by the first responder, its concrete instance could be to verify the presence of TrueCrypt. The knowledge database can be queried for guidelines on conducting the suggested procedure. In our example, it would list the procedures for verifying the presence of TrueCrypt.

4.4 Case Review and Storage

Every procedure performed by the first responder can be reviewed at a later point in time. If a new situation is encountered, its details are added to the case repository and knowledge database as appropriate. Entries can also be flagged as incorrect, incomplete or obsolete. Additionally, upon reviewing and simulating unsuccessful cases, digital forensic experts can identify new procedures that should be added to the databases.

5. Experimental Results

To test the proposed framework, several abstract test cases were built from attributes such as the presence of cryptography, webmail, instant messaging, home banking records, and P2P and social network appli-

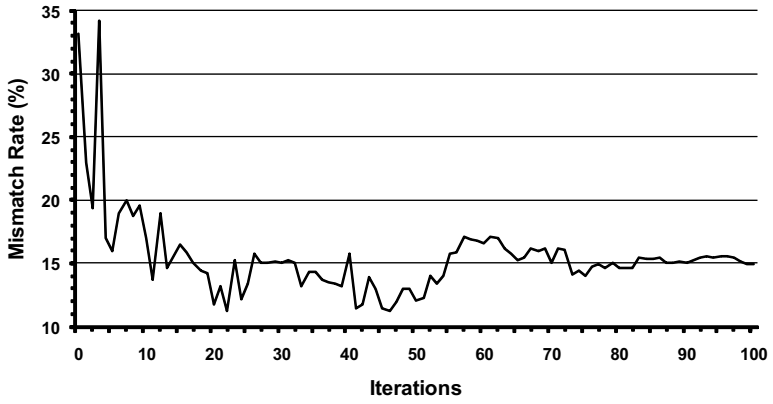

Figure 5. Rate of mismatch during case retrieval.

cations. These attributes were gathered from the forensic examinations management system used by the Brazilian Federal Police. The system contained information relating to 26,187 examinations conducted from 2008 to 2010 on digital storage media and devices such as flash drives, memory cards, cell phones, laptops and desktops.

Concrete instances of each attribute were also defined (e.g., webmail service and P2P software). A set of 1,200 test cases were generated and used to construct the self-organizing map with 32×32 cells (Figure 4). The z-axis value specifies the number of cases in each cell.

Each test case was presented to the decision support system and the results were evaluated. Figure 5 presents the rate of mismatch during case retrieval. A mismatch is deemed to occur when a case from one type of crime is identified as being the most similar to another type of crime. As mentioned above, different types of crime can share characteristics and are treated by the first responder in a similar manner. This means that, although the decision support system may not find a perfect match for the current case, it can suggest previous cases that are useful after some adaptation.

If the suggestions by the decision support system are inadequate, the first responder can perform his/her own procedures, which are then added to the knowledge database. For example, if the first responder encounters encryption software that is unknown to the knowledge database, the decision support system would recommend a new entry to be filled with the specific procedures to be followed for future cases.

Figure 5 shows that as the system learns new cases, the rate of mismatch decreases, eventually stabilizing at around 15%. The main tenet of case-based reasoning is that cases tend to repeat. Therefore, after a

period of time, the system should have sufficient knowledge to retrieve similar cases in most situations. It must also be emphasized that a mismatch does not correspond to an incorrect suggestion – it means that a different type of crime is perceived as being similar to the case at hand.

6. Examples

Three examples are presented using the cycle specified in the proposed framework. For the sake of generality, the names of the tools, systems and software applications are omitted.

Example 1: A household with four persons, one of them an unidentified phishing spam suspect. The arrest order is based on positive evidence of the crime. The computer is powered off.

- Since the computer is powered off, the first responder has no means of collecting live data.
- Based on previous cases, the decision support system suggests interviewing the individuals regarding the use of the computer and cryptography, and taking notes related to possible passwords and login information.

Example 2: A company location that is the workplace of a suspected terrorist. The suspect, who is positively identified, has good technical skills. An arrest order has been issued. The risk of data loss due to remote access and cryptography exists. Physical access is available to the location, which is safe. The computer is expected to be powered on.

- Even before data is collected at the scene, the decision support system retrieves similar cases, which suggest extra caution in securing the location to avoid data loss via the deletion or destruction of evidence.
- Upon arrival, the computer is found to be powered on and data can be collected.
- The decision support system suggests acquiring the contents of the memory for later inspection.
- The decision support system suggests running scripts to detect the presence of encryption software and encrypted data.
- The software is positively identified, as well as an encrypted volume, which is mounted and accessible.
- The decision support system suggests acquiring the contents of the encrypted volume while it is accessible.
- The decision support system suggests collecting other digital media and hard drives for laboratory analysis.

