iJRASET
International Journal For Research in Applied Science and Engineering Technology



INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 4 **Issue:** VIII **Month of publication:** August 2016

DOI:

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)

XML Entity Architecture for Efficient Software Integration

Amol Patwardhan[1], Rahul Patwardhan[2]
Department of Development and Operations, Asset Mark Inc

Abstract— This paper proposed xml entities based architectural implementation to improve integration between multiple third party vendor software systems with incompatible xml schema. The xml entity architecture implementation showed that the lines of code change required for mapping the schema between in house software and three other vendor schema, decreased by 5.2%, indicating an improvement in quality. The schema mapping development time decreased by 3.8% and overall release time decreased by 5.3%, indicating an improvement in productivity. The proposed technique proved that using xml entities and XSLT transforms is more efficient in terms of coding effort and deployment complexity when compared to mapping the schema using object oriented scripting language such as C#.
Keywords— XML, XSLT, Schema Mapping, Business Entities, Architecture, Software Integration.

I. INTRODUCTION

Business logic layer consists of classes representing business entities. The business entities map to a one or more tables and view. In the model view controller architecture, the model is the equivalent of business entities. Sometimes there is a business need to integrate the software with another 3rd party vendor system. Such an integration requires mapping the vendor schema with the business entities of the host and in house software system. A limitation of class level representation for business entities is that the mapping process involves additional processing overhead. This paper proposes the xml business entities based architecture to solve the problem of integration overhead with 3rd party vendor software systems. We used 3 internal projects to compare baseline and new method implementation data. Each project had been implemented in asp.net, C# using N-tier architecture. The business entities were represented by C# classes. The mapping code was implemented using C#.

II. RELATED WORK

The Microsoft Application Architecture Guide [1] described the scenarios when xml business entities can be used. It recommended using xml entities when the presentation layer requires it or when there is specific logic built around the xml element. The guide also recommends using custom xml objects when the system to system message exchange is being done using xml. It also warned against possible high memory usage while processing huge amount of in memory-xml data. We hypothesize that the xml based business entities can be used more extensively for a broader range of scenarios and even as a replacement for custom objects defined by classes. We believe that the implementation advantages exceed the performance concerns.

III. METHODOLOGY

To measure the baseline data, we used following metrics:

A. Number of project files modified.
B. Number of defects fixed per release.
C. Development time to implement the mapping.
D. Development time to implement features and changes to schema.
E. Release time for fixing defects.
F. Deployed lines of code.
G. Memory footprint of data.
H. Deserialization and Serialization time.

First, we measured the metric for an existing project in the solution. The internal project code was DT. The project had been implemented using ASP.NET C#, Web forms and SQL 2008. We obtained the metrics from existing TFS change-sets, shelves, check-ins, review emails, project release documents, project plan and quality assurance tracking system (QA). After the baseline data was established we used the proposed xml entities based architecture for the next implementation of vendor integration modules that involved mapping third party xml schema with the in-house xml schema. The same set of metrics were tracked to

©IJRASET: All Rights are Reserved

Volume 4 Issue VIII, August 2016
ISSN: 2321-9653
International Journal for Research in Applied Science & Engineering Technology (IJRASET)

obtain a comparison with the baseline. The empirical data was obtained over a period of 14 months and included 4 releases and 8 maintenance release cycles. The data was aggregated for 2 new projects (internal project code R1 and CUDL) that implemented the proposed architecture and maintenance cycles for the existing project (internal code DT) that used legacy mapping code.

IV. RESULT

The number of files modified per release decreased when the proposed xml based entities architecture was used in the project as compared to the implementation that used class based business entities representation, as shown in Fig. 1.

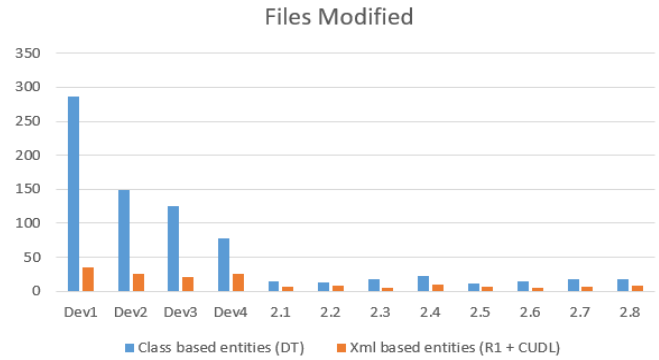


Fig. 1 Number of files modified per release

Fig. 2 shows that the number of defects fixed per release increased when the xml based entities architecture was used. This indicates an improvement in productivity per available development resource (programmers, quality assurance staff).

