

<http://ansinet.com/itj>

ITJ

ISSN 1812-5638

INFORMATION TECHNOLOGY JOURNAL

ANSI*net*

Asian Network for Scientific Information
308 Lasani Town, Sargodha Road, Faisalabad - Pakistan

Study of Reuse Mashup Technology Based on Using Frequency

¹Zhiyuan Shi, ¹Volker Gruhn, ²Yuwan Gu and ²Yuqiang Sun

¹Institut für Informatik Lehrstuhl für Angewandte Telematik/E-Business, Universität Leipzig,
 Leipzig, 04103, Germany

²Changzhou University International Institute of Ubiquitous Computing, Jiangsu,
 Changzhou, 213164, China

Abstract: The features of mashup technology architecture and application technologies of data cache module and warning apparatus module are analyzed. Reuse mashup technology architecture based on using frequency is proposed. Data cache strategy and the determination of information searching the dominating keywords attribute weight is adjusted from the attribute angle of using frequency. Reuse and mashup is implemented using warning apparatus and multi-channel distributed search. Its application effect is explained with example.

Key words: Mashup, data cache, warning apparatus, multi-channel distributed search

MASHUP ARCHITECTURE

Mashup is that creating fire-new web service using content searched from outer cataloging data sources. Mashup possesses the characteristics of reusability and lightweight etc. comparing with traditional network (Di Lorenzo *et al.*, 2009). According to the distinctness of calling data method, Mashup approximately can be divided into two types (Ngu *et al.*, 2010; Bouguettaya *et al.*, 2010): calling other website API (namely application programming interface), calling exterior Web feeds. Moreover it is not difficult to find that two kinds of primary Mashup modes respectively correspond to disposal of data service and software service corresponding to class of content. Mashup is a kind of application form on the Web 2.0 big idea framework, a kind of new software development mode. Mashup architecture is composed of three parts (Daniel *et al.*, 2010; Yu *et al.*, 2009; Auinger *et al.*, 2009) which is detachment on logic and physics as follow Fig. 1, three parts are, respectively API/Content Supplier; MASHUP Site and Terminal Browser. API/Content Supplier provides integration content that is need for Mashup Website, commonly content passes Web protocol (such as RES, web Services and RSS/A TOM), programing and providing using API form. Mashup site is namely Mashup local place, Mashup server encapsulates resources and services of gained or their own to standard package and manages these components, at the same time responds to application of open calling for resources, services. Web browser of

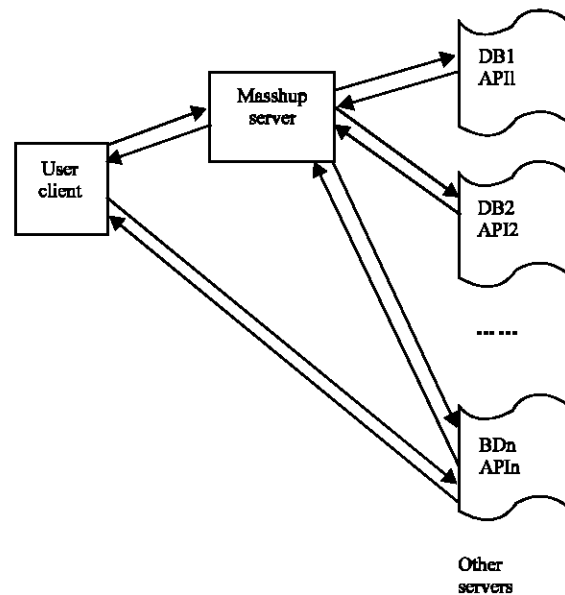


Fig. 1: Mashup system architecture

client server is place of occurring user interaction, Mashup user selects related resources, then it is fused to own portal or other application system and presenting Mashup application results on the mode of graphiel through the client browser. And that according to the different place of data processing it is divided into sever Mashup and client Mashup (Daniel and Matera, 2009; Shi and Zhu, 2010).

Sever Mashup (Lopez *et al.*, 2009) mostly writes and runs Web programme to implement Mashup through

Sever dynamic webpage programming technology (such as ASP, JSP, PHP etc). After the request of Mashup webpage arrives at server, programme code is embedded and called, performing Mashup task on server. Accomplished results pass to browser in the format of static pages, then display.

