JOURNAL OF MECHANICS OF CONTINUA AND MATHEMATICAL SCIENCES
www.journalimcms.org

ISSN (Online) : 2454 -7190 Vol.-15, No.-6, June (2020) pp 104-117 ISSN (Print) 0973-8975

PRIORITIZED INTERVENTION IN E-COMMERCE APPLICATIONS USING LOGICAL OCL SOFTWARE AGENTS (PIE)

Shikha Singh[1], Manuj Darbari[2], Gaurav Kant Shankhdhar[3]

[1]PhD Scholar, Department of Computer Science and Engineering,B.B.DUniversity, Lucknow, Uttar Pradesh, India.

[2]Professor, Department of Computer Science and Engineering,B.B.DUniversity, Lucknow, Uttar Pradesh, India

[3]Assistant Professor, Department of Mathematics and ComputerScience, B.B.D University, Lucknow, Uttar Pradesh, India

[1]shikhasingh2816@gmail.com, [2]manujuma@gmail.com, [3]g.kant.82@gmail.com

Corresponding Author: Shikha Singh

https://doi.org/10.26782/jmcms.2020.06.00009

Abstract

The authors have devised a multi-agent system for management of enormous queries by the customers in an e-commerce website. The paper discusses the phenomenon of having a first visit registration of the customers, extracting the preferences as specified by the customers, accepting the queries for products and applying Affinity Propagation Algorithm in order to obtain the clusters. These clusters are the groups of customers who share common interests in buying products offered by the e-commerce website. So, now the system has segregated the similar types of queries into distinct groups. The queries are then prioritized according to the size of the clusters, that is, the biggest cluster containing maximum number of customers has greatest priority and so on. The queries belonging to same cluster (queries with same priority) are then passed through logical intervention using Object Constraint Language to maximize resource utilization and prevent double payment.

Keywords: OCL, Multi Agent System, e-commerce application, customer query based cluster, Affinity Propagation Algorithm.

I. Introduction

Digitalization has changed the perception of the world miraculously. The main resource in today's world is time and the proper utilization of this crucial and decisive resource has become the solitary factor contributing to its parent

organization. This paper deals with performance intensive and payment secure version of e-commerce. In order to understand this, few definitions are important to be mentioned. Trade is the trading of merchandise and enterprises, particularly for an enormous scope. It incorporates legitimate, financial, political, social, social and innovative frameworks that work in a nation or in global exchange. In the monetary sense, "business" alludes to the lead of exchange among financial operators. Electronics Commerce or popularly known as E-Commerce is a technique of present day business, which tends to the prerequisites of business associations. It tends to be extensively characterized as the way toward purchasing or selling of merchandise or services utilizing an electronic medium, for example, the Internet. A significant change in the buying pattern of the customers has been observed in the recent days. Instead of reaching a market place and going from shop to shop, customers find it more convenient to visit online shops. There are different types of internet business i.e.B2B, B2C and C&B. Motive behind them has ever been the ease of use for any category of client. In the entire e-commerce applications fundamental spotlight is on handling deferral and high throughput regarding result as wanted by the client. OCL was utilized in numerous previous writings however failed to accomplish excellent results as processing time was essentially higher.

All the above proposition have some downside or the other; for instance a customer route need to be guided to right item, however lamentably the time just as the result of the inquiry are fundamental driver of concern. Legitimate converging from the different databases is a major issue. The whole phenomenon is characterized by the design and development of a Multi-Agent System comprising of intelligent software agents that work in proper co-ordination for disciplined working of the complete system.

