
Implementation
To automate the execution of the activities of the proposed digital forensic process we implemented a prototype. An Eclipse plug-in based on EMF [1], GMF Error! Reference source not found., and Xtext [3] provides a graphical designer to represent the general and the concrete model of the crime scene. It also includes a Xtext-based editor to specify domain assumptions, hypotheses and suspicious events in the Event Calculus language. The plug-in defines extensions to automatically configure the proactive and reactive activities of a digital forensic process. The Proactive Collection gathers the monitored events during the system execution and sends them to the Proactive Analysis that will only select those necessary to evaluate the start/stop conditions. The Proactive and Reactive Analysis invokes the Event Calculus Analyzer (Decreasoner [6]) programmatically. The Presentation activity interprets the results of the Reactive Analysis (models generated by Decreasoner) to identify events and predicates abducted, and indeed compute the proportion of satisfied predicates of a hypothesis. If a hypothesis is fully satisfied, the Presentation combines the Event Calculus specification of that hypothesis with the analysis results to generate a forensic argument.

Scenario and Evaluation Set-up
The evaluation was conducted on a public digital forensic case Error! Reference source not found. where a confidential document of a startup company was posted in the technical support forum of a competitor’s website. The document came from the computer of the CFO (Jean). The data-set provides a copy of Jean’s computer hard drive and a copy the document. The investigation aims to understand how did the document get from Jean’s computer to the competitor website. The data-set is big enough to be realistic (~1.46GB), but it is small enough to allow performing an investigation on a desktop computer.

Our evaluation assessed the correctness and effectiveness of our approach compared to traditional digital investigations aided by digital forensics tools. In particular, we verify if, under the assumptions that modeled hypotheses of a crime are correct, our approach leads to the same - or even more accurate - conclusions. Effectiveness is estimated in terms of the amount of evidence an investigator has to analyse that also affects the number of hypotheses to be evaluated. We conducted the same investigation by using available digital forensics tools, our approach without proactive activities, and our approach with proactive activities.

Traditional Digital Investigation
We acquired the image of the computer hard drive by using the Sleuthkit & Autopsy tool [4] and by searching for the Installed Programs from the Autopsy menu we discovered that the installed operating system is Windows XP Service Pack 3 (last written on 6/7/2008T07:21:46). We also used PSTViewer Error! Reference source not found. to analyse sent/received emails and RegRipper [5] to analyse Windows Registry hives and WinMD5 Error! Reference source not found. to compute file hashes. Note that all the dates we used for this investigation are in the GMT timezone.

By searching for one of the strings contained in the confidential document ($1,009,000), the keyword search functionality of Autopsy only gave one result (C:/Documents and Settings/Jean/Desktop/m57biz.xls).

This document was created on 2008-06-16T16:13:51 by Alison Smith and was last accessed on 2008-07-20T01:28:03 by user Jean. From this information, we started formulating a set of hypotheses shown in Figure 1.
1. m57biz.xls is sent via email from a computer user to an internal employee
2. m57biz.xls is sent via email from a computer user to an external recipient
   a. An internal employee was impersonating the external recipient
   b. The credential of the user were stolen
3. m57biz.xls is copied on a USB pen that is mounted by a computer user
   a. The credential of the user were stolen
4. A malware is installed from a blacklisted URL

Figure 1. Investigative Hypotheses.

The amount of evidence inspected (coming from search results) and operations performed for each hypothesis is shown in Figure 2.

Figure 2. # Results/Operations in a Traditional Investigation.

In the first hypothesis we speculate that one of the computer users sends the confidential document as an email attachment by a computer user to an internal employee. We inspected the list of installed
programs (133 programs) and we noticed that Outlook is the only email client available. From Autospy we looked for the Outlook data files available (.pst) and we obtained as a result files (administrator.pst and outlook.pst). Outlook.pst is associated with Jean inbox (222 received emails) and outbox (23 sent emails). Administrator.pst is associated with the Administrator inbox (1 received emails) and outbox (0 sent emails).

After inspecting Jean inbox we noticed that the confidential document was sent as an email attachment by user Jean to an external email address (tuckgorge@gmail.com) on 2008-07-20T01:28:47. Since the attachment is encoded as Base64, we decoded with OpenSSL by using GpGOL plugin of Outlook. Then, after comparing the MD5 code of the attachment (e23a4eb7f2562f53e88c9dca8b26a153) with the one of the original document, we noticed they match. We generated the MD5 code of the confidential document and the attachment by using WinMD5.

This email was a response to another email that Jean received from userid 558838 and mailserver xy.dreamhostps.com. Jean received two emails from the same userID.

Request #1:
Date: 2008-JUL-20 00:39:57
Request #2:
Date: 2008-JUL-20 01:22:45

From this evidence we speculate that Jean was the victim of a phishing attack and hypothesis 1 is not viable.

Hence, the second hypothesis we formulated is that the document is sent by user Jean as an email attachment to an external email address. We also formulated additional explanations - sub-hypotheses. We thought that Jean simulated to be the victim of a phishing attack, i.e. phishing emails were coming from her laptop and she was impersonating tuckgorge@gmail.com (2.a).

However, from the evidence available it was not possible to understand if Jean was impersonating the sender of the phishing email, because from Jean’s sent box we could not find any email having tuckgorge@gmail.com as a sender. We also formulated an additional explanation that Jean’s credentials were stolen (2.b). To demonstrate this sub-hypothesis we analysed the sam Windows registry hive to identify created users (10 users), their last login time and the number of failed logins, as shown below.

User Information
-------------------------
Username        : Administrator [500]
Full Name       :
User Comment    : Built-in account for administering the computer/domain
Account Type    : Default Admin User
Account Created : Tue May 13 22:20:14 2008 Z
Last Login Date : Mon Jul 21 01:22:18 2008 Z
Pwd Reset Date  : Tue May 13 22:23:39 2008 Z
Pwd Fail Date   : Never
Login Count     : 24
  --> Password does not expire
  --> Normal user account

Username        : Guest [501]
Full Name       :
User Comment    : Built-in account for guest access to the computer/domain
Account Type    : Default Guest Acct
Account Created : Tue May 13 22:20:14 2008 Z
Last Login Date : Never
Pwd Reset Date  : Never
Pwd Fail Date   : Never
Login Count     : 0
  --> Password does not expire
  --> Account Disabled
  --> Password not required
---> Normal user account
Username : HelpAssistant [1000]
Full Name : Remote Desktop Help Assistant Account
User Comment : Account for Providing Remote Assistance
Account Type : Custom Limited Acct
Account Created : Tue May 13 21:24:45 2008 Z
Last Login Date : Never
Pwd Reset Date : Tue May 13 21:24:45 2008 Z
Pwd Fail Date : Never
Login Count : 0
--> Password does not expire
--> Account Disabled
--> Normal user account

Username : SUPPORT_388945a0 [1002]
Full Name : CN=Microsoft Corporation,L=Redmond,S=Washington,C=US
User Comment : This is a vendor's account for the Help and Support Service
Account Type : Custom Limited Acct
Account Created : Tue May 13 21:25:56 2008 Z
Last Login Date : Never
Pwd Reset Date : Tue May 13 21:25:56 2008 Z
Pwd Fail Date : Never
Login Count : 0
--> Password does not expire
--> Account Disabled
--> Normal user account

Username : Kim [1003]
Full Name : Kim
User Comment :
Account Type : Default Admin User
Account Created : Wed May 14 05:32:56 2008 Z
Last Login Date : Never
Pwd Reset Date : Never
Pwd Fail Date : Never
Login Count : 0
--> Password does not expire
--> Normal user account

Username : Jean [1004]
Full Name : Jean
User Comment :
Account Type : Default Admin User
Account Created : Wed May 14 05:33:08 2008 Z
Last Login Date : Sun Jul 20 00:00:41 2008 Z
Pwd Reset Date : Never
Pwd Fail Date : Never
Login Count : 80
--> Password does not expire
--> Normal user account

Username : Addison [1005]
Full Name : Addison
User Comment :
Account Type : Default Admin User
Account Created : Wed May 14 05:34:03 2008 Z
Last Login Date : Never
Pwd Reset Date : Never
Pwd Fail Date : Never
Login Count : 0
--> Password does not expire
Among the users, only the Administrator (last login on 2008-07-21T01:22:18), Jean (last login on 2008-07-2-0700:00:41) and Devon (last login on 2008-07-12T03:02:47) logged on the laptop at least once and no failed login attempt was made. In this case, we also assumed that a potential offender has not concealed necessary (e.g., an offender crafted the sam Windows registry hive to hide its logins). Given the identified evidence, we do not have enough elements to demonstrate sub-hypothesis 2.b.

The third hypothesis we formulated is that an offender copied the confidential document on a storage device. To demonstrate this hypothesis we inspected the system Windows registry hive by using RegRipper to obtain mounted devices. In particular, by inspecting the results given by plugin mountdev2, we noticed that the following volumes have been registered (2 removable media).

Among the users, only the Administrator, Jean and Devon logged on. Given the evidence, we do not have enough elements to demonstrate sub-hypothesis 2.b.

The third hypothesis we formulated is that an offender copied the confidential document on a storage device. To demonstrate this hypothesis we inspected the system Windows registry hive by using RegRipper to obtain mounted devices. In particular, by inspecting the results given by plugin mountdev2, we noticed that the following volumes have been registered (2 removable media).
Device: \\??\STORAGE\RemovableMedia\8\&33c9e989\&0\RM\{53f5630d-b6bf-11d0-94f2-00a0c91efb8b\} \\
\\??\Volume\{c3b81074-4b2a-11dd-ad95-000c298d352b\}
Device: \\??\STORAGE\RemovableMedia\8\&38f54984\&0\RM\{53f5630d-b6bf-11d0-94f2-00a0c91efb8b\} \\
\\DosDevices\E: \\
\\??\Volume\{d9cb701a-55fa-11dd-ad9a-000c298d352b\}

We can notice that a floppy disk, a VMWare CD and a Removable media were mounted respectively on drive A, D E. We also identified the id of the removable media that were mounted and their last mount time from the results given by plugin usbdevice of RegRipper.

Disk&Ven_CB&MProd_Flash_Disk&Rev_5.00 [Sun Jul 6 07:11:34 2008] 
S/N: 15003702E152E204&0 [Sun Jul 6 07:11:37 2008] 
FriendlyName : CBM Flash Disk USB Device 
ParentIdPrefix: 8\&33c9e989&0

Disk&Ven_Generic&Prod_USB_Flash_Drive&Rev_1.00 [Sun Jul 20 01:26:18 2008] 
S/N: 7\&162a4319&0 [Sun Jul 20 01:26:21 2008] 
FriendlyName : Generic USB Flash Drive USB Device 
ParentIdPrefix: 8\&38f54984&0

We can notice that among removable media, a USB pen (serial number 7\&162a4319&0) was mounted on E: on 2008-07-20-T01:26:21. Although the confidential document was last accessed near after this USB storage was mounted, we cannot prove that the confidential document was copied onto a USB storage.

The fourth hypothesis we formulated is that a malware was downloaded from a blacklisted URL. However, only one of the installed programs (QQBubbleArena) can be malicious, since its signature does not belong to the NIST National Software Reference Library (NSRL) [7] that contains the files that are known to be good. The search for malicious files is supported by on of the menus of Sleuthkit and Autopsy.

