

СЪЩНОСТ, ИНТЕГРАЦИЯ И АВТОМАТИЗАЦИЯ С МАШЪПИ

Гл. ас. д-р Галина Димитрова Момчева-Гърдева
Варненски свободен университет „Черноризец Храбър”

Резюме: В статията се разглеждат понятието „машъп” и свързаните с него технологии и понятия. Определят се области на приложения, свързани с машъпи, относно интегриране на функционалности на вече съществуващи приложения и автоматизация на дейности с тях.

Ключови думи: машъп, портлет.

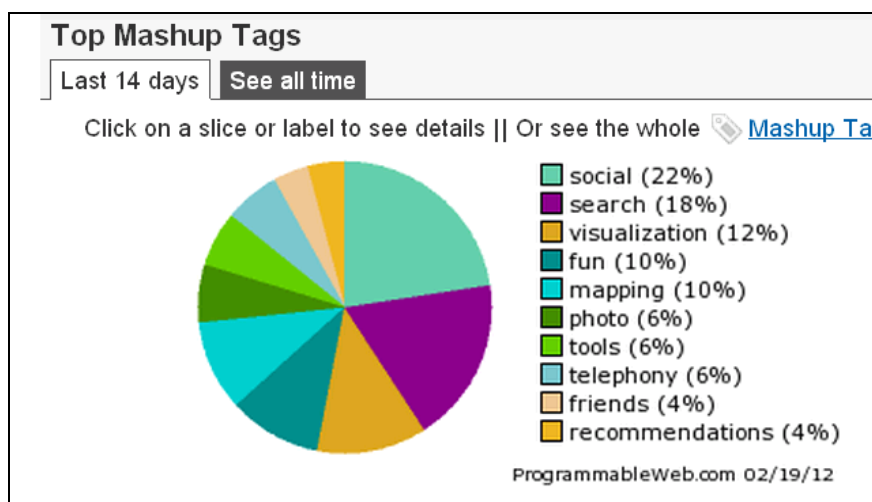
Abstract: The aim of the article is to explore the mashup topic and the technologies connected with it and their applications. Some benefits for integration and automatization with mashups are lined up.

Keywords: mashup, portlet.

Известно е, че думата „mashup” се е появила първоначално в музикалната индустрия, където машъпът представлява комбинация от две или повече песни за създаване на ново усещане. Типичното за този процес е, че вокалът на едната песен се комбинира с инструменталния фон на другата. Технологичната индустрия е разширила това определение, за да обхване и нов род приложения, който описва комбинацията от два или повече източника в интегриран сайт. Този метод на хибридиране на развитието може приблизително да се раздели на две отделни категории: консуматорски машъпи и машъпи за предприятия. Според Ogrinz сайтът

Housingmap.com, създаден от Paul Rademacher и комбиниращ информация от craigslist.com и Google Maps, може да се приеме за първия машъп (Ogrinz, 2009).

Една диаграма от двуседмични статистики в сайта на ProgrammableWeb¹, която може да се разгледа на фигура 1, показва разпределението на машъпи по видове в началото на 2012 г. Сред най-често използваните категории са тези за социални цели, за търсене и за визуализация.



Фиг. 1

В сайта ProgrammableWeb се дава възможност и за публикуване на връзки към нови, създадени от потребители, машъпи.

В публикация в сп. “PC Magazine”² от 2008 г. се прави опит да бъде даден адекватен превод на mashups на български език като „хибридни уеб

¹ <http://www.programmableweb.com>.

² <http://sagabg.net/PCMagazine/article-190.html>.

приложения”, но той остава без отзвук, както и самата технология на създаване на подобни приложения, макар че през 2009 г. е създаден IBM Mashup център.

Машъпите удовлетворяват следните пет критерии:

- Извличане на данни;
- Моделиране на източника;
- Изчистване на данни;
- Интеграция на данни;
- Визуализация на данни (Tuchinda, 2008).

Днес сред най-често срещаните машъпи в глобален план са тези, свързани с използването на географски карти (например за пазаруване). Машъпите могат да бъдат свързани както с продажби, така и с реклама на стоки и услуги. Това са машъпи, свързващи Amazon (Shanahan, 2007) с Flickr (Wilkinson, 2007).

Машъпи, свързани с географските карти, може да се създадат, като се използват средствата Yahoo Maps (Freedman, 2007) и Google Maps (Young, 2007).

При използването на Yahoo Maps може да се използват четири API-та: Ajax API, JS-Flash API, AS-Flash API и Flex API (Freedman, 2007) (Korhonen 2007).

