International Journal of Innovative Computing, Information and Control
Volume **8**, Number **1(A)**, January **2012**

ICIC International ©2012 ISSN 1349-4198
pp. **107–119**

HERMIT: A NEW METHODOLOGY FOR CREATING AUTONOMOUS SOFTWARE DEPLOYMENT PACKAGES

Umar Manzoor, Samia Nefti and AH Jones

School of Computing, Science and Engineering
University of Salford
Salford, Greater Manchester, United Kingdom
umarmanzoor@gmail.com; s.nefti-meziani@salford.ac.uk

Received August 2010; revised December 2010

Abstract. *An Installation that is performed without user interaction and does not display any message or GUI during its progress is known as Silent Unattended Installation. Silent unattended installation plays a key role when installing software over networks and time is of immense importance because it does not require user interaction/intervention during the process and usually skips the non important steps which are usually part of installer wizards. In this paper we have proposed HERMIT: A new Methodology for Creating Autonomous Software Deployment Packages that automates the process of silent unattended installations/un-installations and requires minimal possible level of interaction with the user. HERMIT not only introduces new methodology but generalizes the process of silent unattended installation/un-installation. HERMIT installation package when deployed on the PC or network will follow the same steps as performed by the installer and un-installation package will rollback everything added during the installation without user interaction. An agent based system for activity monitoring on network (ABSAMN) is responsible for the deployment of HERMIT Packages on the specified locations of the network. The process is fully autonomous and does not require any user interaction. HERMIT has been evaluated on a large number of softwares, and results were very promising and support the implementation of the solution.*
Keywords: Autonomous installation, Silent unattended installation, Network installation, Mobile agent for installation, Multi-agent for software deployment

1. **Introduction.** With the ever growing complexity of computer networks it has become a challenging job for an IT professional to manage the network resources and perhaps becoming difficult humanly. In such environments, it becomes a hassle to install the software(s) of need and remove the unnecessary ones. An installer is a program which helps the user to install the software application on the computer [7]. In the past, different software vendors used different standards for installation which resulted in many problems (overwriting or removal of shared file, etc.) while installing multiple softwares [6]. Today, every installer follows the same standard for software installation and common operations performed during the installation including the creation and modification of 1) Shared and Non-Shared Program Files, 2) Directories (Folders), 3) Windows Registry Entries, 4) Configuration Files, 5) Environment variables and 6) Links (Shortcuts) [9]. An uninstaller is opposite of installer and helps the user to remove all or part of the software application from the computer and generally software vendors provide uninstaller with the software (K. Spreitzer (2009)).

The main goal of every software is to provide its user minimal hassle to bring it into use and this gives birth to intelligent installer which facilitates user with easy installation and skips unnecessary steps. An Installation that is performed without user interaction and

does not display any message or GUI during its progress is known as Silent Unattended Installation [27]. The advent of complex computer systems and the human motivation to simplify every process have led to an increased demand for silent unattended installer programs. Silent unattended installation plays a key role when installing software over networks and time is of immense importance because it does not require user interaction/intervention during the process and usually skips the non important steps which are usually part of installer wizards. A silent unattended installer can start multiple installations at the same time regardless of the fact that they are being performed on a network or PC.

Many installers provide the facility of silent installation using a set of switches which are defined within the installer setup [27]. The user has to run the installer setup followed by the particular switch to perform silent unattended installation. The preferences are either set to default or in a separate file. The disadvantage of this technique is that the method is not generalized and cannot be applied to every software because it is dependent on the installer type which was used for deployment.

The need of the time is to develop a tool which automates the process of installation/un-installation of softwares and minimizes the user interaction in this process. In this paper, we have proposed A New Methodology for Creating Autonomous Software Deployment Packages (HERMIT) that automates the process of silent unattended installations/un-installations and requires the minimal possible level of interaction with the user. The aim is to generalize the process of silent unattended installations/un-installations and create a repository of software packages, which when needed can be deployed according to the need on the network.

The proposed HERMIT will not only introduce new methodology but generalize the process of silent unattended installation/un-installation. Computer administrator will train HERMIT by performing the installation of the specific software on a PC. During this phase, Process Monitoring will be performed and all the changes made by the installer to the system will be tracked and stored in a XML Log file. After the training phase, the XML log file will be passed to HERMIT which will filter the information and make silent unattended installation/un-installation packages. This process in turn would result in a repository of software's which would not require the training or the monitoring phase. The software can then just be pulled of the repository and the silent unattended installation/un-installation can be performed on the PC or network. HERMIT installation package when deployed on the PC or network will follow the same steps as performed by the installer and un-installation package will rollback everything added during the installation.