The methodology for agent development followed in this paper is O-MaSE, Organizational Multi-Agent System Engineering. [IV] Depicts the Adaptive O-MaSE (AO-MaSE) process, which gives modelers and designers an organized methodology for testing and iteratively including usefulness in unpredictable, versatile frameworks. O-MaSE coordinates a lot of solid advancements planned for making multi-operator innovation accessible to industry and encouraging across the board acknowledgment. In particular, O-MaSE was made as an adjustable approach that can be adjusted and reached out for a wide assortment of employments [VI]. An epic smart strategy is proposed right now moving component bearing deficiency determination dependent on the versatile element choice procedure and AP clustering algorithm [XXV]. An adaptive feature selection technique is proposed to expel excess data and to choose the ideal highlights, which will extraordinarily lessen the calculation weight and increment the precision of the clustering algorithm.[XII, XVII] Presented, a multi-objective transformative calculation dependent on liking spread to take care of network recognition issues. Right off the bat, the calculation utilizes a similitude measure dependent on signal transmission to change the graph clustering problem into an data clustering problem, and uses the AP technique to get a lot of primer allotments of the system. As AP (Affinity propagation) technique has high exactness and quick bunching speed, we can utilize it to get fulfilling good primer segment

results inside a couple of steps. [XXIV] Demonstrated, a two-stage fast affinity propagation (Fast AP) algorithm. The scale of the similarity matrix is first compressed by selecting only potential exemplars, than further decreased by inadequacy as per the *k* nearest neighbors, unlike from their previous task.

The problem with the current systems providing e-commerce services are by and large the huge number of queries for product purchase at a time interval. The fact is that many queries have similar arguments, choices and most importantly, common preferences by numerous online customers. This commonality in product selection paves ways to group similar queries. When these queries are grouped by usage of clustering algorithm (here, Affinity Propagation Algorithm), the queries are also segregated on the basis of resources ought to be utilized indirectly like quality, cost, delivery time, etc. Once grouped by category, the queries are clustered on the basis of aforementioned preference parameters. Then, clusters are sorted on the basis of cluster size. The biggest cluster is allotted maximum priority and subsequently priority is allotted for other clusters in decreasing order of the cluster size. These prioritized clusters of queries are then passed to intervening logic in OCL to get executed.

The first part of the paper will cover the current problem related to e-commerce website followed by exhaustive review over Object Constraint Language. Further the discussion on the proposed solutions will be taken up along with proof demonstration. And at the last the entire work will be backed up with the conclusion and future scope.

II. Related Work

Mandel and Cengarle in [XIII] examined the behavior and representation of OCL and its relationship with Relational Algebra. They tended to the expressive intensity of OCL as a question language. They explore whether for each RA query q we can fabricate an OCL articulation that profits the equivalent tuples as q. Since OCL 2.0 presented tuple offices, Balsters contended that OCL can encode any RA question made out of the essential RA activities [II], that is: association, distinction, item, renaming, choice and projection tasks. Be that as it may, Balsters worried in his work that, still, OCL isn't identical to RA from a maximal perspective since it is unthinkable in OCL to characterize another activity that gets as information two self-assertive arrangements of tuples, and yields the regular join of them. From a progressively down to earth perspective, Queralt and Teniente proposed in [XV] an interpretation from OCL to space free first-request rationales, which is identical to social variable based math. The part canvassed in their interpretation is expressively equal to OCLFO since they spread OCLCORE. Strikingly, their interpretation is additionally founded on first normalizing an OCL imperative into another made out of less articulations. Another interpretation of a part of OCL into firstorder rationale is proposed by Clavel et al. in [V]. By normally stretching out their interpretation of imbalances to disparities with objects, we can see that their OCL section secured is expressively comparable to OCLFO.

