MULTIPLE USER INTERFACES AND CROSS-PLATFORM USER EXPERIENCE: THEORETICAL FOUNDATIONS

Khalid Majrashi[1], Margaret Hamilton and Alexandra L. Uitdenbogerd[2]

[1]School of Computer Science and Information Technology,
RMIT University, Australia Institute of Public Administration and
Ministry of Higher Education, KSA
majrashik@ipa.edu.sa
[2]School of Computer Science and Information Technology,
RMIT University, Australia
margaret.hamilton@rmit.edu.au,alexandra.uitdenbogerd@rmit.edu.au

ABSTRACT

Evaluating the user experience of cross-platform interactive systems has become a research issue of increasing importance. There is a lack of clear concepts and definitions for testing, evaluating or even teaching cross-platform user experience. In this paper, we review the actual meanings and interpretations of different concepts in the field of Human-Computer Interaction (HCI) relevant to cross-platform service usage. We also investigate the traditional definitions of usability and user experience before extending them to develop precise definitions for cross-platform usability and user experience. Our paper builds on existing theories to establish the theoretical foundations that can help us better conceptualise cross-platform user experience evaluation.

KEYWORDS

Cross-platform, User Experience, Usability, Multiple User Interfaces

1. INTRODUCTION

Nowadays, end-users can interact with a service and information using different types of computing platforms including traditional office desktops, smart TVs, tablets, and mobile phones. This allows users to migrate their tasks from one user interface to another across devices or platforms. For example, a user can search for a restaurant from specific service, and then switch to the service image from their mobile phone to find the restaurant contact information, and then might transit to use a tablet to write a review about the restaurant. This brings a new user experience theme in which a user interacts with Multiple User Interfaces (MUIs) to achieve goals horizontally (across platforms). This type of MUI access is different from traditional user experience involving interaction with a single user interface (vertical interaction) [1]. There are new aspects in MUI interaction, including, switching a process from one user interface to another, migrating knowledge and tasks from one user interface to another. Despite the increased use of MUIs, and the corresponding increase in the need for cross-platform user experience evaluation,

Dhinaharan Nagamalai et al. (Eds) : CCSEA, DKMP, AIFU, SEA - 2015
pp. 43–57, 2015. © CS & IT-CSCP 2015 DOI : 10.5121/csit.2015.50205

there is a lack of explanations, definitions, and discussions of concepts in the context of cross-platform user experience.

In this paper, we review and explain different concepts related to cross-platform service, its characteristics, as well as relevant concepts in HCI. We follow this review by presenting the definitions of traditional forms of usability and user experience, and exploring the differences between them. This is to eliminate possible confusion between the two terms before defining them in the context of cross-platform interaction. Then, we provide comprehensive definitions that explain the concepts for both usability and user experience in the context of cross-platform service.

2. Cross-Platform Service

In this section, we provide an overview of cross-platform service, including, definitions of the terms service and cross-platform usage, approaches for connecting a service, and configuration of cross-platform services.

2.1. What is a Service?

A service refers to software and hardware in which one or more services can be used to support a business's needs and functions. There are two primary types of services: atomic and composite [2, 3]. An atomic service is a self-contained function that does not require the use of other services. A composite service is an assembly of atomic or other composite services that may require the use of another service contained within the same composite service.

2.2. What is a Cross-Platform?

The term cross-platform can be used to characterise different entities in computer science [4]. For example, hardware devices, such as computer monitors, can be described as cross-platform as they can work with any operating system. Similarly, programming languages, such as C, can be described as cross-platform as they can be used to write software for use on any operating system. In addition, the term can be used to refer to software that can operate on more than one platform. For the purpose of this paper, we use the term cross-platform to refer to a service that can be accessed and used on two or more computing platforms.

2.3. Connection of Services

Web services provide the technologies for connecting services together. For cross-platform services, a web service can be defined as a system, which can be designed to support interoperable application-to-application communication over a network [5]. Interoperability can refer to both syntactic interoperability, and semantic interoperability [6, 7]. Syntactic interoperability depends on specified data formats and communication protocols to ensure communication and data exchange between heterogeneous software applications. With syntactic interoperability, there is no guarantee of consistent interpretations of exchanged data from one application to another. Semantic interoperability refers to the ability of various services across platforms to interpret the exchanged information meaningfully and accurately. There are multiple technologies of Web services for connection services, including the use of SOAP, WSDL, UDDI, REST, XML, and JSON, which are explained briefly as follows [8]:

1. Simple Object Access Protocol (SOAP) is a protocol for enabling communication between applications.

2. Web Service Description Languages (WSDL) is used to define web service interfaces, data and message types, and protocol mapping.
3. Universal, Description, Discovery, and Integration (UDDI) is a web service registry and discovery mechanism, used for sorting business information, and retrieving pointers to web service interface.
4. Extendable Markup Language (XML) provides a language for defining data and processing it.
5. Representational State Transfer (REST) is an alternative to SOAP that is developed on a set of principles describing how networked resources are defined and addressed.
6. JSON (JavaScript Object Notation) is an alternative to XML that uses name/value pairs instead of tags as used in XML.

