



## Adaptive Graphical User Interface for Web Applications Using Aspect Oriented Component Engineering

Atif Zaman<sup>‡</sup>, Mudassar Ahmad<sup>£</sup>, Shafique Ahmad<sup>‡</sup>, Tasleem Mustafa<sup>‡</sup>

<sup>‡</sup>Department of Computer Science, University of Agriculture, Faisalabad, Pakistan.

<sup>£</sup>Department of Komputing, Universiti Teknologi Malaysia, Johor Bahru, Malaysia.

atifzaman@live.com, mudassar.utm@gmail.com, mr.shafique@live.com, tasleem@uaf.edu.pk

### ABSTRACT

Graphical User Interface (GUI) is considered to be an essential part in any web applications development. Aspect-Oriented Component Engineering (AOCE) is new approach for developing more and higher quality reusable and adaptable software or web applications components. AOCE uses the idea of providing and requiring services. Adaptable user interface for AOCE based development has not yet been considered to web applications. Simple and easy user interface facilitate users by which application or web interface can be operated effectively. The purpose of this study is to discuss popular user interfaces and suggestions for adaptable GUI designing. The example of adaptation at different levels includes architecture, presentation, extension and composition. Furthermore the AOCE common systemic aspects are discussed for web interfaces adaptivity.

### Indexing terms/Keywords

Graphical User Interfaces, Adaptive User Interface, Human Computer Interaction, Aspect Oriented Component Engineering.

### Academic Discipline And Sub-Disciplines

Web Applications, Software Engineering, User Interfaces

### SUBJECT CLASSIFICATION

Computer Science Subject Classification;

### TYPE (METHOD/APPROACH)

Qualitative Research; Review

---

# Council for Innovative Research

Peer Review Research Publishing System

**Journal:** INTERNATIONAL JOURNAL OF COMPUTERS & TECHNOLOGY

Vol 10, No 2

[editor@cirworld.com](mailto:editor@cirworld.com)

[www.cirworld.com](http://www.cirworld.com), [member.cirworld.com](http://member.cirworld.com)



## INTRODUCTION

Human Computer Interaction (HCI) is playing key role in the designing of novel information, interaction and communication via technology [1]. HCI is difficult task to accomplish. Display system may have different characteristics such as architecture, functionality, size and how the internal workings are performed. User may have different behavior of viewing different aspects of display. There is need of improving HCI design so the user can better understand the interface of an application [2]. Graphical User Interface (GUI) present information in simplified manner on particular machine, electronic devices or computer which make it easier for user to operate. The visual icons such as buttons, menus and other widgets speed up the process of learning. It makes lot easier to provide input to computer. A poorly developed GUI can obscure functionality. If GUI is not organized in clear and understandable manner and if the contents are not arranged properly then user will have to spend significant time to learn how to operate program before user can use it properly. An efficient GUI is the one that does not require user to remember steps need to be performed [3]. Generally the internet user does not have technical knowledge about things to understand and if the GUI is not user friendly then it will make difficult to use particular application efficiently.

The internet has been emerged as a stable application platform. Those days has gone when internet was used to present static information only. Today World Wide Web [4], [5], [6] has ability to provide highly interactive applications and services. Transformation of application development and deployment is growing from desktop to web. Over the internet, the applications are provided as "Software as a Service (SaaS)" [7], [8], [9]. Due to SaaS over internet, the access has become independent of time and physical location [10]. Adaptive user interface [2], [11], [12], [13] is the system which can change aspect details with respect to architecture, functionality or information presentation to suit the requirements of different users or group of users. The adaptation approach will assist the user to exploit the information in any of the context to use. Without the adaptive interface the information cannot be presented in simple and in easy way for users. The most important feature of adaption is to organize the graphical interface to present information in best possible way with respect to the behavior of users. The adaptation achieves the goal of highlight relevant information, organize usage of space and display best representation of information [14].

Component-based systems development is the composition of systems from parts, called software components. Components encapsulate data and functions. They often provide events, are self-describing, and many can be dynamically plugged and played into running applications [15]. Aspect Oriented Component Engineering (AOCE) is a new approach for developing component based applications. AOCE uses the concept of characterizing systemic aspect of components properties, it helps to improve design and implementation [16]. Aspects of components take horizontal slicing across systemic features of application which includes user interface, distribution, persistency and security. AOCE consider the required and provided capabilities of components [17]. In components based development software or web application are developed from existing components, which are developed separately with other applications. Components aspects are separated from other components in AOCE to provide or require services from other components. Reusing other application's components over web application mostly create problem of displaying inappropriate information interface because components have no knowledge of end user graphical interface of other component. In addition part of web application components itself have no knowledge of other components.