In constrast, the interpretation given by Beckert et al. in [III] appears to manage a more extensive subset of OCL. In any case, their interpretation isn't unadulterated first-request rationales since, for example, it utilizes some implicit capacities to check the occasions an item shows up in an assortment freely, which is certifiably not a first-request ability. At last, there are a few apparatuses that actualize interpretations from OCL into SQL. Egea et al. presented MySQL4OCL[VIII], which creates MySQL code for a subset of OCL articulations. In any case, the interpretation characterized plainly drops out of RA since it utilizes MySQL explicit methodology. Another apparatus, some portion of the notable Dresden OCL Toolkit [I], is OCL2SQL. It creates an interpretation in standard SQL, however needs hypothetical reason for complex cases. To be sure, the interpretation depends on some direct examples with no proper confirmation [VII], along these lines, it isn't away from rightness of the interpretation when managing, for example, invalid qualities. Without a doubt, OCL2SQL utilizes SQL not exists articulations which is known to have deceptive conduct when managing invalid qualities, however no conversation on this viewpoint is given.

Franconi [X], identified OCL_{FO} the first order fragmentations of OCL, which covers entire OCL except features like aggregation, recursion etc. They have also shown that OCL_{FO} has the similar expressivity like Relational Algebra, which indicates that any RA expression can be rewritten into an equivalent OCL_{FO}one and vice versa. [XIV] Demonstrated an alternate approach for automatically computing the repairs of an update, solving the problem which arises when updating an information base may lead to same constraint violation. The approach is independent of the language. They have applied UML/OCL as they are widely used in conceptual modeling community.

A comparison among UML/OCL and Ontology was conducted in [XXIII], to bridge the gap between the two approaches, to bring out the best out of them. A method was proposed [XXII] to improve the linking of scattered databases, with the help of clusters of similar queries and hence increasing the throughput of the same.[XVI] Priority based clustering of weighted graph streams, was proposed and demonstrated. They presented a data structure which kept the current state in terms of edges in the network along with a timestamp value for updating them. Algorithm by them claims clustering with high quality and performance compared to previous state-of-the-art evolution-aware clustering algorithms. [IX] Demonstrated the mapping of OCL into SQL expression and automatic code generation at the time of data modeling.

III. Software Agents Development in PIE

For developing the agents in this e-commerce ecosystem the authors have followed the O-MaSE agent development methodology [XI]. In this methodology goals of the organization are identified as the first task. The Goal Model is shown in figure 1. Once the goals are identified, they are allotted to roles in order to be accomplished [IV]. The roles describe the authority that functions to meet the goals. A role can comprise of one or more than one goals according to the similarity and

relationship between the goals. The Role Model as developed by following the O-MaSE methodology are shown in figure 2.