We also inspected the web history (~1470 results) from the functionality provided by Sleuthkit and we did not identify any blacklisted URLs. Indeed, we can only conclude that hypotheses 2 and 3 are likely.

**Reactive Digital Investigation**

As a first step we modeled the crime scene associated with the digital forensic scenario. This is very similar to the one proposed for our previous case study. The general model also includes additional entities to represent urls, programs, more specific programs such as email clients (eClient) and browsers, email accounts and files hashes. Programs can also have a state as they can be Installed. We also included in the general model a function that identifies the signature of a file (MD5) and two untimed predicates that indicate if a program is not a malware (InNSRL) and if a url is blacklisted (IsBlack). The concrete model includes additional monitored events to represent sent emails (Sent_Email), sent emails with attachments (Sent_Attach), file searches (File_Search) and web requests (Web_Request) for urls.

As an investigation starting point we considered the time instant in which the confidential file was created. All programs that were installed before that time were initially considered as installed. At that point in time all the users were considered not logged and all devices not mounted. The generic hypotheses of a crime where modelled as shown in
Each hypothesis was customised depending on the elements represented in bold. For example, hypotheses 1 and 2 are customised depending on the users (Jean, Devon, and Administrator), on the email accounts found in the disk image (e.g., jean@m57.biz, alison@m57.biz, bob@m57.biz, carol@m57.biz, alex@m57.biz, tuckgorge@gmail.com) (6 accounts) and on the email clients (Outlook), for a total of 18 options. Hypothesis 3 is customised depending on the users (3 options), while hypothesis 4 is customised depending on the installed programs (133) and the browsers (Mozilla and IE), for a total of 266 options.

All the evidence that is possible to gather from the disk image was converted into an ordered sequence of events. Only those events necessary to satisfy a hypothesis are employed to verify each potential hypothesis. For example, only events Sys_Login, Sys_Logout, and Sent_Attach are used to satisfy hypothesis 1. Only events Sys_Install and Sys_Uninstall are used to satisfy hypothesis 4.

Sub-hypotheses were only evaluated for the concrete values on which their parent hypotheses are satisfied. For example, since hypothesis 2 is completely satisfied for user Jane, email jean@m57.biz, and email tuckgorge@gmail.com, hypotheses 2.a and 2.b were only evaluated for these values. However, hypotheses 2.a and 2.b are not satisfied.

Hypothesis 3 is 75% likely since only one of the monitored events is abducted (event Sys_Copy). Indeed the results we obtained are coherent with those obtained in a traditional digital investigation and are slightly more accurate. For hypothesis 4 no events were collected because no new program was installed after the confidential document (m57biz.xls) was created.

The concrete examples of potential hypotheses of a crime generated for the generic hypotheses are shown in the appendix.

Forensic requirements make possible to reduce the amount of evidence used to evaluate each hypothesis, since only the events necessary to evaluate the predicates in the hypothesis definition are considered. Figure 4 shows that the amount of evidence considered to evaluate each hypothesis in the reactive case was much smaller than the evidence collected in a traditional digital investigation. For the last hypothesis we did not get any evidence because no new program was installed after the confidential document was created.
Proactive & Reactive Digital Investigation

To evaluate proactive activities we simulated a sequence of possible monitored events that could have been collected proactively. We envisioned two possible situations. In the first case, we crafted an event snapshot that complies with the evidence collected from the disk image and shows that Jean was simulating to be the victim of a phishing attack (i.e., she was sending emails from her laptop as tuckorgre@gmail.com). We assumed that this evidence was not available in the disk image because it was concealed. The evaluation of the crafted event snapshot shows that hypothesis 2.a is completely satisfied. In the second case, we crafted an event snapshot demonstrating that Jean copied the confidential document onto a USB pen. In this case, hypothesis 3 is completely satisfied, since event Sys_Copy is considered in the evaluation of the hypothesis and does not have to be abducted. The aforementioned cases demonstrate the usefulness of proactive investigations when some evidence is concealed by an offender (case 1) or is ephemeral and cannot be retrieved from a disk image (case 2).

We estimated the effectiveness of proactive activities by focusing on case 1. Together with the crafted event snapshot for hypothesis 2.a, we also included 3 event snapshots that did not cause the satisfaction of any hypothesis. In addition, for the time period that is not covered by any crafted event snapshot we randomly triggered a new monitored event every hour, according to a uniform distribution.

Note that we considered that hypotheses 2 and 2.a are both associated with a suspicious event condition that a user is logged on the computer. Indeed events Sys_Login and Sys_Logout will be collected to check the start and stop condition and, additionally, events Sent_Email and Sent_Attach will be gathered during the full evidence collection for both hypotheses 2 and 2.a.

For this set of monitored events the Proactive Collection only preserves the snapshots related to the satisfaction of a suspicious events and avoids gathering all the events necessary to satisfy a hypothesis. The overhead of the Proactive Collection is only given by the events necessary to evaluate the start and stop conditions. Proactive activities provide non negligible savings in the amount of evidence analysed, compared to performing only reactive activities, as shown in Figure 5.

Note that this difference is given by the fact that the evidence that is analyzed when proactive activities are performed is only that included in the event snapshots. While the evidence analyzed when only reactive activities are performed is only that necessary to satisfy the predicates included in the hypotheses. The events snapshots crafted for the first and second case are shown in the Appendix.
Appendix

Potential Hypothesis Example for Hypothesis 1 (user=Jean, email=jean@m57.biz, eClient=MSOutlook)

sort boolean
sort integer
reified sort predicate
reified sort function

sort time: integer
sort offset: integer

reified sort fluent
reified sort event

predicate Happens(event,time)
predicate HoldsAt(fluent,time)
predicate ReleasedAt(fluent,time)
predicate Initiates(event,fluent,time)
predicate Terminates(event,fluent,time)
predicate Releases(event,fluent,time)
predicate Trajectory(fluent,time,fluent,offset)

****************************************************************************
;THIS PART SHOULD BE GENERATED BY THE DOCUMENT CrimeScene.e
;possible elements that can be available in the crime scene
sort user
sort user1
sort desktop
sort file
sort email
sort email1
sort hash
sort eclient

function MD5(file): hash

fluent IsStoredIn(file,desktop)

fluent Logged(user,desktop)

predicate HasEmail(user,email)
predicate HasUser(user,desktop)
predicate IsInternal(email)
predicate IsInstalled(eclient, desktop)

event Login(user,desktop)
event Logout(user,desktop)
event Sys_Login(user,desktop)
event Sys_Logout(user,desktop)
event Sent_Email(email, email, user, eclient, desktop)
event Sent_Email_Attach(email, email, user, file, eclient, desktop)
event Rec_Email(email, email, user,eclient, desktop)
event Rec_Email_Attach(email, email, user, file,eclient, desktop)

;entities modeled in the crime scene
user Jean
user Administrator
user Devon
eclient MSOutlook
email AlisonAtm57Dotbiz
email JeanAtm57Dotbiz
email BobAtm57Dotbiz
email CarolAtm57Dotbiz
email AlexAtm57Dotbiz
email GooglealertsDashNoreplyAtGmailDotCom
email AccountsDashNoreplyAtGmailDotCom
email NprDashAccountsAtNprDotCom
email AdminAtAssociatedcontentDotCom
email NewsLettersAtNDotNprDotCom
email AllsongsAtNDotNprDotCom
email TuckgorgeAtGmailDotCom

desktop M1
hash e23a4eb7f2562f53e88c9dca8b26a153
file Spreadsheet
file Attachment
MD5(Spreadsheet)= e23a4eb7f2562f53e88c9dca8b26a153.
MD5(Attachment)= e23a4eb7f2562f53e88c9dca8b26a153.

HoldsAt(IsStoredIn(Spreadsheet,M1),0).
!HoldsAt(Logged(Jean,M1),0).
HasEmail(Jean,JeanAtm57Dotbiz).
HasUser(Administrator,M1).
HasUser(Jean,M1).
HasUser(Devon,M1).
IsInternal(AlisonAtm57Dotbiz).
IsInternal(JeanAtm57Dotbiz).
IsInternal(BobAtm57Dotbiz).
IsInternal(CarolAtm57Dotbiz).
!IsInternal(AlexAtm57Dotbiz).
!IsInternal(GooglealertsDashNoreplyAtGmailDotCom).
!IsInternal(AccountsDashNoreplyAtGmailDotCom).
!IsInternal(NprDashAccountsAtNprDotCom).
!IsInternal(AdminAtAssociatedcontentDotCom).
!IsInternal(NewsLettersAtNDotNprDotCom).
!IsInternal(AllsongsATNDotNprDotCom).
!IsInternal(TuckgorgeAtGmailDotCom).

mutex Sys_Login, Sys_Logout, Sent_Email, Sent_Email_Attach, Rec_Email, Rec_Email_Attach

;************************************************************

;Domain Assumptions

[user,desktop,time] Initiates(Login(user,desktop),Logged(user,desktop),time).
[user,desktop,time] Terminates(Logout(user,desktop),Logged(user,desktop),time).

[user,desktop,time] Happens(Sys_Login(user,desktop),time) & ([user1]!HoldsAt(Logged(user1,desktop),time)) & HasUser(user,desktop) <->
Happens(Login(user,desktop),time).

[user,desktop,time] Happens(Sys_Logout(user,desktop),time) &
HoldsAt(Logged(user,desktop),time)  <->  Happens(Logout(user,desktop),time).

[user,desktop,time] HoldsAt(Logged(user,desktop),time)  
(user1!=user  ->  !HoldsAt(Logged(user1,desktop),time))).

; This part of the file is filled with the events coming from the monitoring infrastructure

[user] !HoldsAt(Logged(user,M1),0).

Happens(Sys_Login(Devon,M1),0).
[user] (user!=Devon)  ->  !Happens(Sys_Login(user,M1),0).
[user] !Happens(Sys_Logout(user,M1),0).
[email,email1,user,file]  !Happens(Sent_Email_Attach(email,email1,user,file,MSOutlook,M1),0).
[email,email1,user]  !Happens(Sent_Email(email,email1,user,MSOutlook,M1),0).
[email,email1,user]  !Happens(Rec_Email_Attach(email,email1,user,file,MSOutlook,M1),0).
[email,email1,user]  !Happens(Rec_Email(email,email1,user,MSOutlook,M1),0).

Happens(Sys_Logout(Devon,M1),1).
[user] (user!=Devon)  ->  !Happens(Sys_Logout(user,M1),1).
[user] !Happens(Sys_Login(user,M1),1).
[email,email1,user,file]
!Happens(Sent_Email_Attach(email,email1,user,file,MSOutlook,M1),1).
[email,email1,user]  !Happens(Sent_Email(email,email1,user,MSOutlook,M1),1).
[email,email1,user]  !Happens(Sent_Email(email,email1,user,MSOutlook,M1),1).
Happens(Rec_Email_Attach(email, email1, user, file, MSOutlook, M1), 1).
[email, email1, user] !Happens(Rec_Email(email, email1, user, MSOutlook, M1), 1).