In this paper, the approach is discussed at architecture and presentation level which will allow extension and composition of component's aspects. This could possibly aid other components to determine graphical user interface with other components.

## RELATED WORK

There are many solutions exists for designing web interfaces. These approaches are language based, grammar based and widget based. Web application based user interfaces techniques are discussed in the past which focus on constructing web interfaces for application processing, to manipulate data and information efficiently. The proposed approach was set of type-oriented building blocks [18] for constructing complex type of interfaces easily. This approach focuses on complexity of user interfaces but this approach is acceptable in case of complex components based development. The distributed web application's are coupled of different components and ran on different platforms. Thus the reduction of web interface complexity is most important in case of components based development. In [19], discusses the approach for the development of distributed components at architecture level. Iterative and incremental development [20] process was used for the development purpose. The techniques of distributed components implementation have no adaptability of user interfaces in distributed environments.

In [21], the work focuses on adaptable interface for components systems. The components adaptability was achieved through the extension, composition and reconfiguration of components in application's interface aspects. This work is good example of adaptability for user interface aspects. But this work has no sufficient details about end user adaptability over web based application. In addition the adaptability has not been considered for the other aspects of AOCE such as distribution, persistency and security in web application environment. The adaptive graphical user interface has gained lot of attention over web applications and extensive work has been done in the past and some frameworks have been developed which allows dynamically building of interface. All the work belongs to the development of simple or components based application. But adaptation for AOCE for web application has not yet been considered. Adaption of aspect-oriented components can achieve better implementation of application and information system over web environment.



## TYPES OF COMMON USER INTERFACES

Emerging developments in HCI technologies are narrowing down the gap of human and machine interaction with the help of human machine interfaces [22]. Due to the increasing change in the development of technology during the last decade. Such as internet, pervasive technology, laptops, smart phones, wireless communication and track devices. These techs have created lots of possibilities for encouraging, extending and growing user experiences, communication and interactions. As an ultimate result the interfaces are becoming ubiquitous. Computer based interactions has taken place over many different types of surfaces at various levels [23]. The main concept behind HCI is usability and modality for user interaction with machine. HCI techniques are based on graphic displaying, keyboard and mouse but recently there is more advancement in HCI techniques such as vision, sound, speech recognition and projective displays [24].

Currently the popular types of user interfaces are:

- I. Command Line Interface (CLI): It is old interface used for interacting computer programs. CLI accepts commands as an input by typing with keyboards and perform appropriate operating function. CLI is less widely used for normal computer users but it is favored by most advance users, as it provide the powerful means to control operating system or program.
- II. Graphical User Interface (GUI): It is a human-computer interface which uses windows, menus and icons. It is manipulated with mouse and keyboard. GUI allows user to take advantage of multitasking environment. GUI has become standard in human-computer interaction [25].
- III. Web User Interface (WUI): It accepts input and gives output by creating web pages which are accessed over internet by using web browser [26] programs. Latest technologies such as Java, JavaScript, AJAX, Adobe Flex and Microsoft .NET provide real time access to a program without refreshing bowers as in HTML based. In WUI the administrative interface for web servers, servers and networked computers are called control panels [27].
- IV. Touch User Interface (TUI): It is pointing technology which is based on sense of touch (haptics), while GUI relies on the sense of sight. TUI provides a touchscreen-based display which provides input and output is generated in response. TUI based interface provide reduction in time whatever user want to access in digital world [28].
- V. Attentive User Interface (AUI): It is the interface that is sensitive towards the attention of user is called Attentive User Interface. AUI uses a range of novel input, interpretation and output techniques to implement. AUI utilize extra channels for input which measures the characteristics of user attention [29].
- VI. Gesture Interface: It is a graphical user interface that takes input in the form of hand or mouse gestures. The gestures are sketched with computer mouse or a stylus [27].
- VII. Intelligent User Interface (IUI): IUIs are human machine interfaces. Its prime aim is to improve the effectiveness, efficiency and naturalness of human-machine interaction by reasoning, representing and acting on the models of the user, task, domain and media. The next generation interfaces often called intelligent. IUI provide many additional advantages to the users, it includes adaptivity, context sensitive and task assistance [30].
- VIII. Multi Screen Interface: Integrate multiple screens for display. It shows both screens separately where mouse can move seamlessly from one screen to other. It also allows to split windows across screens. It shows same as they are formed of large single display interface [31].
- IX. Object-Oriented User Interface (OOUI): It is user interface type which is based on object oriented programming metaphor. OOUI allows the user to interact explicitly with objects which represents entities in the domain about which application is concerned. Most of the vectors drawing applications have OOUI. The examples of objects are lines, circle and canvases. OOUI facilitate the user to select object explicitly or change its properties such as color and size or invoking of other actions. If OOUI is used in any business application, the user may explicitly select or invoke actions on objects which are representing business domain entries. Examples of business domain entities are customers, products or orders [32].
- X. Text User Interface (TUI): The concept of TUI was coined after the graphical user interface was invented. The core concept of TUI was the way of output that does not match with command line interface. It accepts different form of input unlike CLI [33].
- XI. Voice User Interface (VUI): A human interaction when communicating with a spoken language application. The VUI elements includes prompts, grammar and dialog logic. The inputs are made by speaking or responding verbally [34].
- XII. Natural Languages Interface: It is useful for querying the search engines or on web-pages. As input user types a question and for output wait for the response [27].