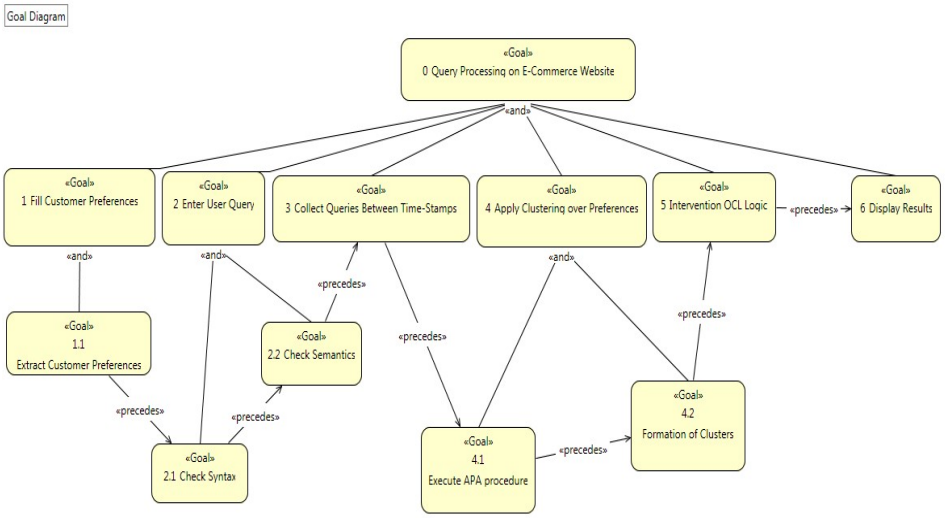


Fig. 1: Goal Model for PIE ecosystem

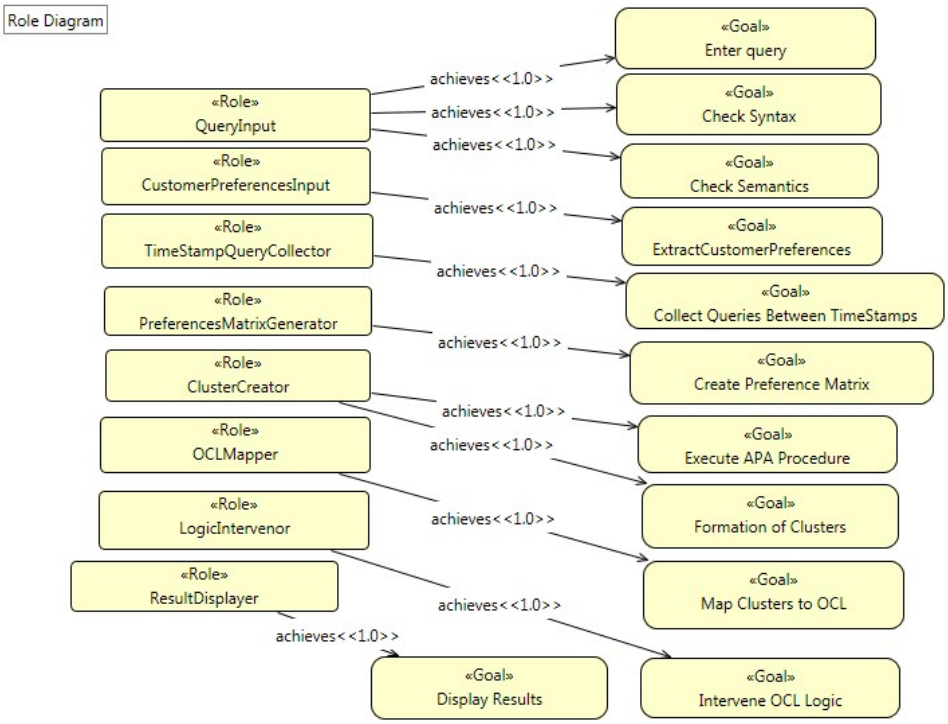


Fig. 2: Role Model for PIE ecosystem

The role diagram decides the identification and design of the software agents. The roles are converted into agents and the roles having similar functionality can even be

grouped into a single agent [VI]. This process is shown in figure 3, where software agents are derived through the roles identified in the previous process.