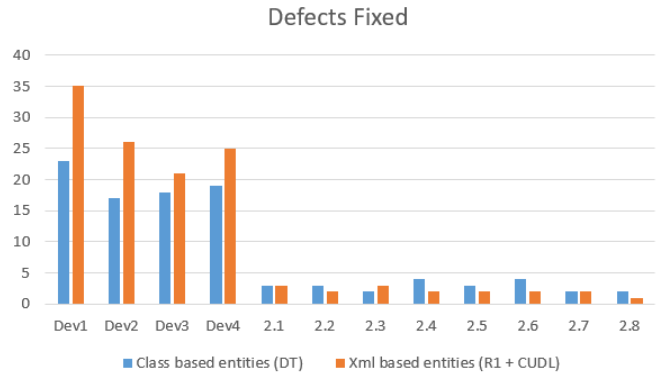


Fig. 2 Number of defects fixed per release

Fig. 3 shows that the schema mapping duration per release decreased when the xml based entities architecture was used. This indicates that the implementation effort reduced and contributed in overall productivity of the programmer.

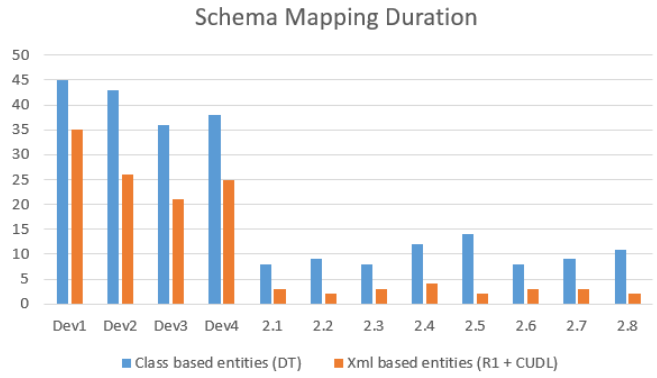


Fig. 3 Schema mapping duration in person-days per release

©IJRASET: All Rights are Reserved

Volume 4 Issue VIII, August 2016
ISSN: 2321-9653
International Journal for Research in Applied Science & Engineering Technology (IJRASET)

Fig. 4 shows that the code change effort in person-days per release decreased when the xml based entities architecture was used. This indicates an improvement in productivity per available development resource (programmers, quality assurance staff).

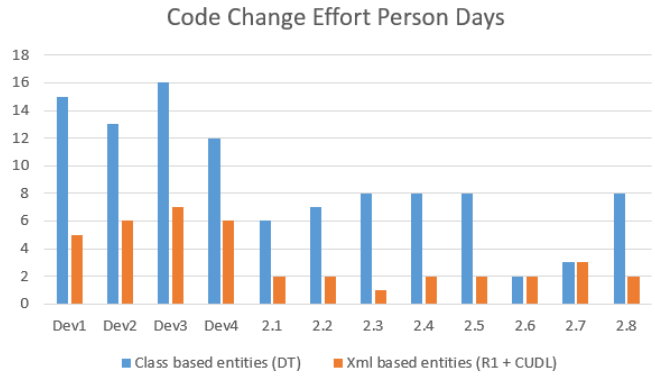


Fig. 4 Code change effort in person-days per release

Fig. 5 shows that the maintenance release duration decreased when the xml based entities architecture was used. This indicates an improvement in efficiency of release management process when the proposed architecture was used. The decrease in release duration between process that used legacy architecture and the new xml based entities architecture was 5.3%.

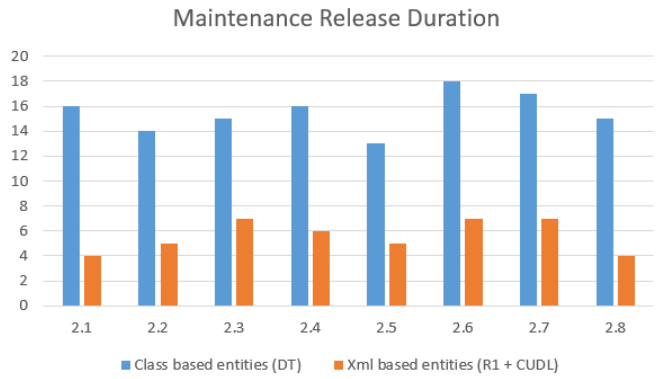


Fig. 5 Maintenance release duration in person-days per release

Fig. 6 shows that the number of deployed lines of code per release increased when the xml based entities architecture was used. This indicates a reduction in code churn and thus decrease in the amount of defect regression and an improvement in the quality of the code. The xml based entities architecture also required very few lines of code changes and was highly flexible in terms of build and deploy. It also reduced the lines of code that had to be code reviewed.