Happens(Sys_Login(Jean, M1), 2).
[email, email1, user, file] !Happens(Sent_Email_Attach(email, email1, user, file, MSOutlook, M1), 2).
[email, email1, user] !Happens(Sent_Email(email, email1, user, MSOutlook, M1), 2).
[email, email1, user, file] !Happens(Rec_Email_Attach(email, email1, user, file, MSOutlook, M1), 2).
[email, email1, user] !Happens(Rec_Email(email, email1, user, MSOutlook, M1), 2).

[email, email1, user, file] !Happens(Sent_Email_Attach(email, email1, user, file, MSOutlook, M1), 3).
[email, email1, user, file] !Happens(Sent_Email_Attach(email, email1, user, file, MSOutlook, M1), 3).
[email, email1, user, file] !Happens(Sent_Email_Attach(email, email1, user, file, MSOutlook, M1), 3).
[email, email1, user, file] !Happens(Sent_Email_Attach(email, email1, user, file, MSOutlook, M1), 3).
[email, email1, user, file] !Happens(Sent_Email_Attach(email, email1, user, file, MSOutlook, M1), 3).
Happens(JeanAtm57Dotbiz, TuckgorgeAtGmailDotCom, Jean, Attachment, MSOutlook, M1), 3).
[email, email1, user] !Happens(Sent_Email(email, email1, user, MSOutlook, M1), 3).
[email, email1, user, file] !Happens(Rec_Email_Attach(email, email1, user, file, MSOutlook, M1), 3).
[email, email1, user] !Happens(Rec_Email(email, email1, user, MSOutlook, M1), 3).

[email, email1, user, file] !Happens(Sent_Email_Attach(email, email1, user, file, MSOutlook, M1), 4).
[email, email1, user, file] !Happens(Sent_Email_Attach(email, email1, user, file, MSOutlook, M1), 4).
[email, email1, user, file] !Happens(Sent_Email_Attach(email, email1, user, file, MSOutlook, M1), 4).
[email, email1, user, file] !Happens(Sent_Email_Attach(email, email1, user, file, MSOutlook, M1), 4).
[email, email1, user, file] !Happens(Sent_Email_Attach(email, email1, user, file, MSOutlook, M1), 4).
Happens(JeanAtm57Dotbiz, TuckgorgeAtGmailDotCom, Jean, Attachment, MSOutlook, M1), 4).
[email, email1, user] !Happens(Sent_Email(email, email1, user, MSOutlook, M1), 4).
[email, email1, user, file] !Happens(Rec_Email_Attach(email, email1, user, file, MSOutlook, M1), 4).
[email, email1, user] !Happens(Rec_Email(email, email1, user, MSOutlook, M1), 4).

Happens(Sys_Login(Administrator, M1), 5).
[email, email1, user, file] !Happens(Sys_Login(user, M1), 5).
[email, email1, user, file] !Happens(Sys_Login(user, M1), 5).
[email, email1, user, file] !Happens(Sys_Login(user, M1), 5).
[email, email1, user, file] !Happens(Sys_Login(user, M1), 5).
Happens(JeanAtm57Dotbiz, email, Jean, file, MSOutlook, M1), 5).
[email, email1, user] !Happens(Sys_Logout(user, M1), 3) & !Happens(Sys_LogOut(user, M1), 3).

;Jane mail + Jane user
{email, file, time} HoldsAt(Logged(Jean, M1), time) & Happens(Sent_Email_Attach(JeanAtm57Dotbiz, email, Jean, file, MSOutlook, M1), time) & IsInternal(email) & MD5(email) = MD5(Spreadsheet) & IsInternal(email).
Potential Hypothesis Example for Hypothesis 2 (user=Jean, email= jean@m57.biz, eClient=MSOutlook)

sort boolean
sort integer
reified sort predicate
reified sort function

sort time: integer
sort offset: integer

reified sort fluent
reified sort event

predicate Happens(event, time)
predicate HoldsAt(fluent, time)
predicate ReleasedAt(fluent, time)
predicate Initiates(event, fluent, time)
predicate Terminates(event, fluent, time)
predicate Releases(event, fluent, time)
predicate Trajectory(fluent, time, fluent, offset)

;***************************************************************************
;**THIS PART SHOULD BE GENERATED BY THE DOCUMENT CrimeScene.e**
;possible elements that can be available in the crime scene
sort user
sort user1
sort desktop
sort file
sort email
sort email1
sort hash
sort eclient

function MD5(file): hash
;possible predicates that can be expressed in the crime scene

;A file is stored in a machine
fluent IsStoredIn(file, desktop)

;A user is logged in to a machine
fluent Logged(user, desktop)

predicate HasEmail(user, email)
predicate HasUser(user, desktop)
predicate IsInternal(email)
predicate IsInstalled(eclient, desktop)

event Login(user, desktop)
event Logout(user, desktop)
event Sys_Login(user, desktop)
event Sys_Logout(user, desktop)
event Sent_Email(email, email, user, eclient, desktop)
event Sent_Email_Attach(email, email, user, file, eclient, desktop)
event Rec_Email(email, email, user, eclient, desktop)
event Rec_Email_Attach(email, email, user, file, eclient, desktop)

; entities modeled in the crime scene
user Jean
user Administrator
user Devon
eclient MSOutlook
e-mail AlisonAtm57Dotbiz
email JeanAtm57Dotbiz
e-mail BobAtm57Dotbiz
e-mail CarolAtm57Dotbiz
e-mail AlexAtm57Dotbiz
e-mail GooglealertsDashNoreplyAtGmailDotCom
e-mail AccountsDashNoreplyAtGmailDotCom
e-mail NprDashAccountsAtNprDotCom
e-mail AdminAtAssociatedcontentDotCom
e-mail NewsLettersAtNDotNprDotCom
e-mail AllsongsAtNDotNprDotCom
e-mail TuckgorgeAtGmailDotCom
desktop M1
hash e23a4eb7f2562f53e88c9dca8b26a153

file Spreadsheet
file Attachment
MD5(Spreadsheet)= e23a4eb7f2562f53e88c9dca8b26a153.
MD5(Attachment)= e23a4eb7f2562f53e88c9dca8b26a153.

HoldsAt(IsStoredIn(Spreadsheet,M1),0).
!HoldsAt(Logged(Jean,M1),0).

HasEmail(Jean,JeanAtm57Dotbiz).
HasUser(Administrator,M1).
HasUser(Jean,M1).
HasUser(Devon,M1).

IsInternal(AlisonAtm57Dotbiz).
IsInternal(JeanAtm57Dotbiz).
IsInternal(BobAtm57Dotbiz).
IsInternal(CarolAtm57Dotbiz).
!IsInternal(AlexAtm57Dotbiz).
!IsInternal(GooglealertsDashNoreplyAtGmailDotCom).
!IsInternal(AccountsDashNoreplyAtGmailDotCom).
!IsInternal(NprDashAccountsAtNprDotCom).
!IsInternal(AdminAtAssociatedcontentDotCom).
!IsInternal(NewsLettersAtNDotNprDotCom).
!IsInternal(AllsongsAtNDotNprDotCom).
!IsInternal(TuckgorgeAtGmailDotCom).
mutex Sys_Login, Sys_Logout, Sent_Email, Sent_Email_Attach, Rec_Email, Rec_Email_Attach

; Domain Assumptions

Initiates(Login(user,desktop),Logged(user,desktop),time).

Terminates(Logout(user,desktop),Logged(user,desktop),time).

Happens(Sys_Login(user,desktop),time) & ([user1]!=user -> !HoldsAt(Logged(user1,desktop),time)) & HasUser(user,desktop) <=>
Happens(Login(user,desktop),time).

Happens(Sys_Logout(user,desktop),time) &
HoldsAt(Logged(user,desktop),time) -> Happens(Logout(user,desktop),time).

HoldsAt(Logged(user,desktop),time) -> ([user1] (user1!=user -> !HoldsAt(Logged(user1,desktop),time))).

; This part of the file is filled with the events coming from the monitoring infrastructure

Happens(Sys_Login(Devon,M1),0).
Happens(Sys_Login(Devon,M1),0).
Happens(Sys_Login(user,M1),0).

Happens(Sent_Email_Attach(email,email1,user,file,MSOutlook,M1),0).
Happens(Sent_Email(email,email1,user,MSOutlook,M1),0).

Happens(Rec_Email_Attach(email,email1,user,file,MSOutlook,M1),0).
Happens(Rec_Email(email,email1,user,MSOutlook,M1),0).

Happens(Sys_Logout(Devon,M1),1).
Happens(Sys_Logout(user,M1),1).

Happens(Sent_Email_Attach(email,email1,user,file,MSOutlook,M1),1).
Happens(Sent_Email(email,email1,user,MSOutlook,M1),1).

Happens(Rec_Email_Attach(email,email1,user,file,MSOutlook,M1),1).
Happens(Rec_Email(email,email1,user,MSOutlook,M1),1).

Happens(Sys_Login(Jean,M1),2).

Happens(Sent_Email_Attach(email,email1,user,file,MSOutlook,M1),2).
Happens(Sent_Email(email,email1,user,MSOutlook,M1),2).

Happens(Rec_Email_Attach(email,email1,user,file,MSOutlook,M1),2).
Happens(Rec_Email(email,email1,user,MSOutlook,M1),2).

(email ! = JeanAtm57Dotbiz) ->
Happens(Sent_Email_Attach(email,email1,user,file,MSOutlook,M1),3).
[email,email1,user,file] (email1 != TuckgorgeAtGmailDotCom) ->
!Happens(Sent_Email_Attach(email,email1,user,file,MSOutlook,M1),3).
[email,email1,user,file] (user != Jean) ->
!Happens(Sent_Email_Attach(email,email1,user,file,MSOutlook,M1),3).
[email,email1,user,file] (file != Attachment) ->
!Happens(Sent_Email_Attach(email,email1,user,file,MSOutlook,M1),3).
Happens(Sent_Email_Attach(JeanAtm57Dotbiz,TuckgorgeAtGmailDotCom, Jean, Attachment,MSOutlook,M1),3).
[email,email1,user] !Happens(Sent_Email(email,email1,user,MSOutlook,M1),3).
[email,email1,user] !Happens(Rec_Email_Attach(email,email1,user,file,MSOutlook,M1),3).
[email,email1,user] !Happens(Rec_Email(email,email1,user,MSOutlook,M1),3).
[user] !Happens(Sys_Login(user,M1),3) & !Happens(Sys_Logout(user,M1),3).
Happens(Sys_Logout(Jean,M1),4).
[user] !Happens(Sys_Login(user,M1),4).
[email,email1,user,file] !Happens(Sent_Email_Attach(email,email1,user,file,MSOutlook,M1),4).
[email,email1,user] !Happens(Sent_Email(email,email1,user,MSOutlook,M1),4).
[email,email1,user] !Happens(Rec_Email_Attach(email,email1,user,file,MSOutlook,M1),4).
[email,email1,user] !Happens(Rec_Email(email,email1,user,MSOutlook,M1),4).
Happens(Sys_Login(Administrator,M1),5).
[user] !Happens(Sys_Logout(user,M1),5).
[email,email1,user,file] !Happens(Sent_Email_Attach(email,email1,user,file,MSOutlook,M1),5).
[email,email1,user] !Happens(Sent_Email(email,email1,user,MSOutlook,M1),5).
[email,email1,user] !Happens(Rec_Email_Attach(email,email1,user,file,MSOutlook,M1),5).
[email,email1,user] !Happens(Rec_Email(email,email1,user,MSOutlook,M1),5).
;Jane mail + Jane user
{email,file,time} HoldsAt(Logged(Jean,M1),time) &
Happens(Sent_Email_Attach(JeanAtm57Dotbiz, email, Jean, file, MSOutlook, M1),
time) & MD5(file) = MD5(Spreadsheet) & !IsInternal(email).