2.4. Configuration of Cross-Platform Services

A cross-platform service aims to provide pervasive and synergistic support for human activities in different contexts of use. Feiner [9] presented the concept of hybrid user interfaces in which multiple heterogeneous displays and interaction devices are used synergistically to benefit from the features for each of them. Services across devices can be configured based on different user and/ or business needs, considering different device constraints and capabilities. Configuration of a cross-platform service refers to the manner in which devices are organised and the service is delivered across these devices [10]. Before discussing device organisation and service delivery, we need to clarify the concept of synergistic specificity, which is associated with different methods of configuration**.**

2.4.1. Synergistic Specificity

Systems across multiple platforms can reach their complete planned potential advantages when their components are used synergistically. Synergistic specificity is a term introduced by Schilling [11] to describe “the degree to which a system achieves greater functionality by its components being specific to one another” within a specific configuration. Systems with high synergistic specificity may be able to support functionality and user experiences better than segmental systems. These days many systems have core functionalities across platforms that rely on optimal coordination between their components to work with each other. These systems can lose their intended performance or become completely paralysed if their cross-platform components are used in isolation [12]. An example of a system with a high degree of synergistic specificity is a fitness system, whereby a system in a wearable device collects data (e.g., heart rate), and a web service visualises data in a meaningful way.

2.4.2. Device Organisation

In most situations of multi-device service, data and functions are distributed across devices and cannot be completely sourced from a single device. This is due to two main reasons. Firstly, technical constraints of mobile devices prevent accessing the full advantages of a large amount of data and complex functions. Secondly, device-unique capabilities can allow only the access of some functions from a specific device. For example, non-mobile devices may not have mobile device capabilities such as geo-location services, accelerometer, camera, gyroscope, and video recording. Denis and Karsenty [13] outlined three degrees of device redundancy representing levels of data and functions availability across devices.

The first level is redundant, where all the interactive systems across devices can allow access to the same data and functions. In this redundancy level, multi-device service can be classified into two types. Responsive redundant service refers to the multi-device service with the same data and functions adapted to varying screen sizes, resolutions, aspect ratios and orientations, see Figure 1. Independent redundant service refers to multi-device service with the same data and functions with different appearance or structure in each device, see Figure 1.

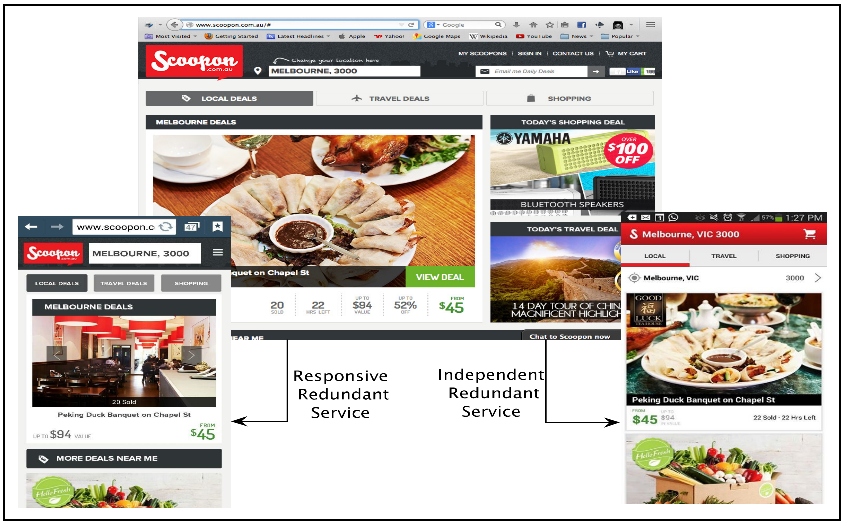

Figure 1: Responsive redundant service versus independent redundant service, for Scoopon service (www.scoopon.com)

The second level of device redundancy is exclusive, where each interactive system in each device gives access to different data and functions. This level of redundancy has the lowest degree of synergistic specificity. An example of this type of redundancy can be found with the Samsung WatchOn multi-device system (www.samsung.com/us/watchon/), which is composed of an interactive TV system and a native mobile app that is used as a remote control for the television service. From the mobile application, users can choose programs directly from their mobile devices to watch on the high-quality display screen of the television. They can also share favourite TV shows with friends.

The third level of device redundancy is complementary, whereby the interactive systems in all devices have a zone of shared data and functions, but one or more of the devices offer access to data or functions that are inaccessible on the other device(s). An example of this redundancy level can be found in Evernote multi-device service (www.evernote.com). The service allows the user to write notes of all types that can then be accessed from different devices. The interactive systems of Evernote across all devices have a shared zone of functions. However, some functions can only be found with mobile devices, such as taking a photo using a device camera to include it in user notes.

As the number of interactive cross-platform systems increases, there is greater probability of more varied configuration of device redundancies of data and functions. Figure 2 shows different degrees of device redundancy of a multi-device service across three devices.

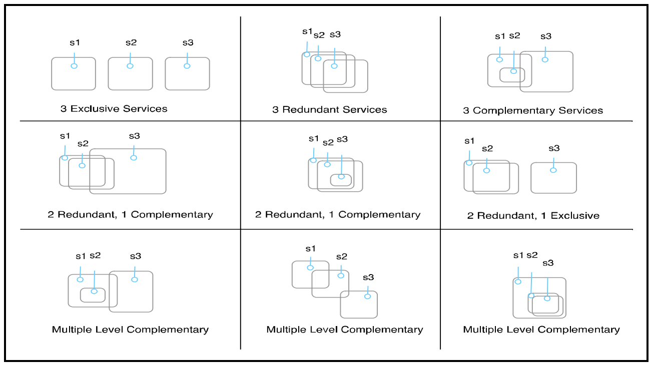

Figure 2: Degrees of device redundancy of a multi-device service across three devices

2.4.3. Service Delivery

Service delivery can be defined as the technique of delivering services to multiple devices. There are three main types of service delivery. The first type is multichannel service delivery. The concept of multichannel service delivery was coined in marketing research to describe all multiple routes (including on and off-line channels) by which customers and business interact with each other [14]. In pervasive computing and HCI, multichannel service delivery refers to the concept of functionality and content being channelled through multiple devices. The aim of this type of service delivery is to provide the practical means for anytime and anywhere customisation of a service for changing user needs and business requirements, and support access to functionality and information from multiple channels [15, 16]. Multichannel service often requires redundant or complementary devices in which core functionalities are supported in different devices. Figure 3 illustrates the conceptual view of a multichannel service delivery in which functionality and content are being channelled through multiple devices into system images.