Most of the installers do not cater for the mass installations/un-installations on the network; in fact the ones which provide do not allow silent unattended installations/un-installations. The approach to this dilemma is to incorporate a network tool which deploys the package created by HERMIT on large distributed network efficiently. HERMIT packages will be deployed on the network using an agent based system for activity monitoring on network (ABSAMN) [4] which is a multi-agent [1,2,5,42,43] based approach for the monitoring of resources over a network. The system is fully autonomous and manages resources on its own with the help of mobile agents [8,28,29,41,44]. Mobile Agent has been used in the areas of intrusion detection, network fault management, load sharing, traffic analysis, mobile devices, network routing, system management, dynamic parking and supply chain [13,18,19,23,32,34,35,39,40].

The paper is organized as follows. The first section discusses the related work done in silent and unattended installation. This section is followed by the detail discussion of the HERMIT: A New Methodology for Creating Autonomous Software Deployment Packages

architecture. At the end, conclusions are drawn and outline some questions for future research.

2. **Related Work.** Many installers provide the facility of silent installation using a set of switches which are defined within the installer setup [11,12,27,37], however, the method is not general and is dependent on the installer used for deployment. Every installer has its own set of switches and scripting language, understanding the switches and writing scripts using the scripting language is nothing less than a job of an expert and the process is nowhere near generalization. Before software installation, the System Environment is in a particular state and once it is installed, the System Environment is in another state (i.e., changes are made to different parts of the System Environment). Therefore, after installation a state change occurs and many installers use the System State Change methodology for making unattended software setups however this technique is not reliable and takes a long time to calculate the differences between the snapshots [14,15].

During software installation, most of the installers tend to generate an installation log which has the track of all the changes made by the installer during installation. This installation log can be used to backtrack all the changes done by the installer and the software can be un-installed using this technique, however, every installer has its own format of installation log so third party tools cannot use the log file for software un-installation [16,17,31]. There is no particular framework by any installer to perform silent unattended installations/un-installations on the network. There are many softwares available which support the unattended installation [25-27] but these softwares are either vendor specific or software specific (not general) and do not provide silent installation feature. Most of the installers/softwares do not cater for the mass installations/un-installations on the network; in fact the ones which provide do not allow silent unattended installations/un-installations.

3. **System Architecture.** HERMIT: A New Methodology for Creating Autonomous Software Deployment Packages is a specialized and advanced silent unattended installation /un-installation package creator. HERMIT generalizes and automates the process of silent unattended installation/un-installation. The packages created can be deployed on the network without user intervention using an existing framework An agent based system for activity monitoring on network (ABSAMN) [4]. The system consists of the following three main modules:

- Installation Monitor Module – IMM
- PID Tracker
- Package Manager – PM

3.1. **Installation monitor module.** Installation Monitor monitors the activity of installer during installation of certain application and generates a log file, which contains the information of system changes made by installer during installation. Initially for monitoring purpose Installation Monitor Module – IMM uses Microsoft Sysinternals Process Monitor [3] for monitoring. Process Monitor is an advanced monitoring tool for Windows Operating System which shows/logs real-time file system, Registry and process/thread activity.

The key role of Installation Monitor is to monitor the activities during the installation and generate a log file in XML format. HERMIT cannot be freely distributed unless it has its own Monitoring tool. However, for the purpose of proof of concept Process Monitor is used. Package Manager needs two types of information from Installation Monitor Module 1) Registry Monitoring – Changes made in registry by the installer and 2) File Monitoring – Changes made in File System by the installer. The installation

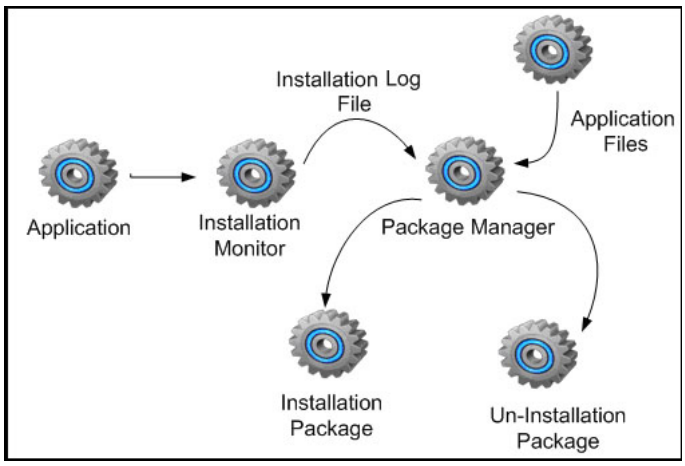

FIGURE 1. Zero level diagram of HERMIT: A new methodology for creating autonomous software deployment packages

and logging of the installation events is a complex process and once it is complete the Installation Monitor generates a XML log file, which contains all the activities (registry, file) regarding the installation process. Figure 2 shows a Sample of log file. In order to standardize the process, a common format is created and the detail of some of the tags used in this format is given below. The key role of Installation Monitor is to monitor the activities during the installation used.