range time 0 6
range offset 1 1

Potential Hypothesis Example for Hypothesis 2.a (user=Jean, email=tuckgorge@gmail.com, eClient=MSOutlook)

sort boolean
sort integer
reified sort predicate
reified sort function

sort time: integer
sort offset: integer
reified sort fluent
reified sort event

predicate Happens(event,time)
predicate HoldsAt(fluent,time)
predicate ReleasedAt(fluent,time)
predicate Initiates(event,fluent,time)
predicate Terminates(event,fluent,time)
predicate Releases(event,fluent,time)
predicate Trajectory(fluent,time,fluent,offset)

;****************************************************************
;THIS PART SHOULD BE GENERATED BY THE DOCUMENT CrimeScene.e
;possible elements that can be available in the crime scene
sort time1
sort user
sort user1
sort desktop
sort file
sort file1
sort email
sort email1
sort hash
sort eclient

function MD5(file): hash

;possible predicates that can be expressed in the crime scene
fluent IsStoredIn(file,desktop)
fluent Logged(user,desktop)

predicate HasEmail(user,email)
predicate HasUser(user,desktop)
predicate IsInternal(email)
predicate IsInstalled(eclient, desktop)

event Login(user,desktop)
event Logout(user,desktop)
event Sys_Login(user,desktop)
event Sys_Logout(user,desktop)
event Sent_Email(email, email, user, eclient,  desktop)
event Sent_Email_Attach(email, email, user, file, eclient,  desktop)
event Rec_Email(email, email, user,eclient, desktop)
event Rec_Email_Attach(email, email, user, file,eclient, desktop)

;entities modeled in the crime scene
user Jean
user Administrator
user Devon
eclient MSOutlook
email AlisonAtm57Dotbiz
email JeanAtm57Dotbiz
email AlexAtm57Dotbiz
email TuckgorgeAtGmailDotCom
email BobAtm57Dotbiz
email CarolAtm57Dotbiz
derktop M1
hash e23a4eb7f2562f53e88c9dca8b26a153

file Spreadsheet
file Attachment

MD5(Spreadsheet)= e23a4eb7f2562f53e88c9dca8b26a153.
HoldsAt(IsStoredIn(Spreadsheet,M1),0).

HasEmail(Jean,JeanAtm57Dotbiz).

HasUser(Administrator,M1).
HasUser(Jean,M1).
HasUser(Devon,M1).

IsInternal(AlisonAtm57Dotbiz).
IsInternal(JeanAtm57Dotbiz).
!IsInternal(AlexAtm57Dotbiz).
!IsInternal(TuckgorgeAtGmailDotCom).

mutex Sys_Login, Sys_Logout, Sent_Email, Sent_Email_Attach, Rec_Email, Rec_Email_Attach

Initiates(Login(user,desktop),Logged(user,desktop),time).
Terminates(Logout(user,desktop),Logged(user,desktop),time).

Happens(Sys_Login(user,desktop),time) & ([user1]!HoldsAt(Logged(user1,desktop),time)) & HasUser(user,desktop) <-> Happens(Login(user,desktop),time).

Happens(Sys_Logout(user,desktop),time) & HoldsAt(Logged(user,desktop),time) <-> Happens(Logout(user,desktop),time).

HoldsAt(Logged(user,desktop),time) -> ([user1] (user1!=user -> !HoldsAt(Logged(user1,desktop),time))).

Domain Assumptions

Happens(Sys_Login(Jean,M1),0).

!Happens(Sys_Login(user,M1),0).
!Happens(Sys_Logout(user,M1),0).

!Happens(Sent_Email(email,email1,user,MSOutlook,M1),0).

!Happens(Sent_Email_Attach(email,email1,user,file,MSOutlook,M1),0).

!Happens(Rec_Email(email,email1,user,MSOutlook,M1),0).

!Happens(Sent_Email_Attach(email,email1,user,file,MSOutlook,M1),1).

Happens(Rec_Email(email1,email1,user,MSOutlook,M1),0).
!

Happens(Sent_Email(JeanAtm57Dotbiz,JeanAtm57Dotbiz,Jean,MSOutlook,M1),1).

!Happens(Sent_Email(email1,email1,user,MSOutlook,M1),1).

!Happens(Sent_Email(email1,email1,user,MSOutlook,M1),1).

!Happens(Sent_Email(email1,email1,user,MSOutlook,M1),1).

!Happens(Sent_Email_Attach(email1,email1,user,MSOutlook,M1),1).
!Happens(Rec_Email_Attach(email, email1, user, file, MSOutlook, M1), 1).
[email, email1, user] !Happens(Rec_Email(email, email1, user, MSOutlook, M1), 1).
[user] !Happens(Sys_Login(user, M1), 1).
[user] !Happens(Sys_Logout(user, M1), 1).

Happens(Sent_Email(JeanAtm57Dotbiz, JeanAtm57Dotbiz, Jean, MSOutlook, M1), 2).
[email, email1, user] email!=JeanAtm57Dotbiz ->
!Happens(Sent_Email(email, email1, user, MSOutlook, M1), 2).
[email, email1, user] email1!=JeanAtm57Dotbiz ->
!Happens(Sent_Email(email, email1, user, MSOutlook, M1), 2).
[email, email1, user] user!=Jean ->
!Happens(Sent_Email(email, email1, user, MSOutlook, M1), 2).
[email, email1, user, file]
Happens(Sent_Email_Attach(email, email1, user, file, MSOutlook, M1), 2).
[email, email1, user, file]
Happens(Rec_Email_Attach(email, email1, user, file, MSOutlook, M1), 2).
[email, email1, user] !Happens(Sys_Login(user, M1), 2).
[email, email1, user] !Happens(Sys_Logout(user, M1), 2).

Happens(Sent_Email(JeanAtm57Dotbiz, AlisonAtm57Dotbiz, Jean, MSOutlook, M1), 3).
[email, email1, user] email!=JeanAtm57Dotbiz ->
!Happens(Sent_Email(email, email1, user, MSOutlook, M1), 3).
[email, email1, user] email1!=AlisonAtm57Dotbiz ->
!Happens(Sent_Email(email, email1, user, MSOutlook, M1), 3).
[email, email1, user] user!=Jean ->
!Happens(Sent_Email(email, email1, user, MSOutlook, M1), 3).
[email, email1, user, file]
Happens(Sent_Email_Attach(email, email1, user, file, MSOutlook, M1), 3).
[email, email1, user, file]
Happens(Rec_Email_Attach(email, email1, user, file, MSOutlook, M1), 3).
[email, email1, user] !Happens(Sys_Login(user, M1), 3).
[email, email1, user] !Happens(Sys_Logout(user, M1), 3).

Happens(Sent_Email(JeanAtm57Dotbiz, AlisonAtm57Dotbiz, Jean, MSOutlook, M1), 4).
[email, email1, user] email!=JeanAtm57Dotbiz ->
!Happens(Sent_Email(email, email1, user, MSOutlook, M1), 4).
[email, email1, user] email1!=AlisonAtm57Dotbiz ->
!Happens(Sent_Email(email, email1, user, MSOutlook, M1), 4).
[email, email1, user] user!=Jean ->
!Happens(Sent_Email(email, email1, user, MSOutlook, M1), 4).
[email, email1, user, file]
Happens(Sent_Email_Attach(email, email1, user, file, MSOutlook, M1), 4).
[email, email1, user, file]
Happens(Rec_Email_Attach(email, email1, user, file, MSOutlook, M1), 4).
[email, email1, user] !Happens(Sys_Login(user, M1), 4).
[email, email1, user] !Happens(Sys_Logout(user, M1), 4).

Happens(Sent_Email(JeanAtm57Dotbiz, AlisonAtm57Dotbiz, Jean, MSOutlook, M1), 5).
[email, email1, user] email!=JeanAtm57Dotbiz ->
!Happens(Sent_Email(email, email1, user, MSOutlook, M1), 5).
[email, email1, user] email1!=AlisonAtm57Dotbiz ->
!Happens(Sent_Email(email, email1, user, MSOutlook, M1), 5).
[email, email1, user] user!=Jean ->
!Happens(Sent_Email(email, email1, user, MSOutlook, M1), 5).
Happens(Sent_Email_Attach(email,email1,user,file,MSOutlook,M1),5).

Happens(Rec_Email_Attach(email,email1,user,file,MSOutlook,M1),5).

Happens(Sys_Login(user, M1),5).

Happens(Sys_Logout(user, M1),5).

Happens(Sent_Email(JeanAtm57Dotbiz,AlisonAtm57Dotbiz,Jean,MSOutlook,M1),6).

Happens(Sent_Email(email,email1,user,MSOutlook,M1),6).

Happens(Sys_Login(user, M1),6).

Happens(Sys_Logout(user, M1),6).

Happens(Sent_Email(JeanAtm57Dotbiz,AlisonAtm57Dotbiz,Jean,MSOutlook,M1),7).

Happens(Sent_Email(email,email1,user,MSOutlook,M1),7).

Happens(Sys_Login(user, M1),7).

Happens(Sys_Logout(user, M1),7).

Happens(Sent_Email(JeanAtm57Dotbiz,AlisonAtm57Dotbiz,Jean,MSOutlook,M1),8).

Happens(Sent_Email(email,email1,user,MSOutlook,M1),8).

Happens(Sys_Login(user, M1),8).

Happens(Sys_Logout(user, M1),8).

Happens(Sent_Email(JeanAtm57Dotbiz,AlisonAtm57Dotbiz,Jean,MSOutlook,M1),9).

Happens(Sent_Email(email,email1,user,MSOutlook,M1),9).

Happens(Sys_Login(user, M1),9).

Happens(Sys_Logout(user, M1),9).
!Happens(Sent_Email(email1, jean, jean, MSOutlook, M1), 9).
!Happens(Sent_Email_Attach(email1, jean, jean, file, MSOutlook, M1), 9).
!Happens(Rec_Email_Attach(email1, jean, jean, file, MSOutlook, M1), 9).
Happens(Rec_Email(jean, jean, jean, MSOutlook, M1), 9).
Happens(Sys_Login(jean, M1), 9).
Happens(Sys_Logout(jean, M1), 9).

Happens(Sent_Email(email1, jean, jean, MSOutlook, M1), 10).
!Happens(Sent_Email(email1, jean, jean, MSOutlook, M1), 10).
!Happens(Sent_Email(email1, jean, jean, MSOutlook, M1), 10).
!Happens(Sent_Email(email1, jean, jean, MSOutlook, M1), 10).
Happens(Sys_Login(jean, M1), 10).
Happens(Sys_Logout(jean, M1), 10).

Happens(Sent_Email(email1, jean, jean, MSOutlook, M1), 11).
Happens(Sys_Login(jean, M1), 11).
Happens(Sys_Logout(jean, M1), 11).