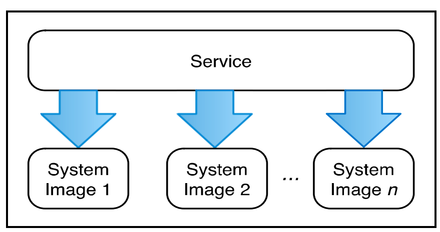

Figure 3: The conceptual view of multichannel service delivery

The second type of service delivery is cross-media (also referred to as transmedia). The term cross-media has been developed in the context of communications research traversing two computer science fields, namely, pervasive computing and human-computer interaction [17]. The term is used in communications research to describe a communication format in which the storyline invites the receiver (user) of media to cross over from one medium to the next in an attempt to achieve a goal or receive a full story [18]. For example, a user finishes watching a TV show and then follows a URL provided at the end of the show to further explore the show. The cross-media concept requires a range of devices including TVs, mobiles, PCs and so on to distribute the content and spread a story across different platforms [19]. With cross-media communication, the systems across devices are designed to be experienced fragmentarily, see Figure 4. Thus, cross-media services are highly synergistic in that users can only achieve a goal if they use the full package of systems as no single system can provide the full package of the content, see example in [20]. In contrast, multichannel services are usually characterised with less synergistic specificity as users can achieve a goal using any number of channels in tandem or in an isolated manner. In comparison with multichannel services, cross-media services tend to employ exclusive device redundancy, and sometimes complementary device redundancy.

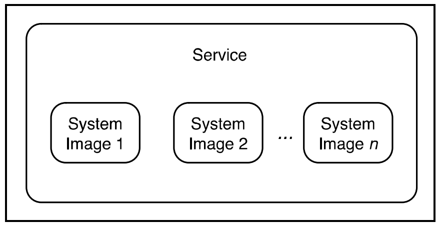

Figure 4: The conceptual view of cross-media service delivery

The third type of service delivery occurs in a cross-channel format where functionality and content is channelled through multiple devices but not in full mode like in multichannel service delivery, see Figure 5. In comparison with cross-media services, users with cross-channel services can achieve a goal within an individual channel without having recourse to other channels. However, in contrast to multichannel services, users cannot interact with all functions and content from a single channel in cross-channel services, which means that there will be at least one central service that includes all content and functions. Cross-channel services often employ complementary device redundancy and have a medium level of synergistic specificity between cross-media and multichannel services. An example of this type of service delivery can be found in the YouTube cross-channel service (wwe.youtube.com), whereby users can access full content and functions when using PCs, and can access fewer functions and content when using the service through Internet TV (e.g., AppleTV: www.apple.com/au/appletv/). Figure 5 illustrates the conceptual view of cross-channel service delivery in which functionality and content are being channelled through multiple devices into system images. The level of functionality and content being channelled to system images can differ from one device to another.

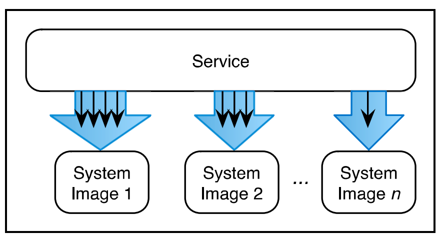

Figure 5: The conceptual view of cross-channel service delivery

3. HCI-RELATED TERMINOLOGY AND CONCEPTS

Several terms associated with cross-platform compilations have been developed in the literature. In the following section, we have reviewed some of these terms.

3.1. Distributed User Interface

There are several concepts associated with the term Distributed User Interface (DUI). One of the early concepts of DUI was migratory applications, introduced by Bharat and Cardelli [21], to describe applications that are capable of roaming on the network instead of being limited to an individual computer. The plasticity of a user interface concept is also associated with DUI, referring to the capability to adapt application interfaces to a new device with different capabilities of input (e.g., touch, stylus, or mouse) and output (e.g., screen sizes in laptop, or mobile) [22]. Multi-device interaction technique is another concept used in the context of DUI for input redirection where input events entered by users from one device are sent to another device in the same environment [23]. An example of multi-device interaction techniques can be found in the multi-display pointers that move across multiple views [24]. As far as output technique is concerned, content redirection is the most common distribution concept relevant to the term DUI. It refers to redirecting content (e.g., graphical output) across several devices [25, 26]. DUI has also been used widely in several publications, to describe interactive systems that extend across devices (see e.g., [27-29]).

3.2. Multiple User Interface

The term Multiple User Interface (MUI) was first introduced by Seffah [30], and has subsequently gained widespread acceptance among HCI researchers (see e.g., [31, 32]). Seffah [30] used MUI to refer different views of the same information and manage the services that can be accessed by users from different computing platforms. Computing platforms in [30] refers to a combination of hardware (e.g. office desktops, laptops, mobile phones, and tablets), operating systems (e.g. iOS, Windows, Mac OS), computing capabilities and UI toolkit. MUIs can support different interaction styles across platforms, which need to take into account constraints of each device [33]. The concept of MUI is different from multi-device user interface. Multi-device user interface is concerned with whether user interface across devices are able to allow a user to interact with them with any input style [33]. This is different from the MUI concept, which is concerned with different views of the same service across platforms.