3.2. **PID tracker.** Log file created by Installation Monitor Module contains information of many other processes as it monitors the whole system activity during software installation. The workaround to this problem is to capture the ID of the process which is of our interest and ignore the rest of the processes. This not only reduces the size of the Log File but also reduces the time for filtering the Log File.

3.3. **Package manager.** Package manager is a bundling tool for HERMIT: A New Methodology for Creating Autonomous Software Deployment Packages. The purpose of this module is to bundle all the necessary information including files, inis, registries, shortcuts and other necessary information to a custom HERMIT Package which will later be used by an HERMIT Installer to perform the installation. The package manager consists of following three modules.

3.3.1. *HERMIT builder.* HERMIT Builder creates the silent unattended installation/un-installation packages for the software. It collects the application files from the system along with software file and registry manifest and creates Installation/UnInstallation Packages. This module is further divided into three more modules.

- **Sift:** Log file created by Installation Monitor Module and filtered using PID Tracker contains valuable and invaluable information which needs to be filtered. The information which Package Manager needs for making HERMIT Packages is shown in Table 1. Sift filters the Log file based on the filtering criteria as shown in Table 1 and creates a new filtered log file.
- **Wrapper:** Sift is responsible for filtering and gathering the required events to perform the installation. Wrapper on the other hand uses that filtered log file to generate commands to perform the required installation as shown in Figure 3. Wrapper will generate the following five files. First three files will be used for silent unattended installation and the last two files will be used for silent unattended un-installation.

<**Event**> Each Activity of File/Registry operation is called an Event
<**Process_Name**> Name of the Process which is performing the event
<**PID**> Process ID
<**Operation**> Operation Name which is performed by the Process (i.e., create file or add registry)
<**Path**> The location where the event has an effect
<**Result**> The result of the event
<**Detail**> Any detail regarding the event

Sample Log XML File

```
<EVENTS>
    <Event>
      <ProcessIndex>52</ProcessIndex>
      <Sequence>41372</Sequence>
      <Time_of_Day>12:30:04.4591462 PM</Time_of_Day>
      <Process_Name>SETUP.exe</Process_Name>
      <PID>3800</PID>
      <Operation>CreateFile</Operation>
      <Path>C:\Program Files\WinZip\EXAMPLE.ZIP</Path>
      <Result>SUCCESS</Result>
      <Detail>Generic Read/Write</Detail>
   </Event>
   <Event>
      <ProcessIndex>52</ProcessIndex>
      <Sequence>41373</Sequence>
      <Time_of_Day>12:30:04.4728681 PM</Time_of_Day>
      <Process_Name>SETUP.exe</Process_Name>
      <PID>3800</PID>
      <Operation>QueryOpen</Operation>
      <Path>C:\Program Files\WinZip\EXAMPLE.ZIP</Path>
      <Result>FAST IO DISALLOWED</Result>
      <Detail></Detail>
   </Event>
</EVENTS>
```

FIGURE 2. Sample log XML file

TABLE 1. Filtering criteria for the log file

	Operation	**Result**
1	CreateFile	SUCCESS
2	SetRenameInformationFile	SUCCESS
3	RegCreateKey	SUCCESS
4	RegSetValue	SUCCESS
5	RegDeleteValue	SUCCESS
6	RegDeleteKey	SUCCESS

IGetter – This file will be used by the Builder module of the Package Manager to collect all the application files. Builder Module just has to run the IGetter.bat file and all the application files will be copied to the specified folder automatically. Sample IGetter file is shown in Figure 4.

ISetter – This file will be bundled in HERMIT Installation Pack by the Builder module and will be used by the HERMIT Installer Module to copy all the application

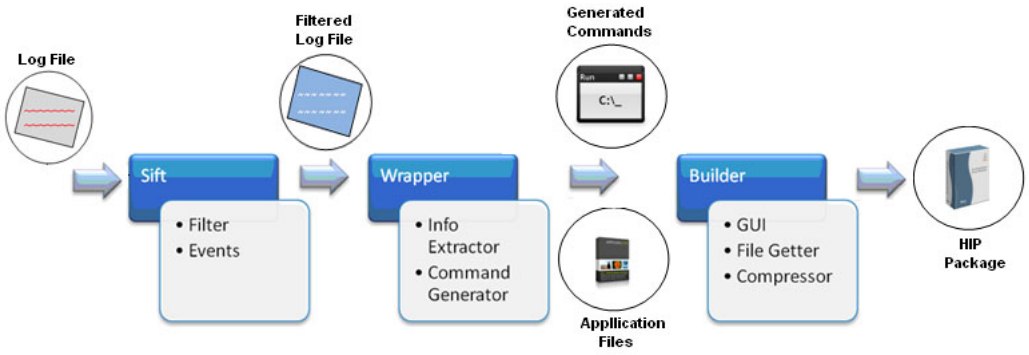

FIGURE 3. HERMIT builder modules and their interaction

```
Sample IGetter File
mkdir "C:\HERMIT\bin\Pack"
cd "C:\HERMIT\bin\Pack"
copy "C:\Program Files\WinZip\EXAMPLE.ZIP"
copy "C:\Program Files\WinZip\WINZIP.CHM"
copy "C:\Program Files\WinZip\WZ.COM"
```

