Smart UI: My Commentary

Stephen Lofthouse

Sheffield Hallam University
Faculty of ACES

Abstract

The rise in e-commerce has provided the retailer with a wealth of customer data that they would have been otherwise unable to collect. Through the use of modern data analysis techniques including Formal Concept Analysis, Content Based Algorithms and Collaborative Filtering Algorithms, it Is possible to mine this wealth of customer data and produce product recommendations that can be delivered to the user via a smart user interface.

Smart UI: My Commentary

This report is a commentary on Smart user interfaces and the integration of user profiles within them. In it I will discuss Smart User Interfaces, their implementation through the use of SAP's Web Dynpro technology and how those smart User Interfaces can be supported and enhanced by the use of user profiles generated using a variety of algorithms including formal concept analysis. I will then go on to offer my conclusions before final presenting a list of references should the reader wish to learn more about any of the topics covered.

In order to provide an effective implementation of a smart user interface we should start by defining our understanding of one. With respect to this report we shall define a Smart User Interface using a revised version of Jacob Nielsen definition of a user experience (Nielsen Norman Group).

> 'The first requirement for an exemplary Smart User Interface is to meet the exact needs of the customer, without fuss or bother. This goes far beyond giving customers what they say they need and takes us into the realm of anticipation, of fulfilling a need before that need is articulated. In order to achieve this there must be a seamless merging of the services of interface design, business intelligence and data mining and warehousing.'
> Stephen Lofthouse - November 2008

The Smart User Interface is one which is enhanced in order to take account of the user who is interacting with it. These enhancements make take many forms dependent upon the needs of the user and the motivation of the business with whom the user is interacting. A business may choose to implement a Smart User Interface for its employees to present them with the real time aggregated business data thereby facilitating an improved performance. Whilst a retailer may also choose to implement a Smart User Interface for the customers of its e-commerce site in order to target its products to those customers based on accumulated customer data and or profiles e.g. Amazon.com. But regardless of the implementation the generation of an interface, based upon predefined profiles or mined historical data is key. There are a number of ways to implement this dynamic interface generation but perhaps the most widely used is the Model View Controller Pattern.

Model View Controller

The model view controller pattern was first described by Trygve Reenskaug, a Norwegian computer scientist, in the late 1970's, (Reenskaug, 2003). A common technique within the field of Object Oriented programming is to split the application into layers or tiers (n-tiers) a common separation being

the Data base tier, the Business Logic Tier and the Presentation Tier, the Model View Controller pattern was perhaps the precursor to this technique. In the Model View Controller pattern the Model represents the business data and business rules enacted upon that data. The view is concerned with the user interface and all the elements it comprises and finally the controller manages the communication of user interaction with the model, communicating to the other layers. Unlike the n-tier pattern which is depicted as linear, the Model View Controller pattern is represented as a triangle (Figure 1).

