

Czech Technical University in Prague
Faculty of Electrical Engineering

Department of Computer Graphics and Interaction

DIPLOMA THESIS ASSIGNMENT

Student: **Bc. Jan Kolařík**

Study programme: Open Informatics
Specialisation: Software Engineering

Title of Diploma Thesis: **Consistent multi-modal user interface prototype
for Windows OS**

Guidelines:

Research the most notable aspects of user interface quirks of built-in applications designed for both mouse/keyboard and touch input in Microsoft Windows 8.1. Perform tests to identify OS users' biggest struggles controlling these applications.

Design and describe a new universal interface merging the touch and desktop parts of Windows 8.1, describe possible ways to make most GUI widgets usable in both modes of input.

Develop UI prototypes of applications from previous tests in accordance with the new universal user interface descriptions and recommendations from initial test results. Test these prototypes with users as well to verify their usability.

Bibliography/Sources:

Windows 8 - User experience guidelines

http://download.microsoft.com/download/C/0/A/C0AEF0CC-B969-406D-989A-4CDAFDBB3F3C/Win8_UXG_GA.pdf

User Experience Interaction Guidelines for Windows 7 and Windows Vista

<http://www.microsoft.com/en-us/download/confirmation.aspx?id=2695>

C. M. Barnum: Usability Testing Essentials. Ready, Set ... Test!, Elsevier - Morgan Kaufmann, 2011

Y. Rogers, H. Sharp, J. Preece - Interaction Design: Beyond Human - Computer Interaction, Wiley, 2011

Diploma Thesis Supervisor: Ing. Ivo Malý, Ph.D.

Valid until the end of the summer semester of academic year 2014/2015



prof. Ing. Jiří Žára, CSc.
Head of Department

prof. Ing. Pavel Ripka, CSc.
Dean

Prague, February 25, 2014

Czech Technical University in Prague
FACULTY OF ELECTRICAL ENGINEERING
DEPARTMENT OF COMPUTER GRAPHICS AND INTERACTION



Diploma Thesis

Consistent multi-modal user interface prototype for Windows OS

Bc. Jan Kolařík

January 5th, 2015

Supervisor:
Ing. Ivo Malý, Ph.D.

Study program:
Open informatics

Specialization:
Software engineering and interaction

ACKNOWLEDGMENTS

I would like to thank my supervisors Ing. Ivo Malý, Ph.D. from Czech Technical University in Prague and Robert Roth, Ph.D. from University of Wisconsin–Madison for supporting my work and providing valuable feedback. I also also appreciate the support of my friends and family.

D E C L A R A T I O N

I hereby declare that this thesis is my own work and that I used only the sources (literature, articles, software) stated in references.

I have no objection to usage of this work in compliance with the Czech act §60 (Zákon č. 121/2000 Sb. – copyright law) and with the rights connected with the copyright law including changes in the law.

In Prague on January 5th, 2015

.....

Bc. Jan Kolařík

Abstract

This thesis researches the history of user interfaces in Microsoft Windows operating systems and describes the historical circumstances that led Microsoft to introduce a multi-modal user interface in Windows 8. Since Windows 8 received criticism from both professional reviewers and all categories of users in terms of usability and user experience, a major part of the thesis analyzes and lists usability drawbacks of Windows 8's interface. Next, these listed drawbacks were used as assumptions and task list base for a qualitative software usability test with users. Verified assumptions were later used to compile a list of design principles for a prototype of a consistent multi-modal user interface for possible future versions of Microsoft Windows.

A low-fidelity and a medium-fidelity prototype were constructed based on these principles and tested with users again to bring final recommendations for possible implementation in an actual operating system.

Keywords: usability, user interfaces, user experience, Windows 8, UI design, UI prototyping

Abstrakt

Tato práce se zabývá historií uživatelských rozhraní v operačních systémech Microsoft Windows a popisuje historické souvislosti, které vedly Microsoft k uvedení multimodálního uživatelského rozhraní ve verzi Windows 8. Jelikož byl systém často kritizován z hlediska použitelnosti uživatelského rozhraní jak profesionálními recenzenty, tak samotnými uživateli různých kategorií, jedna z hlavních částí práce zkoumá a popisuje nedostatky tohoto rozhraní. V další části byl tento seznam nedostatků použit jako předpoklady a předloha k seznamu úkolů v rámci kvalitativního testu použitelnosti softwaru s uživateli. Ověřené nedostatky byly dále použity k vytvoření seznamu zásad návrhu prototypu konzistentního multimodálního uživatelského rozhraní pro možné příští verze systému Microsoft Windows.

Rovněž byly s použitím těchto zásad vytvořeny low-fidelity a medium-fidelity prototypy rozhraní a následně otestovány v dalším testu použitelnosti s uživateli. Výsledky tohoto testu posloužily jako doporučení pro možnou implementaci do skutečného operačního systému.

Klíčová slova: použitelnost, uživatelská rozhraní, Windows 8, návrh uživatelského rozhraní, prototypy uživatelského rozhraní

Table of contents

1 Introduction.....	15
1.1 Motivation.....	16
1.2 Historical circumstances of the introduction of Windows 8.....	17
1.3 The evolution of Windows user interfaces.....	18
1.3.1 Traditional desktop Windows UI.....	18
1.3.2 The standalone “metro” interface.....	21
1.3.3 Multi-modal user interface in Windows 8.....	23
2 Usability and user experience drawbacks of Windows 8.....	27
2.1 Application style inequality.....	27
2.2 Schizophrenic applications.....	29
2.3 Inconsistency of design decisions.....	31
2.4 Limited usability of the entire system with one particular input method.....	31
2.5 Basic PC controls have inconsistent style.....	33
3 Usability test of Windows 8.....	35
3.1 Test tasks.....	36
3.2 Screener survey.....	38
3.3 Other surveys.....	39
3.4 Test analysis.....	41
3.4.1 Participant 1.....	41
3.4.2 Participant #2.....	42
3.4.3 Participant #3.....	43
3.5 Test results.....	45
4 Prototype design.....	47
4.1 UI philosophy layers.....	47
4.2 The “basic controls” feature.....	48
4.3 All applications treated equally.....	50
4.4 Unified visual style.....	51
4.5 No schizophrenic applications.....	52
4.6 Touch mode.....	52
4.7 On-screen keyboard.....	52
4.8 Backwards compatibility rules.....	53
4.9 Touch gestures.....	53
5 Prototype testing.....	55
5.1 Low-fidelity prototype test.....	55
5.1.1 Test tasks.....	55
5.1.2 Screener survey.....	56
5.1.3 Test analysis and results.....	57
5.2 Medium-fidelity prototype test.....	58
5.2.1 Test results.....	59

6 Conclusions.....	61
6.1 Ideas for future thesis expansion.....	62
References.....	63
A Abbreviations listing.....	67
B Testing log sample.....	69
C CD-ROM contents.....	73
D Context notes.....	75

Figures listing

1.1 Presentation of the original thesis idea.....	17
1.2 Inconsistent wizard styles.....	17
1.3 UI philosophy layers of Windows 3.x.....	19
1.4 UI philosophy layers of Windows 95 to 2000.....	19
1.5 “Windows-branded” environment in Windows Vista.....	20
1.6 UI philosophy layers of Windows XP to 7.....	20
1.7 “Metro” interface timeline.....	21
1.8 “Metro” interface visual history.....	22
1.9 UI philosophy layers of Windows 8.....	24
1.10 Windows Explorer and Media Player UI comparison.....	25
2.1 Application types not treated equally.....	28
2.2 Input fields moving out of range.....	32
2.3 The “spinner” UI widget.....	32
2.4 Czech on-screen keyboard.....	32
2.5 Czech (QWERTY) standard keyboard layout.....	33
2.6 SmartScreen vs. User Access Control interface.....	33
3.1 Usability test setup.....	35
4.1 Desktop splitting.....	47
4.2 UI philosophy layers of the prototype.....	48
4.3 The basic controls.....	49
4.4 Prototype Start menu philosophy.....	49
4.5 Window listing feature.....	50
4.6 Full and non-full screen mode applications.....	51
4.7 Microsoft Office 2013 interface.....	51
5.1 Low-fidelity prototype test setup.....	58

Tables listing

1.1 Windows 8 and RT comparison.....	23
2.1 Schizophrenic built-in Windows 8 applications.....	30
3.1 Windows 8 testing screener survey results.....	39
3.2 Windows 8 testing pre-test survey results.....	40
3.3 Windows 8 testing post-test survey results.....	40
3.4 Windows 8 testing conclusion.....	45
5.1 Low-fidelity test participants.....	57
5.2 Verified an unverified assumptions of low-fidelity prototype.....	57
5.3 Medium-fidelity prototype testing participants.....	59
5.4 Medium-fidelity prototype testing results.....	59

Chapter 1

Introduction

Context and terminology note: All Windows 8 testing and prototype designing was done in February and March 2014. The term “Windows 8” used throughout the thesis refers to both Windows 8 (including the Pro edition) and 8.1 (excluding Update 1, which was officially released after the testing). Whenever the term “Windows 8.1” is used, it refers only to Windows 8.1 – excluding the previous Windows 8 RTM release.

This thesis covers the design and implementation process of a multi-modal user interface prototype for Microsoft Windows operating system. The prototype’s multimodality means a combination of several user input methods:

- the traditional method of a mouse or a trackpad and a physical keyboard,
- the capacitive touchscreen method,
- the capacitive touchscreen method combined with a physical keyboard,
- a combination of all above.

Initial part of this process includes an analysis of the most notable usability drawbacks in Microsoft Windows 8’s current multi-modal user interface. Results of the analysis were verified by a series of qualitative usability tests with actual users. These test results served as a base for an early low-fidelity prototype of a new user interface avoiding the verified drawbacks.

As a part of the design process another usability test followed – users performed tasks interacting with these low-fidelity paper prototypes in order to identify potential other usability quirks.

Next, the paper low-fidelity prototype was adjusted in accordance with the second test results and converted into a medium-fidelity one running as a standalone Windows application. A similar qualitative test with users followed to bring conclusions and allow for comparison of the current Windows 8 multi-modal interface with the prototype.

A part of the thesis analyzes the circumstances that led Microsoft to release an operating system containing a multi-modal interface that received generally mixed to negative reviews from critics [1] and cold reception from consumer market users. [2]

1.1 MOTIVATION

Some of the major selling points of desktop Windows have always been the following:

- **Backwards compatibility:** The Windows team has been putting enormous effort into making sure that backwards compatibility for as many applications as possible is kept. Microsoft engineers even directly cooperate with third party Windows software vendors to ensure that their popular software from the past relying on undocumented features runs in new releases of Windows. [3]
- **Large application catalog:** This selling point is a by-product of the above. Thanks to the well-maintained backwards compatibility some very specific applications are available for this platform. [4]

These two selling points were a particularly important aspect in the process of choosing an operating system in the past. Lately, however, due to the success of multi-platform applications, platform independent development frameworks, cloud technologies and virtualization, it matters increasingly less which operating system to choose. [5] This trend is mostly observed in the consumer segment; businesses, though, are starting to follow.

As a long-time power user of Windows as well as a technology enthusiast, for almost 15 years I have been providing technical support to dozens of my friends and acquaintances, often involving the use of desktop Windows. Throughout these years I was able to see what the major issues of consumer segment Windows users are. Many of these issues were usability-related and could have been avoided if Microsoft had implemented what Aero Taskforce [33] (or Windows 7 Taskforce) community members had suggested or designed. Also, some of these usability quirks turned out to be a deal breaker for certain consumers when choosing an operating system for their new computer [6].

At the same time, throughout the years I saw many areas where the concept of desktop Windows excels – which made me think of publishing a thesis on ways desktop Windows should change to stay relevant in the consumer market where the two named Windows selling points are becoming increasingly less significant and competition among operating systems is stiffer than ever (Chrome OS, Mac OS X) [7].

My original thesis idea (figure 1.1) was to examine the entire user interface of Windows 7, find the most notable quirks or elements that do not comply with the official UI guidelines (figure 1.2), and – unlike Windows 7 Taskforce – design high-fidelity prototypes of suggested replacements of the parts of the user interface affected.

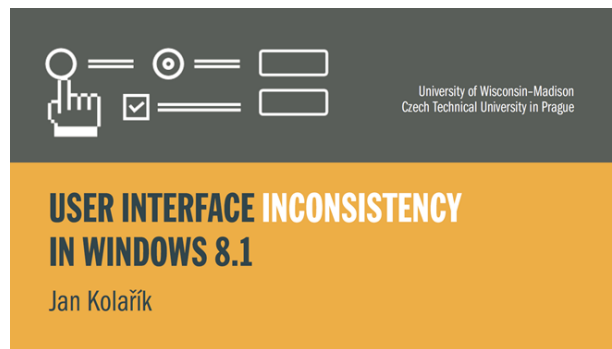


Figure 1.1: Presentation of the original thesis idea

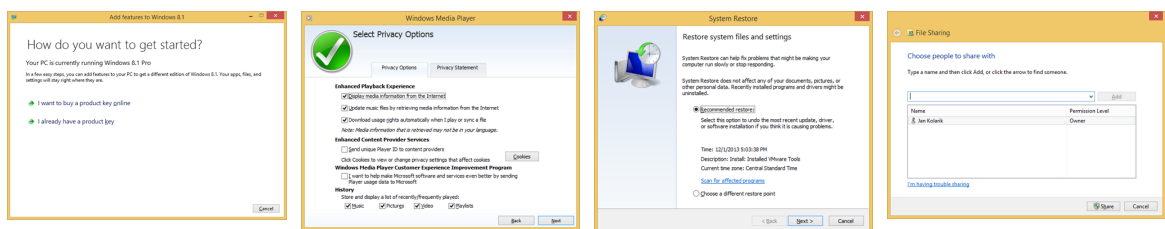


Figure 1.2: Desktop Windows UI elements not complying with the official guidelines – four different Wizard styles (with only the rightmost one being guidelines-compliant)

With the introduction of Windows 8 in 2012, however, Microsoft started using a new multi-modal user interface (for tablets, desktop, and laptop computers) which turned out to be a much more confusing aspect of user interface to consumers. After facing a lot of criticism from the media, with each subsequent update to the OS Microsoft turned away from the initial idea of a multi-modal interface and started fragmenting it again, breaking the original UI philosophy [8].

