https://doi.org/10.48009/2_iis_2009_293-298

USING ENTERPRISE MASHUPS AS A BUSINESS INTELLIGENCE SITUATIONAL APPLICATION

Robert E. Samuel, Robert Morris University, robert.samuel@ieee.org

ABSTRACT

A review of recent literature highlights that web 2.0 technologies are enabling a new programming approach. Business end users are able to analyze disparate data to aid in time sensitive decision-making that was traditionally performed by data warehouse and business intelligence infrastructures. The Internet has enabled the consumer to become aware of the possibility of situated software. Data is no longer purely controlled by enterprise IS organizations. Mashup platforms allow users to create an ad-hoc composition of data sources into a new representation to serve as a business intelligence report. By making all data available, business end-users can aggregate the data that they deem important, not just the data that was made available in existing data warehouse dimensions.

Keywords: mashups, enterprise data services, situated software, situational applications, business intelligence, web 2.0

INTRODUCTION

Enterprise business users are faced with the increased need to make timely decisions based on real-time data available from disparate sources to solve an immediate business problem. Often the majority of the data is available in the existing enterprise structured repositories, however, the addition of unstructured data and publicly available data offers the opportunity for information richness. This problem leads to the scenario where end users that are proficient in a particular industry discipline require an application that can be quickly assembled and made available for a short duration. Traditional software development cycles often prohibit Information Systems (IS) departments to respond to this scenario under the given constraints. Jhingran [7] refers to this scenario as situational applications and can be solved using an enterprise information mashup fabric. Traditional data warehousing and business intelligence approaches do not provide the agile characteristics needed to address this scenario. Madsen [8, p.56] explains "you can develop a mashup in an afternoon if you know what you're doing. To achieve the same result in the BI [business intelligence] environment can take weeks".

The consumer Web experience has altered the way enterprise business users consider the ease to interact with information [8]. Traditionally, for business users to extract useful decision-making information, a data warehouse is created containing the useful elements of data from disparate operational data stores. Using business intelligence tools this data is modeled to yield information related to the users' discipline. As Madsen [8, p.56] states "the frustration of BI often begins with a 'simple request' to add functions or data from another system, work that can take months to deliver. Users don't want to be shown more logical data models or complex analytical tools". Additionally, enterprise users are realizing that "much of their data and intelligence are siloed - either unavailable at all to workers who might benefit from it, unavailable in a timely enough fashion, or locked in the desktop tools used to combine and present analysis" [4, p. 54]. Enterprises need to refocus on creating a culture and mechanisms along with accessible data via a centralized mashup directory [4].

This paper reviews existing literature related to the topic of how the emergence of situational applications, such as mashups, offers a new opportunity for data warehouse and business intelligence solutions. By realizing this opportunity, enterprises can address the tension between the unmet need of business units and the limited IS resources available to create solutions [4]. Enterprise information architects can enable a collaborative platform using unstructured and structured data sources for end users to create tactical, self-service applications. Meanwhile, the information architects can remain focused on providing enterprise, structured data warehouse designs that core business processes depend on for long-term historical analysis.

SITUATIONAL APPLICATIONS

Situated software is "a form of opportunistic software that a small subset of users create to fulfill a specific purpose" [1, p. 50]. Instead of a team of programmers gathering requirements, creating a design, developing code, testing the functionality, and finally deploying to a common infrastructure, end users use exposed applications and data to construct functionality that is unavailable from other

software [6]. Nardi [6, p. 123] refers to this new programming paradigm as end user programming and defines it as:

> end users who themselves have control over computational resources will be able to effectively manage application development and ongoing extension to meet changing needs. Computing work must therefore be distributed across the people in the organization, not bottlenecked at programming professionals.

This task-specific programming approach leverages the user's strength of domain knowledge and skills based on years of training in their chosen field along with basic programming experience. The end users require a perpetual beta development cycle where incremental functional updates are made in rapid progression [6]. Additionally, functionality defects and poor performance are tolerated over traditional enterprise development efforts.

Clay Shirky coined the term, situated software, as he witnessed how a new wave of his students approached software development. He observed that "situated software is in the small pieces category, with the following additional characteristic -- it is designed for use by a specific social group, rather than for a generic set of users" [12]. Thus, while enterprise design practices focus on building applications for the needs of the many and for years of use, situational applications are used by a few for a limited duration but addresses a pressing business need. Shirky [12] concludes that this type of development is:

> a kind of progress, not because situated software will replace other kinds of applications, but because it mostly won't. For all the value we get out of the current software ecosystem, it doesn't include getting an application built for a handful of users to use for a few months. Now, though, I think we're starting to see a new software niche, where communities get form-fit tools for very particular needs, tools that fail most previous test of design quality or success, but which nevertheless function well, because they are so well situated in the community that uses them.