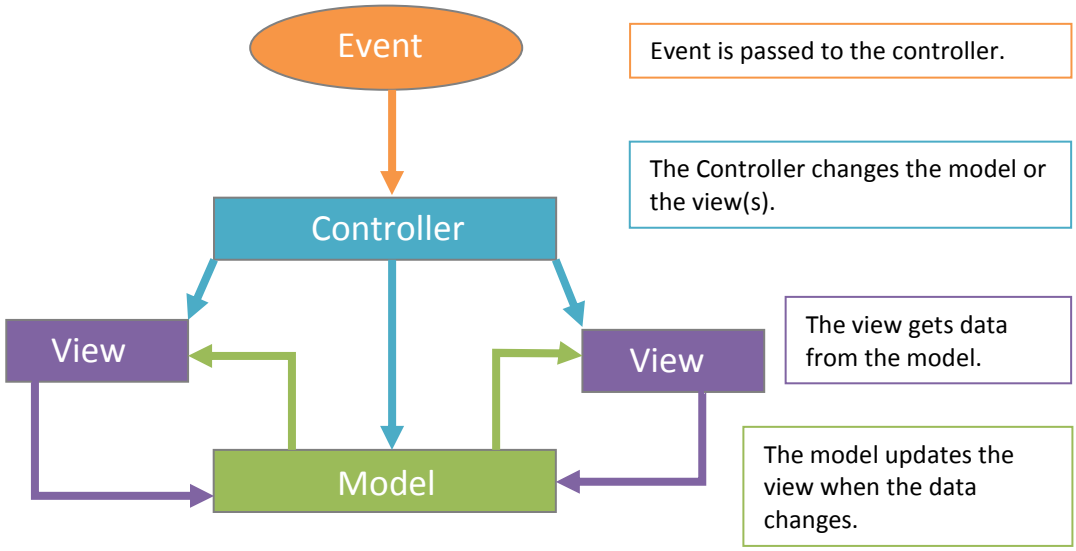

Figure 1- Model View Controller (eNode, 2002)

Because of this clean separation, multiple views and multiple controllers are able to interface with the same model. In addition new types of controller and new views that have previously not existed can interface with a model, without the need to change the models design.
Events would typically cause a controller to change a view or a model or both. All of the views dependent upon a model, are updated when that models data or properties are changed by a controller. Views or parts of them can be dynamically generated based upon the data and business rules within the model i.e. views can be based upon user profiles and business intelligence and the user interface can be generated and updated in line with changes to the users profile and relevant business intelligence data.
Business intelligence and user based profiling

The rapid growth of the internet and e-commerce has presented retailers and consumers with new challenges and opportunities. Consumers are able to shop without leaving the comfort of their homes. They are able to search a number of retailers websites for their desired purchase and are able to easily implement a number of metrics, e.g. item price, supplier reputation, speed of delivery etc, in order to influence their purchase decisions. This has meant that retailers have had to ‘up their game’. Retailers are

no longer presented with a customer who has walked through the door; a physical person with whom the highly trained sales staff can interact, link sell and otherwise build a relationship with. But the rise in e-commerce has also given the retailer the ability to gather vast amounts of accurate data about their customers, data that would have been far harder to gather in a tradition bricks and mortar retail establishment.

Accurate customer profile and web usage information can be used to help attract new customers, retain current customers, improve cross marketing and sales, and increase the effectiveness of promotional campaigns. Data mining techniques can be used to anticipate the user's behaviours in real time. It is possible to develop recommendation systems which analyse, in real time, the user's current navigation path through the site, compare it to previous visits and provide recommended products and links based both on this data and on historical purchases.

Due to the rapid development and decreasing cost of large scale computing hardware and software the option to implement User Based Profiling is, comparatively speaking, not a costly one. Web technologies such as Asynchronous JavaScript and XML (AJAX) allow for information to be transmitted from the web front end to the server back end transparently, without the need to refresh the whole page. The Model View Controller pattern mention earlier facilitates the development of a dynamically generated web pages based on business intelligence. Therefore not only is it possible to track an individual user's passage through a web site but also their interaction with it i.e. which button / link they clicked and dynamically generate links and product offers in real time in line with the business intelligence about that consumer and the consumers demographic grouping.

In order to make purchases from these e-commerce merchants' users must register and provide a range of demographic data to facilitate payment and delivery of goods. Returning customers must log in and make themselves known to the web site. This validated identity can then be used to transform anonymous usage data gathered through the use of cookies and site interactions into data linked to a particular profile. This data together with data from other profiles can then be analysed to generate information. Data analysis would typically focus on three main areas;

Association rules - Used to find associations between web pages/products that frequently appear together in user's sessions.
Sequential patterns - Used to discover navigation patterns and product buying patterns.
Clustering & Prediction - Clustering is used to group together similar patterns and prediction methods are used for forecasting.

This information would be married with data gather from sales history records and used by the business to make better informed decisions and employ targeting of its customer base with increased efficiency.