- XIII. Zooming User Interface (ZUI): It is graphical environment, in which the scale of the viewed area can be changed to see more or less details and browse through different documents. ZUI is a type of Graphical User Interface (GUI) [35].

## Aspect-Oriented Component Engineering

When developing application from components, developer builds components that have been identified and build by functional decomposition. In components based the data of the system and functions are organized in vertical piece(s) to support system functionality. In such applications many systemic features are end up with cross-cutting concern or they impact on many other components in system [36].

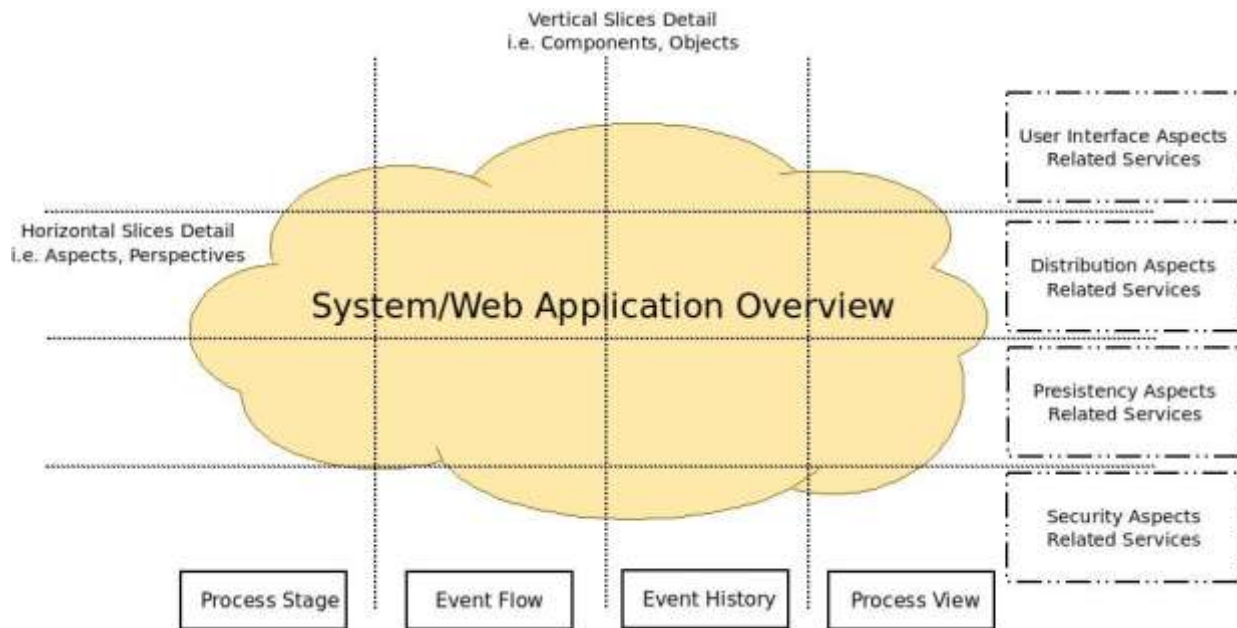


Figure. 1: Components Concepts vs. Aspects Components

Figure. 1 shows the general conceptual difference between the software components know as vertical slices of data and functionality in software system and components aspects which are horizontal slices are known as functional and non functional constraints in the system. Normally in a system one or more aspect services are provided by each component to other components for use and from other components, it also require aspect services which may be required by one or more to help to perform functionality of components [37]. The examples of such things are user interfaces, distribution of data, data persistency and security management. They all have impacts on most of the components, even upon the methods and state. Consequently the components provide functionality belonging to these features. And some other requires it from other components for operating [36]. The term components aspects<sup>2</sup> used to describe horizontally-impacting perspectives on software components in application or system [37], [17].

Figure. 2 [17] illustrates the characterization of tree editor components capabilities using aspects. User interface aspects and configuration property sheets are provided, while it may require viewer. On the other hand in the distribution aspects, it can require event transport and synchronization support and may provide acting capabilities and event generation. In the persistency aspects it may provide serialization of data and deserialization of data. But it can require the capabilities of data storage and retrieval. For security aspect purpose, it may provide service of item locking and highlighting but in response it can require collaborative event generation between users.

