Application Based Extension for Windows Network Monitor Widget

Kumarasinghe C.U.
Department of Computer Science and Engineering
University of Moratuwa, Sri Lanka
Email: chamath.10@cse.mrt.ac.lk

Abstract **- This paper was written to give an idea about the windows network monitor extension which was developed in order to improve system security and application accountability of windows processes and to help the user to be aware of how much bandwidth is used by each process.**

I. Introduction

This document is a guide to refer on how the network monitor extension was developed and how it was implemented. This also includes designs, results and quality assurance work which were done in this project.

The objective of the project was to extend the network monitor widget to an advanced version by including concurrent network monitoring for each application that is using network services on Windows.

Early research suggested that developing a Windows gadget from scratch is too difficult and cannot be completed in the scope of this project. Therefore existing open source projects of similar nature were examined and application based network monitoring functionality was implemented by extending such applications.

This gadget will run only on Windows Vista and Windows 7 operating systems. Future plans include porting the application to Linux operating systems.

The project was developed by the RUP (Rational Unified Process) model in development. The major milestones were the mid evaluation and the final evaluation. Other milestones were the Project Proposal, Project Feasibility Document, Project Vision Document System Requirement Specification, System Architecture and Design.

II. Literature review

The Need for the Project

There are many applications that provide monitoring of network bandwidth usage of application on windows machines but there are no widgets that provide the same functionality.

The widgets (gadgets in windows) that are on the market only provide the view of overall bandwidth usage. It is necessary for the user to see whether any unwanted processes are using bandwidth, as it may be security risk. The user can then take actions to minimize unwanted bandwidth usage.

Technical Requirements

This project required good knowledge on WMI (Windows Management Instrument) [20] classes in the windows framework. These are written in C++ and this was necessary to access the required data on the windows machine. For the development of the windows gadget, knowledge on XML, HTML, JavaScript and CSS was required. The Network Monitor II [19] gadget and many other similar gadgets are in the market but only a select few gadgets are open source.

Importance

Product perspective of the gadget came with the idea of improving user experience by adding accountability to applications which consumes network bandwidth. Product functions that are associated with any other gadgets such as a proper GUI & accurate figures in a timely manner are considered. User characteristics of an average user, a user without high IT knowledge was considered. Time constraint was the key factor in this project as dates of the deliverables of the project are fixed and very limited.

Stakeholder Responsibilities

-Ensures that the project is delivered on time with the required deliverables.
-Ensures that the system will be maintainable.
-Ensures that there will be a market demand for the product’s features.
-Monitors the project’s progress.
-Whether any copyright issues may arrive.
-Whether it is possible to integrate to other systems and distribute it with other products.

Product Requirements

Functionality

Give Details of the Applications That Are Taking Bandwidth: Provide details of the applications that are using bandwidth which are currently running in the computer.

Save Prior Details of Application Bandwidth Usage: The gadget must keep history of application that uses bandwidth with the intension using it for analysis purposes.

Give Graph View of Past Usage of Application: Provide information about peak usage time, peak usage speed & total cumulative bandwidth.

Usability

Skill Level of a User: Basic knowledge on Windows Gadgets is required.

Training Time to Be a Competent User: A day of using the gadget is enough to be competent at using it.

Reliability

Availability: The gadget is available for the user 24/7 and the extension at user request.

Accuracy: The information given, the actual figures are at least 99% accurate.

Failure of Information Storage: Once the gadget is installed it should keep information until uninstall happens or a format occurs.

Performance

Start-up Time: Start-up Time for the gadget should not exceed 10 seconds.

Concurrent Updates: Update on the details of usage should change at least once per second.

Number of Application: At any given moment at least details with regard to 5 applications is displayed.

Hardware Interfaces

This gadget supports any LAN connection module and it will work for any USB Mobile (Dongle) Interface as well as any Wi-Fi network even with proxy servers.

Software Interfaces

It is an extension for an existing gadget Windows Monitor II, it needs to interact with it and open as a flyout from the parent widget. Windows PerfRawData [11] class & similar classes from the Windows API system were accessed in order to get performance related data.