Web Dynpro

SAP's Web Dynpro technology implements the Model View Controller pattern described earlier. It does this using a declarative style of development; the focus being on what should be achieved rather than how it should be achieved. SAP propagates this declarative approached in a number of ways. The Web Dynpro environment permits the developer to define screen flow and management. The developer specifies not only the view and the controls within it but also the navigation and interaction between views. Controls within a view can be data bound to objects within the views controller and the controller objects can be bound to objects within the model. Web Dynpro will generate code to facilitate the automatic flow of data to and from these objects and it does all this without the need to handwrite any code.

Web Dynpro facilitates the development of components. Components are a collection of views and controllers which are packaged as a single entity. These components can contain code that will change the layout of a view, adding or removing controls and generating navigation links at runtime when they become bound to the data structures within the controller or model.

Because of the separation of business logic from front end presentation enacted by Web Dynpro's implementation of the Model View Controller pattern, Web Dynpro is able to cope with a number of different browser clients. Web Dynpro uses JavaScript to render pages at the client side. Handheld devices that do not offer a java scripting engine have their pages generated by the Web Dynpro runtime on the server thereby ensure that the appropriate output for that client is generated. This also means that, due to the implementation of the Model View Controller pattern, the underpinning business logic, which is separated from the presentation layer, does not need to be amended. An additional benefit of this separation is a degree of future proofing on the part of the business. Should a new user interface technology come to the fore in the future only the view need be changed. The model and controller i.e. the business logic and the data are separate and can be retained.

Capturing User Profiles Through Formal Concept Analysis

The construction of an accurate user profile is a key task of any business intelligence system used within an e-commerce setting. The system will only be of value to the business and the user if the profiles it constructs are accurate. The aim of a user profile within the business context is to create a 'sticky' personalized experience for the user and, from the businesses prospective, to increase sales and customer retention.

What to capture

In order to facilitate the construction of user profiles that are of value to both the business and the user, the initial consideration is with regards to what data the business should capture about the user. A user profile will typically comprise two parts, static data and dynamic data. The static data would comprise demographic user details e.g. the users name, address, email address etc. This static data rarely changes.

Dynamic data is data that either changes or increases in volume, sales data would be classified as dynamic in this context. The business may choose to capture only the products a user buys. Or it could extend its capture mechanism to store products a user views, pages a user navigates, the amount of time a user spends on a page, whether the user requests further information about a product, did the user click on any advertisements and if so which ones. All of this data can be gathered from server logs and sales databases and processed accordingly but increasing the volume of data captured will increase the business intelligence processing time. So the more information a business chooses to capture the greater the amount of business intelligence it could produce but the longer it would take to produce that data.

How to capture it

As we have already mentioned there are a number of ways to capture user data. With respect to static user demographics the business can simply ask the user for it. Typically this is done via a registration form or the payment and delivery section of an e-commerce site.

In order to capture data about which products a user buys the business would query the sales database. One could gather the sales history of a specific user as well as performing data mining and analysis techniques on the entire sales database in order to discover patterns and correlations between

purchases. Information about which pages a user navigates through and which pages a user views could be gathered from the web server log files and session history logs.

An additional method of gathering user data is facilitated by the use of AJAX (Asynchronous JavaScript and XML) within a web page. Using this technology it is possible to send data concerning every interaction a user makes on a page, back to the server without the page refreshing. This may be suitable for a small site with a limited number of products but would be unsuitable for a large business given that the amount of data it would generate would preclude its timely analysis.

Regardless of the method used once captured the data will require cleansing and validation before it can be analysed and made use of. This cleaning and validation process would comprise a number of steps and could be undertaken both pre and post transaction. For example when a user completes a registration form, the address they supply could be validated against the Post Office database. The payment information they supply would be validated with a financial institution. When mining sales history data the company may choose to exclude all sales where the product purchased was returned or exclude all products that are no longer stocked. Information gathered from server logs would require analysis to determine which page views are relevant and exclude pages navigated by search engine crawlers.

