A critical review of 7 years of Mobile Device Forensics

Konstantia Barmpatsalou[a], Dimitrios Damopoulos[a], Georgios Kambourakis[a,*], Vasilios Katos[b],

[a] *Info-Sec-Lab Laboratory of Information and Communications Systems Security, Department of Information and Communication Systems Engineering, University of the Aegean, Samos, Greece*
[b] *Information Security and Incident Response Unit, Department of Electrical and Computer Engineering, Democritus University of Thrace, University Campus, Kimmeria, Xanthi,Greece*

Abstract

Mobile Device Forensics (MF) is an interdisciplinary field consisting of techniques applied to a wide range of computing devices, including smartphones and satellite navigation systems. Over the last few years, a significant amount of research has been conducted, concerning various mobile device platforms, data acquisition schemes, and information extraction methods. This work provides a comprehensive overview of the field, by presenting a detailed assessment of the actions and methodologies taken throughout the last seven years. A multilevel chronological categorization of the most significant studies is given in order to provide a quick but complete way of observing the trends within the field. This categorization chart also serves as an analytic progress report, with regards to the evolution of MF. Moreover, since standardization efforts in this area are still in their infancy, this synopsis of research helps set the foundations for a common framework proposal. Furthermore, because technology related to mobile devices is evolving rapidly, disciplines in the MF ecosystem experience frequent changes. The rigorous and critical review of the state-of-the-art in this paper will serve as a resource to support efficient and effective reference and adaptation.

*Corresponding author. Department of Information and Communication Systems Engineering, University of the Aegean, Samos GR- 83200, Greece.

Email addresses: tbarbatsalou@gmail.com (Konstantia Barmpatsalou), ddamop@aegean.gr (Dimitrios Damopoulos), gkamb@aegean.gr (Georgios Kambourakis), vkatos@ee.duth.gr (Vasilios Katos)

October 24, 2013

Keywords: Mobile Device Forensics; Smartphone; Security;

1. Introduction

Internet and Information Technology (IT) are no longer a novelty, but a necessity in almost every aspect concerning people's lives, extending to a great variety of purposes, from business, education and public health to entertainment, commerce and even more. Models and behavioral patterns succumbed to changes in order to adapt to the new conditions that IT has created. It is thus inevitable that delinquent actions and patterns are expected to follow the same direction concerning their evolution and differentiation. Cybercrime, including the involvement of IT infrastructures in minor and major criminal activities, led to the creation of a new discipline, namely Digital Forensics (DF), equivalent to classical forensics where "evidence analysis takes place using data extracted from any kind of digital electronic device" (Harrill and Mislan, 2007). Although a digital device can participate in a crime by different means, "unless hardware itself is contraband, evidence, an instrumentality, or a fruit of crime, it is merely a container for evidence" (Casey, 2011b). Due to different attributes among computering devices, DF has developed several sub-disciplines, including Computer Forensics, Memory Forensics, Multimedia Forensics, Network Forensics, Small Scale Device Forensics or Mobile Device Forensics. (Casey, 2011b).

Technology concerning mobile devices has presented revolutionary growth during the last decade. Mobile phones, enhanced with hardware and software capabilities, not only serve as a means of communication, but also as small-scale portable computers with advanced communication capabilities. For instance, smartphones are able to store a rich set of personal information and at the same time provide powerful services, e.g. location-based services, Internet sharing via tethering, and intelligent voice assistance to name just a few. In addition to the traditional cyber attacks and malware threats that plague legacy computers, smartphones now represent a promising target for malware developers that struggle to expose users sensitive data, compromise the device or manipulate popular services (Damopoulos et al., 2011, 2012a, 2013). Additionally, the number of stolen or lost smartphones has increased rapidly over the last few years. In a research conducted by McMillan et al. (2013) among British legal databases from 2006 to 2011, the involvement of smartphones in delinquent activities presented an average growth of 10 cases

per year. Some of the devices mentioned above may be used as a stepping stone (Chavez, 2008) to spoof the real identity of the attacker or to take advantage of the stored sensitive personal information.