There are four main aspects of MUI [34]. Firstly, MUI allows an individual user or a group of users to interact with server-side services using different interaction styles. Secondly, MUI can be designed to support interrelated tasks that can be achieved using more than one device. Thirdly, although a user interface in each device may have its unique look and feel, MUI can display features, functions, and information that can have the same behaviour across platforms. Finally, MUI refers to various assortments of a single service, for example a user interface, for different computing devices.

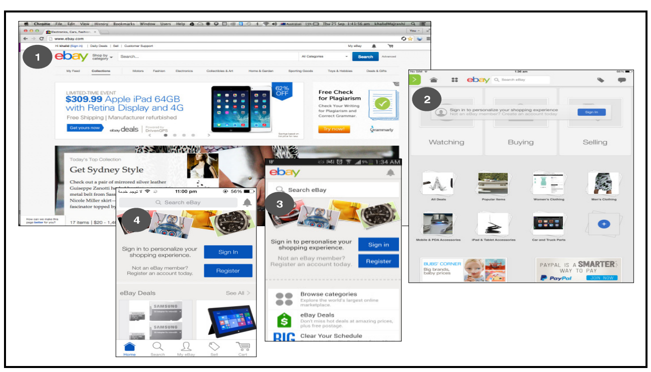

Figure 6: MUIs of eBay service across different devices and interaction styles

Figure 6 shows an MUI to the same system (www.ebay.com) from four different devices (laptop, iPad, Samsung Galaxy and iPhone). MUIs can be a combination of interaction styles [33]. The eBay system across platforms consists of four interaction styles, web based user interface, (1) in Figure 6, native iPad application, (2) in Figure 6, native android application, (3) in Figure 6, and native iPhone application, (4) in Figure 6. Hence, MUI can be a combination of any possible interaction styles that can exist across platforms, see e.g., Figure 7.

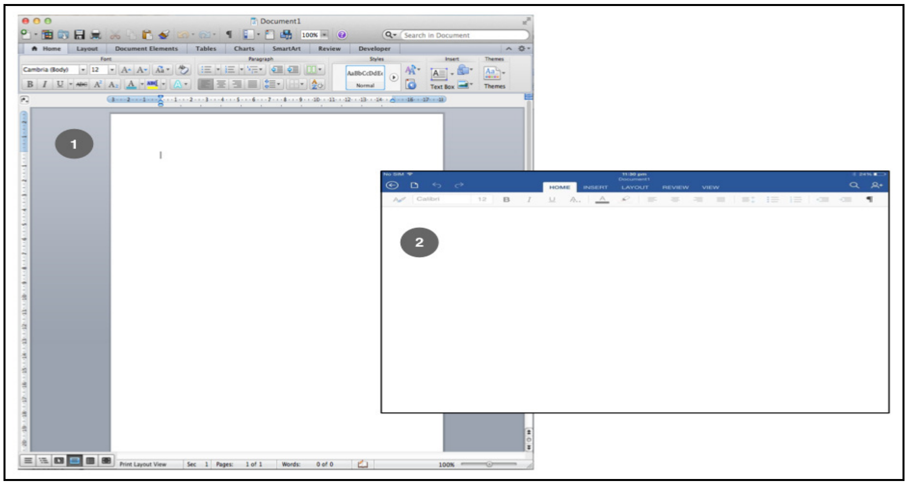

Figure 7: MUIs of Microsoft Word across two interaction styles, where (1) refers to a Graphical User Interface from Laptop running Mac OS and (2) a Native iPad Application from an iPad

There is a lack of research focus in the literature on classifying MUIs. In the following, we have attempted to categorise MUIs into three different models: on-demand model, independent model, and hybrid model. In the on-demand model, the service model can be stored in a single information repository, and delivered to the user on demand, as the user can request the service using a web browser. Figure 8 shows the on-demand model for Amazon's MUIs, which were accessed using web browsers across two devices.

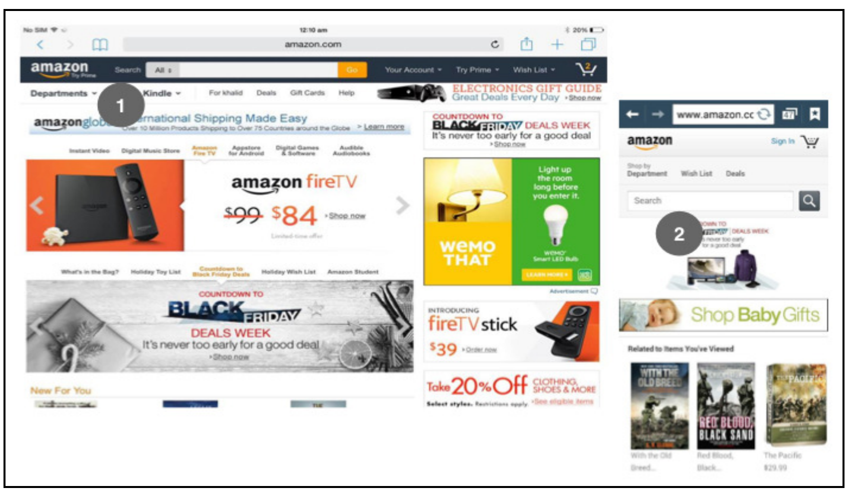

Figure 8: MUIs of Amazon service form a single repository, where (1) refers to the Amazon service accessed via web browser from iPad, and (2) refers to Amazon service accessed via web browser from Samsung Galaxy phone

In the independent model, the service model can be distributed among independent systems, while each view of the MUI can be seen as an all-inclusive user interface for each specific platform that runs it. In this type of MUI, function and information can vary from one platform to another. Figure 9 shows the independent model of Amazon's MUIs for two independent user interfaces installed on two different devices.