Aspect-Oriented Component Engineering (AOCE) used to help software engineer better software components development [38], [16]. Component aspects have broad annotations of categories which are used to describe systemic properties of system that the components provide functions for or they require functions from the other components. Components aspects examples include (which are referred as aspects) user interface, distribution, persistency, security and many other [36].

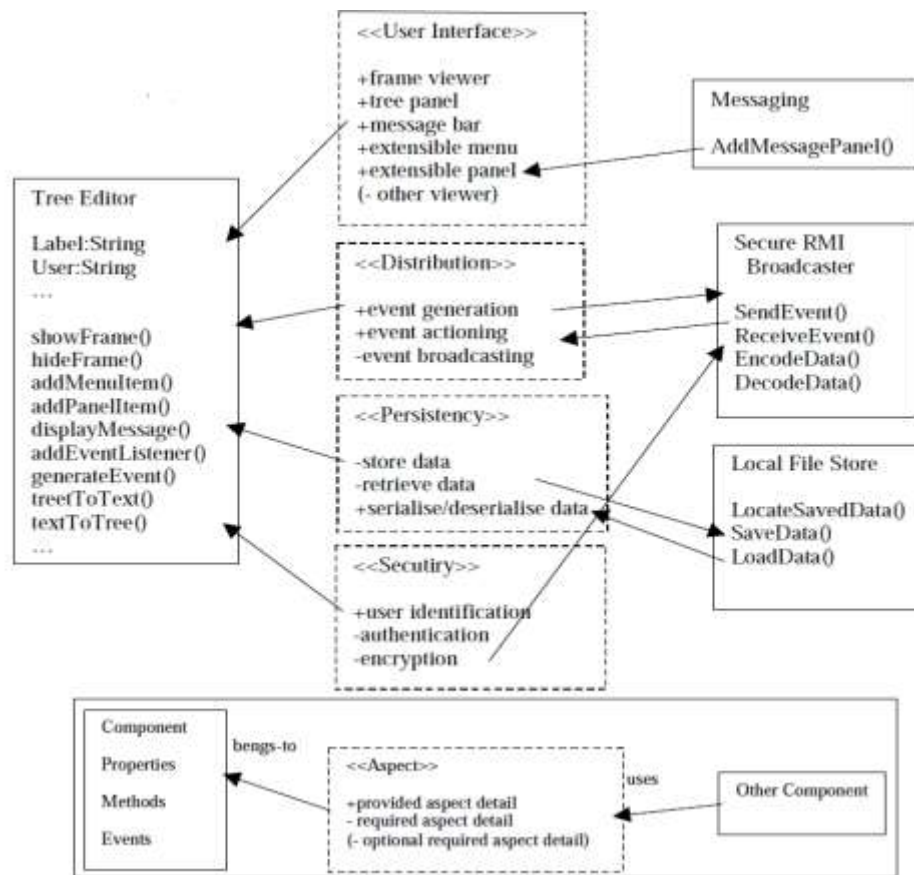


Figure. 2: Visual Representation of aspects and aspects details.

Table 1: [37] some general aspects and aspects details.

Aspect	Details of Aspect	Aspect Description
User Interface	Views Affordances Feedback Extensible Parts	Aspect components support several components by providing or requiring, not only user interface but extensible and composable interfaces.
Distribution	Object Identification Method Invocation Transactions Even Propagation Concurrency Control	Distribution aspect components support the requiring or providing of distributed object management facilities.
Persistency	Locate Data Store/Retrieve Data Lock Data	Aspect components support to provide or require the facilities of persistency management.
Security	Encoding Model Key Distribution Authentication Access Control	This components aspect supports the facilities of providing or requiring inter components security models.