As I – along with several respected technology journalists – believe that the multi-modal interface could become one of the new selling points of Windows [9] in the consumer market, the possibilities of improvement of this multi-modal user interface rather than turning away from it served as the core topic of my thesis instead [10].

1.2 HISTORICAL CIRCUMSTANCES OF THE INTRODUCTION OF WINDOWS 8

As mentioned in the “Motivation” section, the key selling points of Microsoft Windows have been becoming increasingly less important, particularly in the consumer segment. Plus, the sales of traditional personal computers have been decreasing for years now in favor of mobile devices with simplified operating systems. Traditionally, Microsoft’s two core products for the consumer market have been Xbox and desktop Windows with the latter being negatively affected by the personal computer sale decreases [11].

Until 2007, Microsoft had been the leader in mobile operating systems for touchscreen devices. [12] With the introduction of the iPhone, though, the way people interact with touchscreen-enabled mobile devices changed dramatically: user interfaces evolved from being designed for use with styluses to being designed for being controlled with bare fingers. Microsoft was able to adapt to this change relatively quickly on pocket mobile devices with the release of a brand new operating system with a UI designed from the scratch – “metro” on Windows Phone. On larger touchscreen devices – tablets – however, Microsoft tried gaining an advantage over its competitors by modifying an existing operating system – desktop Windows – for use with both touchscreen devices and traditional mouse/trackpad + keyboard devices, making full-featured computing possible on capacitive screen tablets. While the original intentions behind this step were certainly positive, the actual implementation by Microsoft caused a lot of confusion among users when the final product – Windows 8 – was released.

1.3 THE EVOLUTION OF WINDOWS USER INTERFACES

In order to understand the philosophy and Microsoft’s decisions behind Windows 8 user interface, the history of two separate UI philosophies in Microsoft products is important since the current philosophy is a blend of these.

1.3.1 Traditional desktop Windows UI

Note: Only desktop (non-server) versions of Windows are taken into account for this section.

The most fundamental principles of the traditional desktop Windows UI have not changed since the very first release of the OS – Windows 1.0 in 1985. As the name of the OS implies – a core element of the UI philosophy are windows. The UI hierarchy, however, has undergone mostly minor changes throughout the releases.

Windows 3.x and Windows NT 3.x releases featured two basic UI hierarchy layers (figure 1.3). The bottom level layer was represented by a desktop tied to a physical screen [13]. The desktop did not have any interactive elements such as icons for launching applications or menus besides the list of currently open or minimized windows. Windows, then, were the top element in UI hierarchy.

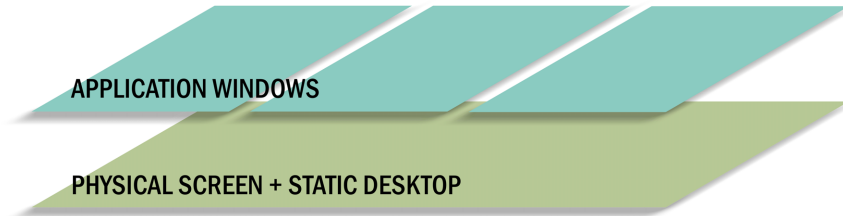
Windows 3.0 | 3.1 | 3.11 | NT 3.1 | NT 3.5 | NT 3.51

Figure 1.3: User interface philosophy layers of Windows 3.x

The release of Windows 95 changed the UI philosophy (figure 1.4) by giving the desktop significantly more functions while still being tied to a physical screen. The desktop received right-click context functionality, started using Windows Explorer calls to display files and folders from a specified location along with special purpose icons such as “My Computer” or “Recycle Bin”. Interestingly, since Windows 95, the “Desktop” has been treated as a root location for the entire Windows Explorer including file selection system dialogs. Another addition to the desktop layer was a taskbar with a notification area and a button for accessing Start menu.

Windows 95 | NT 4.0 | 98 | ME | 2000

Figure 1.4: User interface philosophy layers from Windows 95 to Windows 2000

In 2001, Windows XP slightly changed the UI philosophy (figure 1.6) by adding another layer above the desktop [14] by default. This “Windows-branded” environment (figure 1.5) layer was used for loading screens after video driver had been initialized, for logon screens, lock screens, and, later (since Windows Vista) for the “Ctrl + Alt + Del” menu as well.

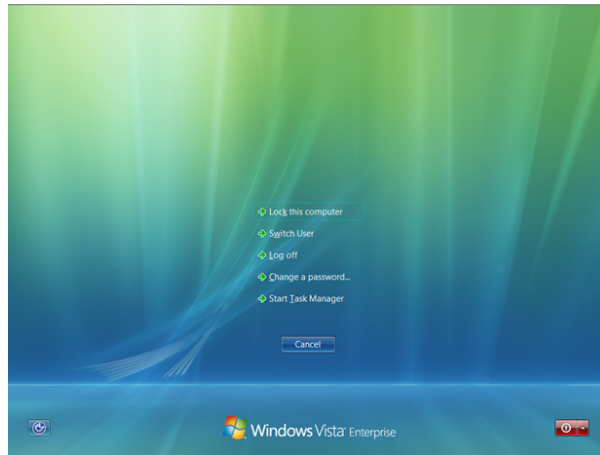


Figure 1.5: Example of the “Windows-branded” environment in Windows Vista

Windows XP | Vista | 7

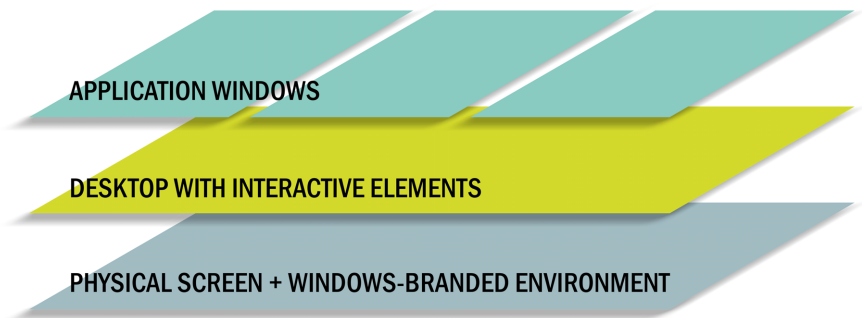


Figure 1.6: User interface philosophy layers of Windows XP/Vista/7

Another important aspect of the entire UI philosophy, apart from the element hierarchy, is the user experience consistency guideline manual for both third party application developers and in-house Windows developers. These guidelines went through many revisions and updates with all major releases of Windows. [15] However, even though these guidelines are very detailed, vast parts of the operating system fail to comply with them – in most cases they do not comply with the latest versions due to developers’ slow reaction times to the guideline updates [16].

1.3.2 The standalone “metro” interface

Note: Until the official introduction of Windows Phone 7 OS, there was no official name for this user interface philosophy/design language. At the unveiling of Windows Phone 7, Microsoft started using the name “metro” for their UI philosophy. In August 2012, though, the company stopped officially using the “metro” term. Although never officially confirmed, various sources state that the reason behind this action was a copyright dispute. For the purposes of this thesis, though, I will be using the term “metro” [17].

Definition: The design style’s main objective is to remove any chrome and extra decoration and allow users to interact directly with content – relying more on typography and less on graphics while keeping icons and graphics in a flat, 2-dimensional style [18].

History (figures 1.7, 1.8): The information of the very first occurrence of the “metro” style in Microsoft products vary depending on sources and the writer’s interpretation of the definition. Very early examples of the “metro” principles, although in a different graphical style, can be seen in the interfaces of the Media Center application in the Windows XP Media Center Edition 2005 operating system and the initial firmware versions of the 2006 Zune portable media players. In late 2006, Microsoft updated the Media Center application appearance with the release of Windows Vista operating system to a graphical style that most sources consider identical to the current “metro”.

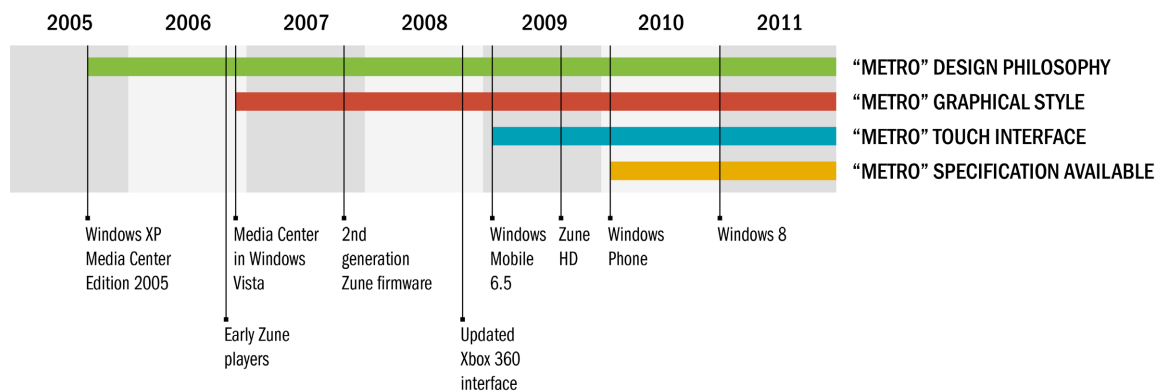


Figure 1.7: Timeline of the “metro” interface

A 2007 firmware update to the existing Zune players along with a new product line and a new version of the Zune PC software also featured this graphical style. On November 19th, 2008, a firmware update brought this style to the interface of Xbox 360 as well. On touchscreen devices, “metro” first appeared with the introduction of Windows Mobile 6.5 Smartphone Edition in February 2009. On September 15th, another touchscreen interface followed the “metro” trend: that of the new Zune HD portable media player. On February 15th, 2010, Microsoft officially defined this UI philosophy with the unveiling of Windows Phone 7 mobile operating system.

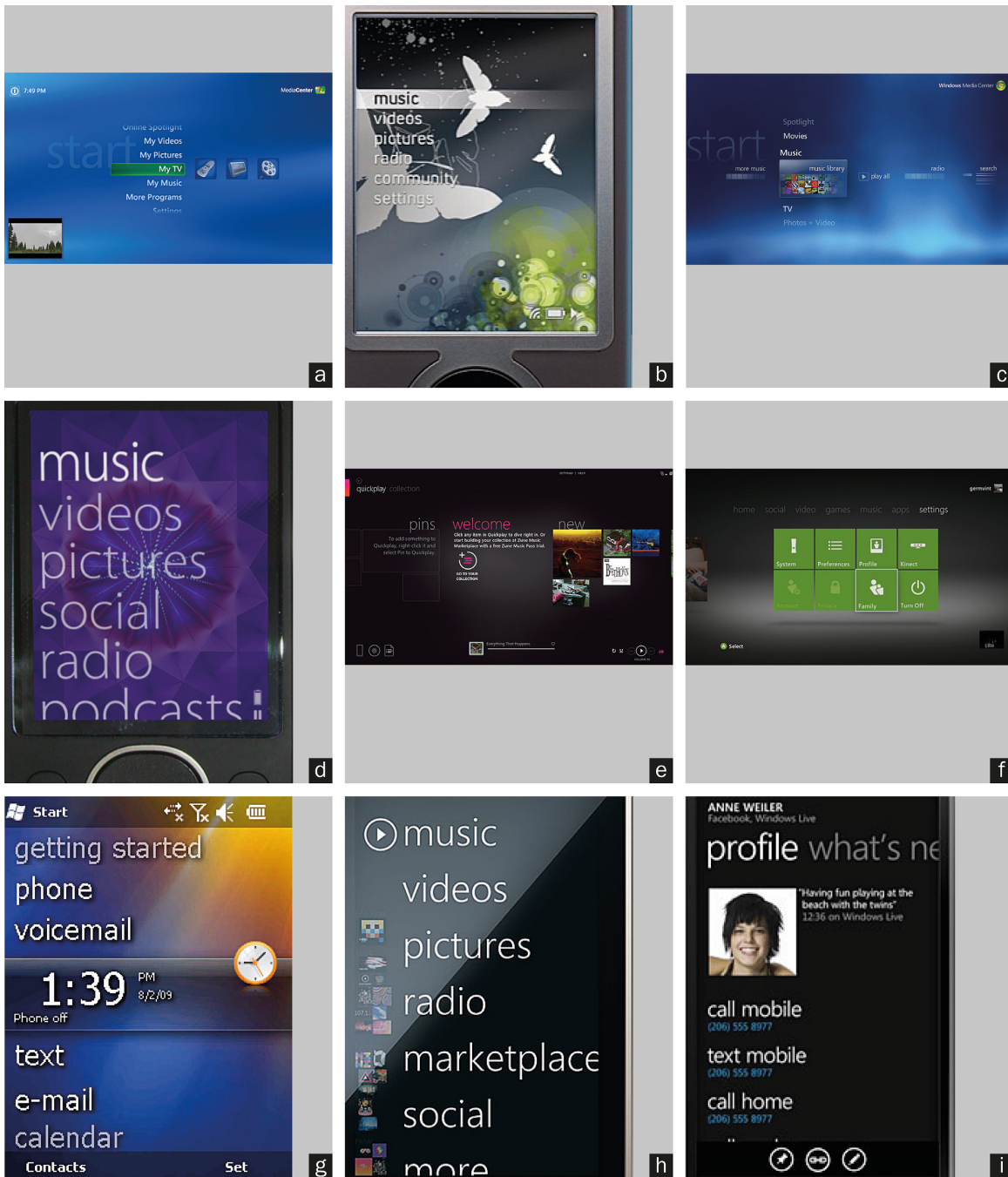


Figure 1.8: The “metro” interface’s visual history – Windows XP Media Center Edition 2005 (a); early Zune players (b); Media Center in Windows Vista (c); updated Zune interface along with a new Zune PC software (d and e); updated Xbox 360 interface (f); Windows Mobile 6.5 Smartphone Edition (g); Zune HD (h); Windows Phone 7 (i) [35]

1.3.3 Multi-modal user interface in Windows 8

The user interface of Windows 8 introduced the design principles of Windows Phone on desktop computers which allowed PC manufacturers to come up with devices lacking traditional input mechanisms (a combination of a physical keyboard and mouse/trackpad), or, on the other hand, combining these mechanisms with touchscreen input methods.

Windows 8 supported, along with traditional Windows API applications (running within a traditional desktop interface), a brand new format of third party applications: “metro-style applications” independent of processor architecture – with limited access to hardware devices – allowing only full screen (or, alternatively, split screen) display.