Всички API-та за карти предоставят следните функционалности: изгледи, приближаване, панорамен изглед, геолокация, маркиране на място и навигация.

Стъпките за създаване на машъп са сравнително прости и важат за повечето машъпи (Feiler, 2008):

1. Изберете цел на машъпа.
2. Определете данните и ключовете.
3. Намерете достъп до данните.
4. Прегрупирайте.
5. Съставете потребителския интерфейс.
6. Имплементирайте машъпа.
7. Имплементирайте началната страница.

При създаването и използването на карти с Google Maps се използват маплети (Maplets). Това са малки приложения, които се изпълняват върху Google картите. Използвайки основната концепция за API, маpletите имат следните особености:

– Това е web приложение, което се показва в Google Maps на <http://maps.google.com>.

– Използва се JavaScript, който ни позволява да управляваме картата <http://maps.google.com>.

– Google използва `<iframe>`, за да вмъкне маплета в <http://maps.google.com>.

– Google кешира маплета на сървъра <http://gmodules.com> (Young, 2007).

Един машъп за създаване на интерактивна среда за изследване на място, въз основа на снимки от хора, е Mappr³. Raymond предлага и списък от машъпи с приложението им и употребяваните от тях средства (Raymond, 2008).

³ <http://www.mappr.com/>.

Осигуряване на достъпа до данните е един от основните проблеми при създаването на машъпи и този проблем е решен до голяма степен с предоставените API-та на сайтове като Google, Yahoo, Flickr, Amazon и др. Използването на API изисква регистрация, по време на която се предоставя регистрационен код. При използване на такова API трябва да се спазват правилата за употребата му. Те са най-често свързани с ограничения в ползването на данните, обикновено в броя транзакции (например Google имат ограничение от 100 на ден). Възможно е данните да са от източник, който имаме възможност да контролираме поради наличие на права върху него, най-често собственост, или да получим данни от източник, който не контролираме, но чиито данни са достъпни публично, без API. Пример за това е четенето на RSS, чиито данни са общодостъпни. В правилата за употреба на някои API-та се среща терминът „честно използване“, което означава „по правилата“.

При избора на средство като портлетите трябва да обърнем внимание на предимствата им като съвкупност, в пакет. В противен случай употребата на това сложно средство с налични ограничения за употреба не е оправдана.

Първият стандарт за портлети е Portlet 1.0 (JSR-168), а през 2008 г. се разработва Portlet 2.0 (JSR-286), който преодолява ограниченията на Portlet 1.0. Портлетите стават особено популярни при разработката на технологии за пренос – мостове (portlet bridges). Последните позволяват разработка на приложения с използването на съществуващите фреймуърци JSF, Wicket, Struts, без да се запознава разработчикът специално с технологията Java Portlet.

Предимството на използването на технологията Java portlet е не само възможността да се изгради бързо портал, но също така възможността да се координират (orchestration) услуги, които се създават на ниво потребителски интерфейс, позволявайки на бизнеса бързо да адаптира промените.

Един портал представлява съвкупност от малки web приложения, наречени портлети. С него се предоставят възможности като персонализация, идентификация и настройки на потребителя (пример за портал е iGoogle). Порталите събират (агрегират) съдържание от различни информационни източници.

Един портлет е компонента от интерфейса на потребителя, обезпечаваща със специфично съдържание, което може да бъде услуга или информация от съществуваща информационна система. Примери за портлети са тези, предоставящи информация за времето за определен град чрез Yahoo.

Основната отговорност на портлета е да генерира фрагмент с език за маркиране на данните като HTML, XML или WML.

Един портлет съдържа сървлети и JSP. В архитектурите, ориентирани към услуги (SOA), съвместната работа на услугите (оркестрацията) прави възможно да се разработват приложения от съществуващи услуги. Тъй като портлетите могат да взаимодействат един с друг на нивото на потребителския интерфейс (вътрешна портлетна комуникация – *inter-portlet communication*), те играят съществена роля в разработката на SOA приложения.

Портлетите може да бъдат използвани за създаване на машъпи чрез събиране (агрегиране) на данни от различни източници на данни и комбинирането им. Важна разлика между машъпите и портлетите е, че портлетите не комбинират информация от различни източници, за да

обезпечат с нова услуга. Вместо това те показват информация от различни източници в различни портлети. Приложение, което претърсва сайтовете за пазаруване за определяне на най-ниска цена, е също пример за машъп.