FIGURE 4. Sample IGetter.bat file

files to the specified folder. HERMIT Installer Module just has to run the ISetter.bat file and all the application files will be copied to the specified folder automatically. Sample ISetter file is shown in Figure 5.

```
Sample ISetter File
mkdir "C:\Program Files\WinZip"
mkdir "C:\Documents and Settings\All Users\Start Menu\Programs\WinZip"
copy "C:\Documents and Settings\Default User\Local Settings\Temp\HS1\WZSHLSTB.DLL"
"C:\Program Files\WinZip"
copy "C:\Documents and Settings\Default User\Local Settings\Temp\HS1\EXAMPLE.ZIP"
"C:\Program Files\WinZip"
```

FIGURE 5. Sample ISetter.bat file

IRegistry – This file will be bundled in HERMIT Installation Pack by the Builder module and will be used by the HERMIT Installer Module to copy all the registry entries in the Windows registry at the specified locations. HERMIT Installer Module just has to run the IRegistry.bat file and all the registry entries will be inserted. Sample IRegistry file is shown in Figure 6.

```
Sample IRegistry File
REG ADD "HKCR\CLSID\{00000000-0000-0000-0000-000000000000}"/v "X13D5SY3FG
7WZ" /t "REG_SZ" /d "0" /f
REG ADD "HKLM\SOFTWARE\Google\NavClient" /v "test" /t "REG_SZ" /d "test" /f
REG ADD "HKCU\Software\Microsoft\Windows\ CurrentVersion\Explorer\User Shell
Folders" /f
REG ADD "HKLM\Software\Microsoft\Windows\ CurrentVersion\Uninstall\WinZip" /f
```

FIGURE 6. Sample IRegistry.bat file

USetter – This file will be bundled in HERMIT UnInstallation Pack by the Builder module and will be used by the HERMIT UnInstaller Module to delete all the application files along with the folders. HERMIT UnInstaller Module just has to run the USetter.bat file and all the application files will be deleted automatically. Sample USetter file is shown in Figure 7.

```
Sample USetter File
DEL "C:\Program Files\WinZip\WZSHLSTB.DLL" /f
DEL "C:\Program Files\WinZip\EXAMPLE.ZIP" /f
RMDIR "C:\Program Files\WinZip" /S /Q
RMDIR "C:\Documents and Settings\All Users\Start Menu\Programs\WinZip" /S /Q
```

FIGURE 7. Sample USetter.bat file

URegistry – This file will be bundled in HERMIT UnInstallation Pack by the Builder module and will be used by the HERMIT UnInstaller Module to delete all the registry entries of the specific application from the Windows registry. HERMIT Installer Module just has to run the URegistry.bat file and all the registry entries will be deleted automatically. Sample URegistry file is shown in Figure 8.

```
Sample URegistry File
REG DELETE "HKLM\SOFTWARE\Microsoft\Windows\ CurrentVersion\App Paths\
winzip.exe" /f
REG DELETE "HKCR\WinZip\DefaultIcon" /f
REG DELETE "HKCR\WinZip\shell\open\command" /f
```

FIGURE 8. Sample URegistry.bat file

- **Builder:** Builder will use the IGetter.bat file to get the application files and bundle the application files along with the ISetter.bat and IRegistry.bat files generated by the wrapper in HERMIT Installation Package as shown in Figure 3. Builder will bundle USetter.bat and URegistry.bat files together in HERMIT UnInstallation Package.

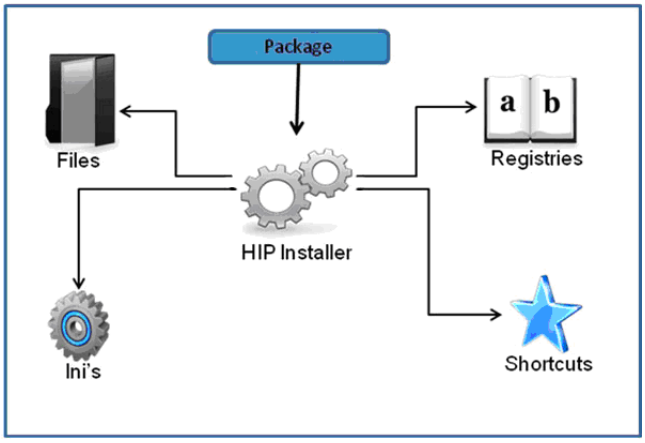

FIGURE 9. HERMIT installer

3.3.2. *HERMIT installer.* HERMIT Installer extracts the Installation package and installs it to the desired location. It works like any other installer but the difference is that it does not need any user interaction and the installation is performed silently. The input to this module is the Installation package created by the HERMIT Builder Module. The installer extracts all the bundled items and copies them to their location throughout the system where necessary as shown in Figure 9.