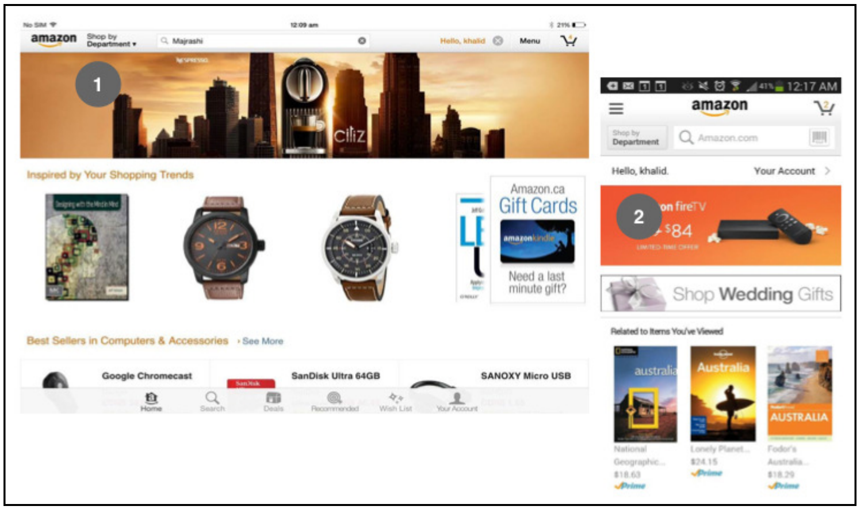

Figure 9: MUIs of Amazon service installed on different independent systems, where (1) refers to Amazon service installed on iPad, and (2) refers to Amazon service installed on Samsung Galaxy phone

In the hybrid model, the MUI can be a combination of on-demand and independent models, including services that can be accessed using web browsers and services installed on computing devices. The combination of web based application, see Figure 8, and native device applications, see Figure 9, represents the hybrid model of Amazon's MUIs.

4. Traditional User Experience and Usability

In this section, we review traditional concepts of user experience and usability and also discuss the differences between them.

4.1. Traditional User Experience

User experience (UX) is a term used broadly by HCI practitioners and researchers to represent a variety of meanings [35]. UX is considered as an umbrella term for a range of dynamic concepts, such as traditional usability (see e.g., [36, 37]), affective, and emotional (see e.g., [38-41]), experiential (see e.g., [35, 42]), hedonic (see e.g., [43, 44]), aesthetic (see e.g., [45]), and values variables. There is also an argument that user experience goes far beyond interaction with user interfaces. For example, Jakob and Don [46] have suggested that people need to separate the association of the broad concept of user experience from the experience with regard to design of User Interface (UI). They see UI as one aspect of several forms of interactions with a service.

In the following, we present some UX definitions from the literature:

- Hassenzahl and Tractinsky [47] defined UX as "a consequence of a user's internal state (predispositions, expectations, needs, motivation, mood, etc.), the characteristics of the designed system (e.g. complexity, purpose, usability, functionality, etc.), and the context (or the environment) within which the interaction occurs (e.g. organisational/social setting, meaningfulness of the activity, voluntariness of use, etc.)"
- Jakob and Don [46] defined UX as "all aspects of the end-user's interaction with the company, its services, and its products".
- Alben [48] defined UX as "all the aspects of how people use an interactive product: the way it feels in their hands, how well they understand how it works, how they feel about it while

they're using it, how well it serves their purposes, and how well it fits into the entire context in which they are using it".
- International Organization for Standardization [49] defined UX as "a person's perceptions and responses that result from the use and/or anticipated use of a product, system or service".

It is clear that all these definitions are concerned about the result of end-user interaction as a means of understanding user experience. The definition by Hassenzahl and Tractinsky [47] explicitly stated the variables that can impact the user experience of end-user interaction, whether it be the user's internal state, the system itself, or the environment where the interaction occurs.

4.2. Traditional Usability

Usability is an important attribute of software quality measured by a range of metrics and techniques to assess how easy a user interface is to use. Although usability has its academic origins in the HCI community, the term has no shared standard definition. Bevan [50] outlined that the term usability has been interpreted differentially by different people using different standards.

A few common definitions of usability are listed below:

- Shackel defined usability as "[a system's] capability in human functional terms to be used easily and effectively by the specified range of users, given specified training and support, to fulfil a specified range of tasks, within the specified range of environmental scenarios" [51].
- IEEE defined usability as "The ease with which a user can learn to operate, prepare inputs for, and interpret outputs of a system or component" [52].
- ISO/IEC 9126 defined usability as "A set of attributes that bear on the effort needed for use and on the individual assessment of such use, by a stated or implied set of users" [53].
- Preece's defined usability as "a measure of the ease with which a system can be learned or used, its safety, effectiveness and efficiency, and the attitude of its users towards it" [54].
- ISO 9241-11 defined usability as "The extent to which a product can be used by specified users to achieve specified goals with effectiveness, efficiency and satisfaction in a specified context of use" [55].

As it can be seen from the definitions above, usability is a combination of multiple attributes, for example, effectiveness, and efficiency. Different interpretations of usability as a term across academic and industry circles may have impacted the identification of standardised usability attributes in a consistent way over time.

In the previous sub-section, some definitions of user experience have included usability as an aspect of user experience. In the following sub-section, we illustrate differences between user experience and usability.