A special edition of Windows 8 – called Windows RT (table 1.1) – was released for computers powered by processors using the ARM architecture. Even though Windows RT looked almost identical to Windows 8 after a fresh installation, there were vast differences in software extensibility options between these two operating systems. While Windows 8 supported both “metro-style” applications and all Windows API applications from Microsoft as well as third party developers, Windows RT only supported third party “metro-style” applications. The only Windows API-based desktop applications supported by Windows RT were built-in system applications from Microsoft and Microsoft Office 2013 [19].

Feature	Included in Windows 8	Included in Windows RT
CPU architectures	x86, AMD 64	ARM
“Metro” interface	yes	yes
Desktop interface	yes	yes
Built-in “metro-style” applications	yes	yes
Support for third party “metro-style” applications	yes	yes
Built-in Windows API applications	yes	yes
Support for third party Windows API desktop applications	yes	no

Table 1.1: Comparison of Windows 8 and Windows RT

User interface philosophy layers in Windows 8 (figure 1.9): The bottommost layer consists of one or multiple physical screens. One layer above is divided into “metro environments”. Each metro environment can take up one or more screens. Another layer above this, each “metro environment” can be divided into sections corresponding with the amount of physical screens it consists of. Also, each of these sections can be optionally divided into halves – making sub-sections. One layer above, each sub-section from the previous layer can either be occupied by a “metro-style” application, or

a desktop. Desktops can contain any amount of application windows – making yet another layer of the UI philosophy.

Windows 8 | 8.1

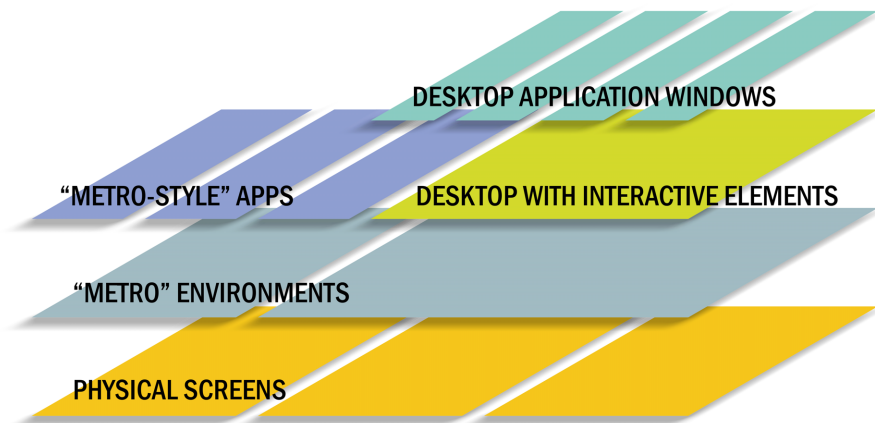


Figure 1.9: User interface philosophy layers of Windows 8

User experience guidelines for Windows 8: With the release of Windows 8, Microsoft designers prepared a detailed manual for third party developers of “metro-style” applications [20]. These guidelines contain all sorts of advice: from general user experience philosophy aspects to the recommended graphical style of icons. However, although the traditional desktop environment for Windows API applications is still the major and most heavily used part of the operating system, Microsoft never released any user experience guidelines for Windows API desktop applications in Windows 8 – leaving the Windows 7 user experience guidelines from 2010 the most recent. When it comes to the integral components of the desktop part of Windows 8, all of these components are philosophy-wise identical to those in Windows 7: either being obsolete (inherited from Windows versions older than Windows 7 and Windows Vista), or complying with the aforementioned 2010 guidelines. Graphical style-wise, however, Microsoft changed the appearance of most UI widgets and window borders to reflect the flat style of the “metro” environment. Thus, built-in applications and dialogs that use system calls to load UI widgets and bitmap images/icons, match with the overall appearance of the OS. However, numerous built-in applications and dialogs using custom UI widgets and bitmap images/icons were not updated to reflect these visual changes (these widgets remain in the Windows 7 visual style, e. g. Windows Media Player, figure 1.10), while some others were (e. g. Windows Explorer, – figure 1.10 Task Manager), contributing to the already inconsistent user experience. What’s more, these visual changes were never documented in any official articles or

– as mentioned above – guideline documents, leaving developers who wish to keep their desktop application visual styles consistent with those that were built into the system only with outdated Windows 7 guidelines and a need for improvisation.

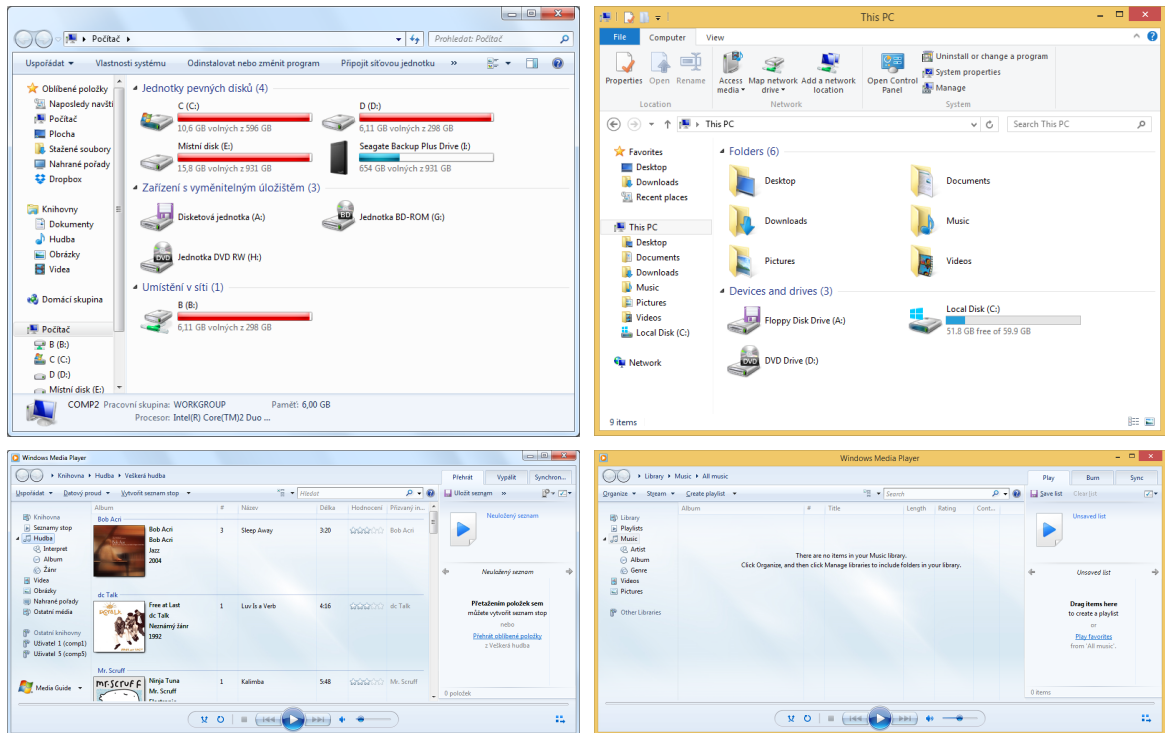


Figure 1.10: Windows Explorer and Windows Media Player from Windows 7 on the left; the same applications from Windows 8.1 on the right – showing the UI inconsistency of desktop applications in Windows 8 (Windows Explorer on Windows 8 using undocumented visual design principles).

Chapter 2

Usability and user experience drawbacks of Windows 8

This chapter lists the major confusion aspects of Windows 8 user experience and usability based on my own experience providing technical support to friends and acquaintances, various internet discussions [21] and articles [22]. These aspects will be used as hypotheses to be later verified in tests with users to serve as a base for a new multi-modal user interface prototype avoiding the most notable usability quirks of current Windows.

2.1 APPLICATION STYLE INEQUALITY

As mentioned in the previous chapter, from the developer's point of view, Windows 8 distinguishes between two types of applications: "metro-style" applications independent of processor architecture, and traditional Windows API desktop applications. However, the system behavior forces users to distinguish between these types of applications as they are not treated equally, causing usability problems and confusion. Following are the user differences between "metro-style" and desktop applications:

- **Installing and updating:** "Metro-style" applications can be installed and automatically updated through a centralized repository – Windows Store as well as "sideloaded" manually, while desktop applications can be installed manually only – leaving the installation technology and update notifications/distribution entirely up to the developers.
- **Uninstalling:** "Metro-style" applications are not displayed in the "Programs and features", only desktop applications are – with no mention of "metro-style" applications. On the other hand, "metro-style" applications are listed in a section of the "metro-style" control panel along with a link to "Programs and features" for uninstalling desktop applications.
- **New system-wide features:** "Metro-style" applications have access to new system features such as search, sharing (figure 2.1), centralized settings access, or notifications. Desktop applications, for instance, need to generate notifications manually – mixing two areas of notifications.
- **Switching between applications:** When using a "metro-style" application, task switcher lists only other running "metro-style" applications, plus an entire desktop as one item.

When using a desktop application, task switcher lists all running desktop application windows plus all running “metro-style” applications as separate items.

- **Pinning to screen:** All “metro-style” applications can be pinned to screen while most desktop applications cannot be – the feature depends only on third party developers’ custom solutions.

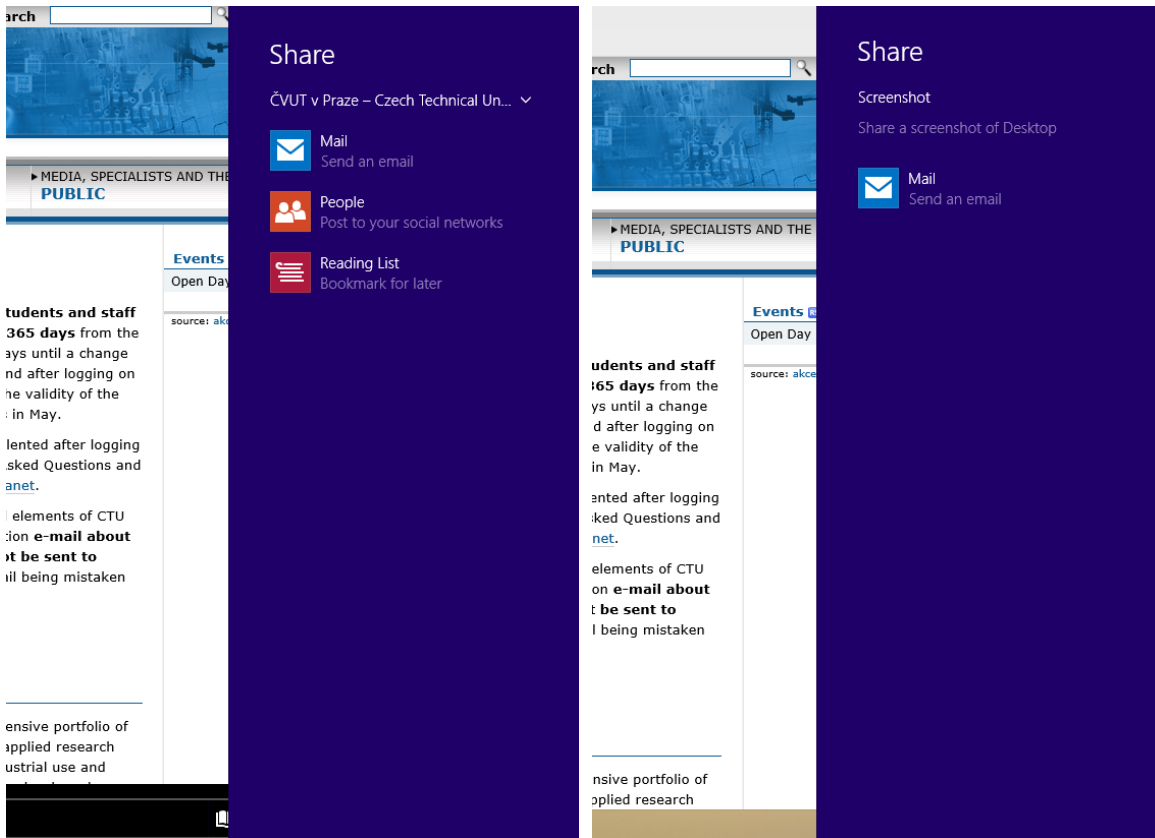


Figure 2.1: Example of the “metro-style” Internet Explorer (left) and desktop style Internet Explorer (right) not having access to the same system-wide features (sharing in this case)

2.2 SCHIZOPHRENIC APPLICATIONS

With Windows 8, Microsoft dramatically changed what applications are built into the operating system. While removing some applications such as games, many applications were added, especially “metro-style” applications. These “metro-style” applications often duplicate the functionality of other built-in desktop applications (table 2.1). [23] Some of these application pairs share data, some do not. For instance, the “metro-style” Music application does not share library and settings with the desktop Windows Media Player and behaves rather as a separate application: both technically and from a user interaction point of view. On the contrary, the “metro-style” Internet Explorer and desktop Internet Explorer are technically two separate applications. They do share data (favorites, browsing history, ...), though, and from the user’s point of view they act as one in some cases. Internet Explorer lets users choose preferred user interface style or perform a one way switch from the “metro-style” application to the desktop one while preserving opened tabs. Also, what is true for both kinds of application pairs, is that feature sets of applications within these pairs differ – usually the “metro” style application’s feature set is more limited.

Applications	Issues
Help + Tips Windows Help	Two separate Help applications, each covering different topics.
Reader	Only a “metro-style” application is available for viewing PDF files, making reading on primarily desktop computers inconvenient.
Skype	Two separate applications that do not share data and do not allow switching between each other. The “metro-style” Skype is built into the system, the desktop edition needs to be downloaded for free separately.
Calculator	Two separate applications for an identical purpose, system allows for running both simultaneously.
Sound Recorder	Two separate applications for an identical purpose, system allows for running both simultaneously.
Xbox Games Game Explorer	Two separate applications, each may display different games.
Music & Video Windows Media Player	Separate applications for an identical purpose that do not share data, settings and most features. For file associations they act also separately which causes the fact that for instance opening sound files from “metro-style” applications can result in the desktop Windows Media Player being called, or when opening sound files from a desktop application the “metro-style” player may be called. Also, Windows Media Player’s visual style has not changed since Windows 7.
Photos Image Viewer	Separate applications that act separately for file associations as well.
Internet Explorer	While technically the “metro-style” Internet Explorer and desktop Internet Explorer are separate applications, for protocol and file associations they act as one, avoiding confusion mentioned with Windows Media Player. These applications also share data and allow for limited switching between each other. Feature set, however, is not symmetric.
Mail Windows Live Mail	While there is only one email client built into the operating system, which uses the “metro” style (thus for most users the issues are comparable to those with Reader), Microsoft offers Windows Live Mail desktop client as a free download from the Windows Essentials package. From schizophrenia perspective, this pair of applications acts exactly like the aforementioned Skype – no data sharing is supported and both applications act completely separately.
PC Settings Control Panel	Operating system settings are divided into two applications: The “metro-style” PC Settings and desktop style Control Panel. Some settings such as Windows Update controls are available in both, while others are available only in one. For instance, what makes user experience confusing is that desktop wallpaper can be changed from Control Panel only. Lock screen wallpaper, on the other hand, can only be set from within PC Settings.

