Interface adaptation by content analysis: Literature Review

Jonathan Wasson 463117

April 21, 2014

UNIVERSITY OF THE WITWATERSRAND

SCIENTIA ET LABORE

JOHANNESBURG

Contents

1 Introduction

As computers begin to encroach on every aspect of everyday life. It has become more and more apparent that the diversity of people using computers is also increasing. This dependence on technology permeates most modern businesses and households. From doctors to store clerks, all these people will have to make use of a computer at some point in their everyday lives [Beaudry and Pinsonneault 2005]. Take doctors for example; Doctors must make use of software that can keep track of patient records as well as machinery that performs complicated procedures on patients. New technology has been developed that allows surgeons to operate over long distances through the use of the internet and augmented reality. Allowing countries without the necessary skilled labour to import it from overseas almost instantaneously in cases of emergency.

However, this progress has left a few behind. Older people, the disabled and often those who are not technologically savvy are struggling to adapt to the rigours of modern life. Nowadays, we can see old people who would have once refused to use things such as email or phones are now being forced to use these services and devices on a daily basis. Disabled people [Gajos *et al.* 2008] are also being forced to adapt to this change since most devices are designed in a way that is not suitable for their use. Indeed, this ever increasing reliance on technology has forced people to adapt to modern times and it can be said that the modern worker must be able to use a computer or face the risk of unemployment.

To justify this as a research topic we must analyse all available media that provides insight into this topic. The following paper will discuss material relevant to this question and then use it to justify this as a research topic.

2 Literature Review

2.1 Justification

As shown in the introduction the need for this research was justified by saying that more people are using technology on a daily basis but not all of these people are capable of using technology efficiently. How then can this issue be tackled to help these people improve their performance when using computers? It was shown in ”Understanding user responses to information technology: A coping model of user adaptation.” [Beaudry and Pinsonneault 2005] how users of varying technological ability responded to a new technology being introduced to their work environment. The paper found that although peoples reactions and abilities to adapt vary widely for each individual, it is possible for these adaptation ’Strategies’ to be grouped. The research concluded by observing actual people in real life scenarios that there are four broad adaptation strategies that cover all individuals coping mechanisms to new technology. This is highly important since it shows that if it is possible to label an individuals coping mechanism and behaviour then it is possible to develop a method to help any individual cope with new technology.

2.2 Implementation

How though can we implement this? How is it possible to create something that will automatically fit a person to effectively minimise the time they take to cope with new technology? There are several papers that have provided a solution to this problem. They propose that to optimise a users performance, one must simply adapt the interface of said program to perfectly suit the user [Lavie and Meyer 2010]. Whether or not this is applicable is another problem. ”Benefits and costs of adaptive user interfaces” by Lavie Talia and Meyer Joachim gives an outline of an experiment where this is explored. Through the use of an experiment involving people of different ages operating vehicles with different interfaces and then measuring their response times. They came to the conclusion that adaptive interfaces are beneficial in situations where computation time is not an issue. They also came to the conclusion that older people’s performances improved drastically when using adaptive interfaces but younger people’s performance either remained unchanged or dropped drastically. In these cases the interfaces actually impeded user performance.

This shows that it is possible to create an adaptive interface which can improve user performance but it was noted in [Lavie and Meyer 2010] that further research was required to explore and expound this topic. Which comes to my research hypothesis.

Hypothesis: The content and interface of a computer program can be updated or changed to automatically improve the users performance.

How then can an adaptive user interface be generated? There are several methods to accomplish this. The method that has the lowest apparent computation time and most apparent usefulness makes use of what is called a Markov decision process [Barto 1998]. ”Reinforcement learning: An introduction” [Barto 1998] explains in detail the applications of a field of computer science called reinforcement learning. Reinforcement learning is a machine learning process whereby the environment learns from the users interaction with it. The paper also covers a very broad range of applications, seeing use in topics such as genetic algorithms, psychology, control engineering etc. However there are several challenges with reinforcement learning. The major challenge is the trade-off between what is referred to as exploitation and exploration. To achieve a high reward, reinforcement learning must exploit as many decisions it has observed in the past. However, to obtain these decisions it must explore as much as possible which might lead to massive training data. The paper then goes on to explain that there is no easy way to pick an optimal route between exploitation and exploration.