HERMIT Installer Module makes use of Installer.bat file to unwrap the Installer Package. Sample of HERMIT Installer is shown in Figure 10.

```
Sample Installer File
rmdir "C:\Documents and Settings\Default User\Local Settings\Temp\Test" /S /Q
mkdir "C:\Documents and Settings\Default User\Local Settings\Temp\Test"
rar e %1 "C:\Documents and Settings\Default User\Local Settings\Temp\Test"
call "C:\Documents and Settings\Default User\Local Settings\Temp\Test\ISetter.bat"
call "C:\Documents and Settings\Default User\Local Settings\Temp\Test\IRegistry.bat"
rmdir "C:\Documents and Settings\Default User\Local Settings\Temp\Test" /S /Q
```

FIGURE 10. Sample installer file

3.3.3. *HERMIT uninstaller.* The input to this module is the UnInstallation package created by the HERMIT Builder Module. HERMIT UnInstaller extracts the UnInstallation package and removes all the files, registry entries, ini's, shortcut etc. of the application. HERMIT UnInstaller Module makes use of UInstaller.bat file to unwrap the UnInstaller Package. Sample of HERMIT UnInstaller is shown in Figure 11.

```
Sample UnInstaller File
rmdir "C:\Documents and Settings\Default User\Local Settings\Temp\Test" /S /Q
mkdir "C:\Documents and Settings\Default User\Local Settings\Temp\Test"
rar e %1 "C:\Documents and Settings\Default User\ Local Settings\Temp\Test"
call "C:\Documents and Settings\Default User\Local Settings\Temp\Test\USetter.bat"
call "C:\Documents and Settings\Default User\Local Settings\Temp\Test\URegistry.bat"
rmdir "C:\Documents and Settings\Default User\Local Settings\Temp\Test" /S /Q
```

FIGURE 11. Sample uninstaller file

HERMIT packages will be deployed on the specified locations of the network using an agent based system for activity monitoring on network (ABSAMN) [4]. The process will be autonomous and does not require any user interaction. ABSAMN uses Mobile Agent to monitor the activities of the network autonomously. The term mobile agent refers to a process that can transport its state from one environment to another, with its data intact, and still being able to perform appropriately in the new environment [10]. Network administrator will interact with Master Controller Agent (MCA) which will be responsible for the deployment of HERMIT packages on the network with the help of Mobile Agents.

4. **Test Case.** HERMIT has been evaluated on a large number of softwares, and results were very promising and support the implementation of the solution. Some of the popular softwares on which HERMIT was evaluated are [20-23,33,36,38]. The Test case of WinZip is given below.

```
Partial WinZip XML Log File
<EVENTS>
  <Event>
     <ProcessIndex>49</ProcessIndex>
     <Sequence>14801</Sequence>
     <Time_of_Day>10:52:00.1600448 PM </Time_of_Day>
     <Process_Name> winzip90.exe</Process_Name>
     <PID>1848</PID>
     <Operation>CreateFile</Operation>
     <Path>C:\WINDOWS\Prefetch\WINZIP90.EXE-31621C60.pf
     </Path>
     <Result> NAME NOT FOUND </Result>
     <Detail> Desired Access: Generic Read, Disposition:
     Open, Options: Synchronous IO Non-Alert,
     Attributes: n/a, ShareMode: None, AllocationSize:
     n/a</Detail>
  </Event>
  <Event>
     <ProcessIndex>49</ProcessIndex>
     <Sequence>15036</Sequence>
     <Time_of_Day>10:52:00.1808645 PM </Time_of_Day>
     <Process_Name>winzip90.exe</Process_Name>
     <PID>1848</PID>
     <Operation>QueryOpen</Operation>
     <Path> C:\WINDOWS\system32\imm32.dll</Path>
     <Result>SUCCESS</Result>
     <Detail> CreationTime: 8/4/2004 5:56:44 AM,
     LastAccessTime: 6/11/2008 10:52:00 PM, 9:42:38 PM,
     LastWriteTime: 8/4/2004 5:56:44 AM, ChangeTime:
     2/9/2008 AllocationSize: 110,592, EndOfFile:
     110,080, FileAttributes: A</Detail>
  </Event>
   ...
   ...
  </EVENTS>
```

FIGURE 12. Partial WinZip XML log file

4.1. **Step 1 (installation monitoring).** In Step 1, Installation Monitor monitors the activity of WinZip installation and generates a Log file. The size of the Log file is 3.66 MB and the partial WinZip Log file is shown in Figure 12.