Happens(Sent_Email(email1, jean, jean, MSOutlook, M1), 12).
!Happens(Sent_Email(email1, jean, jean, MSOutlook, M1), 12).
!Happens(Sent_Email(email1, jean, jean, MSOutlook, M1), 12).
!Happens(Sent_Email(email1, jean, jean, MSOutlook, M1), 12).
Happens(Sys_Login(jean, M1), 12).
Happens(Sys_Logout(jean, M1), 12).

Happens(Sent_Email(email1, jean, jean, MSOutlook, M1), 13).
!Happens(Sent_Email(email1, jean, jean, MSOutlook, M1), 13).
!Happens(Sent_Email(email1, jean, jean, MSOutlook, M1), 13).
!Happens(Sent_Email(email1, jean, jean, MSOutlook, M1), 13).
Happens(Sent_Email(email@email,input,email1,user,MSOutlook,M1),13).
[email,email1,user] user!=Jean ->
Happens(Sent_Email(email,email1,input,email,MSOutlook,M1),13).
[email,email1,input] !Happens(Sent_Email_Attach(email,email1,input,RSOutlook,M1),13).
[email,input,RSLogin] !Happens(Sys_Login(input,RSLogin1),13).
Happens(Sent_Email(JeanAtm57Dotbiz,AlisonAtm57Dotbiz,Jean,MSOutlook,M1),14).
[email,email1,input] email!=JeanAtm57Dotbiz ->
Happens(Sent_Email(email,email1,input,RSOutlook,M1),14).
[email,input] !Happens(Sent_Email_Attach(email,input,email1,RSOutlook,M1),14).
[email,input] !Happens(Rec_Email_Attach(email,input,email1,RSOutlook,M1),14).
[email,input] !Happens(Sys_Login(input,RSLogin1),14).
Happens(Sent_Email_Attach(JeanAtm57Dotbiz,TuckgoreAtGmailDotCom,Jean,Attachment,MSOutlook,M1),16).
[email,input] !Happens(Sent_Email(email,input,input,MSOutlook,M1),16).
[email,input] !Happens(Rec_Email_Attach(email,input,input,MSOutlook,M1),16).
[email,input] !Happens(Sys_Login(input,RSLogin1),16).
Happens(Sent_Email(JeanAtm57Dotbiz, TuckgorgeAtGmailDotCom, Jean, MSOutlook, M1), 17).
[Email, email1, user] email!=JeanAtm57Dotbiz ->
!Happens(Sent_Email(email, email1, user, MSOutlook, M1), 17).
[Email, email1, user] email1!=TuckgorgeAtGmailDotCom ->
!Happens(Sent_Email(email, email1, user, MSOutlook, M1), 17).
[Email, email1, user] user!=Jean ->
!Happens(Sent_Email(email, email1, user, MSOutlook, M1), 17).
[Email, email1, user, file]
!Happens(Rec_Email_Attach(email, email1, user, file, MSOutlook, M1), 17).
[Email, email1, user] !Happens(Rec_Email(email, email1, user, MSOutlook, M1), 17).
[Email, email1, user] !Happens(Sys_Login(user, M1), 17).
[Email, email1, user] !Happens(Sys_Logout(user, M1), 17).

Happens(Sent_Email(JeanAtm57Dotbiz, AlisonAtm57Dotbiz, Jean, MSOutlook, M1), 18).
[Email, email1, user] email!=JeanAtm57Dotbiz ->
!Happens(Sent_Email(email, email1, user, MSOutlook, M1), 18).
[Email, email1, user] email1!=AlisonAtm57Dotbiz ->
!Happens(Sent_Email(email, email1, user, MSOutlook, M1), 18).
[Email, email1, user] user!=Jean ->
!Happens(Sent_Email(email, email1, user, MSOutlook, M1), 18).
[Email, email1, user, file]
!Happens(Rec_Email_Attach(email, email1, user, file, MSOutlook, M1), 18).
[Email, email1, user] !Happens(Rec_Email(email, email1, user, MSOutlook, M1), 18).
[Email, email1, user] !Happens(Sys_Login(user, M1), 18).
[Email, email1, user] !Happens(Sys_Logout(user, M1), 18).

Happens(Sent_Email(JeanAtm57Dotbiz, BobAtm57Dotbiz, Jean, MSOutlook, M1), 20).
[Email, email1, user] email!=JeanAtm57Dotbiz ->
!Happens(Sent_Email(email, email1, user, MSOutlook, M1), 20).
[Email, email1, user] email1!=BobAtm57Dotbiz ->
!Happens(Sent_Email(email, email1, user, MSOutlook, M1), 20).
[Email, email1, user] user!=Jean ->
!Happens(Sent_Email(email, email1, user, MSOutlook, M1), 20).
[Email, email1, user, file]
!Happens(Rec_Email_Attach(email, email1, user, file, MSOutlook, M1), 20).
[Email, email1, user] !Happens(Rec_Email(email, email1, user, MSOutlook, M1), 20).
Happens(Sys_Login(user, M1), 20).
Happens(Sys_Logout(user, M1), 20).

Happens(Sent_Email(JeanAtm57Dotbiz, CarolAtm57Dotbiz, Jean, MSOutlook, M1), 21).
  [email, email1, user] email!=JeanAtm57Dotbiz ->
  Happens(Sent_Email(email, email1, user, MSOutlook, M1), 21).
  [email, email1, user] email1!=CarolAtm57Dotbiz ->
  Happens(Sent_Email(email, email1, user, MSOutlook, M1), 21).
  [email, email1, user] user!=Jean ->
  Happens(Sent_Email(email, email1, user, MSOutlook, M1), 21).
  [email, email1, user, file]
  Happens(Sent_Email_Attach(email, email1, user, file, MSOutlook, M1), 21).
  [email, email1, user, file]
  Happens(Rec_Email_Attach(email, email1, user, file, MSOutlook, M1), 21).
  [email, email1, user] !Happens(Rec_Email(email, email1, user, MSOutlook, M1), 21).
[user] !Happens(Sys_Login(user, M1), 21).
[user] !Happens(Sys_Logout(user, M1), 21).

Happens(Sent_Email(JeanAtm57Dotbiz, BobAtm57Dotbiz, Jean, MSOutlook, M1), 22).
  [email, email1, user] email!=JeanAtm57Dotbiz ->
  Happens(Sent_Email(email, email1, user, MSOutlook, M1), 22).
  [email, email1, user] email1!=BobAtm57Dotbiz ->
  Happens(Sent_Email(email, email1, user, MSOutlook, M1), 22).
  [email, email1, user] user!=Jean ->
  Happens(Sent_Email(email, email1, user, MSOutlook, M1), 22).
  [email, email1, user, file]
  Happens(Sent_Email_Attach(email, email1, user, file, MSOutlook, M1), 22).
  [email, email1, user, file]
  Happens(Rec_Email_Attach(email, email1, user, file, MSOutlook, M1), 22).
  [email, email1, user] !Happens(Rec_Email(email, email1, user, MSOutlook, M1), 22).
[user] !Happens(Sys_Login(user, M1), 22).
[user] !Happens(Sys_Logout(user, M1), 22).

Happens(Sys_Logout(Jean, M1), 23).
[user] (user != Jean) -> !Happens(Sys_Logout(user, M1), 23).
[user] !Happens(Sys_Login(user, M1), 23).
  [email, email1, user] !Happens(Sent_Email(email, email1, user, MSOutlook, M1), 23).
  [email, email1, user, file]
  Happens(Sent_Email_Attach(email, email1, user, file, MSOutlook, M1), 23).
  [email, email1, user, file]
  Happens(Rec_Email_Attach(email, email1, user, file, MSOutlook, M1), 23).
  [email, email1, user] !Happens(Rec_Email(email, email1, user, MSOutlook, M1), 23).

Happens(Sys_Login(Administrator, M1), 24).
[user] (user != Administrator) -> !Happens(Sys_Login(user, M1), 24).
[user] !Happens(Sys_Logout(user, M1), 24).
  [email, email1, user] !Happens(Sent_Email(email, email1, user, MSOutlook, M1), 24).
  [email, email1, user, file]
  Happens(Sent_Email_Attach(email, email1, user, file, MSOutlook, M1), 24).
  [email, email1, user, file]
  Happens(Rec_Email_Attach(email, email1, user, file, MSOutlook, M1), 24).
  [email, email1, user] !Happens(Rec_Email(email, email1, user, MSOutlook, M1), 24).

Happens(Sent_Email(JeanAtm57Dotbiz, CarolAtm57Dotbiz, Jean, MSOutlook, M1), 25).
  [email, email1, user] email!=JeanAtm57Dotbiz ->
!Happens(Sent_Email(email, email1, user, MSOutlook, M1), 25).
[email, email1, user] email1!-CarolAtm57Dotbiz ->
!Happens(Sent_Email(email, email1, user, MSOutlook, M1), 25).
[email, email1, user] user!=Jean ->
!Happens(Sent_Email(email, email1, user, MSOutlook, M1), 25).
[email, email1, user, file]
!Happens(Sent_Email_Attach(email, email1, user, file, MSOutlook, M1), 25).
[email, email1, user, file]
!Happens(Rec_Email_Attach(email, email1, user, file, MSOutlook, M1), 25).
[email, email1, user] !Happens(Rec_Email(email, email1, user, MSOutlook, M1), 25).
[user] !Happens(Sys_Login(user, M1), 25).
[user] !Happens(Sys_Logout(user, M1), 25).
Happens(Sent_Email(JeanAtm57Dotbiz, BobAtm57Dotbiz, Jean, MSOutlook, M1), 26).
[email, email1, user] email!=JeanAtm57Dotbiz ->
!Happens(Sent_Email(email, email1, user, MSOutlook, M1), 26).
[email, email1, user] email1!=BobAtm57Dotbiz ->
!Happens(Sent_Email(email, email1, user, MSOutlook, M1), 26).
[email, email1, user] user!=Jean ->
!Happens(Sent_Email(email, email1, user, MSOutlook, M1), 26).
[email, email1, user, file]
!Happens(Sent_Email_Attach(email, email1, user, file, MSOutlook, M1), 26).
[email, email1, user, file]
!Happens(Rec_Email_Attach(email, email1, user, file, MSOutlook, M1), 26).
[email, email1, user] !Happens(Rec_Email(email, email1, user, MSOutlook, M1), 26).
[user] !Happens(Sys_Login(user, M1), 26).
[user] !Happens(Sys_Logout(user, M1), 26).

{email, file, time} HoldsAt(Logged(Jean, M1), time) & (Happens(Sent_Email(TuckgorgeAtGmailDotCom, email, Jean, MSOutlook, M1), time) | Happens(Sent_Email_Attach(TuckgorgeAtGmailDotCom, email, Jean, file, MSOutlook, M1), time)).

range time 0 27
range offset 1 1

Potential Hypothesis Example for Hypothesis 3 (user=Jean)

class boolean

class integer

reified class predicate

reified class function

class time: integer
class offset: integer

reified class fluent

reified class event

predicate Happens(event, time)
predicate HoldsAt(fluent, time)
predicate ReleasedAt(fluent, time)
predicate Initiates(event, fluent, time)
predicate Terminates(event, fluent, time)
predicate Releases(event, fluent, time)
predicate Trajectory(fluent, time, fluent, offset)
sort user
sort user1
sort desktop
sort file
sort storage
fluent IsStoredIn(file,desktop)
fluent Logged(user,desktop)
fluent Mounted(storage,desktop)
predicate HasUser(user,desktop)

event Login(user,desktop)
event Logout(user,desktop)
event Sys_Login(user,desktop)
event Sys_Logout(user,desktop)
event Mount(storage,desktop)
event Unmount(storage,desktop)
event Sys_Mount(storage,desktop)
event Sys_Unmount(storage,desktop)
event Sys_Accessed(file,desktop)

;entities modeled in the crime scene
user Jean
user Administrator
user Devon

desktop M1
file Spreadsheet
storage 7&162a431960

HasUser(Administrator,M1).
HasUser(Jean,M1).
HasUser(Devon,M1).