The Internet has enabled the consumer to become aware of the possibility of situated software. Data is no longer purely controlled by enterprise IS organizations. Web 2.0 provides the framework for information sharing, customization and personalization encouraged by data feeds and web services concepts. The primary assumption is that content/information is readily available for consumption of the situational application developer. Two trends are influencing the need for situational business applications: first is the ability to exploit data residing outside of traditional enterprise data repositories (such as data warehouses) and second is availability of lightweight programming and deployment technologies to mix and publish aggregated data [13].

Situational applications are dynamically constructed for a transient need that are typically not covered by existing architectures such as Enterprise Information Integration (EII) or requires a lengthy IT department project to create [6]. Watt [15] further expands situational applications to include a social network aspect and lack of requirements to make them the highest quality possible. As IT departments develop a services-oriented architecture (SOA) infrastructure and exposes more information, the ability for business users to create situational software increases. This increase is dependent on the creation of visual representations for services, the ability to relate services to each other, and a common catalog for sharing [15]. Thus, a social network of business users emerges to collaborate on useful data feeds and situational applications.

Original situational applications were created using technology that was still primarily skills obtained by IS engineers such as shell and Perl scripts. The rise of SOA adoption is contributed to technology standardization, an easy to learn programming model, and a collaborative mentality [15]. Nestler [10] highlights that the SOA's loosely-coupled, standards-based, and protocol independent characteristics have not only aids in the proliferation of service-to-service use within an enterprise but also service-to-user communication. Service-to-user functionality, such as mashups, "can result in a more sufficient usage of knowledge specific for their domain and raise their productivity" [10, p. 551].

ENTERPRISE DATA MASHUPS

Simmen et al [13, p. 1172] "envision the corporate intranet steadily evolving into a platform of readily consumable resources, and lightweight integration technologies which can be exploited by business users to create 'enterprise mashups' in response to situational business needs. The lines between the intranet and the Web will progressively blur as enterprise mashups reach outside corporate firewall to exploit data and services on the Web". Several researchers offer similar but slightly varying definitions of mashups. A mashup is "a type of situational application that's comprised of two or more disparate components that have been connected to create a new integrated experience" [15]. Mashups assemble data, content or functionality from disparate sources and delivers the result in a Web interface where the source's owner has no participation in this activity [8]. Mashup platforms "aggregate and visualize heterogeneous web resources" and allow users to create an ad-hoc composition of data outputs of existing services or API's into a new representation and can be deployed within common Web browsers or as a standalone widget [10, p. 2]. Regardless of which definition is preferred, all mashups have the same three basic characteristics of combination, visualization, and integration [8].

Based on the wide range of potential solutions that can be addressed with mashups, categories and patterns have been identified [16, 11]. Wong and Hong [16, p. 35] state "mashups are inspiring and interesting because they are an expression of how users make use of existing systems in new and innovative ways". The way mashup are used can be grouped into categories such as aggregation, alternative UI & in-situ use, personalization, focused view of data, and real-time monitoring [16]. In Orginz's book, *Mashup Patterns: Design and Examples for the Modern Enterprise*, he highlights a series of patterns to "describe what is possible; you must use your unique focus to shape them into what is needed" [11, p. 39]. The examples of patterns include harvest, enhance, assemble, manage, and testing [11].

Mashups are placed into three distinct groupings: presentational layer mashups, process mashups, and data mashups [4]. Presentational layer mashups are the most common and easiest to develop as represented by the popular consumer web solutions showing a unified visualization. Process mashups consist of a complex arrangement of invoking business processes to take action on aggregated data [4]. Data mashups "provide content filtering, aggregation, and other data transformations to present reusable output as opportunistic assets with high functional and technical fitness, but with low QoS [quality of service] due to data quality and security issues" [5, p. 74]. Data mashups provide the task-specific programming approach to enable end users to select and aggregate relevant data into a conceptual space to address a problem [14]. For enterprises, being able to offer data services for mashups to utilize are simply only part of the solution. Enterprise data mashups need to have hubs similar to how business intelligence solutions have data warehouses. This common platform contains the five design principles including the enterprise mashup stack, emerging intermediaries, mass collaboration, lightweight resource composition, and perpetual betas. The enterprise mashup stack consists of four significant components including resources, application programming interfaces (APIs), widgets, and the mashup [6]. Watt [15] offers a similar mashup ecosystem to help frame the development tasks consisting of five layers, which are content (or information), data service interface, widget, mashup maker, and the mashup.