Table 2.1: Schizophrenic built-in Windows 8 applications listing

2.3 INCONSISTENCY OF DESIGN DECISIONS

In terms of built-in applications, with Windows 7 Microsoft made a decision to turn back from the previously practiced philosophy of increasing their amount. The operating system was made much more lightweight and applications such as an email client, photo gallery or video editor were made optional downloads – often as a part of the Windows Live Essentials suite, or completely removed. In contrast to that, Windows 8 brought more new built-in applications than were removed from Windows Vista in Windows 7. [24] Other applications, such as games, though, disappeared from the OS with the introduction of Windows 8 [25].

2.4 LIMITED USABILITY OF THE ENTIRE SYSTEM WITH ONE PARTICULAR INPUT METHOD

The design of Windows 8 makes using “metro-style” parts of the UI as well as the majority of “metro-style” applications limitedly usable with the combination of a mouse (or trackpad) and a physical keyboard. [27] Basic controls of the system rely on touch gestures, which were not very intuitively replaced when using mouse – making the use of a manual or tutorial necessary. Since applications such as Weather or Reader only exist in “metro-style”, users are forced to use those even though they may not own a computer with a touchscreen.

Similarly, many parts of the Windows functionality or built-in apps are available in desktop style only, making a Windows tablet owner’s user experience worse. For instance:

- Even with DPI settings higher than usually recommended for a particular pixel density screen, UI widgets are small for use with a finger. [28] This applies to form elements with not enough spacing around them, concrete UI widgets such as the “spinner” (figure 2.3).
- The operating system contains UI elements relying on the “mouse hover” event which cannot be triggered using a capacitive touchscreen.
- No touchscreen gestures are supported within the desktop environment.
- Text input within the desktop in standard system UI widgets is not optimized for use with the virtual on-screen keyboard. Standard system text input elements inside the desktop environment do not provide information on the input format to the keyboard (such as whether the text field contains a number, a password, ...) which forces users to manually switch character sets. Text selection, copying and pasting within these widgets is also different than in “metro” environment requiring users to exactly emulate the tasks they would perform with a mouse and physical keyboard. Also, the on-screen keyboard does not show up automatically when a standard system text input UI widget gains focus – a behavior that is completely different from within “metro” environment.

- Other virtual on-screen keyboard flaws: The on-screen keyboard’s default behavior within the desktop environment resizes the entire environment when displayed. This makes application windows smaller if needed, causing the text input field with focus to often move out of range (figure 2.2). This requires users to scroll within the application manually in order to display the focused text input field again. Also, the on-screen keyboard is unfriendly to Latin script languages using more than four alpha-numeric rows – such as Czech, displaying special, yet frequently used, characters on non-standard spots (figures 2.4, 2.5). [29] All keys, though, are large enough to allow for their shrinking in order to add a fifth row of characters.

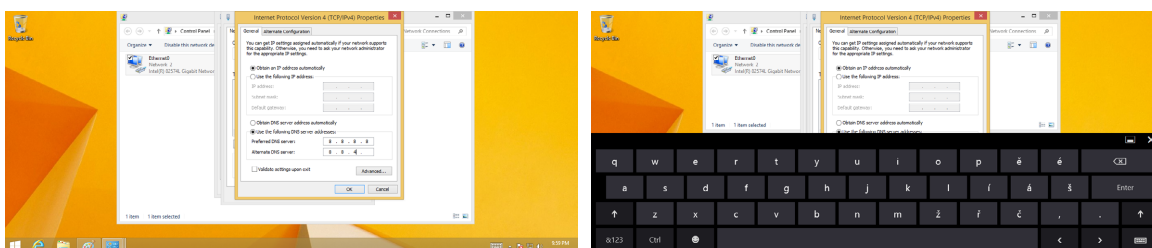


Figure 2.2: Example of a text input field moving out of range when on-screen keyboard is displayed

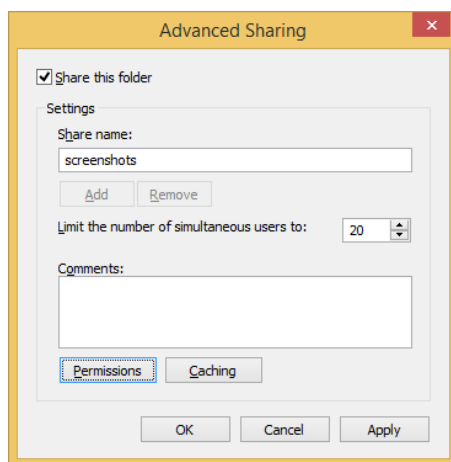


Figure 2.3: A dialog containing the “spinner” UI widget with buttons not large enough for touch control

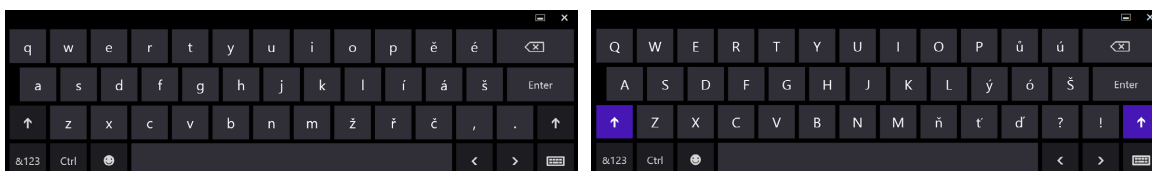


Figure 2.4: Usability drawbacks of the Czech on-screen keyboard – accented letters being in non-standard spots



Figure 2.5: The correct Czech (QWERTY) keyboard layout

2.5 BASIC PC CONTROLS HAVE INCONSISTENT STYLE

These are some of the examples of inconsistent style within basic PC controls and system-wide notifications:

- When the device's battery is running low, the message is displayed in “metro-style”. However, battery settings and energy plan switching is only available from within the desktop.
- The list of available wireless networks is only available in “metro-style”, while other settings accessed from the taskbar's notification area (such as clock or volume) use the Windows 7-like desktop style.
- Even when accessed from within the desktop, the selection dialog of what application a file should be opened with is always displayed in “metro-style”. When adding a custom application to this list, the following dialog uses desktop style.
- When a storage device is plugged into the computer, the notification displays in “metro” style. When safely removing the device, the interface for that displays in desktop style, as well as the notification after successful device removal.
- Messages generated by the Windows SmartScreen filter use the “metro” style, while similar messages generated by the User Account Control component are in desktop style (figure 2.6).

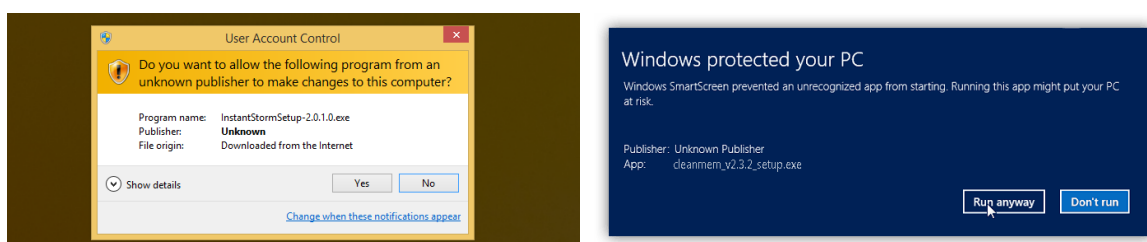


Figure 2.6: User Access Control (left) and SmartScreen (right): similar features using completely different interfaces for notifications [34]

Chapter 3

Usability test of Windows 8

In order to verify assumptions gathered from the previous chapter, a standard qualitative usability test with users was performed. Each user was supposed to perform a set of tasks on an Acer E7 tablet running Windows 8.1 with an attachable physical keyboard – first using the touchscreen input method only, then only with a mouse and a physical keyboard, ignoring the touchscreen. Tasks were selected to reflect everyday use of a computer in home and office environment, emphasizing the need to also verify as many assumptions as possible from chapter 2. The test was performed in my apartment's living room (figure 3.1), participants were recorded and their interaction with the operating system was later transcribed and analyzed.

Before the tasks were finalized, a random participant was asked to perform a preliminary test in order to avoid inadequately difficult, frustrating or confusing tasks in the final list.

In the final tests, participants were asked to comment on every action they perform, thinking out loud. Also, all of them were assured that they are not treated as test subjects, instead the software is – and any mistakes they make are considered mistakes of the software. [26]



Figure: 3.1 Test setup

3.1 TEST TASKS

Participants were supposed to complete the following tasks using only the device's touchscreen (part A):

- 1) **Launch Internet Explorer.** The only Internet Explorer shortcut on the Start screen launches the “metro-style” version. At the same time, this shortcut only appears when Internet Explorer is set as a default web browser. (*verifies 2.2*)
- 2) **Navigate to cvut.cz.**
- 3) **Add the current page to favorites.**
- 4) **Navigate to zelpage.cz.** This is necessary for further steps since “metro-style” Internet Explorer remembers open tabs after quitting.
- 5) **Quit Internet Explorer.** The basic touch gestures that include quitting “metro-style” applications are not intuitive and require multiple attempts. (*verifies a fact from 2.4*)
- 6) **Re-launch Internet Explorer and, using favorites, navigate to cvut.cz.** Favorites are hard to find in the “metro-style” browser since there is no dedicated button for accessing them (unlike the desktop style browser). User needs to open a blank tab first. (*verifies 2.1 and 2.2*)
- 7) **Set the currently loaded page as homepage.** This step requires users to switch to the desktop style Internet Explorer since the option to change homepage is only available in that version. However, the change is applied to both versions of the browser as they do share data. Also, accessing and using the Internet Options dialog in desktop style Internet Explorer with a touchscreen only is not convenient. (*verifies 2.2 and 2.4*)
- 8) **Quit Internet Explorer.**

Participants were supposed to complete the following tasks using only the device's touchscreen (part B):

- 9) **Plug in a camera and import pictures from its memory card to the PC. Name the folder “Šotovýlet”.** Typing most special Czech characters (such as “ý” and “š” in this case) on the Windows 8 virtual on-screen keyboard is rather inconvenient and confusing since the characters are located neither in the standard spots corresponding to physical keyboards (inconvenience), nor in the spots of corresponding basic letters requiring long press (which is the convention used in other touchscreen operating systems – confusion). Instead, Microsoft decided to put most of these special characters in random spots on the right side of the keyboard, while the rest has to be accessed using a long press, or even a combination of the Shift key and a long press. (*verifies 2.4*)
- 10) **Rotate pictures that require it.** Typically using the “metro-style” Photos application.

11) **Set one of the transferred pictures as lock screen wallpaper.** This can either be done from the “metro-style” (verifies 2.2) Photos application directly, or from the PC Settings application. (verifies 2.2)

12) **Set the same picture as desktop wallpaper.** This cannot be done from the “metro” environment, however, there are several ways to do so from the desktop. However, these methods may be inconvenient when using a touchscreen. (verifies 2.4)

Participants were supposed to complete the following tasks using only the device’s physical mouse and keyboard – ignoring the touchscreen (part A):

13) **Launch Internet Explorer.** By default (if there is no desktop shortcut to the application or no icon pinned to the taskbar) Internet Explorer needs to be launched from the Start screen. Since this screen has the “metro” style, the “metro-style” Internet Explorer also launches even though user uses keyboard and mouse to launch it. (verifies 2.1 and 2.2)

14) **Navigate to cvut.cz.** When using mouse, user has to right click on the page to show the address bar. (verifies 2.4)

15) **Add the current page to favorites.**

16) **Navigate to zelpage.cz.** This is necessary for further steps since “metro-style” Internet Explorer remembers open tabs after quitting.

17) **Quit Internet Explorer.** Quitting “metro-style” applications using a keyboard and mouse is more confusing than doing so with a touchscreen. User has to imitate the exact same gesture as they used with a touchscreen. (verifies 2.4)

18) **Re-launch Internet Explorer and, using favorites, navigate to cvut.cz.** Favorites are hard to find in the “metro-style” browser since there is no dedicated button for accessing them (unlike the desktop style browser). User needs to open a blank tab first, this time using right click. (verifies 2.1, 2.2 and 2.4)

19) **Set the currently loaded page as homepage.** This step requires users to switch to the desktop style Internet Explorer since the option to change homepage is only available in that version. In order to access this version, user needs to right click again. (verifies 2.2 and 2.4)

Participants were supposed to complete the following tasks using only the device’s physical mouse and keyboard – ignoring the touchscreen (part B):

20) **Plug in a camera and import pictures from its memory card to the PC. Name the folder “Šotovýlet”.** The “connected device” notification is in “metro” style only, as well as the photo importing feature. (verifies 2.5) The notification does not have the appearance to indicate its mouse clickability. (verifies 2.2)

21) **Rotate pictures that require it.** After importing, the “metro-style” Photos application is

automatically launched, even though it was triggered using mouse. (*verifies 2.2*) The application also requires right clicking in order to access the rotation feature. (*verifies 2.4*)

22) **Set one of the transferred pictures as lock screen wallpaper.** This can either be done from the “metro-style” (*verifies 2.2*) Photos application directly, or from the PC Settings application. There is no way to do so within the desktop environment, for instance from the same control panel as desktop wallpapers are set. (*verifies 2.5*)

23) **Set the same picture as desktop wallpaper.** User needs to leave the “metro” environment to do so. (*verifies 2.2*)

3.2 SCREENER SURVEY

The purpose of a screener survey in usability testing is to choose the right group of participants. A screener survey contains 3 to 5 brief questions identifying the potential testers’ qualities for the test. For the testing of Windows 8 I was looking for three participants – at least one male participant and at least one female participant. None of these participants should be a student of computer science or related fields, or have a computer-related job. None of these participants should be using a Windows 8 tablet – however, using a tablet with a different operating system is recommended. At the same time, at least one of the participants should be using Windows 8 on a non-touchscreen device.