Разработката на Java портлети не е трудна с използването на фреймуърци като Spring и Hibernate.

Portlet 2.0 въвежда идеята за събития в портлети, въз основа на които се осъществява вътрешната комуникация между портлети.

Ajax също поддържа портлети. При използването на Ajax към сървъра се изпраща не цялата портална страница, а само необходимото запитване. Така се намалява мрежовият трафик и се осъществява асинхронна обработка.

Друго възможно улеснение при работа е използването на Dojo. А създаването на портлети в реално време може да се осъществи с помощта на Polling и Comet. Един специален вид портлет е iFrame портлетът. С него можем да вграждаме HTML страница на всяко място в портала.

Съществуват и други средства за създаване на портлети. Това са IBM WebSphere Portlet Factory, SAS 9.3 Intelligence Platform и др. Един свободен за употреба портален сървър е GateIn Portal⁴. С него може да се сливат портални проекти за JBoss Portal и eXo Portal.

За създаване на машъпи се използват и платформи, които играят ролята на бизнес интегратори, като Convertigo⁵. Чрез него се създават динамични

⁴ <http://www.jboss.org/gatein>.

⁵ <http://www.convertigo.com>.

приложения от вече съществуващи web приложения, web услуги и бази данни.

Освен средствата, които разширяват съществуващи функционалности с машъпи, съществуват и средства за реално програмиране, които могат да бъдат използвани за автоматизиране на някои дейности, като филтриране, извличане, сравнение, уведомление – подходящи за работа дори за неспециалисти в разработката на софтуер. Това са всъщност средства от категорията за визуално програмиране и проектиране. Сред най-разпространените са две средства от Yahoo: Yahoo Pipes⁶ е средство за създаване на машъпи, а dapper⁷ е средство от групата на scraper-ите, което представлява преглеждане на страницата и обработка на съдържанието чрез скрипт.

Друго средство за създаване на машъпи с визуални средства, предоставяно за употреба от Майкрософт, е Microsoft Popfly.

За научни изследвания: Bibliogo⁸ е приложение за статии в списания, което ще ви подпомогне да следите какво се случва в области, свързани с работата ви. Bibliogo ви уведомява за нови публикации, осигурява достъп до тях и съдейства за съвместната работа и споделянето.

Друго по-ново средство като ifttt⁹ (**if this than that**) позволява да се автоматизират процеси от ежедневието. Например с изпращане на алармиращи съобщения, при прогноза за лошо време или при публикуване на снимка в някоя от социалните мрежи. Автоматизирането на ежедневието в случая може да се изпълни благодарение на свързаността на данните в

⁶ <http://pipes.yahoo.com/pipes/>.

⁷ <http://dapper.net/>.

⁸ <http://www.popfly.ms/2011/12/02/bibliogo-mashup-for-scientific-research/>.

⁹ <http://ifttt.com>.

системите, като социалните мрежи, от една страна, а от друга, благодарение на лесните начини за свързване на функционалности на софтуер на приложно ниво. При тези средства потребителят не използва определени програмни средства, а технологиите на приложно ниво. Това обаче, което трябва да знае, е какво иска. Отговорът на този прост въпрос за ifttt се крие зад описанието на възникналия бизнес процес, т.е. потребителят трябва да знае какво иска и да може да определи компонентите на ifttt събитието: тригер (this) и действие (that). Нещо повече, резултатите от това действие, записани с номер в сайта на ifttt, се документират като рецепта (receipt) и може да бъдат използвани свободно от всички желаещи.

Използвана литература

- Feiler, J.* How to do everything with Web 2.0 Mashups, McGraw-Hill, 2008.
- Freedman, C.* Yahoo!® Maps Mashups, Wiley, 2007.
- Korhonen, C. etc.* Creating Mashups with Adobe Flex and AIR, Friendsof, 2008.
- Ogrinz, M.* Mashup Patterns Designs and Examples for the Modern Enterprise, Addison Wesley, 2009.
- Sarin, A.* Portlets in Action, Manning, 2012.
- Shanahan, F.* Amazon.com® Mashups, Wiley, 2007.
- Tuchinda, R.* Building Mashups By Example, IUI, 2008.
- Wilkinson, D.* Flickr™ Mashups, Wiley, 2007.
- Yee, R.* Pro Web 2.0 Mashups Remixing Data and Web Services, Apress, 2008.
- Young, M.* Google Maps Mashups with Google Mapplets, Apress, 2007.