Without a doubt, the widespread use of portable, small scale devices significantly increases the likelihood of a such devices being involved in a criminal activities. According to Jansen and Ayers (2007), MF is the science of recovering digital evidence from a mobile device under forensically sound conditions using accepted methods. The field of MF is challenging by default, due to the fact that smartphones have limited processing and memory resources, different CPU architecture and a variety of well-secured Operating System (OS) versions compared to those of a personal computer, making forensic processing a complex task. These challenges are compounded by the rapid rate of change in mobile device technology. While some forensic methods may be effective for a certain device or OS version, they may be useless for its successor(s). The variety of models and OSs can also raise a barrier concerning usage training. Investigators charged with the task of interacting with the devices have to be advanced users, in order to minimize the risk of human-driven errors. On the other hand, the amount of acquired data from small-scale devices may be considerably less than the amount of data retrieved from personal computers (Yates, 2010). Finally, when it comes to power consumption matters, which leads to resources vanishing quicker than in devices such as notebooks, since forensic labs are equipped with power supply cables, the only challenge occurs when the battery is almost depleted upon seizure. If the device shuts down before or during an *on-the-fly* acquisition, crucial information will probably be lost, including volatile data. Preservation of volatile data is becoming particularly important in modern devices in general, due to the proliferation of Web 2.0 and the underlying technologies. A representative example is *Volatility* (Volatile Systems, 2011), a memory analysis tool where addons have been developed in order to scavenge twitter posts and Facebook related data directly from RAM on mobile devices. Such data are dynamic and do not exist in the non-volatile storage.

Additionally, Lessard and Kessler (2009) mentioned that, one of the major difficulties in the field of MF is the general lack of hardware, software and/or interface standardization within the industry. This fact makes forensic processing a hard task, especially for unified research.

Attempting to cover the complete history of research and development in MF from the very beginning would be a time-consuming and outdated procedure, particularly because literature related to early developments in

the field is scarce and older generation devices are no longer in use. The complexities of functionality in smartphones, combined with the growing number of such devices that are in use worldwide led to the decision to limit this research solely to the smartphones area.

Based on these facts, this work reviews and categorizes the main milestones, methodologies and significant studies for MF based on several factors, aiming to provide a comprehensive view of the state-of-the-art, by performing an in-depth study of the field.

More specifically, this work provides a thorough overview of the field of MF, by reviewing and presenting a detailed assessment of the actions and methodologies taken throughout the last 7 years. As further explained in Section 4.1, the decision to concentrate on the past 7 years of research and development in MF was based on the type and interdependence of the various contributions in the particular time frame and in an effort to provide a holistic view of the state-of-the art in MF area. In short, an effort was made to focus the time period to the minimum possible – this is critical for this rapidly evolving area – but doing so without omitting important parts that would make difficult to assemble the whole MF puzzle. A schematic timeline of the most significant studies so far is given, in order to provide a quick but complete way of observing the milestones and trends within the field. By doing so we offer an analytical progress report concerning the evolution of MF. Since standardization activities in the area are quite far from being mature, this work can help shape a common framework for MF. As already pointed out, technology concerning mobile devices is evolving at a rapid pace, and MF must continue to adapt to these frequent changes. A strong fundamental infrastructure will help support efficient and effective adaptation. As a result, newcomers or even experts of the MF field will be able to have a compact image of the state-of-the-art. Also, this work exposes existing problems in the field of MF, thus providing motivation and future direction for further research in the area.

While there is a satisfying number of studies throughout literature (Yates, 2010; Hoog and Gaffaney, 2009; Grispos et al., 2011; Satheesh Kumar et al., 2012; Casey et al., 2010) that have cross-evaluated commercial forensic suites for extracting conclusions about the retrieved data, this is the first work that presents a different approach focusing more on new methods, *but not totally neglecting the former*. For instance, in some types of smartphone platforms, the academic work is practically nonexistent while commercial tools have an excellent coverage. Therefore, due to the fact that the development of

commercial forensic tools is at large based on elements of the main acquisition methodologies (addressed in this paper), we did not dedicate an extended session on them.

The rest of this work is structured as follows. Section 2 enumerates, classifies and analyzes the criteria that will be used throughout this work, providing this way the necessary background knowledge. Section 3 surveys the state-of-the-art in the field. Section 4 elaborates on the research work done so far contributing a complete categorization of the major MF approaches. The last section draws a conclusion.

2. Preliminaries

Considering the guidelines presented by ACPO (ACPO, 2007) and the EDRM model (EDRM LLC, 2013), which are dealing with forensically sound practices, this work classified the context of MF into the following three categories that are suitable for conducting the analysis and comparisons used throughout this research: Evidence *acquisition methods*, *Operating Systems*, and Acquired *Data Types*.