The enterprise data mashup adoption maturity level is at a nascent stage of development, but with the recent introduction of commercial vendor products, corporations are investing and applying first generation solutions [11]. A series of issues are being addressed with the second generation of commercial products. One issue with great attention is security. Web browser security policies enhancements to guard against malicious attacks and server-based identity and access authorization controls for access to corporate data are being addressed. The mashup inclusion of untrusted, or partially trusted, components enable malicious content that "can compromise the confidentiality and integrity of the user's session" [2, p. 83].

BUSINESS INTELLIGENCE MASHUPS

The rate of business intelligence requests is increasing over time [9]. Enterprise Information architectures are challenged to provide a solution for business users to aggregate structured and unstructured data in a single approach. Business intelligence solutions and search-derived approaches have provided limited success due to the limitations of uncertainty of implicit information from unstructured information and also access control [7]. Chowdhury states that data warehousing has the limitation that "static information did not provide end-users with the most recent information in order to perform business analyses on real-time or near

real-time data" [3, p. 22]. Mashups provide a real-time view of the data which today's business intelligence tools and data warehouses are not designed to perform [9].

The long tail of enterprise end-user business intelligence and data warehousing design requests is the ad-hoc need for data mart creation to aid in business problem analysis [6]. Often this data is located in non-traditional data silos, unstructured files (e.g. spreadsheets, presentations, email, HTML pages, etc.), or outside the corporate boundaries. By making all data available, business end-users can aggregate the data that they deem important, not just the data that was made available in data warehouse dimensions. Through web services, the cost to access the infrequently needed data is much lower than the traditional method of identifying, modeling, performing extract-transformation-load (ETL), and creating cubes. Finally, enterprises can manage the data mashups through a centralized repository where end-users upload new mashups, view previously used mashups, and share data source feeds.

Business intelligence mashups is the next iteration of ad-hoc query. Similar to the ETL process, the manipulating and combination and copying data from different sources for the purpose of data warehousing and reporting, business intelligence mashups leverage the virtual assembly (leaving data in source and combining in memory), data federation (formal process and formal definition), and the addition of informal, unstructured data to achieve similar results. Business Intelligence mashups mirror the repeatable behavior of Microsoft Excel manipulation but add an enterprise class performance adding to other business intelligence functionality. Ogrinz suggests a mashup pattern, called the splinter mashup, which "takes a single data source and make its contents available as separate new streams of data" [11, p. 240] to ease the assemble task for reporting. The burden of learning a complicated data extraction technique, as often performed with data warehouses, has been removed. This pattern can be used to create enterprise dashboards and reports [11].

Ideally, business intelligence mashups are saved to a centralized server for other colleagues to reuse and modify. This allows end users to self-service opportunities but still provides the required IS department controls of security and audit [14]. Business intelligence mashups provide the operational reporting solution that users require in a complex environment where the most accurate data is distributed across unconventional enterprise data repositories and local file systems.

MASHUP VENDOR PRODUCTS

Mashup development is composed of recently available Web technologies for both the visual representation and the data blending. Technologies such as Representational State Transfer (REST), Really Simple Syndication (RSS)/Atom as XML (extensible markup language) feeds in combination with reusable Web-based APIs uses data sources to create applications that differ from the original intent of the data. Generally, a Web browser is used to display the visual output. To ease the burden to users and developers, vendors have provided products and frameworks to assemble, manage, and share mashups. Popular examples of mashups include the free real estate information application from Zillow (www.zillow.com) and the crime mapping application from Wikinova called WikiCrimes (www.wikicrimes.org). ProgrammableWeb.com provides a directory of available mashups that a community of developers is sharing. Vendor products and APIs, also called mashup editors, are predominately available for the consumer Web development for little or no cost but restrict companies from using them to create a commercial service. Therefore, some vendors are have recently introduced products designed for enterprise development. Table 1, Mashup Vendor Products, lists vendor products for both the consumer Web and enterprise adoption along with a short description and website.

SUMMARY

More akin to Kimball's data warehousing approach of a bottom-up design, enterprise data mashups provide an end user perspective to data assembly and reporting. Using available mashup editors, business users could create situational applications that provide basic business intelligence functionality. While business intelligence mashups will not provide the advanced features of forecasting and multi-dimensional analysis, they are viable alternatives to respond to the corporate demand of timely decision-making. As unstructured data is used in conjunction with structured data repositories to provide operational perspectives for business users, the mashup approach will continue to emerge as an alternative solution to fill the gap between spreadsheet pivot tables and robust business intelligence and data warehousing solutions.

REFERENCES

1. Balasubramaniam, S., G. Lewis, S. Simanta, and D. Smith. (2008). Situated software: Concepts, motivation, technology, and the future. *IEEE Software*. November/December, 2008. p. 50-55.

2. Barth, A., C. Jackson, and J. C. Mitchell. (2009). Securing frame communication in browsers. *Communications Of The ACM. 52(6)*. p. 83-91.