4.3. User Experience and Usability

In its published notes on user experience, International Organization for Standardization [49] has stated that "User experience includes all the users' emotions, beliefs, preferences, perceptions, physical and psychological responses, behaviours and accomplishments that occur before, during and after use". This note incorporates usability within user experience inexplicitly, whereby "behavior and accomplishments" can include two important usability attributes; efficiency (time to execute task) and effectiveness (completion of task). Therefore, user experience can be seen an umbrella for different concepts. This judgment is supported by different definitions of user experience, such as that given by Hassenzahl and Tractinsky [47], which defined user experience as a consequence of multiple factors including the characteristics of the designed system such as usability. The usability criteria can also be used as a metric to assess user experience [49, 56]. Based on the reviewed definitions of user experience and usability and what we have discussed on the overlapping meaning between the two terms in this section, we attempt to illustrate differences and relationships between user experience and usability, see Figure 10.

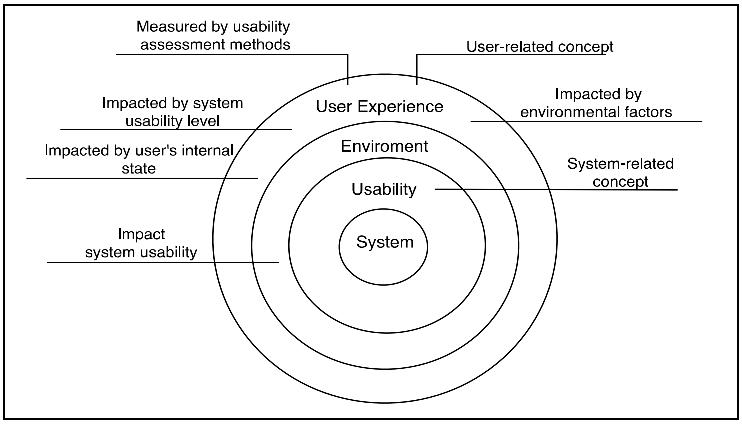

Figure 10: Differences and relationships between user experience and usability

In summary, user experience and usability can be conceptualized in different ways on basis of the following points:

- User experience is a broad term encompassing multiple factors including system usability [47].
- User experience is associated with user perception [47, 49], however, usability is more about the design of a system ISO [55].
- Usability can be impacted by environmental factors (including social and organisational factors) [51, 55]. While system usability can influence user experience negatively or positively [47], the impact of environmental factors on usability can lead to direct effect on user experience.
- Users' internal states, such as beliefs and exceptions, can impact how they use a system [57]. Thus, usability level as perceived by users can be impacted by their internal states. This can also lead to negative or positive user experience based on whether system conforms to user's mental state.
- Usability can be assessed with objective measures (e.g., time to execute task) and or subjective measures (e.g., satisfaction rate). User experience can be measured through usability assessment methods [49], based on subjective measures.

Having discussed traditional concepts of user experience and usability, we now turn to cross-platform usability, and user experience in the next two sections.

5. CROSS-PLATFORM USABILITY

These are some issue that we need to go through before defining cross-platform usability:

- Current usability metrics, for example, task execution time, focus on measuring usability of a single user interface. Thus, they need to be reconsidered for horizontal interaction across multiple user interfaces, for instance, execution time when attempting a task across platforms.

- Horizontal interaction involves using multiple user interfaces, in which every single user interface can be employed in a specific context of use. Factors that can affect usability in each context of use need to be considered when investigating usability across platforms.

After reviewing traditional definitions of usability from different standards and models, we identified characteristics of multiple user interfaces to arrive at the following definition of Cross-Platform Usability (CPU):

The extent to which a service across platforms can be used by specified users to achieve specified horizontal goals in specific or different contexts of use with acceptable level of several measurable factors including efficiency, effectiveness, and satisfaction.

6. Cross-Platform User Experience

There are multiple variables that can impact end-user perceptions when interacting with a service across platforms. These variables are listed below:

- The design of multiple user interfaces must be considered, particularly, given the fact that each user interface may have a unique design.
- Computing platforms used to interact with multiple user interfaces can have different characteristics. Computing platforms refers to a combination of hardware (e.g., office desktops, laptops, mobile phones, and tablets), operating systems (e.g., iOS, Windows, Mac OS), and computing capabilities. Each portion of this combination can have characteristics that can impact user experience across platforms. Some examples of these characteristics are:
    - Devices (e.g., input style, display size)
    - Operating systems (e.g., display, design, feature).
    - Computing capabilities (e.g., capabilities of processors, storage)

- Environments in which interactions with the multiple user interfaces occurs is also important.

We have adopted the traditional definition of user experience by Hassenzahl and Tractinsky [47] and modified to incorporate the variables and characteristics of cross-platform user interaction stated above. Thus, we define Cross-Platform User Experience (CPUX) as:

The consequence of a user's internal state (predispositions, expectations, needs, motivation, mood, etc.), the characteristics of the designed systems across platforms (e.g. service cohesion, composition, horizontal usability, distributed functionality, etc.), the characteristics of the computing platforms used to allow interactions with the systems (devices [display size, input style etc], operation systems [display, design, feature etc], computing capabilities [capabilities of processing, storage etc]) and the contexts (or the environments) within which the multiple interactions occurs (e.g. organisational/social setting, meaningfulness of the activity, voluntariness of use, user situation [seating, standing, driving] etc).

7. Conclusion

In this paper, we have provided a thorough discussion of different concepts that need to be considered in the context of cross-platform user experience. This includes concepts and practical approaches relevant to cross-platform service, and its relevant terms in the field of HCI. We have also investigated the definitions and characteristics of traditional usability and user experience. Then, we extended on these traditional concepts to develop a definition of cross-platform usability and user experience based on characteristics of user interaction across platforms. It is hoped that the definitions and discussions in this paper have contributed in building the necessary theoretical foundations for further study on cross-platform user experience evaluation.