A Markov decision process is a mathematical framework that can easily represent decision making when some decision may be random. Its most common representation is that of a directed graph where states and actions are nodes and the connections between these are probabilities of certain events occuring. Through the use of an MDP, we can easily create an iterative method that updates states and actions so that an optimal policy is possible. In conclusion the paper shows that reinforcement learning can generate agents whose performance improves over time through computational effort.[White III 1991]

These MDP's have been used in [Ramamoorthy *et al.*] and [Rosman *et al.* 2014] to generate interfaces. The MDP's were represented using the list data structure with look-up times increased using hash tables. In [Andrade *et al.* 2005] a representation of a basic MDP is shown. They used a list structure with each state mapping to several actions as well. Using this, the program can measure user's actions and then store them over several 'games'. This data can be averaged to show the behaviour of the user. Knowing which actions have better results will allow us to tweak the user interface so that the user picks better action more often.

So it has now been shown that adaptive interfaces can indeed be made and that through the use of an MDP we can generate a user profile for each user. All that needs to be shown is how to actually create and decide on interfaces for users. The method for doing this quite clear cut [Dessart *et al.* 2011]. All that must be done to adapt the interface is to create a transition function which is done as follows:

- Find what action has been changed by the user.
- Find which element of the user interface is affected
- Perform one of the following operations on the element: Resize, Relocate, Image transformation, Widget transformation, Widget splitting.

There is one more thing that needs to be shown. Once a user profile is selected and an appropriate interface selected, the user is then given the newer interface to use. After this the program will hone in on an optimal interface till the user is always performing his/her optimal actions [Dessart *et al.* 2011] [Andrade *et al.* 2005].

3 Further Study

An adaptive interface isn't exactly easy to implement due to computation constraints. Further study can be done in creating adaptive interface algorithms more efficiently. A computer game usually requires a large amount of computation power. This has large implications for the tech industry since several devices require a calibration stage before usage. An adaptive interface can be used to circumnavigate this and instead perform the 'calibration' on the fly. Take for example the computer game system called the xbox. Users upon first starting the console up must perform several basic adjustments to alter the interface to suit themselves. Most users find this highly annoying and often write complaints on thiss very issue to Microsoft as a result.

Overall, this research has never been fully explored and has only been implemented a few times in small contained experiments such as those in [Andrade *et al.* 2005], [Ramamoorthy *et al.*] and [Rosman *et al.* 2014]. Since this approach does not simply have to be used in the generation of user interface but can be used to map user preferences and behaviours. Due to this fact, there is a lot of room for improvement and exploration. Thought has been given to use this approach in the generation of RSS feeds on the internet. Even advertisements can use this method to create user profiles and then a mapping from one advert to another.
This research can, quite literally, be used in almost every aspect of computer systems known to man since these computer systems have several different users with varying abilities. This research will explore and tackle this in a way that can be implemented anywhere.

4 Conclusion

Therefore it has been proposed that an interface can be created which will increase the users performance. Which can be summarised as follows:

Hypothesis: The content and interface of a computer program can be updated or changed to automatically improve the users performance.

The commercial and societal impact of this research has been shown to have merit by showing that several companies would benefit directly from this research. Lack of research in this area of computer science indicates that more research is required to further this field. Many new and novel ways have also been suggested to generate adaptive interfaces but have not been fully explored.

References

[Andrade *et al.* 2005] Gustavo Andrade, Geber Ramalho, Hugo Santana, and Vincent Corruble. Challenge-sensitive action selection: an application to game balancing. In *Intelligent Agent Technology, IEEE/WIC/ACM International Conference on*, pages 194–200. IEEE, 2005.