2.1. Acquisition Methods

Forensic acquisition from devices is divided into three categories: manual, logical and physical. Each one uses different attributes of the device for extracting the desired amount of data. Manual acquisition is defined as whatever an individual is capable of acquiring by interacting with the device itself. This procedure may consist of two separate phases: keeping a log of the actions taken (Grispos et al., 2011) and interacting with installed applications to copy the existing data (Mokhonoana and Olivier, 2007). Additional means, such as cameras can be used in order to record the device state (Grispos et al., 2011). Since the probability of human error is very high and crucial elements can be bypassed, this method should be used as supplementary. Due to the fact that manual acquisition is the only technique returning data in human interpretable format, it is necessary to take place simultaneously with the other two kinds. As a result, it will not be examined as a separate category, but will be integrated in to the other two.

Logical acquisition retrieves a bitwise copy of entities such as files and directories that reside inside a logical storage means and "provides context information for the formerly mentioned objects, such as date-time stamps and location within the file system of the target mobile device" (Casey, 2011a). It

mainly concerns data that has not been deleted and is achieved by accessing the file system of the device (Hoog, 2011). Nevertheless, information that is not practically deleted , but "disguised as" available space for further overwriting within databases may be retrieved by *file system* access. Data that has already been deleted are less likely to be acquired. Logical acquisition techniques and tools interact with the file system whereas physical acquisition methods access lower areas. This leads to the conclusion that physical and logical acquisition show different strengths and weaknesses concerning the files they retrieve. For instance, physical acquisition retrieves deleted files, whereas logical acquisition is more efficient for recovery of user data (call and SMS logs, contacts) (Grispos et al., 2011). "Sometimes logical acquisition is not possible, for instance when the device is broken beyond repair, or when the device does not have a standard interface to do the logical acquisition over"(Klaver, 2010). Physical acquisition may also be conducted before logical, if there is no other way to bypass user security mechanisms such as passwords and screenlocks (Breeuwsma, 2006). Summarizing, logical acquisition can be divided to the following categories: partition imaging, copying files-folders, content provider and Recovery Mode (Hoog, 2011; Vidas et al., 2011; Son et al., 2013).

On the other hand, physical acquisition is solely related to the physical storage medium. Such a technique is also mentioned as a bitwise copy of the internal flash memory (Grispos et al., 2011; Quick and Alzaabi, 2011; Thing and Chua, 2012). This kind of acquisition is more likely to retrieve deleted data (Husain et al., 2011), which is treated as unallocated but still exists in memory. However, physical acquisition procedures are more likely to damage the device while it is being dismantled. According to (Klaver, 2010) "True physical acquisition can either mean physically removing memory from the device, using hardware techniques like Joined Test Action Group (JTAG) (Breeuwsma, 2006) to extract data from the device or use an (adapted) bootloader to gain low level access to the device". These kinds of techniques "are not only technically challenging and require partial to full disassembly of the device, but they require substantial post-extraction analysis to reassemble the file system (Hoog, 2011). Nevertheless, it is generally acceptable in a forensic context that physical acquisition prevails over logical, because it allows deleted files and any data remnants present to be examined (Jansen and Ayers, 2007). Sometimes, however, as in the case of the Windows Mobile OS (Klaver, 2010), research has led to the development of alternate acquisition methods that lie somewhere in between a physical

acquisition and a logical one (usually referred to as *pseudo physical*).

2.2. Operating Systems

A factor of heterogeneity which is an impediment against the development of a common MF framework is the existence of different OSs (mobile platforms). Current market share gives Android and iOS the prevailing percentages (Becker et al., 2012). Other OSs, such as Blackberry and Windows Phone remain also a popular choice. In the past, the need to exploit vulnerabilities in these operating systems in order to perform physical acquisitions posed a challenge to admissibility in court (ACPO, 2007; Jansen and Ayers, 2007). However, such concerns have decreased as MF techniques became more mature and better understood.