Example 3: A household with one person, who is suspected of being a child molester. The arrest order is based on positive evidence of the crime. The computer is probably powered on.

- Upon arrival, the computer is found to be powered on and data can be collected.
- The decision support system suggests acquiring the contents of the memory.

- The decision support system suggests searching for the hash values of known child porn images and acquiring the files from folders containing positive hits.
- Files are found and extracted.
- The decision support system suggests searching for instant messaging software.
- Instant messaging software is found. The decision support system provides specific procedures contained in the knowledge database to extract the logs.
- The instant messaging logs appear to be encrypted.
- The decision support system suggests listing strings in memory to use as a dictionary in attempting to decipher the logs.
- The decision support system suggests searching for P2P software and known DLLs in memory.
- File sharing software is found. The software is not listed in the knowledge database, so the decision support system cannot suggest specific procedures.
- The first responder analyzes the software, folders and configurations.
- The decision support system suggests extracting files being shared in the P2P network.
- The decision support system suggests listing the open ports to find any ongoing file sharing.
- No ongoing file sharing is found.
- Due to the incriminating evidence that is found, the suspect is arrested immediately.

In Example 1, although live forensics cannot be performed, the decision support system still provides useful recommendations regarding general forensic procedures. In Example 2, due to the presence of cryptography, the decision support system suggests procedures to ensure that the maximum amount of relevant evidence is collected. In Example 3, a reduced set of files is acquired, which reduces the amount of data to be processed at the forensic laboratory. Additionally, a situation unknown to the decision support system is encountered, so the procedures performed by the first responder are reviewed and stored for future use.

7. Conclusions

This case-based reasoning framework for live forensics uses data collected by first responders to adapt previous cases to the current situation. The experimental results demonstrate the feasibility of the framework. In particular, the framework suggests the appropriate procedures to be used in a live analysis, reducing the time required to perform the analysis and enhancing the quality of the analysis. These improvements also increase the throughput at the forensic laboratory by reducing the volume of seized data and the risk of finding encrypted data.

Future work will extend the framework to laboratory examinations. Without the strict time limitations imposed on live analysis, a wider range of procedures can be performed in a laboratory environment. These procedures must also consider the nature of the case and the characteristics of the evidentiary items, which means that knowledge about previous analyses can be reused to good effect. In addition, real-time collaboration options will be introduced to enable expert and novice first responders to exchange information during a large, coordinated police operation, helping them overcome technical difficulties and correlate data as the operation unfolds.

References

[1] A. Aamodt and E. Plaza, Case-based reasoning: Foundational issues, methodological variations and system approaches, *Artificial Intelligence Communications*, vol. 7(1), pp. 39–59, 1994.

[2] F. Adelstein, Live forensics: Diagnosing your system without killing it first, *Communications of the ACM*, vol. 49(2), pp. 63–66, 2006.

[3] B. Carrier, Risks of live digital forensic analysis, *Communications of the ACM*, vol. 49(2), pp. 56–61, 2006.

[4] B. Hay, M. Bishop and K. Nance, Live analysis: Progress and challenges, *IEEE Security and Privacy*, vol. 7(2), pp. 30–37, 2009.

[5] B. Hoelz, C. Ralha and R. Geeverghese, Artificial intelligence applied to computer forensics, *Proceedings of the ACM Symposium on Applied Computing*, pp. 883–888, 2009.

[6] T. Kohonen, The self-organizing map, *Proceedings of the IEEE*, vol. 78(9), pp. 1464–1480, 1990.

[7] J. Kolodner, *Case-Based Reasoning*, Morgan Kaufmann, San Mateo, California, 1993.

[8] W. Kruse and J. Heiser, *Computer Forensics: Incident Response Essentials*, Addison-Wesley, Boston, Massachusetts, 2002.

[9] D. Leake (Ed.), *Case-Based Reasoning: Experiences, Lessons and Future Directions*, AAAI Press, Menlo Park, California, 1996.

[10] C. Waits, J. Akinyele, R. Nolan and L. Rogers, Computer Forensics: Results of Live Response Inquiry vs. Memory Image Analysis, Technical Note CMU/SEI-2008-TN-017, Software Engineering Institute, Carnegie Mellon University, Pittsburgh, Pennsylvania, 2008.

[11] A. Walters and N. Petroni, Volatools: Integrating volatile memory forensics into the digital investigation process, presented at the *2007 Black Hat DC Conference* (www.blackhat.com/presentations/bh-dc-07/Walters/Paper/bh-dc-07-Walters-WP.pdf), 2007.