This is what the screener survey for this test looked like:

- Are you male or female?
- Is your field of study or work related to computers?
- Do you use a tablet? If you do, what operating system is it running?
- Do you use Windows 8 on a non-touchscreen device?

I posted this survey on a social network and got 15 responses, I chose four potential participants, three of which were available for testing (table 3.1):

Participant	Gender	Computer-related study/job	Tablet OS used	Uses Windows 8 on a non-touchscreen device
#1	male	no	Android	yes
#2	female	no	Android	no
#3	male	no	none	no

Table 3.1: Windows 8 testing screener survey results

3.3 OTHER SURVEYS

Selected participants were asked to fill out two more surveys:

- A pre-test survey preceding the actual testing process in order to be able to provide them with potential task clarifications.
- A post-test survey following the actual testing to get subjective information supplementing the test results.

Pre-test survey questions:

- **Pre 1** – Are you familiar with the term “metro” interface?
- **Pre 2** – What web browsers do you use?
- **Pre 3** – What tools or approach do you use to organize digital photo on your computer?
- **Pre 4** – Did you ever participate in software usability testing?
- **Pre 5** – How are you feeling before the test?

Pre-test survey results (table 3.2):

Question	Participant #1's answer	Participant #2's answer	Participant #3's answer
Pre 1	yes	no	no
Pre 2	IE (desktop), Firefox (desktop), Chrome (desktop + mobile)	Opera (desktop), Chrome (mobile)	IE (desktop), Firefox (desktop), stock Android browser
Pre 3	folders	Google Picasa	folders
Pre 4	yes	no	no
Pre 5	Fresh.	Curious.	Excited.

Table 3.2: Windows 8 testing pre-test survey results

Post-test survey questions:

- **Post 1** – How did you find the Windows 8 interface?
- **Post 2** – Would you buy a hybrid Windows 8 device after this test?
- **Post 3** – Which task was the hardest for you?
- **Post 4** – Was the task list clear?

Post-test survey results (table 3.3):

Question	Participant #1's answer	Participant #2's answer	Participant #3's answer
Post 1	Mess.	Confusing.	Awful.
Post 2	no	no	no
Post 3	Typing on virtual keyboard.	Setting homepage.	Accessing favorites.
Post 4	yes	yes	yes

Table 3.3: Windows 8 test post-test survey results

3.4 TEST ANALYSIS

3.4.1 Participant 1

Using only touchscreen – part A:

This participant had no trouble navigating to the initial website. In order to find the button to add the current page to favorites, he systematically went through all button elements in the browser's context menu until he found the "pin" icon. After that, he was not able to quit the browser, minimizing it instead. As for re-opening a web page from favorites, participant was struggling to find a way to access the browser context menu. When he found it, he quickly stumbled upon the list of favorites. However, he claimed that he didn't know how he had done it, and may struggle to find the same menu again.

When setting a website as homepage, he was trying to find this feature in the "metro-style" Internet Explorer. Later, he was advised that the "metro" environment has to be left in order to achieve the task. Even after this advice, finding a way to switch to the desktop style Internet Explorer took him a long time. When he successfully switched to the desktop style Internet Explorer, he intuitively knew where to change the homepage. However, the participant accidentally tapped the Favorites icon next to Options, causing confusion, making himself think that his intuitive approach was wrong. A while later, he retried tapping the Options icon, and successfully changed the homepage. Also, when seeing the contents of the Options menu, he remarked that he "wonders what Add to title screen" means.

Using only touchscreen – part B:

After plugging in the camera, the participant was not able to tap the new storage device notification on time, forcing him to re-plug the camera and try to tap the notification again. When entering the folder title, he does not intuitively find the character "š". As for finding the character "ý", the participant systematically browsed through all options until he found the key, without understanding the context. To rotate the pictures, he first tried rotating them using context menu in the listing of multiple pictures. Then he opened a particular picture, tried out touch gestures known from other touchscreen devices. Finally, he tried opening the context menu of a single picture using a gesture and rotating worked. The same way he set the picture as a lock screen wallpaper. Expecting to set a desktop wallpaper similarly, he navigated through options of the "metro-style" Photos application, until he was advised to try leaving the "metro" interface. In the desktop interface, he intuitively long-tapped the picture icon, expecting the right-click context menu to appear. However, he slightly moved the finger, which made him think the gesture does not exist in Windows 8. A moment later, he retried this gesture and successfully changed the desktop wallpaper.

Using only physical keyboard and mouse – part A:

When adding a page to favorites, the participant intuitively tried right mouse button as a replacement of the context menu touch gesture. Although the right-click action was correct, participant was made to think otherwise, since the context menu could not display while the page was still loading. A while later, he retried right-clicking the screen, and completed the “add to favorites” task. Later, instead of quitting Internet Explorer, he minimized it again. In order to get to the desktop version of Internet Explorer, he first navigated to the desktop from the Start screen, but quickly realized to repeat the same actions as he had performed using the touchscreen.

Using only physical keyboard and mouse – part B:

After plugging in the camera, participant commented that if he had not performed the same task with a touchscreen before, he would not have thought the notification is “clickable”. Transferring the pictures went smoothly; however, when he was rotating one of the pictures, he tried to do so from within the listing of pictures again.

As for setting the picture as desktop wallpaper, participant stopped for a while to think how to get to the desktop environment.

3.4.2 Participant #2

Using only touchscreen – part A:

Test flow was smooth until Internet Explorer had to be closed. The participant minimized the browser using a physical Windows logo key on the tablet. Then she struggled with finding the context metro within a “metro-style” application. When she was supposed to change the browser’s homepage, she was unable to find the feature within the “metro-style” browser – due to which she was advised to leave the “metro” environment in order to complete the task. After that, it took her quite long to figure out how to switch the browser to the desktop mode. In desktop mode, she first tried the context menu touch gesture from metro to see if it works. After realizing it does not work in desktop, she browsed through all options of the browser until she finally found the option to change homepage. Also, while trying out various browser options, she accidentally tapped “Developers’ console” instead of “Internet options”. Participant also remarked that she had thought “Add to title screen” might have been the solution to changing homepage.

Using only touchscreen – part B:

Before plugging in the camera, participant opened the Start screen. When importing the pictures, she did not have much trouble finding the character “Š”. However, it took her extremely long to find the character “Ÿ”. As for rotating pictures – she first tried a touch gesture known from other tablet platforms. It took her fairly long to find a gesture for opening the context menu. After that, finally

rotating the picture and setting it as a lock screen wallpaper was quick. Participant tried to set the desktop wallpaper the same way as lock screen wallpaper – but failed. Later she was advised to leave the “metro” environment – allowing her to open Windows Explorer and find the picture using search. When in Windows Explorer, participant double tapped the picture icon, but was taken back to the “metro-style” Photos application. After returning to the desktop, she intuitively guessed to long-tap the picture icon to emulate right mouse button, and was able to set the picture as desktop wallpaper.

Using only physical keyboard and mouse – part A:

First, participant did not expect that the “Start” on-screen button opens the “metro” environment. Two following steps – navigating to a page and adding it to favorites – went smoothly. When trying to quit Internet Explorer, she, again, minimized it instead. In order to access the context menu in “metro” environment, she tried imitating the touch gesture with a mouse – which did not work, and she managed to pin the application to a half of the screen instead. Later she realized to use right mouse button to open the context menu. Switching to the desktop style Internet Explorer was participant’s biggest struggle in this part: she did not remember how she did it in the touchscreen part of the test, and spent 5 minutes looking for the option. After successfully finding it, the rest of this part of the test went smoothly.

Using only physical keyboard and mouse – part B:

No major struggles were detected in this part. A minor problem was caused by the fact that participant did not realize opening context menu in the “metro-style” Photos app would work identically to the “metro-style” Internet Explorer.

3.4.3 Participant #3

Using only touchscreen – part A:

First, participant comments that he does not understand why the confirmation key of the virtual on-screen keyboard is labeled “Go” when editing the web browser’s address bar. Later, he struggles to quit Internet Explorer – closing the current tab instead of the entire browser repeatedly. After a few of these attempts, he guesses a touch gesture to switch between applications, and gets to the desktop environment. After re-opening the “metro-style” Internet Explorer, it takes him fairly long to find where favorites are located – remarking that he remembers that in the desktop version they have a dedicated button right below the browser’s title bar.

Participant was also struggling to find out how to change homepage, due to which he was advised to leave the “metro” environment. It took him, however, a lot of searching to find the option to switch to desktop mode. When he was in desktop mode, he changed the homepage quickly; not without an accidental tap on a different menu item, though.

Using only touchscreen – part B:

Right after plugging in the camera, participant commented that he almost did not tap the notification on time. When he tapped it, an animation of it disappearing was just starting. As for using the virtual on-screen keyboard, participant had a lot of trouble finding both “š” and “ý” characters. A systematic approach of trying different buttons one by one helped, though. However, participant commented that the placement of characters on the keyboard was highly illogical.

Rotating the picture went smoothly as the participant remembered to use a touch gesture to access context menu. Thanks to this he was able to change lock screen wallpaper quickly. After spending too much time finding where to change the desktop wallpaper, participant was advised to leave the “metro” environment. When he opened Windows Explorer in the desktop environment to access the Pictures folder, he double tapped the picture’s icon only to find out that the “metro-style” Photos app re-appeared. After re-accessing the desktop, participant spent some time figuring out how to imitate the right mouse click using touchscreen. After succeeding, he was easily able to set the wallpaper.

Using only physical keyboard and mouse – part A:

Participant intuitively realized to use right mouse button as a replacement for the “context menu” touch gesture. When he was supposed to quit Internet Explorer, he minimized the application instead. Again, it took him very long to find where favorites are located, although knowing how to access the context menu. As for changing the browser homepage, he remembered to have to access the desktop Internet Explorer, however, he opened desktop from the Start screen instead, only to realize that he has to return to the “metro-style” Internet Explorer and tap the “Show on desktop” option – complaining about the menu item’s terminology. Participant explained that he would expect the option “Show on desktop” to for instance set the current page as desktop wallpaper. The rest of this part of the test went smoothly.

Using only physical keyboard and mouse – part B:

Participant did not have any major issues with this part of the test except for being slightly confused from by the “metro-style” Photos application’s interface. In the late steps of this part of the test he accessed the desktop environment by opening the Start screen and clicking the “Desktop” tile.

3.5 TEST RESULTS

The following table indicates how many times each of the tested assumptions were verified (table 3.4).

Assumption	Verified in
2.1 – Application style inequality	2 out of 3 tests
2.2 – Schizophrenic applications	3 out of 3 tests
2.4 – Limited usability with one input method	3 out of 3 tests
2.5 – Basic PC controls’ inconsistent style	2 out of 3 tests

Table 3.4: Windows 8 test conclusion

The test results show that assumptions were correct and should be taken into consideration when designing the multi-modal UI prototype.

Chapter 4

Prototype design

Since all of the usability and user experience drawbacks confirmed as valid, their list – along with the test results – was used to compile another list: that of design principles of the new multi-modal user interface prototype for a possible future version of Windows. This list is presented as sections of chapter 4.

4.1 UI PHILOSOPHY LAYERS

Similarly to 1.3, user interface philosophy layers (figure 4.2) were used to express the basic semantics behind the new user interface. These semantics were designed to keep the philosophy clean, straightforward and unambiguous, with as little exceptions as possible.

The bottommost layer consists of one or multiple physical screens. Each of these physical screens can be, optionally, divided into two parts: either vertically, or horizontally (figure 4.1).

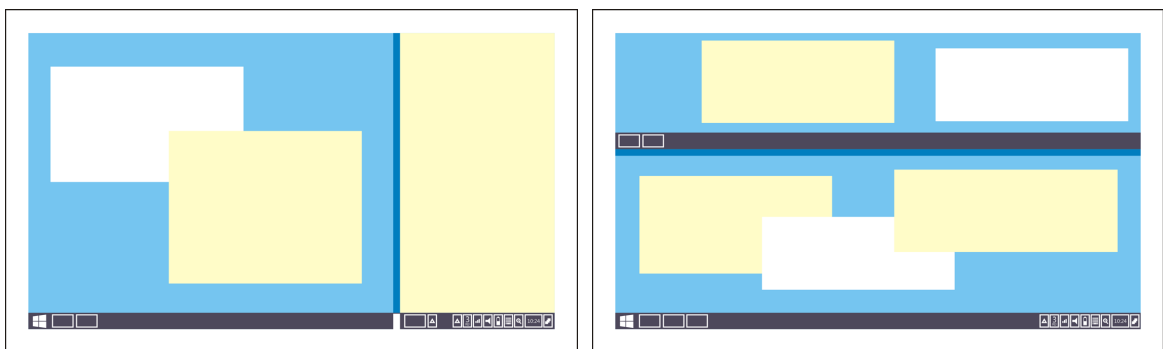


Figure 4.1: A horizontally split desktop on the left, a vertically split desktop on the right (this one running an application in full screen mode)

Above the “physical screens + common controls” layer is the “desktops” layer. A desktop can either correspond to a half (philosophy-wise, not absolute dimensions-wise) of a physical screen, or to an entire physical screens. If there are two desktops on a physical screen, the basic controls (such as Start menu) are displayed only once. Taskbars, however, are repeated for each “virtual” desktop displaying application windows belonging to that particular desktop, and behave identically to taskbars introduced with Windows 7.