Analysis

Previously businesses have used data mining techniques and prediction algorithms to create product recommendations. These prediction algorithms could be classified into one of two methods. Content Based recommendation algorithms and Collaborative Filtering based algorithms.

Content Based algorithms

Information about a user's preferences is maintained, either by completion of a relevant form at registration of by maintaining a history of user purchases. Recommendations are made by taking into account the users purchase history and recommending products which match the user's preference and stored profile. For example a music retailer may recommend a user purchase the new album from band X as they have previously purchased an album by that band.

Collaborative Filtering algorithms

The aim of these algorithms is to calculate similarities and dissimilarities between users and recommend products based upon them. For example users who bought this also bought that.

The suitability of each of these methods differs based upon the product being recommended and the way in which the user profile is constructed and maintained. But in both cases the methods suffered from the same inherent problems;

Sparcity

Even a very active user would, comparatively speaking, purchase just a few of the available products. Due to the manner in which the algorithms function, high degrees of sparcity can be detrimental.

Scalable

Recommendation algorithms require data mining and analysis methods which are computationally expensive and time consuming. These costs will grow with an increasing number of users and products within the database. In order to facilitate the provision of recommendations within an acceptable time scale, advanced and scalable architectures are required.

Cold-Start

Products cannot be recommended until they have been purchased by a number of users. This factor is especially detrimental to new or obscure products or indeed users with unusual purchase histories.

A more advanced alternative that would go some way to negating the above problems could be achieved through the use of a domain specific ontology together with formal concept analysis of both the user data and the sales data and the production of concept lattices based upon the results.

Developed in the 1980's by Rudolf Willie, Formal Concept Analysis is a method of analysing data, representing knowledge and managing information and has it's roots in mathematical lattice theory.

A simple example would be the creation of a list of customers who bought a certain product. By creating a number of these lists and representing them diagrammatically as a lattice, it becomes possible to detect patterns and buying habits with relative ease. By calculating the similarity between 2 customers it becomes possible to recommend products to a customer on the basis that they have been purchased by customers whom the system has calculated to be similar.

With respect to an e-commerce site a domain ontology could comprise all the products available for sale by the site. Using standard data mining techniques together with formal concept analysis using the ontological definitions as concept nodes a concept lattice could be created that comprised the domain ontology together with a rating for each product. This rating could simply be the amount of that item sold over the past 12 months. This would result in a structure similar to that shown in figure 2 below. The business may choose to regenerate this ontological rating lattice each month to take account of the previous month's sales data thereby keeping the lattice up to date and effectively managing the data.

Similarly a user lattice could be produced in the same way using data mined from the user's sale history. Again this sales history data would use concepts derived from the domain ontology for its nodes. As this lattice is relatively small it could be constructed on the fly when the user presents themselves at the e-commerce site. This would reduce the problems associated with scalability as only the users concept lattice is generated on demand.