## ADAPTIVE GRAPHICAL USER INTERFACE DESIGNING

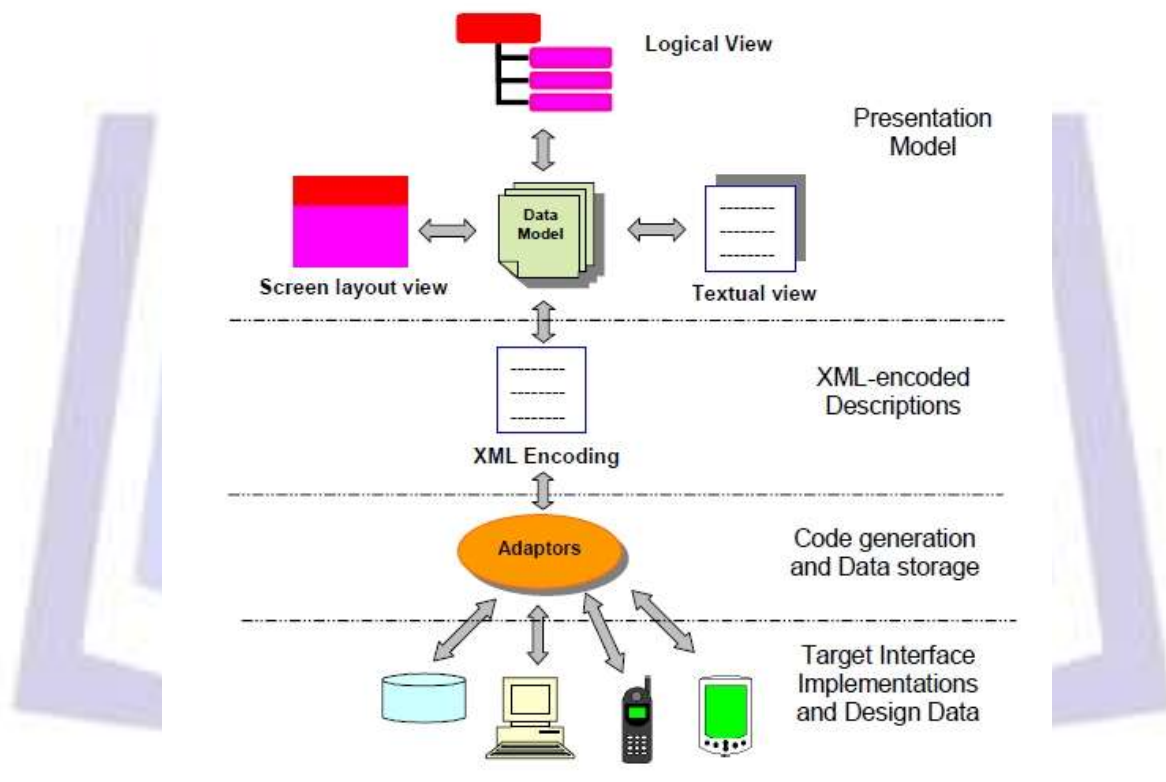
The important and end purpose of adaptation is to present easy and efficient user interfaces. To properly present effective interfaces to millions of user is not an easy job because it is very difficult to write such code for an application. User uses different parts of the system that are related to them but don't learn the whole system. The conman and basic knowledge is shared by the user about the functionality of the system. By the use of adaptive system the use time is changed to

different kind of design time [39]. "An adaptive user interface is to select more suitable interface for user which improves its skills and ability of interaction having partial experience of a user."

Adaptive systems are capable to adjusting the user interface by the preference, knowledge and skills of user. In the adaptable systems user can control or make changes for adjustment. Implicit model of users and tasks are necessary for achieving the adaptivity. For the designing of adaptive graphical user interface it demands for an interface that can be adapted. It is important to design adaptation strategy to take place it [40]. In this section the proposed techniques are described for designing adaptable GUI:

### Adaptive UI Architecture

The adaptive architecture for AOCE user interfaces components comprises of sub components aspect. This adaptation is based on two approaches. The aspects for user interfaces should be developed according to the AOCE horizontal tasks and style aware manner. It means aspects components design knowledge and components alternatives of tasks decomposition both vertical and horizontal aspects. All the AOCE based components must be clearly presented in actual user interfaces. A decision knowledge mechanism should be created, it will help in administrating, maintaining and evaluating the rules of adaption. The decision knowledge mechanism should be responsible for constricting interfaces. And it should be able to propagate adaptation decision at architecture level by generating adaptation triggers.