4.2. **Step 2 (filtering and generating batch files).** In Step 2, Installation Log File will be filtered and commands (batch files) will be generated to perform the installation. The size of the log file is reduced to 16KB after filteration. Partial Batch files are shown in Figures 13-17.

4.3. **Step 3 (package generating batch files).** After creation of Commands Builder Module will make Install and Uninstall Packages which will be used by Installer and Un-Installer Modules of HERMIT for Silent Unattended Installation. The results yield the

```
WinZip Partial IGetter File (Size 8KB)
mkdir "C:\HIP\bin\Pack"
cd "C:\HIP\bin\Pack"
copy "C:\Program Files\WinZip\EXAMPLE.ZIP"
copy "C:\Program Files\WinZip\WINZIP.CHM"
copy "C:\Program Files\WinZip\WZ.COM"
copy "C:\Program Files\WinZip\WZ.PIF"
copy "C:\Program Files\WinZip\WZ32.DLL"
copy "C:\Program Files\WinZip\WZCAB.DLL"
copy "C:\Program Files\WinZip\WZCAB3.DLL"
copy "C:\Program Files\WinZip\WZINST.CHM"
copy "C:\Program Files\WinZip\WZPOPUP.HLP"
...
...
```

FIGURE 13. WinZip partial IGetter.bat file

```
WinZip Partial ISetter File (Size 7KB)
mkdir "C:\Program Files\WinZip"
mkdir "C:\Documents and Settings\All Users\Start Menu\Programs\WinZip"
copy "C:\Documents and Settings\Default User\Local Settings\Temp\WinZip\
WZSHLSTB.DLL" "C:\Program Files\WinZip"
copy "C:\Documents and Settings\Default User\Local Settings\Temp\WinZip\
EXAMPLE.ZIP" "C:\Program Files\WinZip"
"C:\Documents and Settings\Default User\Local Settings\Temp\WinZip\WINZIP.CHM"
"C:\Program Files\WinZip"
"C:\Documents and Settings\Default User\Local Settings\Temp\WinZip\WZ.COM"
"C:\Program Files\WinZip"
"C:\Documents and Settings\Default User\Local Settings\Temp\WinZip\WZ.PIF"
"C:\Program Files\WinZip"
"C:\Documents and Settings\Default User\Local Settings\Temp\WinZip\WZ32.DLL"
"C:\Program Files\WinZip"
"C:\Documents and Settings\Default User\Local Settings\Temp\WinZip\WZCAB.DLL"
"C:\Program Files\WinZip"
"C:\Documents and Settings\Default User\Local Settings\Temp\WinZip\WZCAB3.DLL"
"C:\Program Files\WinZip"
"C:\Documents and Settings\Default User\Local Settings\Temp\WinZip\WZINST.CHM"
"C:\Program Files\WinZip"
...
...
```

FIGURE 14. WinZip partial ISetter.bat file

limitations of HERMIT which are 1) the training phase should be performed on a fresh PC and 2) HERMIT cannot be used to make silent unattended packages for Operating Systems or applications which perform inter process communication during installation. One extension of HERMIT is to extend the existing architecture to support the silent unattended packages for Operating Systems and complex applications which involve inter process communication during installation.

5. **Conclusions.** The main goal of every software is to provide its user minimal hassle to bring it into use and this gives birth to intelligent installer which facilitates user with

```
WinZip Partial IRegistry File (Size 37KB)
REG ADD “HKCR\CLSID\{00000000-0000-0000-0000-000000000000}” /v “X13D5SY3FG
7WZ” /t “REG_SZ” /d “0” /f
REG ADD “HKLM\SOFTWARE\Google\NavClient” /v “test” /t “REG_SZ” /d “test” /f
REG ADD “HKCU\Software\WinZip Computing” /v “WinZip Computing” /t “REG_SZ”
/d “Please look in the Nico Mak Computing section for WinZip keys, values, and settings.”
/f
REG ADD “HKCU\Software\Microsoft\Windows\CurrentVersion\Explorer\User Shell Fold-
ers” /f
REG ADD “HKLM\Software\Microsoft\Windows\ CurrentVersion\Uninstall\WinZip” /f
REG ADD “HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall\WinZip”
/v “DisplayName” /t “REG_SZ” /d “WinZip” /f
REG ADD “HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall\WinZip”
/v “UninstallString” /t “REG_SZ” /d “\”C:\Program Files\WinZip\WINZIP32.EXE \
“ /uninstall” /f
REG ADD “HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall\WinZip”
/v “InstallLocation” /t “REG_SZ” /d “C:\PROGRA~1\WINZIP\\” /f
REG ADD “HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall\WinZip”
/v “Publisher” /t “REG_SZ” /d “WinZip Computing, Inc.” /f
REG ADD “HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall\WinZip”
/v “DisplayVersion” /t “REG_SZ” /d “9.0 SR-1 (6224)” /f
...
...
```