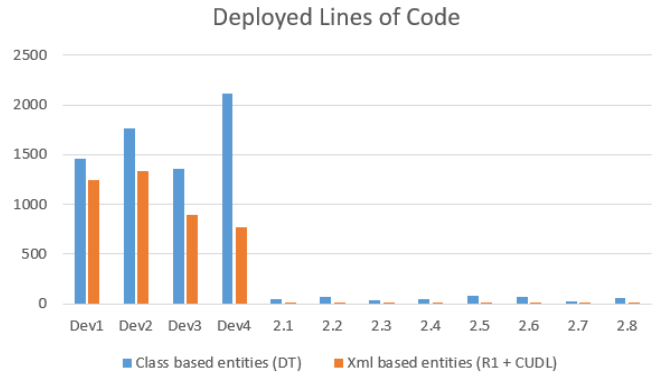


Fig. 6 Deployed lines of code per release

Fig. 7 shows that the available memory in megabytes (MB) per release increased when the xml based entities architecture was used.

©IJRASET: All Rights are Reserved

Volume 4 Issue VIII, August 2016
ISSN: 2321-9653
International Journal for Research in Applied Science & Engineering Technology (IJRASET)

This indicates that the memory required by the xml entities based architecture was higher than the class based business entities representation. But this increase was only 0.48% and could be easily handled by available memory resources on the production as well as development servers.

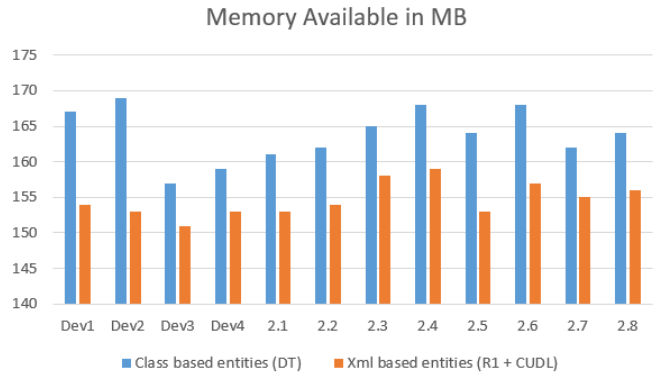


Fig. 7 Memory available in MB per release

Fig. 8 shows that the processing time for mapping the schema per release decreased when the xml based entities architecture was used. This indicates an improvement in processing performance and faster software response. The improvement was 0.875% and was significant in terms of business-to-business (B2B) system communication over the network.



Fig. 8 processing time in milli-seconds per release

V. CONCLUSIONS

Enterprise level software development requires integration with third party APIs, services and software systems over the internet. The proposed xml entities based software implementation improves the integration process in terms of coding effort and release time when compared to integration with software containing classes to represent business entities. The change management and bug fix process also greatly improves in terms of coding and testing effort when xml based entities are mapped to third party vendor software schema using XSLT transforms. The xml based entities architecture also provides a foundation to extend the code with minimal change. As a future scope the feasibility of the xml entities architecture in document generation and code comprehension must be investigated.