REFERENCES

[1] K. Majrashi and M. Hamilton, "A Cross-Platform Usability Measurement Model," Lecture Notes on Software Engineering, vol. 3, 2015.
[2] M. Bell, Service-oriented modeling (SOA): Service analysis, design, and architecture: John Wiley & Sons, 2008.
[3] M. Rosen, B. Lublinsky, K. T. Smith, and M. J. Balcer, Applied SOA: service-oriented architecture and design strategies: John Wiley & Sons, 2008.
[4] The Linux Information Inc, "Cross-platform Definition," 2005.
[5] A. T. Manes, Web Services: A Manager's Guide: Addison-Wesley Longman Publishing Co., Inc., 2003.
[6] Q. Yu and A. Bouguettaya, Foundations for Efficient Web Service Selection: Springer, 2009.
[7] N. Ide and J. Pustejovsky, "What does interoperability mean, anyway? Toward an operational definition of interoperability for language technology," in Proceedings of the Second International Conference on Global Interoperability for Language Resources. Hong Kong, China, 2010.
[8] G. Alonso and F. Casati, "Web services and service-oriented architectures," in Data Engineering, 2005. ICDE 2005. Proceedings. 21st International Conference on, 2005, p. 1147.
[9] S. K. Feiner, "Environment management for hybrid user interfaces," Personal Communications, IEEE, vol. 7, pp. 50-53, 2000.
[10] M. Wäljas, K. Segerståhl, K. Väänänen-Vainio-Mattila, and H. Oinas-Kukkonen, "Cross-platform service user experience: a field study and an initial framework," in Proceedings of the 12th international conference on Human computer interaction with mobile devices and services, 2010, pp. 219-228.
[11] M. A. Schilling, "Toward a general modular systems theory and its application to interfirm product modularity," Academy of management review, vol. 25, pp. 312-334, 2000.
[12] H. A. Simon, The architecture of complexity: Springer, 1991.
[13] C. Denis and L. Karsenty, "Inter-usability of multi-device systems: A conceptual framework," Multiple user interfaces: Cross-platform applications and context-aware interfaces, pp. 373-384, 2004.
[14] H. Wilson, R. Street, and L. Bruce, The multichannel challenge: integrating customer experiences for profit: Routledge, 2008.
[15] F. G. Kazasis, N. Moumoutzis, N. Pappas, A. Karanastasi, and S. Christodoulakis, "Designing Ubiquitous Personalized TV-Anytime Services," in CAiSE Workshops, 2003.
[16] P. Fraternali, A. Bozzon, M. Brambilla, V. Croce, K. Hammervold, E. Moore, et al., "Model-driven development of personalized, multichannel interfaces for audiovisual search: the PHAROS approach," NEM Summit, Saint Malo, France, 2009.
[17] C. Wiberg, K. Jegers, and J. Bodén, "Cross media interaction design," 2007.
[18] L. V. L. Filgueiras, D. O. Correa, J. S. O. Neto, and R. P. Facis, "X-gov planning: how to apply cross media to government services," in Digital Society, 2008 Second International Conference on the, 2008, pp. 140-145.
[19] J. Boumans, "Cross-media E-Content Report 8," Published in a series of E-Content Reports by ACTeN (http://www. acten. net), 2004.
[20] K. Segerståhl, "Crossmedia systems constructed around human activities: a field study and implications for design," in Human-Computer Interaction–INTERACT 2009, ed: Springer, 2009, pp. 354-367.
[21] K. A. Bharat and L. Cardelli, "Migratory applications," in Proceedings of the 8th annual ACM symposium on User interface and software technology, 1995, pp. 132-142.
[22] D. Thevenin and J. Coutaz, "Plasticity of user interfaces: Framework and research agenda," in Proceedings of INTERACT, 1999, pp. 110-117.
[23] B. Johanson, G. Hutchins, T. Winograd, and M. Stone, "PointRight: experience with flexible input redirection in interactive workspaces," in Proceedings of the 15th annual ACM symposium on User interface software and technology, 2002, pp. 227-234.

[24] M. A. Nacenta, D. Aliakseyeu, S. Subramanian, and C. Gutwin, "A comparison of techniques for multi-display reaching," in Proceedings of the SIGCHI conference on Human factors in computing systems, 2005, pp. 371-380.

[25] J. T. Biehl, W. T. Baker, B. P. Bailey, D. S. Tan, K. M. Inkpen, and M. Czerwinski, "Impromptu: a new interaction framework for supporting collaboration in multiple display environments and its field evaluation for co-located software development," in Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, 2008, pp. 939-948.

[26] J. R. Wallace, R. L. Mandryk, and K. M. Inkpen, "Comparing content and input redirection in MDEs," in Proceedings of the 2008 ACM conference on Computer supported cooperative work, 2008, pp. 157-166.

[27]K. Luyten and K. Coninx, "Distributed user interface elements to support smart interaction spaces," in Multimedia, Seventh IEEE International Symposium on, 2005, p. 8 pp.

[28] M. Bång, A. Larsson, E. Berglund, and H. Eriksson, "Distributed user interfaces for clinical ubiquitous computing applications," in International Journal of Medical Informatics vol. 74, ed, 2005, pp. 545-551.

[29] K. Segerståhl and H. Oinas-Kukkonen, "Distributed user experience in persuasive technology environments," in Persuasive Technology, ed: Springer, 2007, pp. 80-91.

[30] A. a. F. Seffah, Peter, "Workshop on multiples user interfaces over the Internet: engineering and applications trends," In: HM-HCI: French/British Conference on Human Computer Interaction, Lille, France, 2001.