Licensing Requirements

The existing gadget has not attached with it any licensing information but holds copyrights for Igor "Igogo" Bushin. [19]

III. DESIGN & IMPLEMENTATION

A. *Use cases*

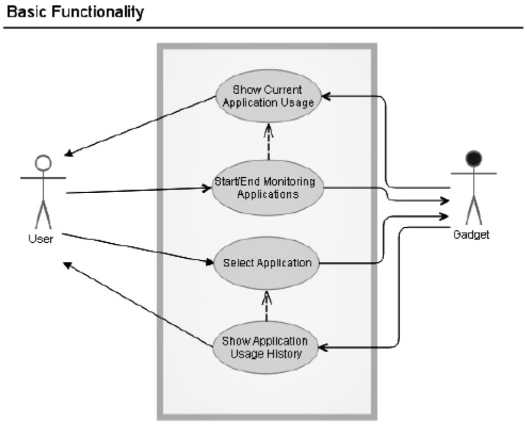

Basic Functionality
Show Current Application Usage
Start/End Monitoring Applications
Select Application
Show Application Usage History
User
Gadget

Fig 1 : Use Case 1

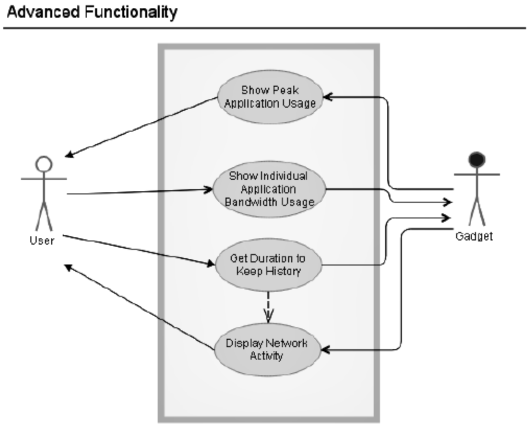

Advanced Functionality
Show Peak Application Usage
Show Individual Application Bandwidth Usage
Get Duration to Keep History
Display Network Activity
User
Gadget

Fig. 2 : Use case 2

B. *View*

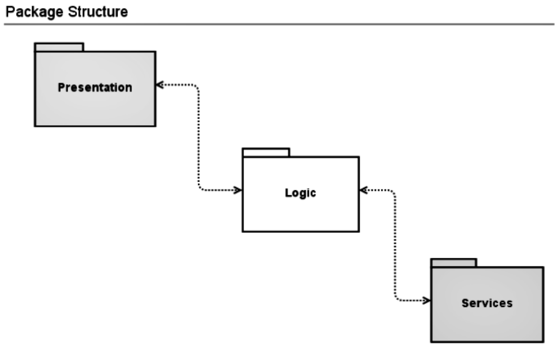


Fig. 3 : View

C. Implementation

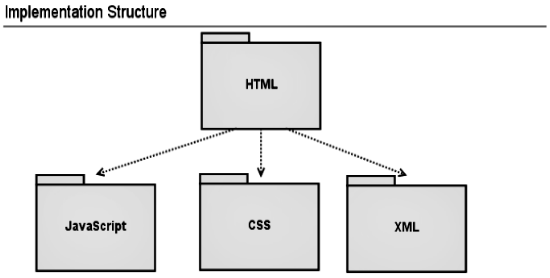


Fig. 4 : Implementation

The existing gadget gives a platform to build the project's new features. It was used to deploy the extended flyout. The default classes in the Windows Performance Counter were used as there is no other foreseeable way to do the implementation. As the main programming language was JavaScript, architectural design was limited to the scope of that domain. Main goal of this architecture was to be a guideline for the implementation process where necessary.

D. GUI

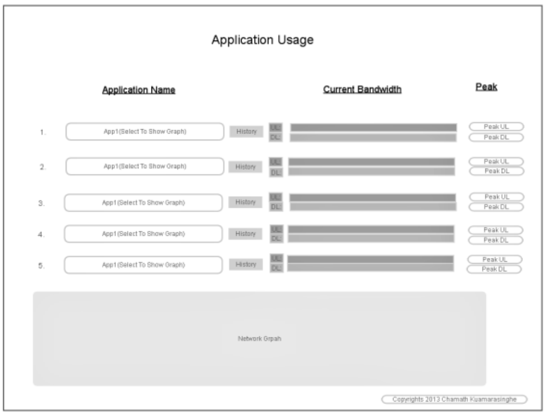
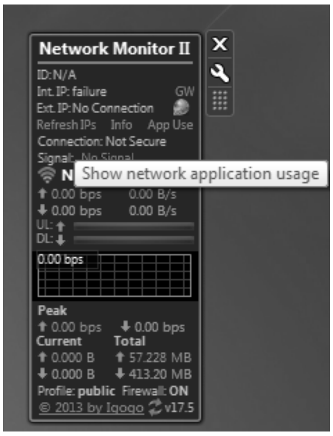