**Figure. 3: Adaptive Graphical User Interface Design Architecture**

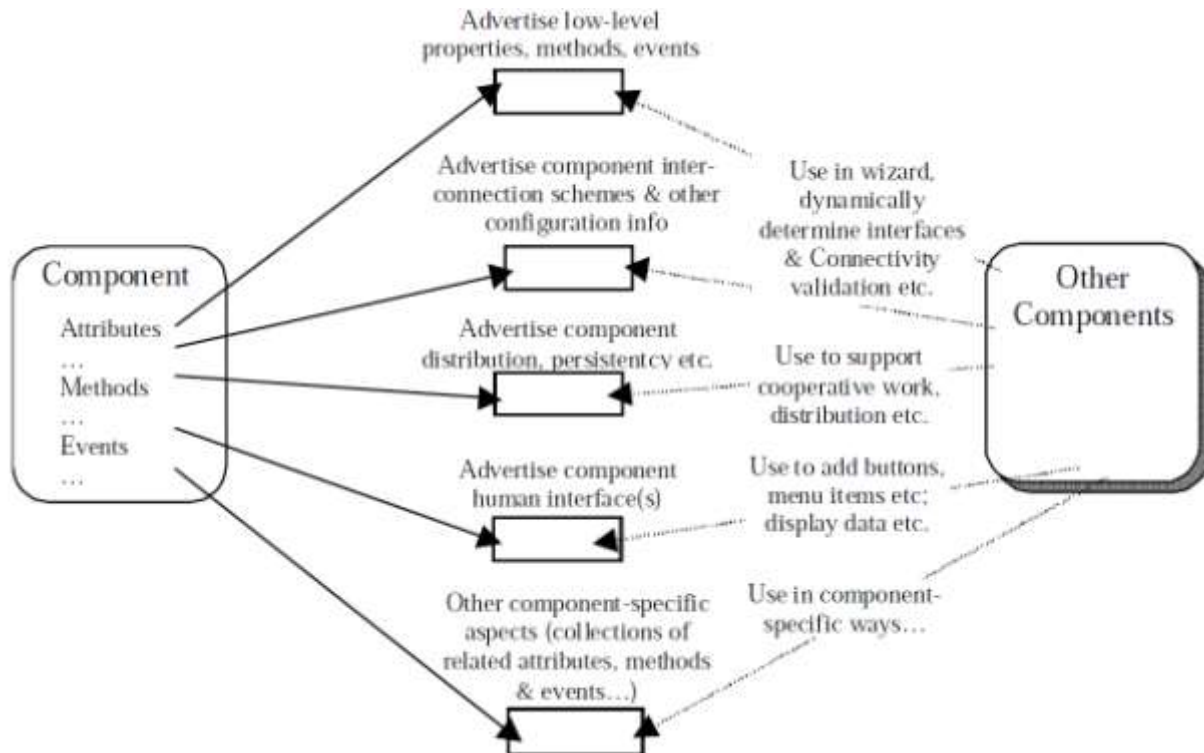
Figure. 3 [41] shows the architecture of prototype design, which will be used to fulfill for the designing of adaptive GUI. With this specification user and designers may be able to view and edit the user interface design. This architecture provides the possibility for the implementation of wide range of interfaces.

### Adaptive UI Presentation

The presentation of adaptive UI should be based on the two approaches. The split interface mechanism should be implemented for components aspects. The horizontal aspects interfaces should split into horizontal manner with respect to vertical components. The user should be able to choose using original interfaces or reconfiguration adaptive interfaces. If more components are added then functionality should be created to provide space for new components by adjusting other components. The moving interface concept can be applied to the components. The idea is to add functionality to move components into pop-up panes, menus or tool-bar by allowing it to configure it in acceptable manner. And by moving components, space should be available for more components.

## Adaptive UI Extension and Composition

The aspect components should be able to share interfaces. It can be achieved by expendable affordance. Where one component allows extending other components. The expendable components should provide access to their own functionality and data from web server with other components interface. Extension avoids in presenting multiple composed components aspects and inconsistency to end user for internet based web application. A mechanism should be implemented for the composition of components over web browser. The composition can help in accessing and modifying components for users. The composition of components should be in composite panel rather than using individual panel, which may save extra load for running web application from web server and space.



**Figure. 4: Implementation of Aspects**

Figure. 4 [38] shows how a component aspects are supported by publicizing its information. It also describes how they can be used. For example there should be a set of some methods which should allow the extension and modification of user interface aspects. So it could facilitate the other components to query it for required aspects from them and can utilize them appropriately upon need. Figure. 4 is implementation of aspects for JViews [16], [37], [17], [42], [43] components. It help components developer and users to access aspects and can verify if all the components are properly linked and are communicating appropriately by the use of aspect information [38].

## CONCLUSION AND FUTURE WORK

The purpose of this research is to explain an approach for better development of adaptable graphical user interface (GUI). In this paper the new GUI design approach discussed for adaptive interface. This adaptation has been achieved at architecture and presentation level. The approach for extension and composition of AOCE based components also disused in distributed environment. At high level the properties of components use elements, including provided and required elements are specified. This adaptive approach is applicable for both software and web based application. For future work our aim is to implement agent-based intelligent techniques for adaptable user interface over web applications.

## REFERENCES

- [1]. D. Fallman, "Design-oriented human-computer interaction," in Proceedings of the SIGCHI conference on Human factors in computing systems. ACM, 2003, pp. 225–232.
- [2]. D. Benyon, "Accommodating individual differences through an adaptive user interface," HUMAN FACTORS IN INFORMATION TECHNOLOGY, vol. 10, pp. 149–149, 1993.
- [3]. B. H. Toby, "EXPGUI, a graphical user interface for GSAS," Journal of Applied Crystallography, vol. 34, no. 2, pp. 210–213, 2001.