For the moment, client Mashup (Yu and Woodard, 2009) programme mostly dynamic generates and updates Mashup webpage using AJAX technology; website embeds JS code for Mashup webpage calling on webpage. Mashup webpage as client dynamic webpage, after arriving at terminal browser, embedded or called Javascript programme starts to work on the browser of supported Javascript, directly calling outer cataloging data sources.

At present, mass Mashup applications implement data integration or service integration through open, easy to understand and describe Web service, even merely carrying through simple page layout integration based on visual experience (such as portal technology). Its architecture is expressed using 3-Tier hierarchical model. The presentation layer Mashup is mostly for portals and page personalized custom; the service layer Mashup mostly aims at software service (SaaS) (Jung 2012), Such as Mashup of commodity information and Googlemap is implemented and that data layer Mashup mostly aims at data service (DaaS) (De Vrieze *et al.*, 2011), Such as data mining of database based on more network authorized.

Mashup technology is general name of Mashup application programme involved web service technology, open integration technology etc technologies. On the software engineering, software structure is replaced by architecture. A word of architecture is more softening than structure; a new characteristic is appeared on the net age of software engineering.

APPLICATION OF WAMING APPARATUS

Warning apparatus (Cappiello *et al.*, 2010): Using intelligent monitoring agent to detect resource change and passing on appropriate notification.

Price will be changed, time will be elapsed and new product will be issued, employee will be joined and leaved. Management can not depend on outdated information, so it is important to keep consistent with all these change. It is a pity; the range of potential data can be monitored is commonly very large. Now that the scale of the problem is large, well then company must restrict they concerned field. This is long tail on another environment: 20% projects that are likely to occurring change obtain most of the attention. Residual 80% projects that are not likely to occurring change will be ignored, however, if it is changed, they are important too. Mashup has no use for adding value through regularly

interact with end user. Through scheduling intelligent agent can automatically monitor all kinds of states and immediately inform sudden change to other objects through e-mail, mobile short message or other channels. The virtual observers are implemented through carrying through API enabling for expected resources and establishing a process that examines content changed or not. Warning apparatus can implement disposal of 20/80 problem of long tail on tens of thousands of resources and that not adding too much labor cost.

Warning apparatus commonly tracks two data points: Currently read and the moat new during historical read (may be inexistence it shows lacking of forestall state). If the warp of two values is not during presetting value, then it will be triggered a warning. Warning apparatus can implement monitoring and notification service with many other modes. For example, if the content of a specific resource changes it can trigger other implementation of mashup.

REUSE MASHUP TECHNOLOGY BASED ON USING FREQUENCY

In the paper, reuse Mashup technology architecture based on using frequency (Fig. 2) adds data cache module, warning apparatus module and multi-channel distributed search module on the basic of original Mashup technology architecture.

Data cache module: Searched information is saved on database as cache; cache is directly used on the next time if sending the same request it improves performance. On the every time of obtaining searched information, corresponding processing time is saved. The time is one of parameters of service matching. Searched information is called at regular intervals, insuring to get exact information, corresponding processing time is saved. The time is one of parameters of service matching. There is counter on the data cache module, the role of counter is that computing the times of directly using information which is saved on data cache. The times arrive at a certain threshold and the information is not readjusted on a certain period of time, then it will trigger warning apparatus, sending a warning to warning apparatus, warning it that the information need to search again, in order to obtaining the latest and the most accurate information.

Warning apparatus module: Giving a warning for information which is often used and is not readjusted on a certain period of time, warning Mashup server to search again for the information, the information which is searched again may be the same with the information which is existed on the data cache, although it wastes time, the most important is that it ensures the accuracy of information.