HoldsAt(IsStoredIn(Spreadsheet,M1),0).

mutex Sys_Login, Sys_Logout,Sys_Mount, Sys_Unmount, Sys_Accessed

; Domain Assumptions

;Predicate Logged
[user,desktop,time]
Initiates(Login(user,desktop),Logged(user,desktop),time).
[user,desktop,time]
Terminates(Logout(user,desktop),Logged(user,desktop),time).

[user,desktop,time] Happens(Sys_Login(user,desktop),time) &
([user1]!HoldsAt(Logged(user1,desktop),time)) & HasUser(user,desktop) <=>
Happens(Login(user,desktop),time).
[user, desktop, time] Happens(Sys_Logout(user, desktop), time) &
HoldsAt(Logged(user, desktop), time) <-> Happens(Logout(user, desktop), time).

[user, desktop, time] HoldsAt(Logged(user, desktop), time) -> ([user1]
(user1! user -> !HoldsAt(Logged(user1, desktop), time))).

; Predicate Mounted
[storage, desktop, time]
Initiates(Mount(storage, desktop), Mounted(storage, desktop), time).
[storage, desktop, time]
Terminates(Unmount(storage, desktop), Mounted(storage, desktop), time).

[storage, desktop, time] Happens(Sys_Mount(storage, desktop), time) &
!HoldsAt(Mounted(storage, desktop), time) <->
Happens(Mount(storage, desktop), time).
[storage, desktop, time] Happens(Sys_Unmount(storage, desktop), time) &
HoldsAt(Mounted(storage, desktop), time) <->
Happens(Unmount(storage, desktop), time).

;***************************************************************************
**********
; This part of the file is filled with the events coming from the monitoring infrastructure

[user] !HoldsAt(Logged(user, M1), 0).
[storage] !HoldsAt(Mounted(storage, M1), 0).
Happens(Sys_Login(Devon, M1), 0).
[user] (user!=Devon) -> !Happens(Sys_Login(user, M1), 0).
[user] !Happens(Sys_Logout(user, M1), 0).
[storage] !Happens(Sys_Mount(storage, M1), 0) &
!Happens(Sys_Unmount(storage, M1), 0).
Happens(Sys_Logout(Devon, M1), 1).
[user] (user!=Devon) -> !Happens(Sys_Logout(user, M1), 1).
[user] !Happens(Sys_Login(user, M1), 1).
[storage] !Happens(Sys_Mount(storage, M1), 1) &
!Happens(Sys_Unmount(storage, M1), 1).
Happens(Sys_Login(Jean, M1), 2).
[user] (user!=Jean) -> !Happens(Sys_Login(user, M1), 2).
[user] !Happens(Sys_Logout(user, M1), 2).
[storage] !Happens(Sys_Mount(storage, M1), 2) &
!Happens(Sys_Unmount(storage, M1), 2).
[file] !Happens(Sys_Accessed(file, M1), 2).
[user] !Happens(Sys_Login(user, M1), 3) & !Happens(Sys_Logout(user, M1), 3).
[file] !Happens(Sys_Accessed(file, M1), 3).
Happens(Sys_Mount(7&162a4319&0, M1), 3).
[storage] !Happens(Sys_Unmount(storage, M1), 3).
Happens(Sys_Accessed(Spreadsheet, M1), 4).
[user] !Happens(Sys_Login(user, M1), 4) & !Happens(Sys_Logout(user, M1), 4).
[storage] !Happens(Sys_Mount(storage, M1), 4) &
!Happens(Sys_Unmount(storage, M1), 4).
Happens(Sys_Logout(Jean, M1), 5).
[user] (user!=Jean) -> !Happens(Sys_Logout(user, M1), 5).
Happens(Sys_Login(user,M1),5).
Happens(Sys_Mount(storage,M1),5) &
Happens(Sys_Unmount(storage,M1),5).
Happens(Sys_Accessed(file,M1),5).
Happens(Sys_Login(Administrator,M1),6).
Happens(Sys_Accessed(file,M1),6).

Happens(Sys_Login(user,M1),6).
Happens(Sys_Logout(user,M1),6).
Happens(Sys_Mount(storage,M1),6) &
Happens(Sys_Unmount(storage,M1),6).
Happens(Sys_Accessed(file,M1),6).

{storage,time} HoldsAt(Logged(Jean,M1),time) &
Happens(Sys_Accessed(Spreadsheet, M1), time) & Mounted(storage,M1) &
HoldsAt(IsStoredIn(Spreadsheet,M1),time).

range time 0 7
range offset 1 1

Potential Hypothesis Example for Hypothesis 4 (program = QQBubbleArena)

sort boolean
sort integer
reified sort predicate
reified sort function

sort time: integer
sort offset: integer

reified sort fluent
reified sort event

predicate Happens(event,time)
predicate HoldsAt(fluent,time)
predicate ReleasedAt(fluent,time)
predicate Initiates(event,fluent,time)
predicate Terminates(event,fluent,time)
predicate Releases(event,fluent,time)
predicate Trajectory(fluent,time,fluent,offset)

;*************************************************************************
;*************************************************************************
;THIS PART SHOULD BE GENERATED BY THE DOCUMENT CrimeScene.e
;possible elements that can be available in the crime scene
sort desktop
sort file
sort program : file
sort hash

function MD5(file): hash
predicate InNSRL(hash)

fluent IsStoredIn(file,desktop)
fluent Installed(program,desktop)
event Sys_Accessed(file,desktop)
event Sys_Install(program,desktop)
event Sys_Uninstall(program,desktop)
event Install(program,desktop)
event Uninstall(program,desktop)

desktop M1
hash 6f03dd8528212edfdab18791116056

program QQBubbleArena
program VMWare
program MicrosoftVisualC
program AimPlugin
program QQGames
program AIMTunes
program AIMToolbar
program ViewPointMediaPlayer
program AIM6
program AOLOCP_Y
program AOLDiagnostics_N
program WebFldrsXP
program MPlayer2
program Firefox
program ShockWaveFlash
program Office
program UpdateForWinXP
program SecurityUpdateForWinXP
program WIC
program WinGenAdValTool
program AdobeFlashPlayerPlugin
program DXM_Runtime
program PCHealth
program AddressBook
program DirectAnimation
program ICW
program NetMeeting
program OutlookExpress
program DirectDrawEx
program Fontcore
program IE40
program IE4Data
program IESBAKEX
program MobileOptionPack
program SchedulingAgent
program ConnectionManager
program AdobeFlashPlayerActiveX

file Spreadsheet

MD5(QQBubbleArena) = 6f03dd8528212edfdab18791116056.

!InNSRL(6f03dd8528212edfdab18791116056).
HoldsAt(IsStoredIn(Spreadsheet,M1),0).
mutex Sys_Accessed, Sys_Install, Sys_Uninstall

***************************************************************************
Domain Assumptions

[program,desktop,time]
Initiates(Install(program,desktop), Installed(program,desktop),time).
[program,desktop,time]
Terminates(Uninstall(program,desktop), Installed(program,desktop),time).

[program,desktop,time] Happens(Sys_Install(program,desktop),time) &
!HoldsAt(Installed(program,desktop),time) <->
Happens(Install(program,desktop),time).
[program,desktop,time] Happens(Sys_Uninstall(program,desktop),time) &
HoldsAt(Installed(program,desktop),time) <->
Happens(Uninstall(program,desktop),time).

***************************************************************************
**********
; This part of the file should be filled with the events coming from the
monitoring infrastructure
; necessary to verify the hypotheses
; NOTE THAT ONLY THE EVENTS THAT ARE SUPPOSED TO COME FROM THE MONITORING
INFRASTRUCTURE
; ARE PUT ON THIS SECTION

; We assume that the evidence that for the evidence
; that is not proactively collected nothing happens

HoldsAt(Installed(VMWare,M1),0).
HoldsAt(Installed(MicrosoftVisualC,M1),0).
HoldsAt(Installed(QQBubbleArena,M1),0).
HoldsAt(Installed(AimPlugin,M1),0).
HoldsAt(Installed(QQGames,M1),0).
HoldsAt(Installed(AIMTunes,M1),0).
HoldsAt(Installed(AIMToolbar,M1),0).
HoldsAt(Installed(ViewPointMediaPlayer,M1),0).
HoldsAt(Installed(AIM6,M1),0).
HoldsAt(Installed(AOLOCP_Y,M1),0).
HoldsAt(Installed(AOLDiagnostics_N,M1),0).
HoldsAt(Installed(WebFldrsXP,M1),0).
HoldsAt(Installed(MPlayer2,M1),0).
HoldsAt(Installed(Firefox,M1),0).
HoldsAt(Installed(ShockWaveFlash,M1),0).
HoldsAt(Installed(Office,M1),0).
HoldsAt(Installed(UpdateForWinXP,M1),0).
HoldsAt(Installed(SecurityUpdateForWinXP,M1),0).
HoldsAt(Installed(WIC,M1),0).
HoldsAt(Installed(AdoaeFlashPlayerPlugin,M1),0).
HoldsAt(Installed(DXM_Runtime,M1),0).
HoldsAt(Installed(PCHealth,M1),0).
HoldsAt(Installed(AddressBook,M1),0).
HoldsAt(Installed(DirectAnimation,M1),0).
HoldsAt(Installed(ICW,M1),0).
HoldsAt(Installed(NetMeeting,M1),0).
HoldsAt(Installed(OutlookExpress,M1),0).
HoldsAt(Installed(DirectDrawEx,M1),0).
HoldsAt(Installed(Fontcore,M1),0).
HoldsAt(Installed(IE40,M1),0).
HoldsAt(Installed(IE4Data,M1),0).
HoldsAt(Installed(IESBAKEX,M1),0).
HoldsAt(Installed(MobileOptionPack,M1),0).
HoldsAt(Installed(SchedulingAgent,M1),0).
HoldsAt(Installed(ConnectionManager,M1),0).
HoldsAt(Installed(AdobeFlashPlayerActiveX,M1),0).