- [4]. E. Al-Masri and Q. H. Mahmoud, "Investigating web services on the world wide web," in Proceedings of the 17th international conference on World Wide Web. ACM, 2008, pp. 795–804.
- [5]. P. Morville and L. Rosenfeld, Information architecture for the World Wide Web: Designing large-scale Web sites. O'Reilly Media, 2008.
- [6]. L. Rosenfeld and P. Morville, Information architecture for the world wide web. O'Reilly Media, Inc., 2002.
- [7]. M. Godse and S. Mulik, "An approach for selecting software-as-a-service (SaaS) product," in Cloud Computing, 2009. CLOUD'09. IEEE International Conference on. IEEE, 2009, pp. 155–158.
- [8]. S. Aulbach, D. Jacobs, A. Kemper, and M. Seibold, "A comparison of flexible schemas for software as a service," in Proceedings of the 35th SIGMOD international conference on Management of data. ACM, 2009, pp. 881–888.
- [9]. A. Elfatry and P. Layzell, "Software as a service: a negotiation perspective," in Computer Software and Applications Conference, 2002. COMPSAC 2002. Proceedings. 26th Annual International. IEEE, 2002, pp. 501–506.
- [10]. S. Pietschmann, M. Voigt, and K. Meissner, "Dynamic composition of service-oriented web user interfaces," in Internet and Web Applications and Services, 2009. ICIW'09. Fourth International Conference on. IEEE, 2009, pp. 217–222.
- [11]. J. Liu, C. K. Wong, and K. K. Hui, "An adaptive user interface based on personalized learning," Intelligent Systems, IEEE, vol. 18, no. 2, pp. 52–57, 2003.
- [12]. J. He and I.-L. Yen, "Adaptive user interface generation for web services," in e-Business Engineering, 2007. ICEBE 2007. IEEE International Conference on. IEEE, 2007, pp. 536–539.
- [13]. Others, "ADAPTIVE USER INTERFACE," Apr. 22 2010, uS Patent 20,100,097,331.
- [14]. G. Viano, A. Parodi, J. Alty, C. Khalil, I. Angulo, D. Biglino, M. Crampes, C. Vaudry, V. Daurensan, and P. Lachaud, "Adaptive user interface for process control based on multi-agent approach," in Proceedings of the working conference on Advanced visual interfaces. ACM, 2000, pp. 201–204.
- [15]. J. Grundy and J. Hosking, "Developing software components with aspects: Some issues and experiences," Aspect-Oriented Software Development, pp. 585–604.
- [16]. J. Grundy, "Supporting aspect-oriented component-based systems engineering," in Proceedings of 11th International Conference on Software Engineering and Knowledge Engineering, 1999, pp. 16–19.
- [17]. J. C. Grundy, "An implementation architecture for aspect-oriented component engineering," in Proceedings of the 5th International Conference on Parallel and Distributed Processing Techniques and Applications: Special Session on Aspect-oriented Programming, Las Vegas. Citeseer, 2000.
- [18]. M. Hanus, "Type-oriented construction of web user interfaces," in Proceedings of the 8th ACM SIGPLAN international conference on Principles and practice of declarative programming. ACM, 2006, pp. 27–38.
- [19]. W. Emmerich, "Distributed component technologies and their software engineering implications," in Proceedings of the 24th international conference on Software engineering. ACM, 2002, pp. 537–546.
- [20]. C. Larman and V. R. Basili, "Iterative and incremental developments. a brief history," Computer, vol. 36, no. 6, pp. 47–56, 2003.
- [21]. J. Grundy and J. Hosking, "Developing adaptable user interfaces for component-based systems," Interacting with computers, vol. 14, no. 3, pp. 175–194, 2002.
- [22]. E. Hudlicka, "To feel or not to feel: The role of affect in human–computer interaction," International Journal of Human-Computer Studies, vol. 59, no. 1, pp. 1–32, 2003.
- [23]. Y. Rogers, "New theoretical approaches for human-computer interaction," Annual review of information science and technology, vol. 38, no. 1, pp. 87–143, 2004.
- [24]. C. Von Hardenberg and F. B'erard, "Bare-hand human-computer interaction," in Proceedings of the 2001 workshop on Perceptive user interfaces. ACM, 2001, pp. 1–8.
- [25]. T. L. I. Project. (2004, Oct.) GUI Definition. [Online]. Available: <http://www.linfo.org/gui.html>
- [26]. R. M. Rogers and K. C. Lagarde, "Web browser system," Aug. 11 1998, uS Patent 5,793,964.
- [27]. T. F. E. Wikipedia. (2013, May) User Interface. [Online]. Available: [https://en.wikipedia.org/wiki/User\\_interface](https://en.wikipedia.org/wiki/User_interface)
- [28]. Wikipedia. (2013, May) Touch user interface. [Online]. Available: [http://en.wikipedia.org/wiki/Touch\\_user\\_interface](http://en.wikipedia.org/wiki/Touch_user_interface)
- [29]. R. Vertegaal, "Attentive user interfaces," Communications of the ACM, vol. 46, no. 3, pp. 30–33, 2003.
- [30]. M. T. Maybury and W. Wahlster, Readings in intelligent user interfaces. Morgan Kaufmann, 1998.
- [31]. K. S. Booth, B. D. Fisher, C. J. R. Lin, and R. Argue, "The mighty mouse multi-screen collaboration tool," in Proceedings of the 15th annual ACM symposium on User interface software and technology. ACM, 2002, pp. 209–212.
- [32]. T. F. E. Wikipedia. (2013, May) Object-oriented user interface. [Online]. Available: [http://en.wikipedia.org/wiki/Object-oriented\\_user\\_interface](http://en.wikipedia.org/wiki/Object-oriented_user_interface)
- [33]. Wikipedia. (2013, May) Text-based user interface. [Online]. Available: [http://en.wikipedia.org/wiki/Text-based\\_user\\_interface](http://en.wikipedia.org/wiki/Text-based_user_interface)
- [34]. M. H. Cohen, M. H. Cohen, J. P. Giangola, and J. Balogh, Voice user interface design. Addison-Wesley, 2004.