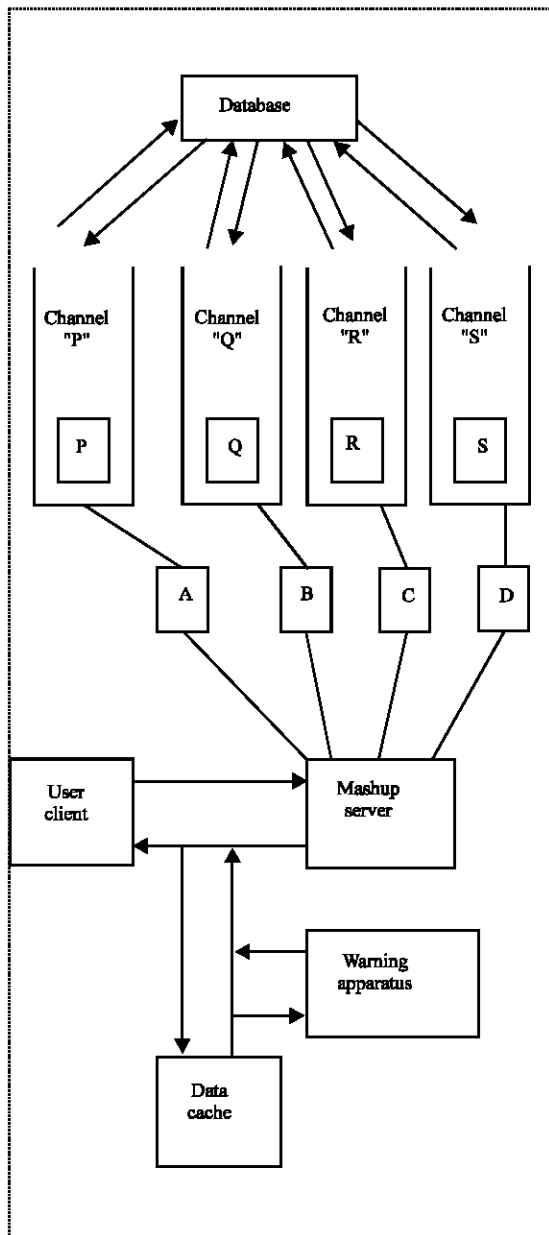


Fig. 2: Reuse mashup technology architecture based on using frequency

Multi-channel distributed search module: taking over keywords that is inputted by client, then carrying through semantic analysis for request of client, semantic description of Mashup is implemented through micro content of self description in common and the predicate description is the most common way of the self description of resources. Data service is represented by connotation (namely attribute) and extension (namely value). Mashup request is expanded according to value. A value is searched on one channel, that is to say all the values of the keywords can be searched at the same time.

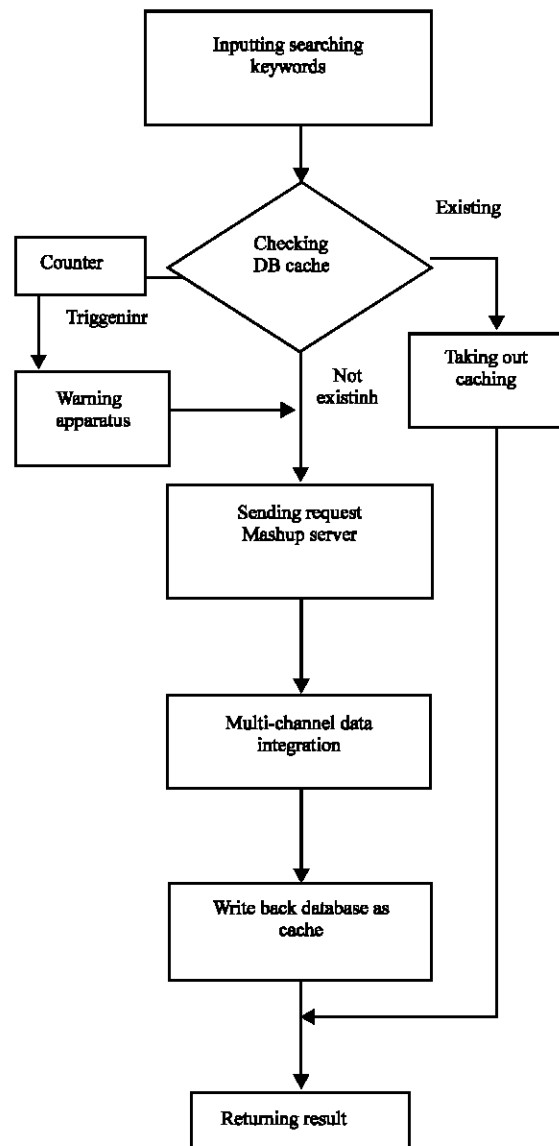


Fig. 3: Flow chart according to Figure 2

Returning searched information of all channels to user, user can obtain all aspects of information.