FIGURE 15. WinZip partial IRegistry.bat file

```
WinZip Partial USetter File (Size 3KB)
DEL “C:\Program Files\WinZip\WZSHLSTB.DLL” /f
DEL “C:\Program Files\WinZip\EXAMPLE.ZI” /f
DEL “C:\Program Files\WinZip\WINZIP.CHM” /f
DEL “C:\Program Files\WinZip\WZ.COM” /f
DEL “C:\Documents and Settings\All Users\Start Menu\ Programs\WinZip\WinZip 9.0
SR-1.lnk” /f
DEL “C:\Documents and Settings\All Users\Start Menu\ Programs\WinZip\Uninstall
WinZip.lnk” /f
RMDIR “C:\Program Files\WinZip” /S /Q
RMDIR “C:\Documents and Settings\All Users\Start Menu\Programs\WinZip” /S /Q
...
...
```

FIGURE 16. WinZip partial USetter.bat file

easy installation and skips unnecessary steps. The advent of complex computer systems and the human motivation to simplify every process have led to an increased demand for silent unattended installer programs. In this paper, we have proposed HERMIT: A New Methodology for Creating Autonomous Software Deployment Packages that automates the process of silent unattended installations/un-installations and requires the minimal possible level of interaction with the user. The aim is to generalize the process of silent unattended installations/un-installations and create a repository of software packages, which when needed can be deployed according to need on the network without user interaction. The work can be extended in many directions. One possible extension of HERMIT

```
WinZip Partial URegistry File (Size 5KB)
REG DELETE "HKCR\CLSID\{00000000-0000-0000-0000-000000000000}"/v "X13D5SY
3FG7WZ" /f
REG DELETE "HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\App Paths\
winzip32.exe" /f
REG DELETE "HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\App Paths\
winzip.exe" /f
REG DELETE "HKCR\WinZip\DefaultIcon" /f
REG DELETE "HKCR\WinZip\shell\open\command" /f
REG DELETE "HKCR\WinZip\shell\open" /f
REG DELETE "HKCR\WinZip\shell\print\command" /f
...
...
```

FIGURE 17. WinZip partial URegistry.bat file

is to extend the existing architecture to support the silent unattended packages for Operating Systems and complex applications which involve inter process communication during installation.

REFERENCES

[1] S. Park and V. Sugumaran, Designing multi-agent systems: A framework and application, *Expert Systems with Applications*, vol.28, no.2, pp.259-271, 2005.

[2] C. Carrascosa, J. Bajo, V. Julian, J. M. Corchado and V. Botti, Hybrid multi-agent architecture as a real-time problem-solving model, *Expert Systems with Applications*, vol.34, no.1, pp.2-17, 2008.

[3] M. Russinovich and B. Cogswell, *Microsoft Sysinternal Process Monitor*, http://technet.microsoft.com/en-us/sysinternals/bb896645.aspx, 2008.

[4] U. Manzoor and S. Nefti, An agent based system for activity monitoring on network – ABSAMN, *Expert Systems with Applications*, vol.36, no.8, pp.10987-10994, 2009.

[5] J. Ota, Multi-agent robot systems as distributed autonomous systems, *Advanced Engineering Informatics*, vol.20, no.1, pp.59-70, 2006.

[6] rm.com, *Application Installation and Management*, http://www.rm.com/secondary/products/story.asp?cref=ps189212, 2009.

[7] R. Lipton, *What is an Installer*, http://radio.weblogs.com/0100059/stories/2002/02/14/ whatisaninstaller.html, 2009.

[8] J. Wong, G. Helmer, V. Naganathan, S. Polavarapu, V. Honavar and L. Miller, Smart mobile agent facility, *Journal of Systems and Software*, vol.56, no.1, pp.9-22, 2001.

[9] Altenmann, *Installation (Computer Programs)*, http://en.wikipedia.org/wiki/software_installation, 2009.

[10] G. Weiss, *Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence*, The MIT Press, London, 1999.

[11] *InstallShield Installer*, http://www.installshield.com, 2009.

[12] *Microsoft MSI Packages*, http://support.microsoft.com/kb/257718, 2009.

[13] D. Gavalas, G. E. Tsekouras and C. Anagnostopoulos, A mobile agent platform for distributed network and systems management, *Journal of Systems and Software*, vol.82, no.2, pp.355-371, 2009.

[14] S. Shim, W. Lee and C. Park, *An Efficient Snapshot Technique for Ext3 File System in Linux 2.6*, http://whitepapers.techrepublic.com.com/abstract.aspx?docid=295978, 2006.