Windows prototype from this thesis

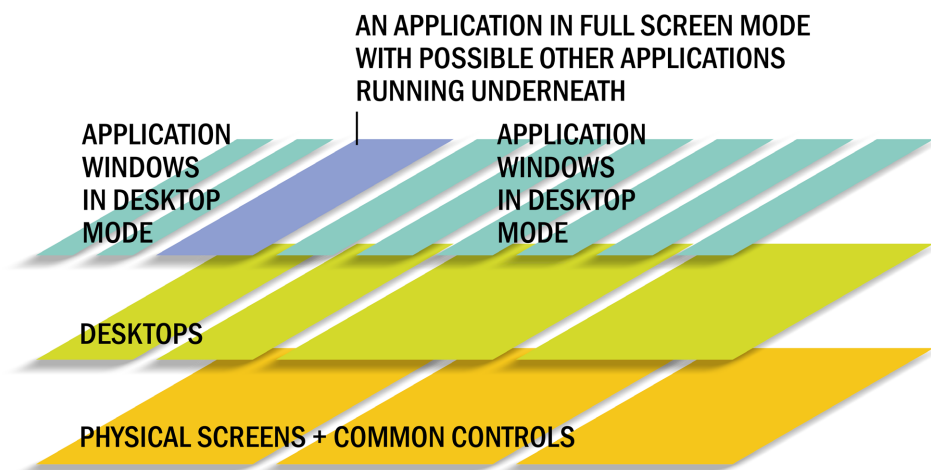


Figure 4.2: User interface philosophy layers of the prototype

4.2 THE “BASIC CONTROLS” FEATURE

Basic controls are a vital part of the user interface. They replace the “Charms bar” from Windows 8’s “metro” interface, the “Options” section of this bar, as well as the “notification area” from traditional desktop Windows. These basic controls are a set of icons with a recommended flat, single-colored graphical style, located in the rightmost area of each physical screen’s primary bar for taskbars (figure 4.3). Built-in icons are the following: window listing, clock, zoom button, keyboard button, battery indicator, volume control. The icon meanings will be explained below. Third party icons are also displayed in the same area, combining the notification area icons from a traditional Windows desktop environment and notifications from the “metro” environment. Unlike built-in icons, these third party basic controls application icons are displayed on all desktops (for cases when they belong to particular application windows). If there is no specific application window corresponding to these icons, they are repeated on every physical screen’s primary basic controls area. If there is not enough space for all icons, the area can be expanded to an interface element with appearance similar to Windows 8’s “Charms bar”. Searching and sharing are available from this very interface as well.

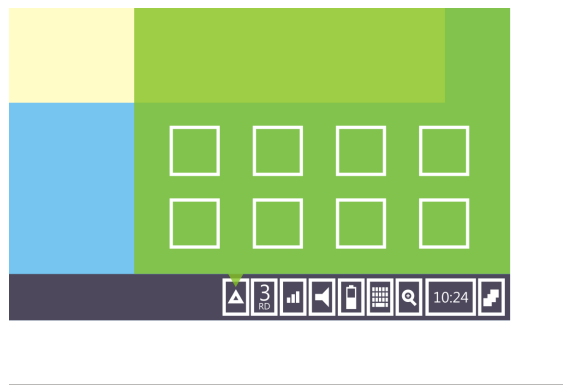


Figure 4.3: The basic controls

Start menu: A Start menu icon is an exception – while it is treated as a built-in basic controls icon, it is located in the leftmost area of each physical screen’s primary bar for taskbars. The menu itself opens taking up majority of the screen estate (figure 4.4) – resembling the appearance of a Windows 8 Start screen, while keeping the basic philosophy of Windows 7 Start menu.

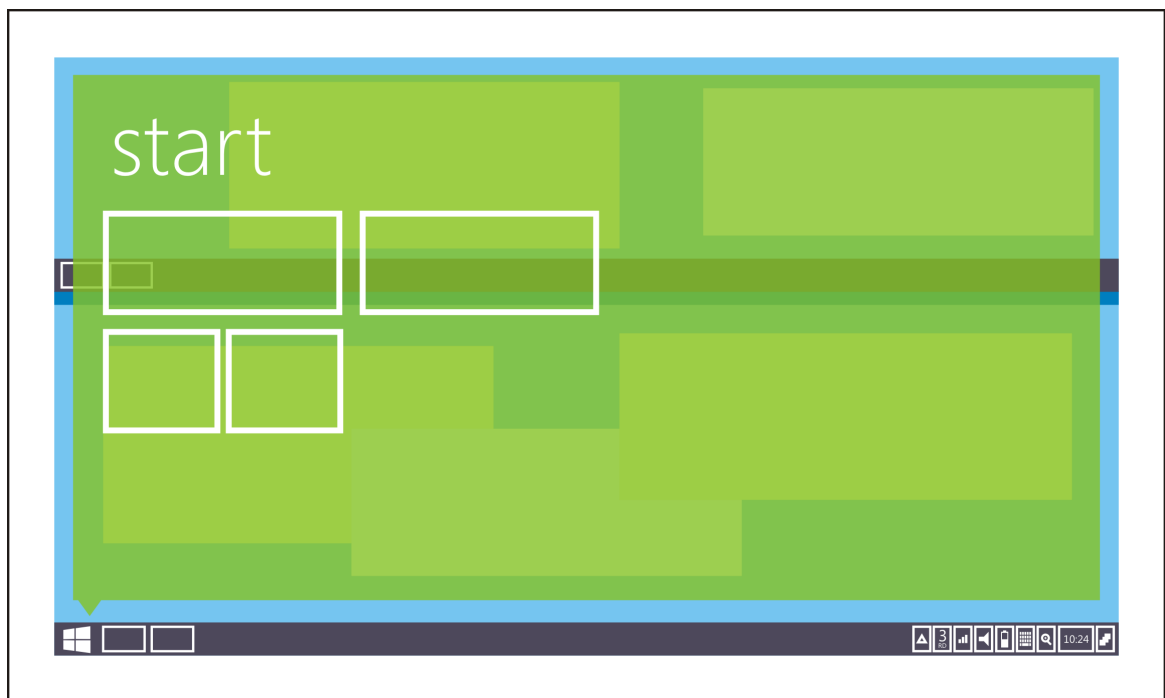


Figure 4.4: The Start menu philosophy

Window listing: The window listing feature (figure 4.5) is activated by clicking/tapping the window listing button in the basic controls area – and after activation it displays thumbnails of all desktops and running application windows throughout them on one screen – the screen from which the activation was made. Similarly to Apple’s Exposé feature in OS X, this allows for rearranging

applications among desktops, or, newly, splitting desktops into halves, or merging split desktops together. While window listing is being displayed, bars and application windows on other screens are temporarily hidden.

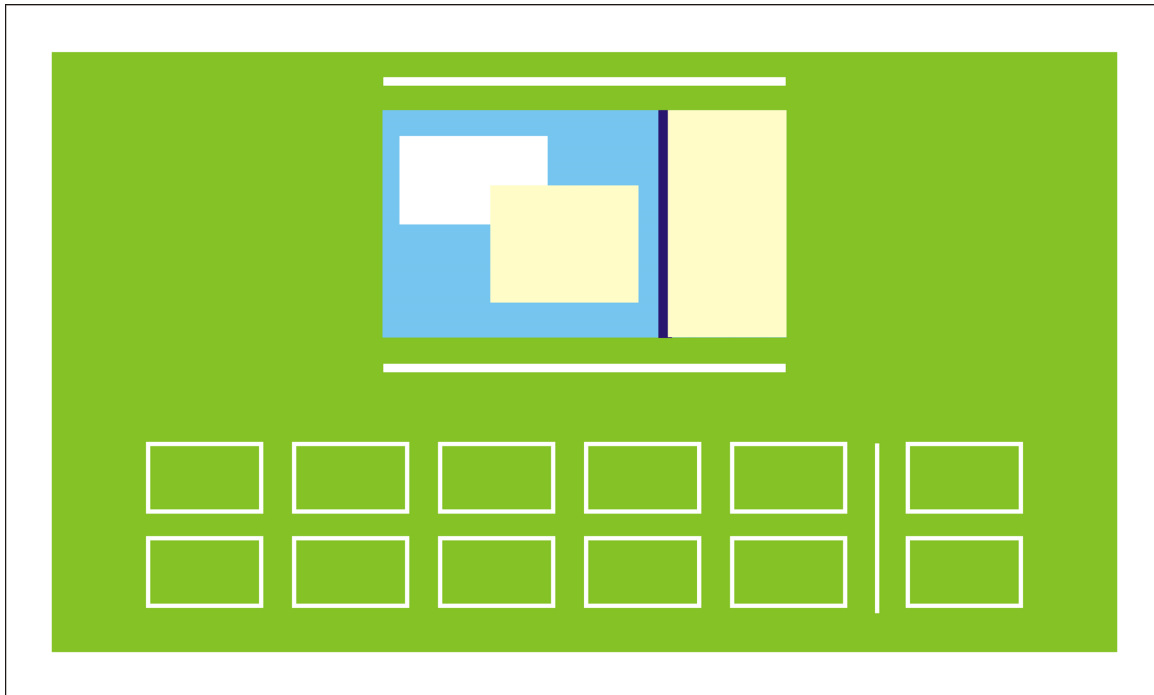


Figure 4.5: Window listing

4.3 ALL APPLICATIONS TREATED EQUALLY

The prototype no longer distinguishes between “metro-style” applications and desktop style applications. Instead, all applications are designed primarily for keyboard and mouse/trackpad controls according to the philosophies from 2010 Windows 7 guidelines [15]. However, with taking touch controls into account – i. e. keeping all elements’ sizes large enough, using proper margins, not relying on the hover event, minimizing the use of non-standard UI widgets, etc. Optionally, applications can support a full screen mode (along with the ability to switch between the full screen and non-full screen modes on the fly without losing data), which is designed primarily for touchscreen controls in accordance with the philosophies from Windows 8 “metro-style” application guidelines [20].

toggling full screen mode: Each application window supporting full screen gets a new title bar button (figure 4.6). When in full screen mode, the title bar is missing; however, the top part of the application still contains buttons from the title bar (figure 4.6), allowing for minimization, closing,

or leaving full screen mode without confusion using any input method. Applications in full screen mode are not limited to opening primarily touch-designed dialogs, either, and the taskbar remains displayed.

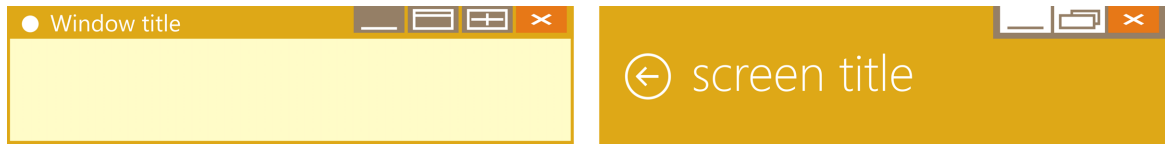


Figure: 4.6 An application in non-full screen mode (left) and full screen mode (right)

Installing, uninstalling and updating: All Windows applications are installed, updated and uninstalled primarily using a centralized application store, while also allowing for sideloading applications manually without restrictions (mostly due to backwards compatibility and specific software distribution cases) – imitating the approach of Android operating system.

4.4 UNIFIED VISUAL STYLE

While the philosophy of the operating system’s fundamental controls is based on Windows 7, the visual style is united in the style of “metro” – due to being an aspect mostly praised in the operating system’s reviews [27] [31] – the very same approach that Microsoft applied with Office 2013 (figure 4.7), or, the Task Manager in Windows 8.

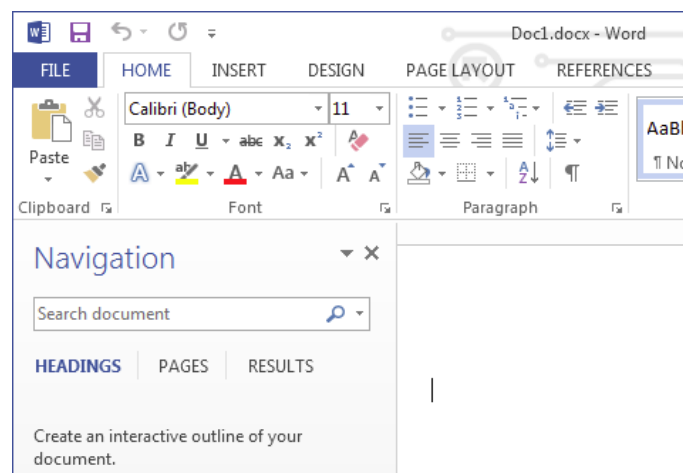


Figure 4.7: Example of the “metro” graphical style inside a desktop application – Microsoft Office 2013

4.5 NO SCHIZOPHRENIC APPLICATIONS

Applications built into the prototype do not duplicate one another's functionality. Instead, they are all designed multi-modally as described in 4.3. System-wide messages (such as laptop battery notifications or SmartScreen filter messages, default application selection) are all united in the philosophy of 2010 Windows 7 guidelines [15] while using the “metro” graphical style. All PC settings are located in one control panel application based on the Windows Explorer philosophy – leaving out all traces of outdated UI elements as mentioned in 1.3.1.

4.6 TOUCH MODE

Each physical screen with the support for touch has a “zoom” icon in the basic controls area which triggers the touch mode. Optionally, touch mode can be toggled by plugging/unplugging a mouse or trackpad on hybrid input devices.

When touch mode is on, the system DPI settings are increased to 1.5 times of what is the default DPI value, making all UI elements larger and touch-friendlier. This change, however, does not apply to full screen modes, which are already designed primarily for touch controls.

When certain UI widgets are activated using touch input (regardless of touch mode status), they act differently than when activated using mouse: for instance the spinner widget displays a popup window with larger buttons, text input fields support text selection using touch, or combo boxes display a popup with larger options listing.

4.7 ON-SCREEN KEYBOARD

The on-screen keyboard in this prototype avoids usability drawbacks listed in 2.4: adds another row of keys for some Latin script languages such as Czech, can trigger scrolling within applications to make UI elements using it always visible. Also, when touch mode is on and no physical keyboard are plugged in, the on-screen keyboard automatically displays when a text input field is touched. Similarly, when a text input field loses focus under these conditions, the on-screen keyboard automatically closes.

Each physical touchscreen can have only one on-screen keyboard displayed.

4.8 BACKWARDS COMPATIBILITY RULES

Since one of the key selling points of Windows has always been its well-maintained backwards compatibility (1.1), the prototype principles take both older “metro-style” and desktop style applications into account.

When an older desktop style application is run, standard system UI widgets are replaced with new ones, allowing application vendor or Microsoft to optionally disable this behavior if the replacement causes issues. Full screen mode is not available for these applications, though, the full screen gesture (4.9) triggers window maximization. If the system detects insufficient spacing between elements, a magnifying glass is displayed, similarly to for instance Chrome browser for Android.

When an older “metro-style” application is run, it is displayed in a window with specified minimum size. Also, both “maximize” and “toggle full screen mode” actions trigger the same action. “Metro-style” application notifications are displayed as basic controls area icons (4.2).