In generalized terms, *low-level modifications* grant access to system areas which were by default protected by each OS manufacturer. The privileges users are gaining after the application of a low-level modification vary among different OSs. Low-level modifications can have a variety of names depending on the OS they are applied to. They are either known as Rooting (Android, Windows Mobile), Jailbreak (iOS) or Capability Hack (Symbian). For example, Android users are able to install and run applications that require access to the root directory, such as backup features. In addition to root privileges, iOS users can install applications not available in the AppStore. *Capabilities* on Symbian devices are security mechanisms that can be bypassed by installing a root certificate and thus allowing users to install and execute unsigned applications. A brief overview of the OSs characteristics will be made in the next paragraphs, alongside with their impact to forensic acquisition.

Android was first released in 2007 and in less than five years achieved to be the dominant OS in the mobile handsets market. The OS runs on a Linux 2.6 - based kernel, which serves for supporting fundamental functions, such as device drivers, network infrastructure and power management (Yates, 2010; Hoog, 2011; Vidas et al., 2011). The next level of the Android architecture is the domain of the libraries, split to application and Android runtime ones. The former category provides the appropriate infrastructure for applications to run properly, such as binaries and graphics support, while the latter consists of the Dalvik Virtual Machine (DVM) and the core libraries that provide the available functionality for the applications (Yates, 2010). Its main purpose is the creation of a stable and secure environment

for applications execution. Each application runs in its own sandbox (virtual machine). Therefore, it is not affected by other applications or system functions. Using certain resources is only permitted by special privileges. This way, a satisfying level of security is preserved. While the Android Runtime Libraries are written in Java (Yates, 2010), DVM translates Java to a language that the OS can perceive (Simao et al., 2011). The rest of the architecture consists of the Applications Framework and the Applications Layer that manage general application structure, such as containers, alerts and the applications themselves.

Due to the small chip size, non-volatile nature and energy efficiency, NAND flash memory was selected to equip Android devices for storage purposes (Hoog, 2011; Zimmermann et al., 2012). NAND flash memory needed a file system that was "aware of the generic flash limitations and take these into account on the software level when reading and writing data from and to the chip"(Zimmermann et al., 2012). Yet Another Flash File System 2 (YAFFS2) was the first file system implemented for devices running Android. After some years of actual use on the other hand, many issues concerning system performance, velocity of input/output actions and large files coverage occurred. As mobile devices architecture tends to follow the path of desktop computers and acquire multiple core processors, another obstacle arises, since YAFFS2 cannot support the specific technology (Kim et al., 2012). Right before the release of ver. 2.3 of the OS (Gingerbread), the file system was replaced to EXT4. The specific file system, apart from successfully coping with the weak points of YAFFS2, is enhanced with the *journaling event function* (Kim et al., 2012), which provides recovery options and facilitates acquisition of unallocated files.

Android provides potential developers with the SDK (Software Development Kit), which includes a very important tool for forensic and generic purposes, the Android Debug Bridge (adb). Adb uses a TCP or USB connection between a mobile device and a computer. The appropriate software is installed at both sides in order to acquire debugging information, start a shell session with the provided interface, initiate file transactions and add or remove applications (Hoog, 2011; Simao et al., 2011; Vidas et al., 2011). Since adb grants a terminal interface, actions like rooting and memory image extraction can be easily performed.

NAND flash memory was incompatible to the Linux-based kernel. A new technique had to be implemented to provide the software components with the ability to access the flash memory areas (Vidas et al., 2011). The Memory

Technology Devices (MTD) system was one of the facilities serving as an intermediary between the kernel and the file system and is present in many Android devices. Handsets that do not support the MTD system usually utilize the plain Flash Transaction Layer (FTL) that enables communication between the two parts (Hoog, 2011). Although there are no restrictions concerning the MTD numbers or types, a certain standard had been adopted from many device manufacturers (Lessard and Kessler, 2009; Hoog, 2011; Vidas et al., 2011). MTDs are divided to several partitions, according to the type of information they store. They can contain information about booting, recovery, user data, configurations, cache and system files.