Aim: Through use of reinforcement learning techniques, this paper presents a method to create intelligent agents that can adjust game difficulty.
Style: Conference paper
Cross References: This paper is used extensively in [Rosman et al. 2014] and in [Ramamoorthy et al.] to help explain and identify how to create an adaptive agent.
Keywords: Adaptation, Intelligent agent, Adaptivity, Dynamic Skill Adjustment
Summary: This paper deals with providing a dynamic skill adjustment feature to programs. Specifically for users of varying skills and capacity for adapting. In particular the paper focuses on providing this dynamic skill adjustment in computer games.
A method is laid out and tested, whereby an agent is created and tested versus various opponents. The steps involved to accomplish this are as follows:

- An agent is created which will learn from the human opponent in the specific game being tested.
- The agent creates a table of available moves for each state. Over successive generations; based on the success of these moves; the action set is updated. Where the decisions learned by unskilled players are added along with those of the skilled players. All this means is that the agent can pick actions based on skill level.
- Once trained the agent will always start at a middle skill level and based on the performance of the user it will slowly change and adapt the actions chosen to meet the player at his or her appropriate level.

In the paper, this method is used to create 3 difficulty levels for the game 'Knock 'Em' which is basically a simulated boxing game where players can move around a 'ring' using the WASD keys, they can jump using the space bar and aim the camera using the mouse. Points in this game are awarded whenever a successful punch is landed on the opponent.
To test the performance of this agent, three seperate agents of varying skill levels were created. The adaptive agent was then tested over 30 games versus each of these other agents. It was found that the adaptive agent honed in to match each specific agent's performance, eventually reaching a state of fluctuation where average score remained around the 50/50 win/lose ratio. Effectively showing that an agent can be created that adapts to the skill level of the user.

[Barto 1998] Andrew G Barto. *Reinforcement learning: An introduction.* MIT press, 1998.

Aim: This paper explores the concept of reinforcement learning and explains it thoroughly.
Style: Explanatory/ Theoretical
Summary: Reinforcement learning is a machine learning process whereby the environment learns from the users interaction with it. It also covers a very broad range of applications, seeing use in topics such as genetic algorithms, psychology, control engineering etc. However there are several challenges with reinforcement learning. The major challenge is the tradeoff between what is refered to as exploitation and exploration. To achieve a high reward, reinforcement learning must exploit as many decisions it has observed in the past. However, to obtain these decisions it must explore as much as possible which might lead to massive training data. The paper then goes on to explain that there is no easy way to pick an optimal route between exploitation and exploration. Markov decision processes theoretical nature and definition is then covered extensively.

A Markov decision process is a mathematical framework that can easily represent decision making when some decision may be random. Its most common representation is that of a directed graph where states and actions are nodes and the connections between these are probabilities of certain events occuring. Through the use of an MDP, we can easily create an iterative method that updates states and actions so that an optimal policy is possible.
In conclusion the paper shows that reinforcement learning can generate agents whose performance improves over time through computational effort.

[Beaudry and Pinsonneault 2005] Anne Beaudry and Alain Pinsonneault. Understanding user responses to information technology: A coping model of user adaptation. *Mis Quarterly*, pages 493–524, 2005.

Aim: This paper explores and explains thoroughly the behavioural and cognitive efforts that users perform to handle and cope with technological events that occur in their work environment as well as explores the psychlogical impact of these events.
Style: Theoretical
Summary: The paper posits that people use two processes to cope with an event. Firstly they assess the problem's nature and formulate how this event affects themselves. They then perform the relevant actions to deal with the problem. These two processes can then be broken up into several subcategories such as emotion-focused coping and problem-focused coping.
It then further explores the actions humans perform to deal with these problems. The actions are laid out as follows:

- Benefits maximising strategy: Maximises benefits at the detriment or risk
- Benefits satisficing strategy: Minimises risk but keeps benefits at a satisfactory level
- Disturbance handling strategy: Lowers risk when high control is available
- Self-Preservation strategy: Lowers risk as much as possible

To show the affects of all these human cognitive processes, a case study was presented wherein a new computer system was introduced to two banks. The banks remain annonymous and are simply refered to as Bank A and Bank B. The new system is called Link, very little detail is given on this system. It is simply stated to be a large change from each banks previous systems. Statements and reviews of this system are taken from all the banks employees. This data is then thoroughly examined and specific instances of the above coping mechanisms are pointed out.