- [35]. T. F. E. Wikipedia. (2013, May) Zooming user interface. [Online]. Available: [http://en.wikipedia.org/wiki/Zooming\\_user\\_interface](http://en.wikipedia.org/wiki/Zooming_user_interface)
- [36]. J. Grundy, T. Panas, S. Singh, and H. Stöckle, "An approach to developing web services with aspect-oriented component engineering," in Proceedings of the 2nd Nordic Conference on Web Services, 2003.
- [37]. J. Grundy, "Multi-perspective specification, design and implementation of software components using aspects," International Journal of Software Engineering and Knowledge Engineering, vol. 10, no. 06, pp. 713–734, 2000.
- [38]. J. Grundy, R. Mugridge, J. Hosking, and M. Apperley, "Tool integration, collaboration and user interaction issues in component-based software architectures," in Technology of Object-Oriented Languages, 1998. TOOLS 28. Proceedings. IEEE, 1998, pp. 299–312.
- [39]. G. Fischer, "User modeling in human–computer interaction," User modeling and user-adapted interaction, vol. 11, no. 1-2, pp. 65–86, 2001.
- [40]. S. A. Horgen, "Essay on Intelligent User Interfaces and adaptivity," Division of Intelligent Systems, NTNU, 2001.
- [41]. J. Grundy and B. Yang, "An environment for developing adaptive, multidevice user interfaces," in Proceedings of the Fourth Australasian user interface conference on User interfaces 2003-Volume 18. Australian Computer Society, Inc., 2003, pp. 47–56.
- [42]. Y. Wang, S. Singh, J. Hosking, and J. Grundy, "An aspect-oriented UML tool for software development with early aspects," in Proceedings of the 2006 international workshop on Early aspects at ICSE. ACM, 2006, pp. 51–58.
- [43]. M. Pinto, L. Fuentes, and J. M. Troya, "A dynamic component and aspect-oriented platform," The Computer Journal, vol. 48, no. 4, pp. 401–420, 2005.

### Author' biography with Photo



**Atif Zaman** is Software Engineer & Web Developer, currently working at Global IT Soft as IT Director. He is student of MS (Computer Science) in University of Agriculture, Faisalabad, Pakistan. He is Member of IEEE (Member NO. 92462214) and ACM (Association for Computing Machinery) (Member NO. 1026328). His research interest includes Software Engineering, Web Application Development, Aspect-Oriented Components Engineering and Human Computer Interaction.



**Mudassar Ahmad** has completed his MS (Computer Sciences) in 2007 from University of Agriculture, Faisalabad. Now he is doing Ph.D. in Computer Sciences from Universiti Teknologi Malaysia, Johor Bahru, Malaysia. His core research area is Network Protocols.



**Shafique Ahmad** is Web Designer and Developer, He has completed his M.Sc computer science from University of Agriculture, Faisalabad, Pakistan. He is currently running a web development company. His main area of include implementation of programming languages, object oriented programming, web data management and database management.



**Tasleem Mustafa** received his MS(CS) from University of Agriculture, Faisalabad, Pakistan, MSC(CS) from University of Engineering and Technology, Lahore, Pakistan, MSC(Mathematics) from University of Punjab, Lahore, Pakistan and MIS from USA. He is working as a Chairman/Assistant Professor, Department of Computer Science, Faculty of Sciences, University of Agriculture, Faisalabad, Pakistan. His research of interest includes software engineering and database technologies.