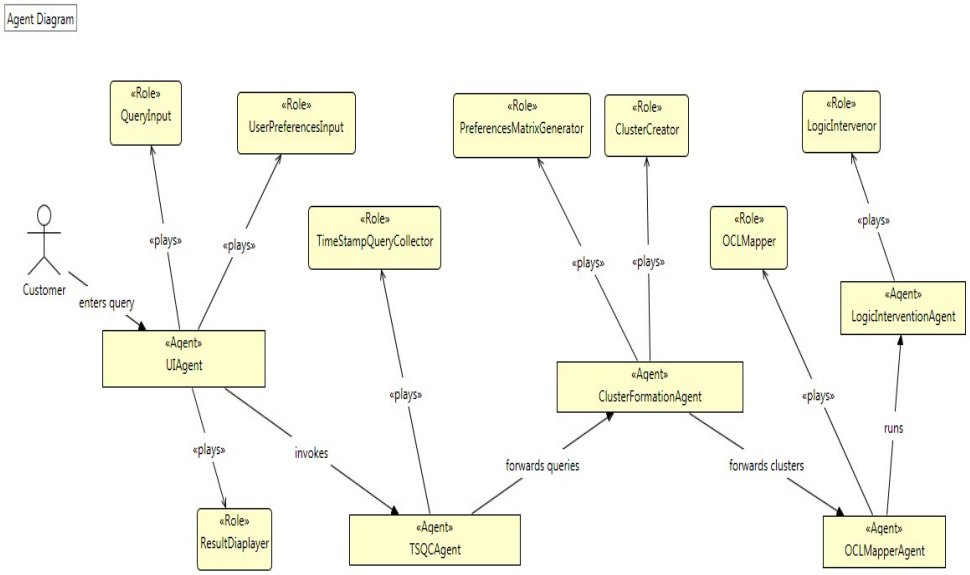


Fig. 3: Agent Model for PIE ecosystem

IV. Software Agents Designed in PIE Ecosystem

The software agents derived in the PIE ecosystem are shown in figure 3. The authors here describe the capabilities of the identified software agents [XVII]. The software agents along with their capabilities are given below:

UIAgent

The UIAgent speaks with the individual legitimately through the information and yield of the UI or interface to nature. An interface specialist can watch activities taken by the client in an immediate control interface, can even detect the items that the client sees on the screen, and would itself be able to take activities by summoning the orders gave by the interface. It can use speech input or output, or communicate via other sensory streams. The main capability of the UIAgent is storing preferences of the customers or visitors and accepting queries for products, the customer wants to purchase online. The UIAgent stores the preferences of the users to be later used by the TSQCAgent and the ClusterFormationAgent. The UIAgent is also responsible to display the results of the queries posted by the users.

TSQCAgent

The TSQCAgent stands for the TimestampQueriesCollectorAgent. This agent works in collaboration with the UIAgent in collecting queries within a start timestamp Ti and end timestamp Tj. Once all the queries are collected in the timestamp interval, they are forwarded to the ClusterFormationAgent.

ClusterFormationAgent

This agent is the most vital agent in this ecosystem because it applies the Affinity Propagation Algorithm in order to create clusters of the customers on the basis of preferences supplied by the UIAgent through the TSQCAgent. When the APA is applied in this manner, it creates clusters of various sizes. The use of APA is incorporated due to the fact that in this scenario, the authors don't know beforehand, the number of clusters that will be generated. Then, the clusters are arranged in decreasing order of their sizes. The cluster having the maximum size is allotted the maximum priority, and so do the subsequent clusters with decreasing cluster size are allotted respectively decreasing priorities [XVI]. Once the priorities are allotted the clusters are provided to the OCLMapperAgent.

The R Language code to implement the APA is shown below:

```
library(apcluster)
cust_prefs<-matrix(sample.int(8, size = 100*8, replace = TRUE), nrow = 100, ncol = 8)
print(cust_prefs)
sim<-negDistMat(cust_prefs[,1:8],r=2)
apclust<-apcluster(sim,details=TRUE)
show(apclust)
plot(apclust,as.matrix(cust_prefs[,1:2]),xlab="Cost", ylab="Quality", zlab="Delivery Time")
```

In the above code, the library for APA called, apcluster, is added [XVII], [XIX]. The preferences of 100 customers is taken in the form of a 100X8 matrix called, cust_prefs, where there are 8 preferences in total, namely, cost, quality, delivery time, flexibility, size, style, age of customer and education level of the customer. As a first step a similarity matrix is generated using negDistMat method. Finally, the clusters are formed by invoke of the apcluster method. The cluster are shown textually by calling the show method. The output of the show method is given below:

```
APResult object
Number of samples     = 100
Number of iterations  = 137
Input preference      = -80
Sum of similarities   = -2365
Sum of preferences    = -960
Net similarity        = -3325
Number of clusters    = 12
```

Exemplars:

2 5 9 23 33 36 57 62 64 77 80 90

Clusters:

Cluster 1, exemplar 2:

2 8 28 44 51 67 95 98

Cluster 2, exemplar 5:

4 5 7 30 35 53 60 66 73 97

Cluster 3, exemplar 9:

9 15 65 83 93

Cluster 4, exemplar 23:

6 14 19 22 23 32 42 61 79

Cluster 5, exemplar 33:

13 16 24 33 37 38 39 41 69 71 86

Cluster 6, exemplar 36:

10 12 20 29 36 45 48 54 70 75 88 89

Cluster 7, exemplar 57:

21 43 57 76 85 96

Cluster 8, exemplar 62:

18 46 59 62 81 99

Cluster 9, exemplar 64:

11 40 52 55 64 78 87 92

Cluster 10, exemplar 77:

3 17 31 56 77 84

Cluster 11, exemplar 80:

1 27 34 47 63 80 82 91 94 100

Cluster 12, exemplar 90:

25 26 49 50 58 68 72 74 90

The clusters generated in the form of graph using plot method, considering Cost on X-axis and Quality on Y-axis as shown:

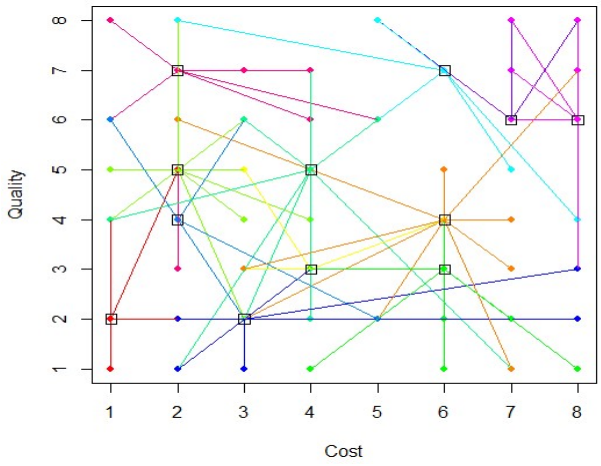


Fig. 4: APA clusters (in R Language)

For cost, quality and delivery time the plots are as below:

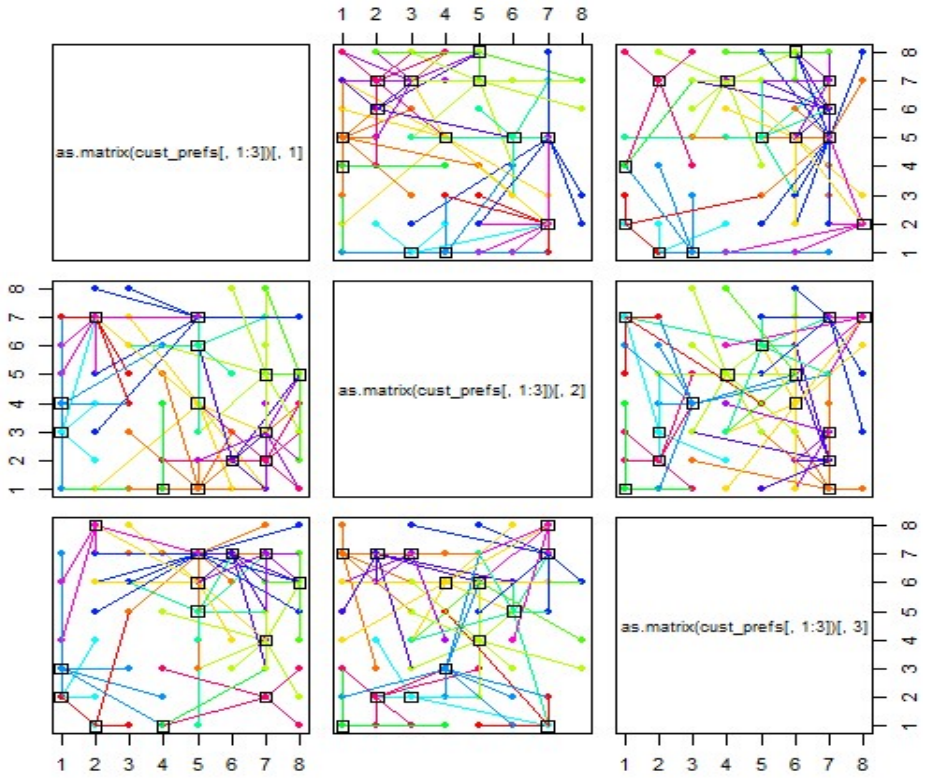


Fig. 5: APA Cluster formation with plot method using cost, quality and delivery time