[Dessart *et al.* 2011] Charles-Eric Dessart, Vivian Genaro Motti, and Jean Vanderdonckt. Showing user interface adaptivity by animated transitions. In *Proceedings of the 3rd ACM SIGCHI symposium on Engineering interactive computing systems*, pages 95–104. ACM, 2011.

Aim: This paper lays out the method used to make transition functions and reinforces the reasons for their usage.
Style: Theoretical
Cross-references: This paper is used in [Rosman et al. 2014] and in [Ramamoorthy et al.] to help explain the method used to create transition functions and will be used extensively in my research to achieve the same. These will be used to create a new user interface at each iteration.
Keywords: Adaptation, adaptivity, transition operation, selection mechanism, transition operation, visual transition.
Summary: The method used to create a transition function is the crux of this paper and as such I will simply summarise this method in point form below:

- Find what action has been changed by the user.

- Find which element of the user interface is affected
- Perform one of the following operations on the element: Resize, Relocate, Image transformation, Widget transformation, Widget splitting.

[Gajos *et al.* 2008] Krzysztof Z Gajos, Jacob O Wobbrock, and Daniel S Weld. Improving the performance of motor-impaired users with automatically-generated, ability-based interfaces. In *Proceedings of the SIGCHI conference on Human Factors in Computing Systems*, pages 1257–1266. ACM, 2008.

Aim: The paper evaluates two methods to generate user interfaces that specifically cater to a disabled person.
Style: Experimental
Cross-references: This paper illustrates the usefulness of my research topic and gives a real world applicable use for it.
Keywords: Ability-based user interfaces, Adaptability, Motor impairments.
Summary: A way to automatically generate a user interface to perfectly cater the needs of the specific user is presented. In this paper the example of disabled people is specifically explored. The real world applicabilty of this specific scenario in content adaptation is why this paper has been explored (i.e. To validate the applicability of my research).
Several participants with a range of disabilities were chosen to take part in developing this software. They were each asked to pick a device they would like to use; objects such as a mouse, tracking ball, keyboard etc. A few tasks were then presented to them for completion. Each task mapped a specific motor ability; things such as dragging, pointing, list selection and multiple clicking. All of which are basic functions able bodied people can do without thought. The participants were then asked to alter the devices settings to a level that most improved their performance and that they were most comfortable with.
User profiles for each specific device and disability were created. An algorithm that uses reinforcement learning was then made to generate interfaces that were most suitable for each user profile. Software was then created using this data that would then generate a user interface best suited to each user based on the training data used to generate each individual user profile.
Testing was then done for the software using people with similar disabilities. The paper then concluded that motor impaired participants overall average performance improved by 26.4 percent. Which showed that automatic generation of user interfaces is feasible and that these interfaces improve performance and the quality of computer interaction.

[Lavie and Meyer 2010] Talia Lavie and Joachim Meyer. Benefits and costs of adaptive user interfaces. *International Journal of Human-Computer Studies*, 68(8):508–524, 2010.

Aim: Due to the cost of adaptive processes, this paper explores the benefits and costs of an adaptive process to establish whether or not it is viable.
Style: Conference paper
Summary: An explanation of Adaptive user interfaces or AUI's is given. With a list of situations given where this useful. The example of an interface where the menu can be split into smallers submenus is given. A hypothesis is made saying the following: "(1) In routine situations, the fully adaptive system will be more beneficial... (2) In non-routine situations, the manual and more routine situations will be more beneficial ..." which seems to point out that in an unpredictable scenario it would be better not to use an AUI due to the time taken to compute the system whereas the opposite would be true for a predictable scenario since computation time won't be an issue.
To test this hypothesis an experiment is laid out in the paper whereby several people are asked to drive cars on a test track. The cars have varying degrees of adaptivity in their controls ranging from fully manual to fully adaptive. Where fully adaptive would entail

the car being able to predict certain actions based on the users input etc. Performance was measured in the time taken to respond to obstacles.
The hypothesis was then validated later on in the paper when people who were driving manual cars were better able to avoid random interruptions. Through successive iterations the developers then added the feature to adaptive cars to switch to manual in the event of a crisis. Performance increased drastically after this addition.
The tests were then performed purely on a group of older participants. The results differed completely, in that the older people had higher performances when using adaptive interfaces in non-routine situations, the exact opposite of what was shown above. The paper concludes that the adaptive interfaces were useful in certain scenarios and required tweaking in others. It was noted that this conclusion was entirely dependent on this specific research study and that further research will be required to explore this topic further.