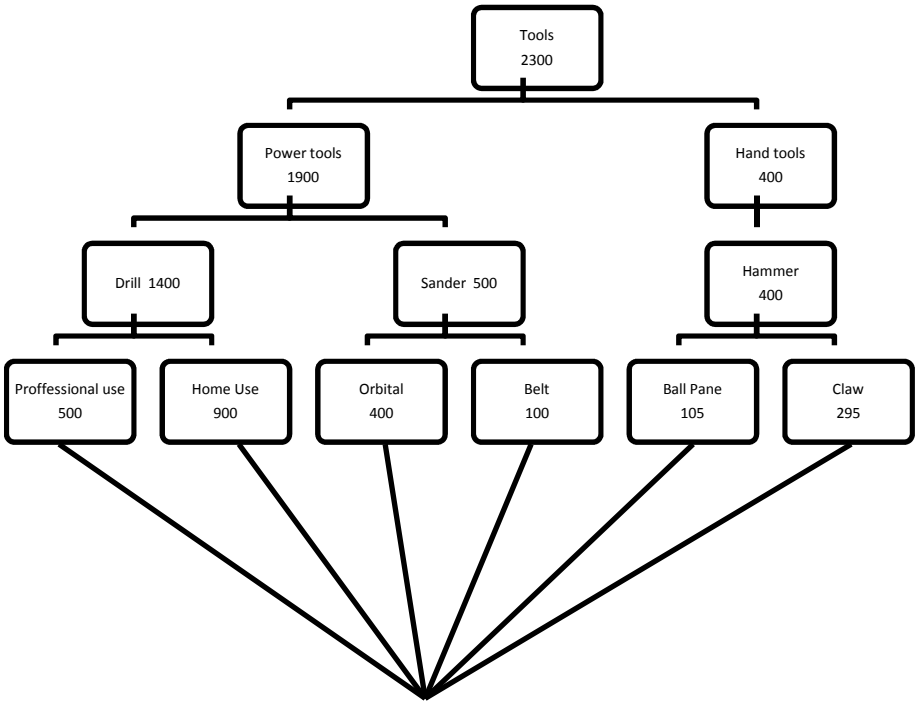

Figure 2 – Ontology based concept lattice with product ratings

The use of a domain ontology facilitates the overlay of one lattice on the other; it is then a relatively straight forward task for the business intelligence component to perform content based recommendation of products to the user.

The use of the domain ontology would also enable the business intelligence component to recommend products which had little or no sales data purely by navigating the ontology thereby eliminating the Cold-Start problem. As concept lattices facilitate both querying and navigation of the data they enable other search strategies for example ontology climbing, space pruning and partial views. The merits and performance of utilising FCA and Concept Lattices for enacting hybrid search strategies has been supported in a number of academic studies. And these methods continue to facilitate the mining of information that would be difficult to acquire using traditional retrieval methods

Whilst the use of FCA and concept lattices can go some way to eliminating the problems associated with the more traditional forms of business intelligence the data used to generate the lattices still has to be obtained. Ideally user sales data would be stored in a data structure which for all intense and purposes already was a concept lattice. This would eliminate the time needed to mine the data and generate the lattice as the lattice would already exist. A data structure of this type would also permit easy navigation and discovery of related concepts and product recommendations. Such a novel data structure does exist in the form of Mazzagatti's Triadic Continuum. This data structure stores its data in a series of tree like triangular nodes. This node based, tree like structure is self organising and facilities the discovery of business intelligence quickly and with little computational overhead.

Integrating Formal Concept Analysis Based User Profiles Into The Smart User Interface

As we have already stated the use of the model view controller pattern facilities the development of a user interface that is dynamic. The elements within the interface can be created based upon business intelligence data. We are therefore able to produce an interface or elements of it that are unique to the user who is currently interacting with the interface.

Numerous studies have shown that users feel they get the most benefit out of e-commerce recommendation systems when these systems recommend products that the user had not considered. Using our user profile concept lattice, together with our ontological sale ratings lattice we are able to discover what products a user has purchased and suggest other products the user may wish to purchase.

Our dynamic interface would allow us to create a web page with these recommendations prominently displayed and render that page on the browser.

Indeed SAP's Web Dynpro technology, through its implementation of the model view controller pattern facilitates this kind of development. Additionally, as we have already stated, Web Dynpro will automatically generate code to facilitate the transfer of data to and from the rendered page so we are able to continually update our business data.

So in order to bring everything together and implement our FCA based User profiles within a smart user interfaced e-commerce site we would;

i. Decide what data we wanted to store about our users (taking into account the relevant data protection laws and regulations set down by the credit card companies.)
ii. Develop data storage
iii. Create ontology's for our domain.
iv. Develop our web front end such that the store functions and relevant user data is transmitted to the back end storage system.
v. Develop our data mining, Formal Concept Analysis, Concept Lattice creation algorithms and business intelligence software to create user profiles from the data we are accumulating.
vi. Integrate the User profile based product recommendations into the web front end.
vii. Evaluate its performance
viii. Update components which are not performing as expected.
ix. Evaluate.