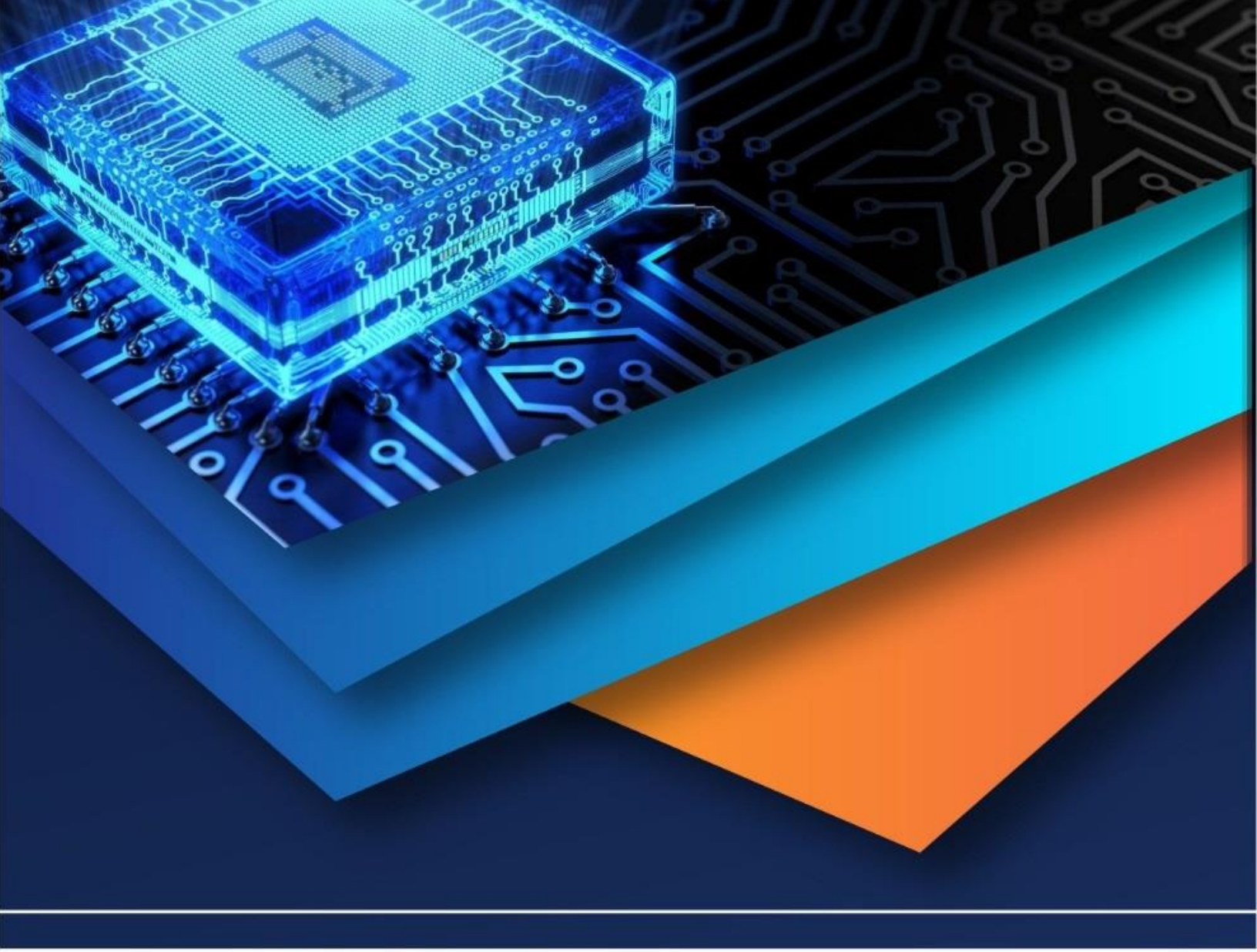
REFERENCES

[1] (2009) Microsoft Application Architecture Guide, 2nd Edition. [Online]. Available: https://msdn.microsoft.com/en-us/library/ee658106.aspx
[2] P. D. Manuel and J. AlGhamdi, “A Data-Centric Design for N-Tier Architecture,” Information Sciences, vol. 150, no. 34, April. 2001, pp. 195-206, doi:10.1016/S00200255(02)00377-8.
[3] A. Aleti, B. Buhnova, L. Grunske, A. Koziolek and I. Meedeniya, “Software Architecture Optimization Methods: A Systematic Literature Review," IEEE Transactions on Software Engineering, vol. 39, no. 5, pp. 658-683, May. 2013. doi: 10.1109/TSE.2012.64.
[4] A. S. Patwardhan and G. M. Knapp, “Affect Intensity Estimation Using Multiple Modalities,” Florida Artificial Intelligence Research Society Conference, May. 2014.
[5] A. S. Patwardhan and G. M. Knapp, “Multimodal Affect Analysis for Product Feedback Assessment,” IIE Annual Conference. Proceedings. Institute of Industrial Engineers-Publisher, 2013.
[6] A. S. Patwardhan, “An Architecture for Adaptive Real Time Communication with Embedded Devices,” LSU, 2006.