[Ramamoorthy *et al.*] Subramanian Ramamoorthy, MM Hassan Mahmud, Benjamin Rosman, and Pushmeet Kohli. Latent-variable mdp models for adapting the interaction environment of diverse users.

Aim: To create a personalized inteface such that it adapts to the users input over time to form an optimal performance enhancing interface.
Style: Experimental
Cross-references: The following papers were consulted to aid in the production of this paper: [Gajos et al. 2008],[Dessart et al. 2011],[Beaudry and Pinsonneault 2005] and [Andrade et al. 2005].
Summary: The paper explained how to select an agent type to use when generating an interface. A latent variable Markov Decision Process was chosen as the transition and reward of actions are included in the formulation of an MDP. An experiment was then formulated to illustrate the use of a latent variable MDP. The experiment is as follows:

- A 20x20 grid was created to represent a game world. The goal of this game is for the users to move a dot from a fixed start to a fixed goal.
- Different contollers are presented to each user which then represent skill level.
- At each step the user chooses a firection to move in. The algorithm then makes a prediction of where the user will move through Baysian probability.
- Performance was then measured for users of varying skill levels and control structures. This one done for 100 episodes each.

The paper concluded that it had indeed used a method to create a Markov decision process that represents an action-set selection.

[Rosman *et al.* 2014] Benjamin Rosman, Subramanian Ramamoorthy, MM Hassan Mahmud, and Pushmeet Kohli. On user behaviour adaptation under interface change. 2014.

Aim: The following problem is presented in this paper: Given an action set/interface, users eventually learn policies that will make them efficient in the usage of that interface. What happens when the interface is changed or a new one is introduced? Experiments are devised to explore this. This paper aims to create interface adaptation algorithms that are capable of taking the users personality traits into account.
Style/ Type: Conference paper
Cross References: The following papers were consulted to aid in the production of this paper: [Lavie and Meyer 2010],[Gajos et al. 2008],[Dessart et al. 2011],[Beaudry and Pinsonneault 2005] and [Andrade et al. 2005].
Keywords: Input and Interaction technologies, usability research, user interface design and usability testing **Summary:** Five experiments are implemented in this paper. All of which relate to the game of moving a ball through a field of circular obstacles to a fixed

goal. Users actions and time taken are recorded for each game. The experiments were conducted for a set of tasks with a specific interface then for a different interface. In total there are 3 interfaces. Each of the five experiments relates to a different ordering and combination of these interfaces.

The purpose of this is to measure users response times for each combination to determine whether or not a change of interface alters performance. The obvious thought is that changing interface will lower performance as the user must now adapt to a newer different interface that he/she is not used to. However the experiments results indicate that this is not entirely true. The experiments showed that performance in some cases decreased and in some cases increased.

Further discussion was then held on how to mitigate these affects. With the conclusion being reached that migratory or intermittent interfaces can be used to give the user a more gradual change between interfaces allowing for time to adapt.

[White III 1991] Chelsea C White III. A survey of solution techniques for the partially observed markov decision process. *Annals of Operations Research*, 32(1):215–230, 1991.

Aim: This paper explains what a Markov decision process is and how to implement it in a virtual environment.

Style: Theoretical

Summary: A Markov decision process is a mathematical framework that can easily represent decision making when some decision may be random. Its most common representation is that of a directed graph where states and actions are nodes and the connections between these are probabilities of certain events occuring. One of the best ways to represent this in code is to use a linked list with states and the respective actions weights listed. It then further explores and expands on the intracacies of achieving this.