...{hash,time} HoldsAt(Installed(QQBubbleArena,M1),time) & hash = MD5(QQBubbleArena) & InNSRL(hash).

range  time 0 1
range  offset 1 1

Events Snapshot Crafted for the Evaluation of Proactive Activities – Case 1

Snapshots generated for hypotheses 1

Snapshot 1
Sys_Login(Jean,M1, 6/07/2008, 6:10:56)
Sent_Email(Jean@m57.biz, Jean@m57.biz, Jean, Outlook, M1) 6/07/2008 8:39:00)
Sent_Email(Jean@m57.biz, Jean@m57.biz, Jean, Outlook, M1), 6/7/2008 8:48:00)
Sent_Email(Jean@m57.biz, Jean@m57.biz, Jean, Outlook, M1), 6/07/2008 8:55:00).
Sys_Logout(Jean, M1, 06/07/2008, 09:07:54)

Snapshot 2
Sys_Login(Jean,M1, 7/07/2008, 5:40:32)
Sent_Email(Jean@m57.biz, Alison@m57.biz, Jean, Outlook,M1), 7/7/2008 6:24:00)
Sent_Email(Jean@m57.biz, Alison@m57.biz, Jean, Outlook,M1), 7 July 6:25:00)
Sent_Email(Jean@m57.biz, Alison@m57.biz, Jean, Outlook,M1), 7 July 6:25:00)
Sent_Email(Jean@m57.biz, Alison@m57.biz, Jean, Outlook,M1), 7/7/2008 6:26:00)
Sent_Email(Jean@m57.biz, Alison@m57.biz, Jean, Outlook,M1), 7/7/2008 6:26:00)
Sent_Email(Jean@m57.biz, Alison@m57.biz, Jean, Outlook,M1), 7/7/2008 6:27:00)
Sys_Logout(Jean, M1, 7/07/2008, 8:35:32)

Snapshot 3
Sys_Login(Jean, M1, 20/07/2008 00:00:41)
Sent_Email_Attach(Jean@m57.biz, Tuckgorge@Gmail.Com, Jean, m57biz, Outlook,M1),20/07/2008 2:28:00).
Sys_Logout(Jean, M1, 21/07/2008 01:06:43)

Snapshots generated for hypothesis 2

Snapshot 1
Sys_Login(Jean,M1, 6/07/2008, 6:10:56)
Sent_Email(Jean@m57.biz, Jean@m57.biz, Jean, Outlook, M1) 6/07/2008 8:39:00)
Sent_Email(Jean@m57.biz, Jean@m57.biz, Jean, Outlook, M1), 6/7/2008 8:48:00)
Sent_Email(Jean@m57.biz, Alison@m57.biz, Jean, Outlook, M1), 6/07/2008 8:55:00).
SysLogout(Jean, M1, 06/07/2008, 09:07:54)

Snapshot 2
Sys_Login(Jean,M1, 7/07/2008, 5:40:32)
Sent_Email(Jean@m57.biz, Alison@m57.biz, Jean, Outlook, M1), 7/7/2008 6:24:00)
Sent_Email(Jean@m57.biz, Alison@m57.biz, Jean, Outlook, M1), 7 July 6:25:00)
Sent_Email(Jean@m57.biz, Alison@m57.biz, Jean, Outlook, M1), 7/7/2008 6:26:00)
Sent_Email(Jean@m57.biz, Alison@m57.biz, Jean, Outlook, M1), 7/7/2008 6:26:00)
Sent_Email(Jean@m57.biz, Alison@m57.biz, Jean, Outlook, M1), 7/7/2008 6:27:00)
SysLogout(Jean, M1, 7/07/2008, 8:35:32)

Snapshot 3
Sys_Login(Jean, M1, 20/07/2008 00:00:41)
Sent_Email_Attach(Jean@m57.biz, Tuckgorge@Gmail.Com, Jean, m57biz, Outlook, M1), 20/07/2008 2:28:00).
SysLogout(Jean, M1, 21/07/2008 01:06:43)

Snapshots generated for hypothesis 2.a

Snapshot 1
Sys_Login(Jean,M1, 6/07/2008, 6:10:56)
Sent_Email(Jean@m57.biz, Jean@m57.biz, Jean, Outlook, M1) 6/07/2008 8:39:00)
Sent_Email(Jean@m57.biz, Jean@m57.biz, Jean, Outlook, M1), 6/7/2008 8:48:00)
Sent_Email(Jean@m57.biz, Alison@m57.biz, Jean, Outlook, M1), 6/07/2008 8:55:00).
SysLogout(Jean, M1, 06/07/2008, 09:07:54)

Snapshot 2
Sys_Login(Jean,M1, 7/07/2008, 5:40:32)
Sent_Email(Jean@m57.biz, Alison@m57.biz, Jean, Outlook, M1), 7/7/2008 6:24:00)
Sent_Email(Jean@m57.biz, Alison@m57.biz, Jean, Outlook, M1), 7 July 6:25:00)
Sent_Email(Jean@m57.biz, Alison@m57.biz, Jean, Outlook, M1), 7/7/2008 6:26:00)
Sent_Email(Jean@m57.biz, Alison@m57.biz, Jean, Outlook, M1), 7/7/2008 6:26:00)
Sent_Email(Jean@m57.biz, Alison@m57.biz, Jean, Outlook, M1), 7/7/2008 6:27:00)
SysLogout(Jean, M1, 7/07/2008, 8:35:32)

Snapshot 3
Sys_Login(Jean, M1, 20/07/2008 00:00:41)
Sent_Email(Jean@m57.biz, Alison@m57.biz, Jean, Outlook, M1), 20/7/2008 00:31:00).
Sent_Email(Alex@m57.biz, Jean@m57.biz, Jean, Outlook, M1), 20/7/2008 00:31:30).
Sent_Email(Alex@m57.biz, Jean@m57.biz, Jean, Outlook, M1), 20/07/2008 00:31:45).
Sent_Email(Alex@m57.biz, Jean@m57.biz, Jean, Outlook, M1), 20/07/2008 00:32:00).
Sent_Email(Alex@m57.biz, Jean@m57.biz, Jean, Outlook, M1), 20/07/2008 00:32:16).
Sent_Email(Alex@m57.biz, Jean@m57.biz, Jean, Outlook, M1), 20/07/2008 00:32:27).
Sent_Email(Alex@m57.biz, Jean@m57.biz, Jean, Outlook, M1), 20/07/2008 00:32:46).
Sent_Email(Alex@m57.biz, Alison@m57.biz, Jean, Outlook, M1), 20/07/2008 00:44:00).
Sent_Email(Alex@m57.biz, Alison@m57.biz, Jean, Outlook, M1), 20/07/2008 00:44:00).
Sent_Email(Jean@m57.biz, Alison@m57.biz, Jean, Outlook, M1), 20/07/2008 00:44:00.
Sent_Email(Jean@m57.biz, Alison@m57.biz, Jean, Outlook, M1), 20/07/2008 00:46:00.
Sent_Email(Jean@m57.biz, Alison@m57.biz, Jean, Outlook, M1), 20/07/2008 00:46:00.
Sent_Email(Tuckgorge@Gmail.Com, Jean@m57.biz, Jean, Outlook, M1), 20/07/2008 02:22:00.
Sent_Email_Attach(Jean@m57.biz, Tuckgorge@Gmail.Com, Jean, m57.biz, Outlook, M1), 20/07/2008 02:28:00.
Sent_Email(Tuckgorge@Gmail.Com, Jean@m57.biz, Jean, Outlook, M1), 20/07/2008 06:02:50.
Sent_Email(Jean@m57.biz, Tuckgorge@Gmail.Com, Jean, Outlook, M1), 20/07/2008 06:04:00.
Sent_Email(Jean@m57.biz, Alison@m57.biz, Jean, Outlook, M1), 21/07/2008 00:51:00.
Sent_Email(Jean@m57.biz, Alison@m57.biz, Jean, Outlook, M1), 21/07/2008 00:57:00.
Sent_Email(Jean@m57.biz, Bob@m57.biz, Jean, Outlook, M1), 21/07/2008 00:58:00.
Sent_Email(Jean@m57.biz, Carol@m57.biz, Jean, Outlook, M1), 21/07/2008 01:01:00.
Sent_Email(Jean@m57.biz, Bob@m57.biz, Jean, Outlook, M1), 21/07/2008 01:04:00.
Sys_LogOut(Jean, M1, 21/07/2008 01:06:43)

Snapshot 4
Sys_Login(Administrator, M1, 21/07/2008 01:22:18)
Sent_Email(Jean@m57.biz, Carol@m57.biz, Jean, Outlook, M1), 21/07/2008 01:45:00.
Sent_Email(Jean@m57.biz, Bob@m57.biz, Jean, Outlook, M1), 21/07/2008 01:46:00.

Snapshots generated for hypothesis 3

Snapshot 1
Sys_Login(Jean, M1, 6/07/2008, 6:10:56)
Sys_LogOut(Jean, M1, 06/07/2008, 09:07:54)
Sys_Login(Jean, M1, 7/07/2008, 5:40:32)
Sys_LogOut(Jean, M1, 7/07/2008, 8:35:32)
Sys_Login(Jean, M1, 7/07/2008, 16:55:00)
Sys_LogOut(Jean, M1, 8/07/2008, 2:10:17)
Sys_Login(Jean, M1, 8/07/2008, 17:10:05)
Sys_LogOut(Jean, M1, 9/07/2008, 2:12:57)
Sys_Login(Jean, M1, 9/07/2008, 16:37:33)
Sys_LogOut(Jean, M1, 10/07/2008, 2:00:17)
Sys_Login(Jean, M1, 10/07/2008, 16:59:03)
Sys_LogOut(Jean, M1, 11/07/2008, 1:56:19)
Sys_Login(Jean, M1, 11/07/2008, 16:02:20)
Sys_LogOut(Jean, M1, 12/07/2008, 2:07:21)
Sys_Login(Devon, M1, 12/07/2008 03:02:47)
Sys_LogOut(Devon, M1, 12/07/2008 05:23:31)
Sys_Login(Jean, M1, 14/07/2008, 16:48:02)
Sys_LogOut(Jean, M1, 15/07/2008, 3:29:59)
Sys_Login(Jean, M1, 15/07/2008, 17:01:32)
Sys_LogOut(Jean, M1, 16/07/2008, 3:10:59)
Sys_Login(Jean, M1, 16/07/2008, 17:34:03)
Sys_LogOut(Jean, M1, 17/07/2008, 3:05:06)
Sys_Login(Jean, M1, 17/07/2008, 16:12:22)
Sys_LogOut(Jean, M1, 18/07/2008, 2:38:13)
Sys_Login(Jean, M1, 18/07/2008, 16:15:14)
Sys Logout(Jean, M1, 19/07/2008, 1:35:06)
Sys_Login(Jean, M1, 20/07/2008 00:00:41)
Sys_Open(m57biz, 20/07/2008 01:05:21)
Sys_Modify(m57biz, 20/07/2008 01:05:21)
Sys_Modify(m57biz, 20/07/2008 01:15:21)
Sys_Mount(7&162a431980, E, M1, 20/07/2008 01:26:21)
Sys_Modify(m57biz, 20/07/2008 01:50:21)
Sys_Accessed(m57biz, 20/07/2008 01:22:00)
Sys_Unmount(7&162a431980,E, M1, 20/07/2008 01:26:21)

Events Snapshot Crafted for the Evaluation of Proactive Activities – Case 2

Snapshots generated for hypotheses 1

Snapshot 1
Sys_Login(Jean, M1, 6/07/2008, 6:10:56)
Sent_Email(Jean@m57.biz, Jean@m57.biz, Jean, Outlook, M1) 6/07/2008 8:39:00)
Sent_Email(Jean@m57.biz, Jean@m57.biz, Jean, Outlook, M1), 6/7/2008 8:48:00)
Sent_Email(Jean@m57.biz, Alison@m57.biz, Jean, Outlook, M1), 6/07/2008 8:55:00).
SysLogout(Jean, M1, 06/07/2008, 09:07:54)