©IJRASET: All Rights are Reserved

[7] M. Shaw and P. Clements, "The Golden Age of Software Architecture," IEEE Software, vol. 23, no. 2, pp. 31-39, March. 2006. doi: 10.1109/MS.2006.58.
[8] P. Kruchten, H. Obbink and J. Stafford, "The Past, Present, and Future for Software Architecture," IEEE Software, vol. 23, no. 2, pp. 22-30, March. 2006. doi: 10.1109/MS.2006.59.
[9] R. L. Nord and J. E. Tomayko, "Software ArchitectureCentric Methods and Agile Development," IEEE Software, vol. 23, no. 2, pp. 47-53, March. 2006. doi: 10.1109/MS.2006.54.
[10] P. Y. Reyes-Delgado, M. Mora, H. A. Duran-Limon, L. C. Rodríguez-Martínez, R. V. O'Connor, R. MendozaGonzalez, "The Strengths and Weaknesses of Software architecture design in the RUP, MSF, MBASE and RUPSOA methodologies: A conceptual review," Computer Standards & Interfaces, vol. 47, pp. 24-41, August. 2016. doi: 10.1016/j.csi.2016.02.005.
[11] L. Grunske, A. Aleti, "Quality Optimisation of Software Architectures and Design Specifications," Journal of Systems and Software, vol. 86, no. 10, pp. 2465-2466, October. 2013. doi: 10.1016/j.jss.2013.06.001.
[12] A. S. Patwardhan and R. S. Patwardhan, "Self-Contained Cross-Cutting Pipeline Software Architecture," International Research Journal of Engineering and Technology (IRJET), vol. 3, no. 5, May. 2016.
[13] A. S. Patwardhan and G. M. Knapp, "Aggressive Action and Anger Detection from Multiple Modalities using Kinect", submitted to
[14] ACM Transactions on Intelligent Systems and Technology (ACM TIST) (in review).
[15] A. S. Patwardhan and G. M. Knapp, "EmoFit: Affect Monitoring System for Sedentary Jobs," preprint, arXiv.org, 2016.
[16] A. S. Patwardhan, "Embracing Agile methodology during DevOps Developer Internship Program", IEEE Software (in review), 2016.
[17] A. S. Patwardhan, "Edge Based Grid Super-Imposition for Crowd Emotion Recognition", International Research Journal of
[18] Engineering and Technology (IRJET), May. 2010.
[19] A. S. Patwardhan, "Human Activity Recognition Using Temporal Frame Decision Rule Extraction", International Research Journal
[20] of Engineering and Technology (IRJET), May. 2010.
[21] A. S. Patwardhan, "Low Morale, Depressed and Sad State Recognition in Confined Spaces", International Research Journal of
[22] Engineering and Technology (IRJET), May. 2011.
[23] A. S. Patwardhan, "View Independent Drowsy Behavior and Tiredness Detection", International Research Journal of Engineering and
[24] Technology (IRJET), May. 2011.
[25] A. S. Patwardhan, "Sensor Based Human Gait Recognition for Drunk State", International Research Journal of Engineering and
[26] Technology (IRJET), May. 2012.
[27] A. S. Patwardhan, "Background Removal Using RGB-D data for Fright Recognition", International Research Journal of Engineering
[28] and Technology (IRJET), May. 2012.
[29] A. S. Patwardhan, "Depth and Movement Data Analysis for Fight Detection", International Research Journal of Engineering and
[30] Technology (IRJET), May. 2013.
[31] A. S. Patwardhan, "HMM Human Action Recognition Classification using HMM and Movement Tracking", International Research
[32] Journal of Engineering and Technology (IRJET), May. 2013.
[33] A. S. Patwardhan, "Feedback and Emotion Polarity Extraction from Online Reviewer sites", International Research Journal of
[34] Engineering and Technology (IRJET), May. 2014.
[35] A. S. Patwardhan, "Call Tree Detection Using Source Code Syntax Analysis", International Research Journal of Engineering and
[36] Technology (IRJET), May. 2014.
[37] A. S. Patwardhan, "Walking, Lifting, Standing Activity Recognition using Probabilistic Networks", International Research Journal of
[38] Engineering and Technology (IRJET), May. 2015.
[39] A. S. Patwardhan, "Online News Article Temporal Phrase Extraction for Causal Linking", International Research Journal of
[40] Engineering and Technology (IRJET), May. 2015.
[41] A. S. Patwardhan, "Online Comment Processing for Sentiment Extraction", International Research Journal of Engineering and
[42] Technology (IRJET), May. 2016.
[43] A. S. Patwardhan, "Analysis of Software Delivery Process Shortcomings and Architectural Pitfalls", PeerJ Computer Science (in
[44] review), 2016.
[45] A. S. Patwardhan, "Multimodal Affect Recognition using Kinect", ACM TIST (in review), 2016.
[46] A. S. Patwardhan, "Augmenting Supervised Emotion Recognition with Rule-Based Decision Model", IEEE TAC (in review), 2016.
[47] A. S. Patwardhan, Jacob Badeaux, Siavash, G. M. Knapp, "Automated Prediction of Temporal Relations", Technical Report. 2014.

©IJRASET: All Rights are Reserved



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089 (24*7 Support on Whatsapp)