[31]J. Vanderdonckt, Q. Limbourg, M. Florins, F. Oger, and B. Macq, "Synchronised, model-based design of multiple user interfaces," in Proc. 2001 Workshop on Multiple User Interfaces over the Internet, 2001.

[32]J. McGrenere, R. M. Baecker, and K. S. Booth, "An evaluation of a multiple interface design solution for bloated software," in Proceedings of the SIGCHI conference on Human factors in computing systems, 2002, pp. 164-170.

[33] A. Seffah and H. Javahery, Multiple user interfaces: cross-platform applications and context-aware interfaces: John Wiley & Sons, 2005.

[34] A. Seffah, P. Forbrig, and H. Javahery, "Multi-devices “Multiple” user interfaces: development models and research opportunities," in Journal of Systems and Software vol. 73, ed, 2004, pp. 287-300.

[35] J. Forlizzi and K. Battarbee, "Understanding experience in interactive systems," in Proceedings of the 5th conference on Designing interactive systems: processes, practices, methods, and techniques, ed, 2004, pp. 261-268.

[36] W. Albert and T. Tullis, Measuring the user experience: collecting, analyzing, and presenting usability metrics: Newnes, 2013.

[37] N. Bevan, "Classifying and selecting UX and usability measures," in International Workshop on Meaningful Measures: Valid Useful User Experience Measurement, 2008, pp. 13-18.

[38] R. W. Picard, "Affective Computing for HCI," in HCI (1), 1999, pp. 829-833.

[39] F. N. Egger, "Affective design of e-commerce user interfaces: How to maximise perceived trustworthiness," in Proc. Intl. Conf. Affective Human Factors Design, 2001, pp. 317-324.

[40] D. A. Norman, Emotional design: Why we love (or hate) everyday things: Basic books, 2004.

[41] H. M. Khalid and M. G. Helander, "Customer emotional needs in product design," Concurrent Engineering, vol. 14, pp. 197-206, 2006.

[42] S. Baurley, "Interactive and experiential design in smart textile products and applications," Personal and Ubiquitous Computing, vol. 8, pp. 274-281, 2004.

[43] P. M. Tsang and S. Tse, "A hedonic model for effective web marketing: an empirical examination," Industrial Management & Data Systems, vol. 105, pp. 1039-1052, 2005.

[44] M. G. Helander, "Hedonomics-affective human factors design," in Proceedings of the Human Factors and Ergonomics Society Annual Meeting, 2002, pp. 978-982.

[45] T. Lavie and N. Tractinsky, "Assessing dimensions of perceived visual aesthetics of web sites," International journal of human-computer studies, vol. 60, pp. 269-298, 2004.

[46] N. Jakob and N. Don, "The Definition of User Experience."

[47] M. Hassenzahl and N. Tractinsky, "User experience-a research agenda," Behaviour & Information Technology, vol. 25, pp. 91-97, 2006.

[48] L. Alben, "Quality of Experience, Interactions," 1996.

[49] International Organization for Standardization, Ergonomics of Human-system Interaction: Part 210: Human-centred Design for Interactive Systems: ISO, 2010.

[50] N. Bevan, "Measuring usability as quality of use," Software Quality Journal, vol. 4, pp. 115-130, 1995.

[51]B. Shackel, "The concept of usability," Visual display terminals: usability issues and health concerns, pp. 45-87, 1984.

[52] J. Radatz, A. Geraci, and F. Katki, "IEEE standard glossary of software engineering terminology," IEEE Std, vol. 610121990, p. 121990, 1990.

[53] ISO/IEC 9126, "Information Technology, Software Product Evaluation, Quality Characteristics and Guidelines for their Use," Geneva, Switzerland: International Organization for Standardization., 1991.

[54] J. Preece, Y. Rogers, H. Sharp, D. Benyon, S. Holland, and T. Carey, Human-computer interaction: Addison-Wesley Longman Ltd., 1994.

[55] ISO 9241-11, "Ergonomic Requirements for Office Work with Visual Display Terminals (VDTs), Part 11: Guidance on Usability," Geneva, Switzerland: International Organization for Standardization., 1998.

[56] K. Majrashi and M. Hamilton, User Experience of University Websites: LAP Lambert Academic Publishing, 2014.

[57] N. Jakob and N. Don, "Mental Models," 2010.

AUTHORS

Khalid Majrashi was born in Jazan, Saudi Arabia, in 1987. He received the B.E degree in education of computer science from the University of Jazan, Jazan Saudi Arabia, in 2008, and the Master degree in computer science from RMIT University, Melbourne, Australia in 2012, and currently is PhD candidate (in the field of computer science) at RMIT University, Melbourne, Australia. His current research interests include software quality, user experience, usability, mobile computing, cloud computing (software as a service), and user interface models.

Margaret Hamilton, BSc (Hons), PhD, University of Wollongong, Australia, is an Associate Professor in the School of Computer Science and Information Technology at RMIT University in Melbourne Australia, where she researches in the field of human computer interaction. She works with new technologies in computer science education and mobility to investigate the human aspects and to research how computer interfaces can be adapted to be more user-friendly, and computer programs to be more relevant, realtime and personalised.

Alexandra L. Uitdenbogerd, BSc, University of Western Australia, Grad. Dip. Ed, Melbourne University, Grad Cert IT, and PhD, RMIT University, Australia, is a senior lecturer in the School of Computer Science and Information Technology at RMIT University in Melbourne Australia. Her current research interests include search engine technology, music/audio applications, information retrieval, computer-assisted language learning, pattern matching, search engines and related areas.