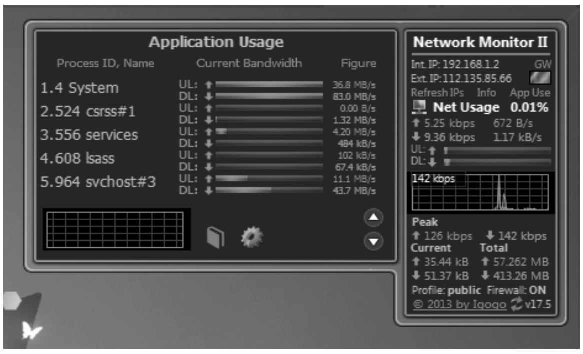
Fig. 5 : GUI

IV. RESULTS



Network Monitor II
ID:N/A
Int. IP: failure GW
Ext. IP:No Connection
Refresh IPs Info App Use
Connection: Not Secure
Signal: No Signal
N
Show network application usage
0.00 bps 0.00 B/s
0.00 bps 0.00 B/s
UL:
DL:
0.00 bps
Peak
0.00 bps 0.00 bps
Current Total
0.000 B 57.228 MB
0.000 B 413.20 MB
Profile: public Firewall: ON
© 2013 by Igogo v17.5

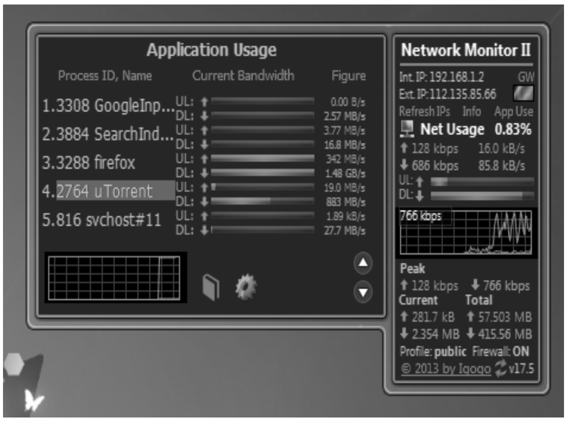
Fig. 6 : Results



Application Usage
Process ID, Name Current Bandwidth Figure
1.4 System UL: 36.8 MB/s DL: 83.0 MB/s
2.524 csrss#1 UL: 0.00 B/s DL: 1.32 MB/s
3.556 services UL: 4.20 MB/s DL: 484 kB/s
4.608 lsass UL: 102 kB/s DL: 67.4 kB/s
5.964 svchost#3 UL: 11.1 MB/s DL: 43.7 MB/s

Network Monitor II
Int. IP: 192.168.1.2 GW
Ext. IP:112.135.85.66
Refresh IPs Info App Use
Net Usage 0.01%
5.25 kbps 672 B/s
9.36 kbps 1.17 kB/s
UL:
DL:
142 kbps
Peak
126 kbps 142 kbps
Current Total
35.44 kB 57.262 MB
51.37 kB 413.26 MB
Profile: public Firewall: ON
© 2013 by Igogo v17.5

Fig. 7 : Results 2



Application Usage
Process ID, Name Current Bandwidth Figure
1.3308 GoogleInp... UL: 0.00 B/s DL: 2.57 MB/s
2.3884 SearchInd... UL: 3.77 MB/s DL: 16.8 MB/s
3.3288 firefox UL: 342 MB/s DL: 1.48 GB/s
4.2764 uTorrent UL: 19.0 MB/s DL: 883 MB/s
5.816 svchost#11 UL: 1.89 kB/s DL: 27.7 MB/s

Network Monitor II
Int. IP: 192.168.1.2 GW
Ext. IP:112.135.85.66
Refresh IPs Info App Use
Net Usage 0.83%
128 kbps 16.0 kB/s
686 kbps 85.8 kB/s
UL:
DL:
766 kbps
Peak
128 kbps 766 kbps
Current Total
281.7 kB 57.503 MB
2.354 MB 415.56 MB
Profile: public Firewall: ON
© 2013 by Igogo v17.5

Fig. 8 : Results 3

The above figure shows how the fly out is initialized by the main gadget. Self-Explaining UI is used to help the user with the operation.

The GUI shows the following attributes.

-List of Applications that are currently doing I/O operations.

-The process ID is given with the process name of the application.

-Upload and Download speed is given in the bar chart with respect to other applications that are shown currently on the flyout.

-Clicking on a process name will activate the graph which will keep history up to 110 seconds.

-Using the scroll button you can view all the applications that are using I/O currently (which has more than 500 I/O operations currently).

-The bar chart displays what processes is taking the most upload or download percentage in the given 5 processes.