4.9 TOUCH GESTURES

Windows 8 introduced touch gestures for some of the features – swiping from screen edges to the center of the screen being the main gestures. The prototype keeps these gesture actions while modifying their meanings:

- **Top edge:** Activates the window listing feature (4.2). If not using touch controls, the same feature can be triggered by clicking an icon in the basic controls area.
- **Right edge:** Expands the basic controls area, displaying all basic control icons (4.2). If not using touch controls, this action can be triggered by clicking an up arrow symbol to the left from the basic controls area in the bar for taskbar.
- **Bottom edge:** For applications in full screen mode (4.3), context menu is toggled (this context menu can also be displayed by clicking an arrow icon in the bottom center of the application’s interface); otherwise no action is performed.
- **Left edge:** Switches between application windows within the physical screen (4.1). If not using touch controls, an alternative to this gesture is pressing Alt + Tab keys.
- **Five finger pinching within an application:** Toggles full screen mode for a particular application’s window. If not using touch controls, the same action can be triggered by clicking a button in the window’s title bar (or, alternatively, in the top right corner of the application interface when in full screen mode).

Unlike Windows 8, this prototype’s interface does not rely on the knowledge of these gestures. Actions they trigger can be intuitively accessed from other parts of the user interface. Also, these gestures can no longer be imitated using mouse moves.

Chapter 5

Prototype testing

In order to verify the usability of a multi-modal operating system user interface constructed with principles from chapter 4 in mind, a sample paper low-fidelity prototype was constructed, as well as a medium-fidelity one on a computer screen [30]. These prototypes were used in two more qualitative tests with users.

5.1 LOW-FIDELITY PROTOTYPE TEST

5.1.1 Test tasks

The tasks were constructed to either verify or deny the usability of a user interface complying with the design principles in chapter 4. Each task was designed to try to verify one or more of these principles, testing all of them except for 4.8, whose nature does not allow for usability testing.

Just like in chapter 3, an initial test with a random participant was performed in order to avoid inadequately difficult tasks. This time no modifications to the task list or task formulations had to be made.

Here is the list of tasks. Participants were also given the same instructions as in chapter 3 if not specified otherwise.

Task list:

Participants were supposed to complete the following tasks using only their finger, simulating the touchscreen:

- 1) **Open the Mail application.** This step verified that the user was able to access the Start menu without knowing any gestures. (*verifies 4.2*)
- 2) **From the inbox, delete an email.** If the user finds out how to enter the full screen mode, deleting an email using a touchscreen will work flawlessly – since no gestures are required to access the context menu and an additional arrow icon is displayed as well. (*verifies 4.6, 4.9, 4.4 and 4.5*)
- 3) **Split the screen and make the Mail application occupy exactly one half of it.** User is required to access the Window listing feature and move application windows. (*verifies 4.2 and 4.6*)
- 4) **Launch Notepad and type in the word “Šotovýlet”.** – The Notepad application does not have

a full screen mode – however, thanks to the design rules, it is usable with a touch screen in touch mode. The text field automatically scrolls if the focused text field gets out of range, plus the keyboard pops up automatically as well. Czech words can be typed easily as the standard Czech keyboard layout with more key rows was adopted for the on-screen keyboard. (*verifies 4.7*)

5) **Close all applications.** Since all applications are now treated equally, closing full screen and non-full screen applications is done similarly. (*verifies 4.3*)

6) **Mute the system audio volume.** Unlike the Windows 8 desktop, this prototype has unified basic PC controls so controlling the sound with touch controls should be flawless. (*verifies 4.2*)

Participants were supposed to complete the following tasks using only a pencil simulating the mouse cursor and an actual physical keyboard (though not connected to anything) – ignoring the touchscreen simulation with fingers. Also, participants were instructed to use the pencil's other end to simulate right mouse button.

7) **Open the Mail application.** This step verified that the user was able to access the Start menu using a button. (*verifies 4.2*)

8) **From the inbox, delete an email.** Deleting an email should work well with a mouse in both full and non-full screen modes. Right mouse click can be used to delete it in non-full screen mode. (*verifies 4.9, 4.4 and 4.5*)

9) **Split the screen and make the Mail application occupy exactly one half of it.** User is required to access the Window listing feature and move application windows. (*verifies 4.2 and 4.6*)

10) **Launch Notepad and type in the word “Šotovýlet”.** – This task should not be an issue to anyone used to using a mouse and keyboard-oriented UI.

11) **Close all applications.** Since all applications are now treated equally, closing full screen and non-full screen applications is done similarly. (*verifies 4.3*)

12) **Mute the system audio volume.** This feature should work identically to Windows 7 and older operating systems when using a mouse. (*verifies 4.2*)

5.1.2 Screener survey

The screening process was identically to 3.2, while also leaving out anyone who participated in the initial testing of Windows 8. 9 participants filled out the screener survey, three of which were chosen for the actual test. The testing process was conducted in the school building (Charles square) of the Faculty of Electrical Engineering of the Czech Technical University in Prague.

Following is the list of participants in low-fidelity prototype testing (table 5.1):

Participant	Gender	Computer-related study/job	Tablet OS used	Uses Windows 8 on a non-touchscreen device
#1	female	no	none	no
#2	female	no	iOS	no
#3	male	no	none	no

Table 5.1: Low-fidelity test participants

5.1.3 Test analysis and results

The testing process showed no significant usability drawbacks of the paper prototype designed in accordance with chapter 4 except for one.

The unverified design principle is 4.1 – the desktop splitting feature in particular, due to which this feature was removed from the medium-fidelity prototype and replaced with an ability to pin any application window or full screen application to a particular desktop, allowing up to two pinned items for each.

Table 5.2 shows the list of verified and unverified assumptions. Figure 5.1 shows the test setup.

Assumption	Verified in
4.1 – UI philosophy layers	1 out of 3 tests
4.2 – Basic controls	3 out of 3 tests
4.3 – Limited usability with one input method	3 out of 3 tests
4.4 – Unified visual style	3 out of 3 tests
4.5 – No schizophrenic applications	3 out of 3 tests
4.6 – Touch mode	3 out of 3 tests
4.7 – On-screen keyboard	3 out of 3 tests
4.9 – Touch gestures	3 out of 3 tests

Table 5.2: Verified and unverified assumptions of low-fidelity prototype testing

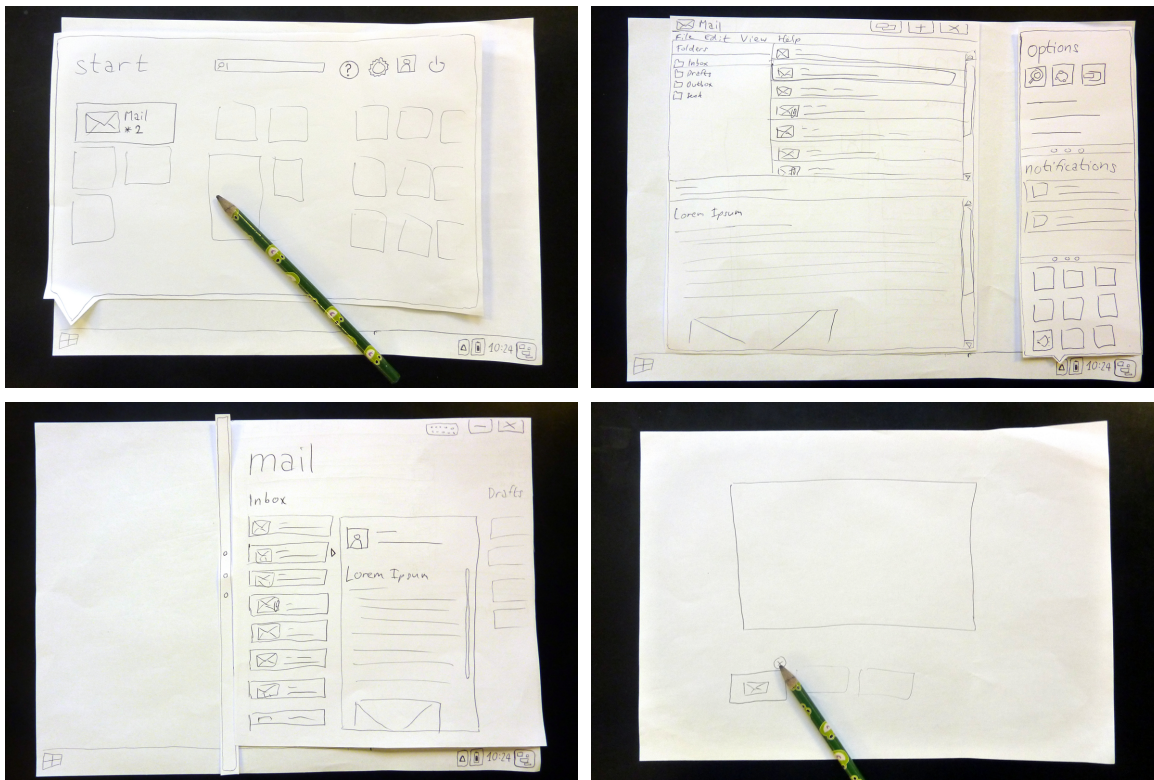


Figure 5.1: Low-fidelity prototype test setup

5.2 MEDIUM-FIDELITY PROTOTYPE TEST

A prototype similar to the one in 5.1 was constructed using Microsoft Visual Studio for a final qualitative test with users – with the exception of a split desktop feature, which was removed, as mentioned in 5.1.3. The prototype was conducted in my apartment’s living room on a Sony VAIO SVF15 laptop with a touchsreen running VMWare Player virtualization software on top of Windows 8.1.

The task list included all tasks from 5.1.1 with the exception of tasks 3 and 9 – which were replaced by tasks asking participants to pin the Mail application to a half of the desktop.

Users from the initial chapter 3 test were invited to re-participate. One of the users had to be left out from the test since he started using a Windows 8.1 tablet since then. All instructions and other test setup aspects remained the same as in chapter 3.

Table 5.3 shows the list of medium-fidelity prototype test participants:

Participant	Gender	Computer-related study/job	Tablet OS used	Uses Windows 8 on a non-touchscreen device
#1	male	no	Android	yes
#2	female	no	Android	no

Table 5.3: Medium-fidelity prototype testing participants

5.2.1 Test results

The test showed no usability drawbacks of the prototype, verifying all of the test assumptions. Table 5.4 shows the test results.

Assumption	Verified in
4.1 – UI philosophy layers	3 out of 3 tests
4.2 – Basic controls	3 out of 3 tests
4.3 – Limited usability with one input method	3 out of 3 tests
4.4 – Unified visual style	3 out of 3 tests
4.5 – No schizophrenic applications	3 out of 3 tests
4.6 – Touch mode	3 out of 3 tests
4.7 – On-screen keyboard	3 out of 3 tests
4.9 – Touch gestures	3 out of 3 tests

Table 5.4: Medium-fidelity prototype testing results

Chapter 6

Conclusions

Both external sources – including professional critics and operating system users – and my own experience (1.3.3) were indicating serious usability drawbacks in the current Windows 8 multi-modal user interface (2). After an analysis of these sources, the following most serious drawbacks were identified:

- **Application style inequality (2.1):** “Metro-style” applications and desktop applications are installed and updated from different parts of the system, do not have equal access to system features, and are displayed differently in task switcher UIs.
- **Schizophrenic built-in applications (2.2):** The operating system includes applications that duplicate other built-in applications’ functionality and purpose. Some of these pairs of applications share data between them, some do not – and often have different feature sets.
- **Inconsistency of design decisions (2.3):** Some decisions Microsoft made involving the inclusion of certain built-in system applications are not consistent throughout the system, which caused experienced users to report confusion.
- **Limited usability of an entire system with one particular input method (2.4):** Schizophrenic nature of the operating system makes very hard to make use of all system features using only one particular input method – either the touchscreen, or a combination of a mouse/trackpad and a physical keyboard. The integrated on-screen virtual keyboard also indicated usability flaws.
- **Basic PC controls have inconsistent style (2.5):** System-wide notifications and dialogs are displayed in two different styles, making it hard to use some with a mouse (battery warnings, for instance), or others with a touchscreen (User Access Control dialogs, for instance).

Four of these listed drawbacks (excluding 2.3 – whose nature does not allow for verification with usability testing) were used as a base for the tasks of a qualitative usability test with users (chapter 3).

All of the assumptions corresponding to drawbacks from chapter 2 were verified (3.5) and served as principles for the design of a new consistent multi-modal user interface prototype for possible future versions of Windows. The principles were all described in chapter 4, with the following being the most important:

- Strictly stick to UI philosophy layers without exceptions (4.1).
- Make basic PC controls such as networking, Start menu, or window management united and usable for both types of user input equally (4.2).
- Treat all applications equally – get rid of the distinguishing between “metro-style” and desktop applications, design all applications primarily for the desktop (with touch controls in mind as well) and add a primarily touch-optimized interface as an optional full screen mode. Never duplicate built-in applications’ purposes. Also, make the installation, management and updating of applications centralized (4.3 and 4.5).
- Use the “metro” graphical style in both non-full screen modes and full screen modes (standardizing the guidelines for non-full screen mode), for instance refrain from using non-flat style icons (4.4).
- Include a “touch mode” feature (only for touchscreen enabled physical displays) which will make use of increased DPI settings. Make some standard system UI widgets act differently when activated using touch (4.6).
- Fix the on-screen virtual keyboard using principles from 4.7.

In chapter 5, principles from chapter 4 were applied to a low-fidelity paper UI prototype and a computer-based medium-fidelity prototype, which, using more usability tests with users, both proved the design from chapter 4 to be usable and fixing usability drawbacks from the current Windows 8.

6.1 IDEAS FOR FUTURE THESIS EXPANSION

Since the thesis includes findings and ideas that be found interesting by the general user base of Microsoft Windows, presenting the thesis results on a dedicated interactive website is recommended.

Creating a sample new guideline document for possible third party software developers is also recommended.