Snapshot 2
Sys_Login(Jean, M1, 7/07/2008, 5:40:32)
Sent_Email(Jean@m57.biz, Alison@m57.biz, Jean, Outlook, M1), 7/7/2008 6:24:00)
Sent_Email(Jean@m57.biz, Alison@m57.biz, Jean, Outlook, M1), 7 July 6:25:00)
Sent_Email(Jean@m57.biz, Alison@m57.biz, Jean, Outlook, M1), 7 July 6:25:00)
Sent_Email(Jean@m57.biz, Alison@m57.biz, Jean, Outlook, M1), 7/7/2008 6:26:00)
Sent_Email(Jean@m57.biz, Alison@m57.biz, Jean, Outlook, M1), 7/7/2008 6:27:00)
SysLogout(Jean, M1, 7/07/2008, 8:35:32)

Snapshot 3
Sys_Login(Jean, M1, 20/07/2008 00:00:41)
Sent_Email_Attach(Jean@m57.biz, Tuckgorge@Gmail.Com, Jean, m57biz, Outlook, M1), 20/07/2008 2:28:00).
SysLogout(Jean, M1, 20/07/2008 01:06:43)

Snapshots generated for hypotheses 2

Snapshot 1
Sys_Login(Jean,M1, 6/07/2008, 6:10:56)
Sent_Email(Jean@m57.biz, Jean@m57.biz, Jean, Outlook, M1) 6/07/2008 8:39:00)
Sent_Email(Jean@m57.biz, Jean@m57.biz, Jean, Outlook, M1), 6/7/2008 8:48:00)
Sent_Email(Jean@m57.biz, Alison@m57.biz, Jean, Outlook, M1), 6/07/2008 8:55:00).
Sys_logout(Jean, M1, 06/07/2008, 09:07:54)

Snapshot 2
Sys_Login(Jean,M1, 7/07/2008, 5:40:32)
Sent_Email(Jean@m57.biz, Alison@m57.biz, Jean, Outlook, M1), 7/7/2008 6:24:00)
Sent_Email(Jean@m57.biz, Alison@m57.biz, Jean, Outlook, M1), 7 July 6:25:00)
Sent_Email(Jean@m57.biz, Alison@m57.biz, Jean, Outlook, M1), 7 July 6:25:00)
Sent_Email(Jean@m57.biz, Alison@m57.biz, Jean, Outlook, M1), 7/7/2008 6:26:00)
Sent_Email(Jean@m57.biz, Alison@m57.biz, Jean, Outlook, M1), 7/7/2008 6:27:00)
Sent_Email(Jean@m57.biz, Alison@m57.biz, Jean, Outlook, M1), 7/7/2008 6:26:00
Sent_Email(Jean@m57.biz, Alison@m57.biz, Jean, Outlook, M1), 7/7/2008 6:27:00
Sys_Logout(Jean, M1, 7/7/2008, 8:35:32)

Snapshot 3
Sys_Login(Jean, M1, 21/07/2008 01:06:43)
Snapshots generated for hypotheses 2.a

Snapshot 1
Sys_Login(Jean, M1, 6/07/2008, 6:10:56)
Sent_Email(Jean@m57.biz, Jean@m57.biz, Jean, Outlook, M1) 6/07/2008 8:39:00
Sent_Email(Jean@m57.biz, Jean@m57.biz, Jean, Outlook, M1), 6/07/2008 8:48:00
Sent_Email(Jean@m57.biz, Alison@m57.biz, Jean, Outlook, M1), 6/07/2008 8:55:00.
Sys_Logout(Jean, M1, 06/07/2008, 09:07:54)

Snapshot 2
Sys_Login(Jean, M1, 7/07/2008, 5:40:32)
Sent_Email(Jean@m57.biz, Alison@m57.biz, Jean, Outlook, M1), 7/7/2008 7:24:00
Sent_Email(Jean@m57.biz, Alison@m57.biz, Jean, Outlook, M1), 7/7/2008 6:25:00
Sent_Email(Jean@m57.biz, Alison@m57.biz, Jean, Outlook, M1), 7/7/2008 6:26:00
Sent_Email(Jean@m57.biz, Alison@m57.biz, Jean, Outlook, M1), 7/7/2008 6:27:00
Sys_Logout(Jean, M1, 7/7/2008, 8:35:32)

Snapshot 3
Sys_Login(Jean, M1, 20/07/2008 00:00:41)
Sent_Email(Jean@m57.biz, Alison@m57.biz, Jean, Outlook, M1), 20/07/2008 00:31:00
Sent_Email(Jean@m57.biz, Alison@m57.biz, Jean, Outlook, M1), 20/07/2008 00:44:00.
Sent_Email(Jean@m57.biz, Alison@m57.biz, Jean, Outlook, M1), 20/07/2008 00:44:00
Sent_Email(Jean@m57.biz, Alison@m57.biz, Jean, Outlook, M1), 20/07/2008 00:44:00
Sent_Email(Jean@m57.biz, Alison@m57.biz, Jean, Outlook, M1), 20/07/2008 00:44:00
Sent_Email(Jean@m57.biz, Alison@m57.biz, Jean, Outlook, M1), 20/07/2008 00:46:00.
Sent_Email_Attach(Jean@m57.biz, Tuckgorge@Gmail.Com, Jean, m57biz, Outlook, M1), 20/07/2008 06:04:00.
Sent_Email(Jean@m57.biz, Tuckgorge@Gmail.Com, Jean, Outlook, M1), 20/07/2008 06:04:00.
Sent_Email(Jean@m57.biz, Alison@m57.biz, Jean, Outlook, M1), 21/07/2008 00:51:00.
Sent_Email(Jean@m57.biz, Alison@m57.biz, Jean, Outlook, M1), 21/07/2008 00:57:00.
Sent_Email(Jean@m57.biz, Bob@m57.biz, Jean, Outlook, M1), 21/07/2008 00:58:00.
Sent_Email(Jean@m57.biz, Carol@m57.biz, Jean, Outlook, M1), 21/07/2008 01:01:00.
Sent_Email(Jean@m57.biz, Bob@m57.biz, Jean, Outlook, M1), 21/07/2008 01:04:00.
Sys_Logout(Jean, M1, 21/07/2008 01:06:43)
Snapshot 4
Sys_Login(Administrator, M1, 21/07/2008 01:22:18)
Sent_Email(Jean@m57.biz, Carol@m57.biz, Jean, Outlook, M1), 21/07/2008 01:45:00.
Sent_Email(Jean@m57.biz, Bob@m57.biz, Jean, Outlook, M1), 21/07/2008 01:46:00).

Snapshots generated for hypotheses 3

Snapshot 1
Sys_Login(Jean,M1, 6/07/2008, 6:10:56)
Sys_Logout(Jean, M1, 06/07/2008, 09:07:54)
Sys_Login(Jean,M1, 7/07/2008, 5:40:32)
Sys_Logout(Jean,M1, 7/07/2008, 8:35:32)
Sys_Login(Jean,M1, 7/07/2008, 16:55:00)
Sys_Mount(USB1,E, 07/07/2008 19:51:07)
Sys_Modify(Doc1, Jean, 07/07/2008 19:47:03)
Sys_Open(Doc2, Jean, 07/07/2008 19:48:48)
Sys_Modify(Doc2, Jean, 07/07/2008 19:55:56)
Sys_Modify(Doc2, Jean, 07/07/2008 19:58:00)
Sys_Modify(Doc1, Jean, 07/07/2008 20:03:18)
Sys_Close(Doc2, Jean, 07/07/2008 20:04:02)
Sys_Close(Doc1, Jean, 07/07/2008 20:04:59)
Sys_Copy(Doc1,E, 07/07/2008 20:05:48)
Sys_Unmount(USB1,E, 07/07/2008 20:06:13)

Snapshot 2
Sys_Logout(Jean,M1, 8/07/2008, 2:10:17)
Sys_Login(Jean,M1, 8/07/2008, 17:10:05)
Sys_Logout(Jean,M1, 9/07/2008, 2:12:57)
Sys_Login(Jean,M1, 9/07/2008, 16:37:33)
Sys_Logout(Jean,M1, 10/07/2008, 2:00:17)
Sys_Login(Jean,M1, 10/07/2008, 16:59:03)
Sys_Logout(Jean,M1, 11/07/2008, 1:56:19)
Sys_Login(Jean,M1, 11/07/2008, 16:02:20)
Sys_Logout(Jean,M1, 12/07/2008, 2:07:21)
Sys_Login(Devon,M1, 12/07/2008 03:02:47)
Sys_Mount(USB2,E, 12/07/2008 03:50:00)
Sys_Modify(Doc2, M1, 12/07/2008 04:02:33)
Sys_Close(Doc2, M1, 12/07/2008 4:05:00)
Sys_Copy(Doc2, C:/Documents and Settings/Devon/Desktop, M1, 12/07/2008 4:15:23)
Sys_Open(Doc3, M1, 12/07/2008 4:18:16)
Sys_Modify(Doc3, M1, 12/07/2008 5:10:56)
Sys_Copy(Doc3,E, 12/07/2008 5:13:31)
Sys_Unmount(USB2,E, 12/07/2008 5:14:59)

Snapshot 3
Sys_Logout(Devon,M1, 12/07/2008 05:23:31)
Sys_Login(Jean,M1, 14/07/2008, 16:48:02)
Sys_Logout(Jean,M1, 15/07/2008, 3:29:59)
Sys_Login(Jean,M1, 15/07/2008, 17:01:32)
Rec_Email(NewsLetters@Npr.Com,Jean@m57.biz, Jean, Outlook, M1, 16/07/2008 00:00:49).
Sys_Logout(Jean,M1, 16/07/2008, 3:10:59)
Sys_Login(Jean,M1, 16/07/2008, 17:34:03)
Sys_Logout(Jean,M1, 17/07/2008, 3:05:06)
Sys_Login(Jean,M1, 17/07/2008, 16:12:22)
Sys_Logout(Jean,M1, 18/07/2008, 2:38:13)
Sys_Login(Jean,M1, 18/07/2008, 16:15:14)
Sys_Logout(Jean,M1, 19/07/2008, 1:35:06)
Sys_Login(Jean, M1, 20/07/2008 00:00:41)
Sys_Open(m57biz, 20/07/2008 01:05:21)
Sys_Modify(m57biz, 20/07/2008 01:05:21)
Sys_Modify(m57biz, 20/07/2008 01:15:21)
Sys_Mount(7&162a431980, E, M1, 20/07/2008 01:26:21)
Sys_Modify(m57biz, 20/07/2008 01:50:21)
Sys_Copy(m57biz, E, 20/07/2008 02:22:00)
Rec_Email(Tuckgorge@Gmail.Com,Jean@m57.biz,Jean,Outlook,M1), 20/07/2008 02:22:45).
Sent_Email_Attach(Jean@m57.biz,Tuckgorge@Gmail.Com,Jean, m57biz,Outlook,M1), 20/07/2008 2:28:00).
Sys_Unmount(7&162a431980,E, M1, 20/07/2008 01:26:21)

References
http://www.eclipse.org/modeling/emf/.
http://digitalcorpora.org/corpora/scenarios/m57-jean.