3. Chowdhury, S. (2007). Trends in database tools and technologies. *The Business Review. 7(1)*. p. 20-25.

4. Fichter, D. and J. Wisniewski. (2009). The grow up so fast: Mashups in the enterprise. *Online*. 33(3). p. 54-57.

5. Gamble, M. T. and R. Gamble. (2008). Monoliths to mashups: Increasing opportunistic assets. *IEEE Software. November/December 2008. 71-79.*

6. Hoyer, V., K. Stanoesvka-Slabeva, T. Janner, and C. Schroth. (2008). Enterprise mashups: Design principles towards the long tail of user needs. *Proceedings of the 2008 IEEE International Conference on Services Computing (SCC2008)*. Honolulu, Hawaii, USA. August 7, 2008.

7. Jhingran, A. (2006). Enterprise information mashups: Integrating information, simply. *Proceedings of VLDB '06*, September 12-15, 2006, Seoul, Korea. p. 3-4.

8. Madsen, M. (2009). Mashups: Here comes the future again; With the consumerization of IT, web 2.0 is leaking into your organization from the edges. *Information Management*. New York: Jan. 1, 2009, 19(1), p. 56.

9. Nardi, B. (2003). *A Small Matter of Programming: Perspectives on End User Computing*. MIT Press.

10. Nester, T. (2008). Towards a mashup-driven end-user programming of SOA-based applications. *ACM iiWAS2008*, November 24-26, 2008, Linz, Austria. p. 551- 554.

11. Ogrinz, M. (2009). *Mashup Patterns: Designs and Examples for the Modern Enterprise.* Addison-Wesley, Boston, MA.

12. Shirky, C. (2004). Situated Software. March 30, 2004. http://www.shirky.com/writings/situated_software.html

13. Simmen, D., M. Altinel, V. Markl, S. Padmanabhan, and A. Singh. (2008) Damia: Data mashups for internet applications. *ACM SIGMOND '08*, June 9-12, 2008, Vancouver, BC, Canada. p. 1171- 1182.

14. TDWI. (2008). TDWI audio report: Mashups, Data and BI. (2008, September 17). Retrieved March 1, 2009, from http://www.tdwi.org/News/display.aspx?ID=9118

15. Watt, S. (2007). Mashups – The evolution of the SOA, Part 2: Situational applications and the mashup ecosystem. (2007, November 8). Retrieved March 1, 2009, from http://www.ibm.com/developerworks/webservices/library

16. Wong, J. and J. Hong. (2008). What do we "mashup" when we make mashups?. *Proceedings of WEUSE IV'08,* May 2008, Leipzig, Germany. p. 35-39.

Table 1. Mashup Vendor Products

Vendor and Product	**Description**	**Website**
Denodo Platform	Server-side data mashup platform that enables access to corporate data sources with extensive transformation and enrichment processing.	www.denodo.com
Google Mashup Editor	A text-based mashup development, test, management, and publishing environment.	editor.googlemashups.com
IBM Lotus Mashup Center	Mashup development and assembly environment that allows for a collaborative sharing of applications based on a common set functionality with security and governance.	www.ibm.com/software/info/mashup-center
InetSoft Style Intelligence	A complete business intelligence platform leveraging data access and mashups, visualization, enterprise reporting and business analytics.	www.inetsoft.com/products/styleIntelligence
Intel Mash Maker	Using a graphical design tool, users can assemble and configure data from disparate websites using a browser extension running on a client machine instead of a server.	mashmaker.intel.com
JackBe Presto	A visual environment that allows users to assemble, configure, and customize mashup using four products to manage and control applications along with security and governance infrastructure.	www.jackbe.com
Kapow Technologies' Enterprise Mashup Server	Focuses on information access, augmentation, and delivery of internal and external data including data not currently available by feed standards.	www.kapowtech.com
Microsoft Popfly	Publicly available mashup assembly and presentation environment that includes a community based approach for sharing applications.	www.popfly.com
Nexaweb Studio	An integrated development environment for mashup development, testing and debugging and provisioning. This product is specifically designed to separate the business logic from the presentation.	www.nexaweb.com
Oracle WebCenter Interaction	Centralizes the management of enterprise mashup infrastructure and shared capabilities such as security audit, administration, provisioning, and activity tracking.	www.oracle.com/products/middleware/user-interaction
TIBCO Spotfire Enterprise Analytics Platform	A highly visual and interactive mashup development environment geared towards business intelligence analytics.	spotfire.tibco.com
Yahoo Pipes	A visual editor for building mashups and executed on Yahoo servers with community based building, sharing, rating, and modifying capabilities	pipes.yahoo.com