The above figure shows how a selection of a particular application would track the I/O usage of that particular application.

V. RELATED WORK

Quality Assurance

It was not be possible to implement a “zero defect policy”. It may be possible to give some guidelines when it comes to quality. Some of these matrices are:

-*Comment Density*. The number commented lines are there in the source code when compared with actual implementation lines. – **7%**

-*Number of Instructions* per Function - ***46***

-*Length of Function* should not be greater than 100 lines - Largest function has **378** statements in it, while median is **4**.

- *Cyclomatic Complexity* - The most complex function has a CC value of **22**.

-Number of *total lines* of code – **1258**

-Number of functions - **26**

B. Unit Testing

-Check number formatting.

-Check max value is taken from All Apps.

-Check application number is read.

-Check max is taken from uploads and downloads.

-Check graph details.

VI. CONCLUSION

It can be said that the project was successful although some requirements could not be achieved with given time and available technology as most of windows source code is private. Considerable testing was done using qunit in order to assure there aren’t any unwanted bugs in the system.

REFERENCES

[1] D. Garlan and M. Shaw, “An Introduction to Software Architecture” Jan 1994.

[2] System.Shell.execute method (Windows). Available: http://msdn.microsoft.com/en-us/library/windows/desktop/ms723232(v=vs.85).aspx.

[3] Start Database window. Available: http://dcx.sybase.com/index.html#sa160/en/sacshelp/server-start-database-dlg.html.

[4] Creating a Database-Driven Vista Gadget – CodeProject. Available: http://www.codeproject.com/Articles/22001/Creating-a-Database-Driven-Vista-Gadget.

[5] Risk I/O — Attack Risk Monitoring & Ranking Service. Available: https://www.risk.io/?utm_source=nmap&utm_medium=sponsorship&utm_content=nmap-spaceship&utm_campaign=homer.

[6] WQL (SQL for WMI) (Windows). Available: http://msdn.microsoft.com/en-us/library/aa394606(v=vs.85).aspx.

[7] WbemFlagEnum (Windows) [Online]. Available: http://msdn.microsoft.com/en-us/library/windows/desktop/aa393980(v=vs.85).aspx.

[8] WbemFlagEnum (Windows) [Online]. Available: http://msdn.microsoft.com/en-us/library/windows/desktop/aa393980(v=vs.85).aspx.

[9] How to Debug a Windows Vista Sidebar Gadget With Visual Studio[Online]. Available: http://www.howtogeek.com/howto/windows-vista/how-to-debug-a-windows-vista-sidebar-gadget-with-visual-studio.

[10] Insert data into MySQL using php, jquery, Ajax[Online]. Available: http://zwaroop.wordpress.com/2011/12/27/insert-data-into-mysql-using-php-jquery-ajax/.

[11] Win32_PerfRawData_PerfProc_Process class (Windows) [Online]. Available: http://msdn.microsoft.com/en-us/library/aa394323(v=vs.85).aspx.

[12] Shape Element (VML) (Windows) [Online]. Available: http://msdn.microsoft.com/en-us/library/bb263897(v=vs.85).aspx.

[13] Gadgets: Build Your Own Windows Vista Sidebar Gadget[Online]. Available: http://msdn.microsoft.com/en-us/magazine/cc163370.aspx.

[14] VML CoordOrigin Attribute (Windows) [Online]. Available: http://msdn.microsoft.com/en-us/library/bb263836(v=vs.85).aspx.

[15] Getting started with Windows 7 Gadgets[Online]. Available: http://www.i-programmer.info/projects/38-windows/2904-getting-started-with-windows-7-gadgets.html?start=2.

[16] Gadgets for Windows Sidebar - Gadget Object Reference (Windows) [Online]. Available: http://msdn.microsoft.com/en-us/library/windows/desktop/bb508511(v=vs.85).aspx.

[17] Developing a Gadget for Windows Sidebar Part 3: Settings and Flyouts (Windows) [Online]. Available: http://msdn.microsoft.com/en-us/library/windows/desktop/bb655904(v=vs.85).aspx#_sidebar_basic_Flyout.

[18] Win32_PerfRawData_Tcpip_NetworkInterface class (Windows) [Online]. Available: http://msdn.microsoft.com/en-us/library/windows/desktop/aa394340(v=vs.85).aspx

[19] Network Monitor II Widget [Online]. Available :http:// http://www.myfavoritegadgets.info/

[20] Windows Management Instrumentation [Online]. Available: http://msdn.microsoft.com/en-us/library/aa394582%28v=vs.85%29.aspx