For example: If the keywords are inputted by client is changzhou, information is searched according to its values of weather, school, restaurant, bus, such as information of changzhou weather is searched by channel P, information of changzhou school is searched by channel Q, information of changzhou restaurant is searched by channel R, information of changzhou bus is searched by channel S. Finally, returning all the information to client.

The flow chart of reuse mashup technology architecture based on using frequency as follow Fig. 3.

Firstly, keywords are inputted by client on client, then database cache is examined, judging that searched information exists in data cache or not, if it exists in data cache, searched information is directly taken out from data cache, returning the result to client; if it not exists in data cache, sending request to Mashup server, Mashup server carries through multi channel data integration according to receiving searched keywords and semantic analysis, matching, as well as integrated result is written to database, saving to data cache, finally, returning the result to client. In order to ensuring data accuracy on the data cache, content on database is carried through renewal over a period of time, renewal also need to set a threshold value, counting the times of directly using information which is saved on data cache using counter, when a cache content arrives at the threshold value, warning apparatus warns Mashup server to search again for the information, then reintegration data is written to cache.

CONCLUSION

The study of Reuse Mashup Technology Based on Using Frequency provides a sort of improved architecture of reuse and mashup, moreover, the validity and practicability of the mode is analyzed in theory. Its implementation of technology and quantitative analysis still needs farther study and validation.

ACKNOWLEDGMENT

Supported by Natural Science Fund in JiangSu (BK2009535) and Jiangsu Province ordinary university innovative research project (CXZZ13_0691).

REFERENCES

- Auinger, A., M. Ebner and D. Nedbal, 2009. Mixing content and endless collaboration-MashUps: Towards future personal learning environments. *Universal Access Hum. Comput. Interact. Appl. Serv.*, 5616: 14-23.
- Bouguettaya, A., S. Nepal, W. Sherchan, Z. Xuan and J. Wu *et al.*, 2010. End-to-end service support for mashups. *IEEE Trans. Serv. Comput.*, 3: 250-263.
- Cappiello, C., F. Daniel, M. Matera and C. Pautasso, 2010. Information quality in mashups. *IEEE Int. Comput.*, 4: 14-22.
- Daniel, F. and M. Matera, 2009. Turning web applications into mashup components: Issues, models and solutions. *Proceedings of the 9th International Conference on Web Engineering*, June 24-26, 2009, San Sebastian, Spain, pp: 45-60.
- Daniel, F., S. Soi and F. Casati, 2010. Chapter 5: From mashup technologies to universal integration: Search computing the imperative way. *Search Comput.*, 5950: 72-93.
- De Vrieze, P., L. Xu, A. Bouguettaya, J. Yang and J. Chen, 2011. Building enterprise mashups. *Future Gener. Comput. Syst.*, 5: 637-642.
- Di Lorenzo, G., H. Hacid, H.Y. Paik and B. Benatallah, 2009. Data integration in mashups. *Sigmod Rec.*, 38: 59-66.
- Jung, J.J., 2012. ContextGrid: A contextual mashup-based collaborative browsing system. *Inform. Syst. Frontiers*, 14: 953-961.
- Lopez, J., F. Bellas, A. Pan and P. Montoto, 2009. A component-based approach for engineering enterprise mashups. *Proceedings of the 9th International Conference, ICWE*, June 24-26, 2009, San Sebastian, Spain, pp: 30-44.
- Ngu, A.H.H., M.P. Carlson, Q.Z. Sheng and H.Y. Paik, 2010. Semantic-based mashup of composite applications. *IEEE Trans. Services Comput.*, 3: 2-15.
- Shi, Z.Y. and M.F. Zhu, 2010. Optimization of semantic MASHUP architecture applied in microlearning environment. *J. Jiangsu Univ. (Nat. Sci. Edn.)*, 31: 339-342.
- Yu, S. and C.J. Woodard, 2009. Innovation in the programmable web: Characterizing the mashup ecosystem. *Service-Oriented Computing-ICSOC 2008 Workshops*, pp: 136-147.
- Yu, Y.H., J.H. Kim, K. Shin and G.S. Jo, 2009. Recommendation system using location-based ontology on wireless internet: An example of collective intelligence by using mashup applications. *Exp. Syst. Appl.*, 36: 11675-11681.