References

- [1] AOL INC. *Windows 8 product overview*. Engadget [online]. [cit. 2015-01-04]. Available at: <http://www.engadget.com/products/microsoft/windows/8/>
- [2] REED, Brad. *Windows 7 has the most to gain from XP's impending demise*. Boy Genius Report [online]. 2014-04-03 [cit. 2015-01-04]. Available at: <http://bgr.com/2014/04/03/windows-7-vs-windows-8-market-share/>
- [3] MURDOCK, Ian. *On the importance of backward compatibility* [online]. 2007-01-14 [cit. 2015-01-04]. Available at: <http://ianmurdock.com/platforms/on-the-importance-of-backward-compatibility/>
- [4] DACHIS, Adam. *Mac vs. Windows: Your Best Arguments*. Lifehacker [online]. 2014-04-30 [cit. 2015-01-04]. Available at: <http://lifehacker.com/mac-vs-windows-your-best-arguments-486125257>
- [5] VAUGHAN-NICHOLS, Steven J. *Operating systems don't matter much anymore*. Computerworld [online]. 2011-12-05 [cit. 2015-01-04]. Available at: <http://www.computerworld.com/article/2550133/operating-systems/operating-systems-don-t-matter-much-anymore.html>
- [6] ETHERINGTON, Darrell. *Mac achieves highest U.S. PC market share ever in Q3 2014*. TechCrunch [online]. 2014-11-07 [cit. 2015-01-04]. Available at: <http://techcrunch.com/2014/11/07/mac-achieves-highest-u-s-pc-market-share-ever-in-q3-2014-according-to-idc/>
- [7] KEIZER, Gregg. *Chromebooks may grab 5 % of PC market, about the same as Apple's Mac*. Computerworld [online]. 2014-08-12 [cit. 2015-01-04]. Available at: <http://www.computerworld.com/article/2491109/windows-pcs/chromebooks-may-grab-5-of-pc-market-about-the-same-as-apple-s-mac.html>
- [8] BRIGHT, Peter. *What does Windows 8.1 offer to desktop die-hards?* Ars Technica [online]. 2013-06-27 [cit. 2015-01-04]. Available at: <http://arstechnica.com/information-technology/2013/06/what-does-windows-8-1-offer-to-desktop-die-hards/>
- [9] RIVARD, S. and HUFF, S. L. *Factors of success for end-user computing*. Communications of the ACM. New York: Association for Computing Machinery, 1988.

- [10] THURROTT, Paul. *Dueling Views on Windows 8 Usability*. Paul Thurrott's SuperSite for Windows [online]. 2012-11-20 [cit. 2015-01-04]. Available at: <http://winsupersite.com/windows-8/dueling-views-windows-8-usability>
- [11] CLARK, Jack. *Worldwide PC sales drop 1.7 % in third quarter, IDC says*. Bloomberg [online]. 2014-10-09 [cit. 2015-01-04]. Available at: <http://www.bloomberg.com/news/2014-10-08/worldwide-pc-sales-drop-1-7-in-third-quarter-idc-says.html>
- [12] HINER, Jason. *Smartphone marketshare showdown in 2008: iPhone vs. Windows Mobile vs. BlackBerry*. TechRepublic [online]. 2007-12-31 [cit. 2015-01-04]. Available at: <http://www.techrepublic.com/blog/tech-sanity-check/smartphone-marketshare-showdown-in-2008-iphone-vs-windows-mobile-vs-blackberry/>
- [13] SULLIVAN, Kent, Alex CALVO, Adam DAWES, Fabio Claudio FERRACCHIATI, Emanuele GAROFALO, Rob CAMERON, Kyle BURNS, Rob CAMERON, Fabio Claudio FERRACCHIATI, Emanuele GAROFALO, Fabio Claudio FERRACCHIATI and Emanuele GAROFALO. *The Windows 95 user interface: a case study in usability engineering*. Proceedings of the 19th international conference on Software engineering – ICSE '97. 1997. DOI: 10.1007/978-1-4302-4138-6_3.
- [14] WORTHINGTON, David and TOMLINSON, James E. *Microsoft Windows XP operating system: usability study*. Bloomsburg University Department of Communication Studies. 2001.
- [15] MICROSOFT CORPORATION. *User Experience Interaction Guidelines for Windows Vista and Windows 7*. Redmond, WA: Microsoft Corporation, 2010. Available at: <http://www.microsoft.com/en-us/download/details.aspx?id=2695>
- [16] ZHENG, Long. *State of the Windows UX Taskforce* [online]. 2008-06-11 [cit. 2015-01-04]. Available at: <http://www.istartedsomething.com/20080611/state-of-the-taskforce/>
- [17] BIDDLE, Sam. *Microsoft: metro isn't metro anymore*. Gizmodo [online]. 2012-08-02 [cit. 2015-01-04]. Available at: <http://gizmodo.com/5931381/windows-metro-isnt-metro-anymore>
- [18] *The metro style design guide*. BLINK INTERACTIVE, Inc. [online]. [cit. 2015-01-04]. Available at: <http://blinkux.com/metro/>
- [19] CHANG, Alexandra. *The differences between Windows 8 and Windows RT explained*. Wired Magazine Online [online]. 2012-10-22 [cit. 2015-01-04]. Available at: <http://www.wired.com/2012/10/windows8-windows-rt-explainer/>

- [20] MICROSOFT CORPORATION. *Windows 8 User Experience Guidelines*. Redmond, WA: Microsoft Corporation, 2012. Available at: <http://www.microsoft.com/en-us/download/details.aspx?id=30704>
- [21] BLODGET, Henry. *If these 5 people who tried Windows 8 are normal, Microsoft has a big problem on its hands*. Business Insider [online]. 2012-10-21 [cit. 2015-01-04]. Available at: <http://www.businessinsider.com/microsoft-has-big-problem-with-windows-8-2012-10>
- [22] NIELSEN, Jakob. *Windows 8 – disappointing usability for both novice and power users*. Nielsen Norman Group. 2012.
- [23] DVORAK, John C. *The schizophrenic nature of Windows 8*. PC Magazine Online [online]. 2012-02-08 [cit. 2015-01-04]. Available at: <http://www.pcmag.com/article2/0,2817,2400010,00.asp>
- [24] PROTALINSKI, Emil. *Windows Live to replace built-in Windows applications in 7*. Ars Technica [online]. 2008-09-23 [cit. 2015-01-04]. Available at: <http://arstechnica.com/information-technology/2008/09/windows-live-to-replace-built-in-windows-applications-in-7/>
- [25] BOTT, Ed. *Introducing Windows 8.1 for IT professionals*. Redmond, WA: Microsoft Press, 2013. ISBN 978-0735684270.
- [26] BARNUM, Carol C. *Usability testing essentials: ready, set... test!* Burlington, MA: Morgan Kaufmann, 2010. ISBN 978-0123750921.
- [27] GARFINKEL, Simson. *Windows 8: design over usability*. MIT Technology Review. 2013, Issue 2013, 3–4.
- [28] FORLINES, Clifton, Daniel WIGDOR, Chia SHEN and Ravin BALAKRISHNAN. *Direct-touch vs. mouse input for tabletop displays*. Proceedings of the SIGCHI conference on Human factors in computing systems – CHI '07. 2007. DOI: 10.1145/1240624.1240726.
- [29] ZIMA, Jiří. *Windows 8 v pracovním nasazení (1): úvod, zápory*. NotebookBlog [online]. 2013-03-07 [cit. 2015-01-04]. Available at: <http://notebookblog.cz/ostatni/software/windows-8-v-pracovnim-nasazeni-1-uvod-zapory/>
- [30] SAUER, Juergen, Andreas SONDEREGGER, Diana RUEDA, René HOTO, Andrés CONEJERO, Jonathan ARNOWITZ, Michael ARENT, Nevin BERGER and Florian RESATSCH. *The influence of prototype fidelity and aesthetics of design in usability tests: Effects on user behaviour, subjective evaluation and emotion*. Applied Ergonomics. 2009, vol. 40, issue 4, DOI: 10.1007/978-3-8349-8683-2_5.

[31] SHARWOOD, Simon. *N00bs vs Windows 8: we lock six people in a room with new OS*. The Register [online]. 2012-10-26 [cit. 2015-01-04]. Available at: http://www.theregister.co.uk/2012/10/26/civilians_test_windows_8/

[32] MACHALANI, Jay. *Fixing Windows 8*. [online]. 2013-12-12 [cit. 2015-01-04]. Available at: <http://jaymachalani.com/blog/2013/12/12/fixing-windows-8>

[33] *Aero Taskforce* [online]. [cit. 2015-01-04]. Available at: <http://www.aerotaskforce.com/>

[34] Image source:

<http://www.xatakawindows.com/windows/seguridad-en-windows-8-smartscreen-filter>

[35] Image sources:

http://en.wikipedia.org/wiki/Windows_XP_Media_Center_Edition

<http://www.amazon.com/Zune-Digital-Media-Player-Black/dp/B000EPLP3C>

<http://carltonbale.com/editing-the-start-menu-in-windows-vista-media-center/>

<http://en.wikipedia.org/wiki/Zune#mediaviewer/File:Zune80and4.jpg>

<http://www.wikihow.com/Upload-CDs-to-a-Zune>

<http://blog.gsmarena.com/xbox-360-getting-a-metro-ui-styled-dashboard-tomorrow/>

http://en.wikipedia.org/wiki/Windows_Mobile_6.5

<http://www.doobybrain.com/2009/05/26/hardware-highlight-the-microsoft-zune-hd/>

<http://engadget.com/2010/02/17/windows-phone-7-series-everything-you-ever-wanted-to-know/>

Appendix A

Abbreviations listing

API	Application programming interface
CD	Compact disc
Ctrl	The Control key
Del	The Delete key
HTML	Hypertext Markup Language
HTTP	Hypertext Transfer Protocol
HTTPS	Secure Hypertext Transfer Protocol
Inc.	Incorporated
IP	Internet Protocol
OS	Operating system
PDF	Portable Document Format
RTM	Release to manufacturing
SDK	Software development kit
USB	Universal Serial Bus
UI	User interface
UX	User experience
WP	Windows Phone
XP	Experience

Appendix B

Testing log sample

Tests in the chapters 3 and 5 were logged in the following format. This excerpt is from the testing with participant #1 in chapter 3.

00:03 Participant unlocks the tablet and is presented the metro-style start screen.
00:13 Participant taps the Internet Explorer tile.
00:21 Participant taps the URL text field and virtual keyboard is displayed.
00:22 Participant types in the URL. Since he only sees the '.cz' suffix on the keyboard, he types '.com' manually.
00:53 The web page starts loading.
01:05 Participant taps the 'Options' icon.
01:10 Participant taps the 'Pin' icon which is still inactive.
01:16 Participant taps the 'Pin' icon.
01:20 Participant complains about not getting any feedback whether the website was bookmarked.
01:41 Participant taps the URL bar to type in the next URL.
01:51 The other website is loaded.
01:57 Participant taps the 'Back' button.
02:00 Participant tries many different gestures.
02:06 Participant opens the metro-style start screen.
02:22 Participant taps the Internet Explorer tile.
02:31 Participant taps the 'Options' icon.
02:40 Participant tries out different gestures again.
02:58 Participant uses the 'back' gesture.
03:10 Participant taps the 'Options' icon.
03:15 Participant tries out different gestures yet again.
03:34 Participant swipes screen from the bottom.
03:40 Participant closes the newly opened menu and does the 'back' gesture.
03:58 Participant taps the 'Pin' icon.
04:03 Participant swipes screen from the bottom.

04:13 Participant taps the 'plus' icon.
04:18 Participant tries to scroll right in the 'Favorites' area with no luck.
04:22 Now he tries scrolling again and is lucky this time.
04:26 Participant taps the bookmarked page's tile.
04:40 Participant swipes screen from the right and opens the start screen.
04:57 Participant taps the Internet Explorer tile again.
05:03 Participant taps the 'Pin' icon.
05:10 Participant tries out various different gestures.
05:16 Participant is instructed that the option is only available in the desktop version of Internet Explorer. Tries to switch to the desktop version.
05:30 Participant swipes screen from the bottom.
05:35 Participant taps the three dots icon.
00:15 Participant swipes screen from the top.
00:30 Participant taps the 'plus' button.
00:35 Participant scrolls the 'pinned and bookmarked pages' tile row to the right.
00:46 Participant reloads the bookmarked page.
00:51 Participant opens the start screen.
00:54 Participant taps the Internet Explorer tile.
01:00 Participant taps the URL text field.
01:20 Participant taps the 'Options' icon.
01:23 Participant taps the 'Show on desktop' option.
01:35 Participant tries to tap the 'Options' icon several times.
01:43 Participant taps the 'Options' icon.
02:07 Participant taps the 'Favorites' icon.
02:16 Participant switches tab in the favorites pane.
02:20 Participant taps 'Favorites options'.
02:38 Participant pins the favorites pane to the left part of the browser window.
02:47 Participant taps the 'Home' icon and nothing happens.
02:55 Participant closes the pinned favorites pane.
02:57 Participant taps the 'Home' icon and home page is loaded.
03:05 Participant taps the back button.
03:28 Participant tries to tap the 'Options' icon.
03:30 Participant taps the 'Options' icon.
03:43 Participant taps the 'Options' icon.

04:16 Participant taps 'Internet Options'.

04:23 Participant taps 'Use current'.

04:29 Participant taps 'Apply' and 'OK'.

04:50 Participant closes Internet Explorer.

Appendix C

CD-ROM contents

- 📁 misc (miscellaneous content)
- 📁 original_thesis_idea (the original thesis idea presentation from 2013)
 - 📄 overview.pdf (an overview of the idea)
 - 📄 slideshow.pdf (a slideshow presenting the idea)
- 📁 prototypes (user interface prototypes)
- 📁 text (text of this diploma thesis)
 - 📁 images (source image renders used in the thesis)
 - 📄 text.odt (source of the thesis in OpenDocument format)
 - 📄 text.pdf (this thesis in PDF format)

Appendix D

Context notes

As mentioned in (1), the testing and prototype design was performed in February and March 2014. Since then, Microsoft announced a new version of Windows – Windows 10 – whose purpose is to remove major usability drawbacks of Windows 8 and make the system more usable and understandable when controlled with a mouse/trackpad and a physical keyboard.

Microsoft promised to preview the system's aspects of user interface multimodality in late January 2015 – it will be interesting to see how similar or different Microsoft's solutions will be compared to this thesis.

Also, one interesting study on Windows 8 interface gained a lot of attention on technology-related websites lately – “Fixing Windows 8” by Canadian UX designer Jay Machalani [32]. While his work focused on the easiness of implementation – calling the prototype “Windows 8.2” – unlike the prototype from this thesis, which modifies some of the fundamental system core aspects, the designer identified very similar user interface drawbacks and some of his solution approaches are similar to this thesis.

Both of these facts can serve as another verification of what test results in this thesis have shown – and emphasize the seriousness of Windows 8's usability issues.