Blackberry OS devices are designed by the Research in Motion (RIM) company and have a diversity of popularity among different countries and groups worldwide. Few things concerning the Blackberry OS itself and its ingredients are known from official sources, since the manufacturer does not provide sufficient documentation. However, substantial amounts of information concerning support were obtained via reverse engineering. These acquisitions are certain to trigger further research. A significant attribute concerning the OS is that it consists of two separate runtime environments, one Java ME-based destined for applications and one MDS-based, destined for network functionality and operations. One of the most forensically interesting elements of the OS is the *Interactive Pager Backup (IPD)* file, a collection of databases where information such as call logs, SMS and other user data are stored (ipddump, 2011). MacOS X is also known to generate IPD .zip compressed files, but with the *.bbb* suffix (Forensics Wiki, 2012). User data, such as contacts, messages, images and OS artifacts are stored in databases, which are the acquisition target of every forensic operation.

iOS was first released in 2007. It is a UNIX-based OS, partially following the architecture of the MacOS X equivalent. The main storage device of a mobile phone running the iOS is divided into two partitions. The first contains the OS fundamental structure and the applications, while the second contains all the user-manipulated data (Husain et al., 2011). The two bottom layers, Core Services and Core OS provide support for low-level data types, network sockets and file access interfaces. The Media Services layer consists of the infrastructure responsible for 2D and 3D graphics, audio and video. Finally, the Cocoa Touch layer contains two subcategories, the UIKit, which is equipped with the appropriate interface material for applications and the Foundation framework, which is supporting file management, collections and network operations (Yates, 2010).

Maemo is a Linux-based, open source OS. Even though it is not widespread and its development has been frozen since Oct. 2011, there are some research-oriented interesting features, such as the fact that user data, OS functions and swap spaces are situated in different partitions (Lohrum, 2012).

Apart from popular brands, massive production (Fang et al., 2012b) is also present in replicas of the former. One of the most popular brand names of this kind is the *Shanzhai* iPhone imitation. Their low cost is a motivation for potential buyers. In addition to the fact that they are not easily tracked down, they make a valuable "weapon" for delinquent actions. Lack of documentation on infrastructure and system manuals provokes impediments for forensic investigations.

Symbian is one of the older OS in the category, with its first release taking place in 1997 as EPOC 32 and discontinued after January 2013. Applications are mainly written in Java, while its native language is Symbian C++ (Mokhonoana and Olivier, 2007). Since many different versions of the OS exist, it is inevitable that slight variations concerning its architecture will also be present. The UI Framework is the upper level and consists of the infrastructure responsible for user interface functionality. Below that resides the Application Services Layer, hosting essential services for applications to run properly. A separate layer is devoted to Java ME, in order to provide compatibility with the OS. It contains the virtual machine and some supportive packages. Networking services, handlers and components, graphic support elements and generic services are combined under the OS Services Layer. Lastly, the lowest level concerns the hardware and kernel infrastructure (Morris, 2006; Yates, 2010).

WebOS is a Linux-based OS, designed especially for HP smartphones and tablets. It can be considered as a hybrid mobile OS, since its partitions have different formats, according to the use they had been destined to (Casey et al., 2011). The user partition, which contained user generated data and multimedia files had an FAT 32 file system format, whereas the system partition had an ext3 file system. The fact that it was supporting a bunch of innovative features (Hewlett-Packard Development Company, L.P., 2011), such as a web browser backbone infrastructure (Hewlett-Packard Development Company, L.P., 2011; Casey et al., 2011) and multitasking was not able to hamper an upcoming fall; from being the default OS for many devices, it was purchased by LG electronics so as to equip Internet TV sets after some modifications.

Similarly to other mobile OSs, it also provided an SDK and a special version giving the developers the opportunity to interact with code situated in lower levels of the system, such as binaries and libraries, mainly written in C and C++.

The *Windows Mobile OS* is the evolution of Windows CE, used mainly on handheld devices, such as palmtops and PDAs (Satheesh Kumar et al., 2012). The Windows Phone OS is its successor, with many structural elements of forensic importance in common, such as EDB files (Kaart et al., 2013). It is a Windows-based system, with similar properties specially modified so as to apply to the nature of mobile devices. One of the basic examples in this category is its file system. The T-FAT file system (Transaction-safe FAT) is a variation of the FAT file system used in desktop versions of Windows, enhanced with recovery options (Klaver, 2010; Yates, 2010). Devices incorporating this OS support NOR and NAND flash chips, so it is possible to either encounter a device running all its functions solely on one of the two categories, or a hybrid running its OS from NOR and storing its user data in NAND. Likely to mobile OSs mentioned before, the architecture of the Windows Mobile OS consists of similar layers. That is, the upper layer, Application UI the median between the user and the applications and the lower layer (above hardware) that provides the appropriate infrastructure for completion of system-oriented routine tasks, such as start-up, networking and other functions (Sasidharan and Thomas, 2011). The Framework and CLR layers contain libraries serving to execution and performance of applications.