[15] Y.-H. Kim, S.-I. Jung and M.-J. Kim, Snapshot technique using snapshot status table for shared storage supporting large capacity, *Proc. of the 6th International Conference on Advanced Communication Technology*, Korea, vol.2, pp.677-682, 2004.

[16] *Advance Uninstaller Pro*, http://www.innovative-sol.com/uninstaller/index.htm, 2009.

[17] *Total Uninstall – Powerful Installation Monitor and Advanced Uninstaller*, http://www.martau.com/tu.php, 2009.

[18] W. A. Jansen, Intrusion detection with mobile agents, *Computer Communications*, vol.25, no.15, pp.1392-1401, 2002.

[19] J. Cao, X. Wang and S. K. Das, A framework of using cooperating mobile agents to achieve load sharing in distributed web server groups, *Future Generation Computer Systems*, vol.20, no.4, pp.591-603, 2004.
[20] *Yahoo Messenger*, http://messenger.yahoo.com/, 2009.
[21] *Msn Messenger*, http://messenger.msn.com/, 2009.
[22] *Fire Fox*, http://www.mozilla.com/en-US/firefox/, 2009.
[23] W. Qu, H. Shen and J. Sum, New analysis on mobile agents based network routing, *Applied Soft Computing*, vol.6, no.1, pp.108-118, 2005.
[24] *Download Accelerator Plus – DAP*, http://www. speedbit.com/, 2009.
[25] *Almeza MultiSet Automatic Program Installer*, http://www.almeza.com/.
[26] *Unattended Installation Procedure for Specific Softwares*, http://www.robvanderwoude.com/unattendedapps.php.
[27] *SourceForge, Silent and Unattended Installer*, http://unattended.sourceforge.net/installers.php, 2009.
[28] G. P. Picco, Mobile agents: An introduction, *Microprocessors and Microsystems*, vol.25, no.2, pp.65-74, 2001.
[29] A. R. Tripathi, T. Ahmed and N. M. Karnik, Experiences and future challenges in mobile agent programming, *Microprocessors and Microsystems*, vol.25, no.2, pp.121-129, 2001.
[30] K. Spreitzer, *UnInstaller*, http://en.wikipedia.org/wiki/Uninstaller, 2009.
[31] *Perfect Uninstaller V6.3*, http://www.perfectuninstaller.com/, 2009.
[32] M. Al-Kasassbeh and M. Adda, Analysis of mobile agents in network fault management, *Journal of Network and Computer Applications*, vol.31, no.4, pp.699-711, 2008.
[33] *Adobe Reader*, http://www.adobe.com/products/acrobat/readstep2.html.
[34] K. Kulesza, Z. Kotulski and K. Kulesza, On mobile agents resistance to traffic analysis, *Electronic Notes in Theoretical Computer Science*, vol.142, pp.181-193, 2006.
[35] M. J. O'Grady and G. M. P. O'Hare, Mobile devices and intelligent agents – Towards a new generation of applications and services, *Information Sciences*, vol.171, no.4, pp.335-353, 2005.
[36] *WinZip*, http://www.winzip.com/, 2009.
[37] *Wise InstallMaster Installer*, http://www.wise.com, 2009.
[38] *Microsoft Office*, http://office.microsoft.com/, 2009.
[39] S. Chou, S. Lin and C. Li, Dynamic parking negotiation and guidance using an agent-based platform, *Expert Systems with Applications*, vol.35, no.3, pp.805-817, 2008.
[40] A. Pan, S. Y. S. Leung, K. L. Moon and K. W. Yeung, Optimal reorder decision-making in the agent-based apparel supply chain, *Expert Systems with Applications*, vol.36, no.4, pp.8571-8581, 2009.
[41] Y.-F. Chung, T.-Y. Chen and C.-H. Liu, Efficient hierarchical key management scheme for access control in the mobile agent, *International Journal of Innovative Computing, Information and Control*, vol.4, no.9, pp.2375-2390, 2008.
[42] J.-Y. Huang, Y.-F. Chung, T.-S. Chen and I-E. Liao, A secure time-bound hierarchical key management scheme based on ECC for mobile agents, *International Journal of Innovative Computing, Information and Control*, vol.6, no.5, pp.2159-2170, 2010.
[43] B.-H. Yuan, H.-Y. Chou and C. C. Yang, A selector index for mobile agent dispatch schedules in context-free Internet applications, *International Journal of Innovative Computing, Information and Control*, vol.6, no.5, pp.2319-2330, 2010.
[44] Y.-L. Lin and C.-L. Hsu, Cryptanalysis and improvement of a hierarchical key management scheme for access control in the mobile agent, *ICIC Express Letters*, vol.4, no.1, pp.183-188, 2010.