Using MongoDB and Mongoose with a MEAN Stack Implementation of the NEXUS-PORTAL-DOORS System

Jason J. Liu, Adam Craig, and Carl Taswell

Introduction

For researchers, primary research is only the first stepping stone to proving or disproving a hypothesis. One successful experiment proves little to nothing in the grand scheme of the scientific community. However, the weakness of a single experiment is covered by the power of the meta-analysis, usually completed by a third party. Meta-analysis requires researchers to analyze results from muliple primary articles and assess whether or not the result is accurate, show if the hypothesis presented are proven or disproven, and provide a degree of error. However, despite the necessity of meta-analyses, it continues to be time consuming and ineffective at connecting with all points of data. The miniscule amount of secondary articles pale in comparison to the literal thousands of primary articles which are published each day. In order to take advantage of all the new information which is provided and to create more effective secondary analyses, a semantic web solution provides the necessary tools in order to take full statistical advantage that meta-analysis offers.

One such implementation of a meta-analysis system is the NEXUS-PORTAL-DOORS (NPD) system. In order to support PORTAL registries and DOORS directories, a database is necessary in order to support and store the data. In order to take full advantage of a tried and true framework, MEAN stack serves as the ideal system by thoroughly integrating client-side, server-side, and database processes.

Methods

For this new NEXUS-PORTAL-DOORS (NPD) system implementation, MEAN (MongoDB, Express, Angular, Node.js) stack is the most updated and contributed fullstack javascript framework. Both backend and front end will rely heavily on node.js, an open-source runtime environment which enables scalable web applications and servers. Used with Express and other packages, node.js enables the user to organize a web application into a MVC (model-view-controller) architecture. In accordance with the "M" from MEAN stack, MongoDB serves as the database for the NPD. As a noSQL database, MongoDB relies on collecetions and documents in contrast to tables and rows. However, MongoDB offers no implementation of queries or updates to the database. Mongoose, a popular object relational mapper for MongoDB, enables connection to a MongoDB database from javascript, allowing model abstraction, large scale queries, data validation, and more. It also enables the user to create schemas, which help generate a structured database.

In the NPDS system, Problem Oriented Registry of Tags And Labels (PORTAL) registers resource labels and tags, inserting or updating them into the database while the Domain Ontology Oriented Resource System (DOORS) publishes resource locations and descriptions with mapping of labels to locations for the semantic web, retrieving stored data from the database. Both PORTAl and DOORS can be optimized for faster read/write if organized in a heirarchical architecture.

Timeline:

- Week 1-2: Research on the semantic web (reading primary and secondary literature, books, videos, articles, etc)
- Week 3-4: Start looking at ideal databases and software for semantic web
- Week 5-6: Work with co-authors to decide on type of database, database, organization, and its role in the application
- Week 7-8: Develop template for a NPDS MVC web application generator
- Week 9-10: Work to assemble NPDS web application on appropriate servers

August:

- Week 1: Settle on specifications for template for NPDS web application
- Week 2: Begin assembling simple template as per components list
- Week 3: Finsih assembling template with all components; begin generating webpages for NPDS read registrar, NPDS write registrar, NEXUS read-only directory, PORTAL read-only registry, DOORS read-only directory
- Week 4: Individualize databases for each generated ; bug fixes

Results

Discussion

discussion text here

Conclusion

conclusion text here

Acknowledgments

Do not acknowledge co-authors; only acknowledge persons who have contributed to or otherwise supported the study in some way but did not get recognized as a co-author. Acknowledgments are also used to identify any outside funding sources other than the affiliated institution identified for authors.

References

1. Barrasa Rodriguez, J., Corcho, . and Gmez-Prez, A. R2O, an extensible and semantically based database-to-ontology mapping language Springer-Verlag, 2004
2. Berners-Lee, T., Hendler, J., Lassila, O. and others The semantic web Scientific american, New York, NY, USA:, 2001, Vol. 284(5), pp. 28-37
3. Dickey, J. Write modern web apps with the MEAN stack: Mongo, Express, AngularJS, and Node. js Pearson Education, 2014
4. MySQL, A. MySQL 2001
5. Pan, Z. and Heflin, J. Dldb: Extending relational databases to support semantic web queries DTIC Document, DTIC Document, 2004
6. Spanos, D.-E., Stavrou, P. and Mitrou, N. Bringing relational databases into the semantic web: A survey Semantic Web, IOS Press, 2012, Vol. 3(2), pp. 169-209
7. Stojanovic, L., Stojanovic, N. and Volz, R. Migrating data-intensive web sites into the semantic web Proceedings of the 2002 ACM symposium on Applied computing 2002, pp. 1100-1107
8. Suehring, S. MySQL bible John Wiley and Sons, Inc., 2002
9. Taswell, C. DOORS to the semantic web and grid with a PORTAL for biomedical computing IEEE Transactions on Information Technology in Biomedicine, IEEE, 2008, Vol. 12(2), pp. 191-204
10. Taswell, C. Portals and doors for the semantic web and grid Google Patents, 2010
11. Taswell, C. A distributed infrastructure for metadata about metadata: The HDMM architectural style and PORTAL-DOORS system Future Internet, Molecular Diversity Preservation International, 2010, Vol. 2(2), pp. 156-189