Conclusion

The use of FCA based user Profiles as the basis for e-commerce recommender systems does bring benefits to both the retailers that implement them and the users that are profiled by them. But there are a number of issues that should be flagged up.

There are growing concerns with regards to the privacy implications of big business holding vast amounts of data about us. With increasingly sophisticated modelling and prediction techniques, retailers are increasingly able to predict our behaviour with alarming accuracy.

Whilst we have stated that the cost to implement BI based user profiling is comparatively small when offset against the benefits it can bring to a business it should be noted that SAP's Web Dynpro is not the cheapest way to implement this functionality. The initial cost to the business to implement SAP'S system can be high. In 2003 the average cost to a business for a 3 year deployment of SAP's software was $10 million (Vance, 2003). Now it should be said that that cost represent a full SAP system but the point is that Web Dynpro is an expensive technology to implement and the same level of functionality can be achieved using alternatives such as Microsoft .Net or PHP for a vastly reduced cost.

References

Abraham, A. (n.d.). Natural Computation for Business Intelligence from Web Usage Mining. Retrieved 11 22, 2008, from Soft Computing: http://www.softcomputing.net/synasc05.pdf

Anand, S. S., Kearney, P., & Shapcott, M. (2007). Generating semantically enriched user profiles for web personalization. ACM Transactions on internet technology , Article 22.

Bamshad, M., & al, e. (2007). Web search personalization with ontological user profiles. Chicago USA: Centre for Web Intelligence - DePaul University .

Carpineto, C., & Romano, G. (2004). Exploiting the potential of Concept Lattices for information retrieval with CREDO. Journal of Universal Computer Science , 985-1013.

eNode. (2002). Model-View-Controller Pattern. Retrieved 11 15, 2008, from Mark Up Language: http://www.enode.com/x/markup/tutorial/mvc.html

Fahle, A. (2005). SAP Web Dynpro Overview. Seminar System Modeling. Hasso-Plattner-Institute for Software Systems Engineering.

Kwon, O., & Kim, J. (2009). Concept lattices for visualizing and generating user profiles for context-aware service recommendations. Expert systems with applications , 1893-1902.

Mobasher, B., & al, e. (2002). Discovery and evaluation of aggregate usage profiles for web personalization. Data mining and knowledge discovery , 61-82.

Nielsen Norman Group. (n.d.). Strategies to enhance the user experience - User Experience - Our Definition. Retrieved 11 22, 2008, from NN/G: http://www.nngroup.com/about/userexperience.html

Papagelis, M., & Plexousakis, D. (2005). Qualitative analysis of user-based and item-based prediction algorithms for recommendation agents. Engineering applications of artificial intelligence , 781-789.

Reenskaug, T. (2003, 08 20). The Model-View-Controller (MVC) Its Past and Present. Retrieved 12 20, 2008, from Trygve M. H. Reenskaug: http://heim.ifi.uio.no/~trygver/2003/javazone-jaoo/MVC_pattern.pdf

SAP. (2004). The basic facts about Web Dynpro. Retrieved 11 22, 2008, from SAP Community network: https://www.sdn.sap.com/irj/scn/go/portal/prtroot/docs/library/uuid/91b8c890-0201-0010-c787-be96f6fade89

SAP. (2007, 12 26). Web Dynpro Overview. Retrieved 11 22, 2008, from SAP: https://www.sdn.sap.com/irj/scn/go/portal/prtroot/docs/library/uuid/1c22cf90-0201-0010-92be-b6ef4c2df31c

Vance, A. (2003, 03 31). SAP costs too much. Retrieved 01 08, 2009, from The Register: http://www.theregister.co.uk/2003/03/31/sap_costs_too_much_customers/

Zuchero, J. (2007, 10). The Best New BI Invention You've Never Heard Of. Retrieved 12 20, 2008, from DMReview.com: http://www.dmreview.com/specialreports/2007_44/10000157-1.html