2.3. Data Types

Data acquired from forensic examinations can be also classified, depending on their types and the entity that has access to them. The first group consists of data handled and altered strictly by OSs, such as connection handlers (GPS, WiFi) and OS defaults and structural elements (IMEI, IMSI). The second group concerns data imported and edited by users, such as text messages, contact lists, pictures and all sorts of customized application data. Data used by applications as background procedures and other similar entries manipulated by applications, form the third category.

Table 1 offers a complete view of the areas where forensically significant data are stored in each mobile platform. Note that Databases and External Storage are present in every OS, thus pointing out that a common framework can be implemented towards their acquisition. Taking into account previous research in the field, it seems than the RAM Heap as source of

information was only researched in Windows Mobile and Android carrier devices (504ensics Labs, 2013) despite the fact that other OSs may also store valuable data in it. The more popular an OS is, the more research effort is anticipated to be devoted to it. However, this not always the case especially between different countries. For instance, while in Canada the BlackBerry platform is still quite popular, relatively little academic research has been devoted to it, but Cellebrite has made significant progress acquiring Blackberry devices physically and decoding the information they contain. Popularity though is only one of the factors that affect potential research activity. The availability of documentation concerning a specific OS also contributes to future decisions on research. For example, Blackberry was known for the lack of distributed documentation, so it was harder, but not impossible to examine.

Table 1: Forensically Significant Data per OS

OS	Databases or Files	External Storage	Shared Preferences	Network	System Logs or Registry	RAM Heap
Android	X	X	X	X	X	X
Blackberry	X	X				
iOS	X	X		X	X	
Maemo	X					
Symbian	X	X				
WebOS	X	X		X	X	
Windows Mobile	X	X		X	X	X

3. State-of-the-art

3.1. Standards Background

Given the fact that MF is a relatively new discipline and presents big deviations from computer forensics, developing standards in this area is challenging. Some early efforts on MF standardization include the series of guides by ASTM International (2009), focusing on *digital and multimedia evidence* principles and Best Practices for Mobile Phone Forensics by SWGDE (2009). Both organizations have been publishing updated versions of the standards

until 2013. One attempt to create an ISO certification was published in Oct. 2012, containing guidelines of general acceptance. "The fundamental purpose of the digital forensics standards ISO/IEC 27037, 27041, 27042 and 27043 is to promote good practice methods and processes for forensic investigation of digital evidence"(ISO/IEC, 2012).

Within his research, Marshall (2011) reviewed the state-of-the-art dealing with standardization. Similarly to ancestors and successors, the author annotates the standardization problem and enumerates attempts of publishing guidelines and standards. One of the most distinctive attributes of the research was the schematic representation of the relationship among the ISO /IEC 27xxx standards. Each entity of the diagram refers to an ISO publication and is mutually connected to the others. Right above the standards, we can find the *Investigation Principles and Processes* level, which has one-way relations to the entities. The ISO upcoming publications are sorted according to the EDRM model (EDRM LLC, 2013). The relevance between ISO publications and model entities are described in Table 2.

Table 2: ISO and EDRM Relevance

ISO Publication	**EDRM Entity**
ISO/IEC 27035 Incident Management	Information Management
ISO/IEC 27037 Identification, Acquisition and Preservation of Evidence	Identification, Collection and Preservation
ISO/IEC 27041 Assuring Suitability and Adequacy of Methods	Processing and Review
ISO/IEC 27042 Analysis and Interpretation of Digital Evidence	Analysis, Production and Presentation

National Institute of Standards and Technology (NIST) has been actively involved in publishing guidelines for MF investigation regulations, in a series of Special Publications (SP). The two most recent publications are SP 800-101 (Jansen and Ayers, 2007) and Reference Materials NIST IR-7617 (Jansen and Delaitre, 2009), which had superseded the obsolete SP 800-72 (Jansen and Ayers, 2004a) and NISTIR 7100 (Jansen and Ayers, 2004b) issues. At least for the time being the work conducted by Jansen and Ayers (2007) is considered a milestone in the field of MF. However, it seems that this version of guidelines is going to be replaced soon by the later candidate that is currently under review (Ayers et al., 2013). For more information about this under preparation publication, refer to Section 4.2.