OCL MapperAgent

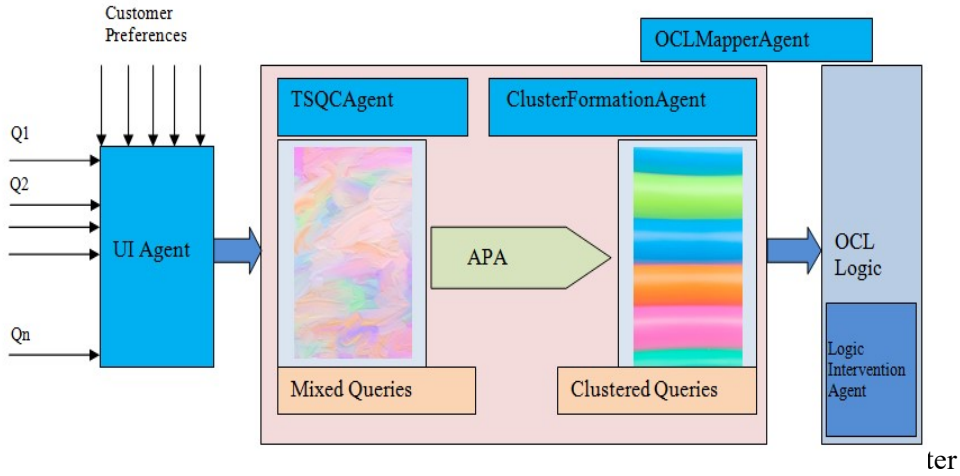
The task of the OCLMapperAgent is to initialize the LogicInterventionAgent and the Object Constraint Language Module for further processing of queries. The prioritized clusters are inputted to the LogicInterventionAgent.

Logic Intervention Agent

This agent uses Argumentative Query to provide much increased output using OCL logic foundations. By embedding theoretical formal syntax providing a Argumentative logic to reach to a final efficient query outcome.

The full process of the PIE eco-system is shown in figure 6.

Fig. 6: Clus



ter

generation of queries with similar preferences

The PIE multi-agent system accepts user queries and preferences through the UIAgent [XVIII]. This UIAgent checks for validity of the queries submitted by the end user. These end user queries are started to be accepted at a time stamp Ti and is open to take the queries till another time stamp Tj. When all the queries are taken up inside the time interval, these time stamped queries are forwarded by the TSQCAgent to the ClusterFormationAgent. The input to the ClusterFormationAgent comprises of mixed queries that are unrelated with mixed preferences from all the customers belonging to the specified time interval. The ClusterFormationAgenttakes these mixed queriesand applies the Affinity Propagation Algorithm (APA) in order to cluster these queries based on the preferences. The APA is applied once the items that have to be bought are associated with some categories. It is not known at prior about the number of clusters going to be formed, the main reason for using APA. Once the clusters are identified, the OCLMapperAgent maps the clusters to the LogicInterventionAgent. The purpose of the LogicInterventionAgent is depicted in

figure 7. In order to maximize resource utilization, the queries of one particular cluster are executed in the order of their specificity.

The basic idea of this model is to break down the searching item query by creating a cluster, as already done, and then applying the cluster constraints to solve the problem. The paper [XXII] suggests how augmentative computational method provides a more formalized and refined method of querying. The problem of time of query is simplified and satisfaction level has significantly increased along with drop in response time with respect to normal query. It was achieved by using cluster in which various sets of work breakdown Ontologies are stored and are retrieved used argumentative centric preference adaption using OCL.

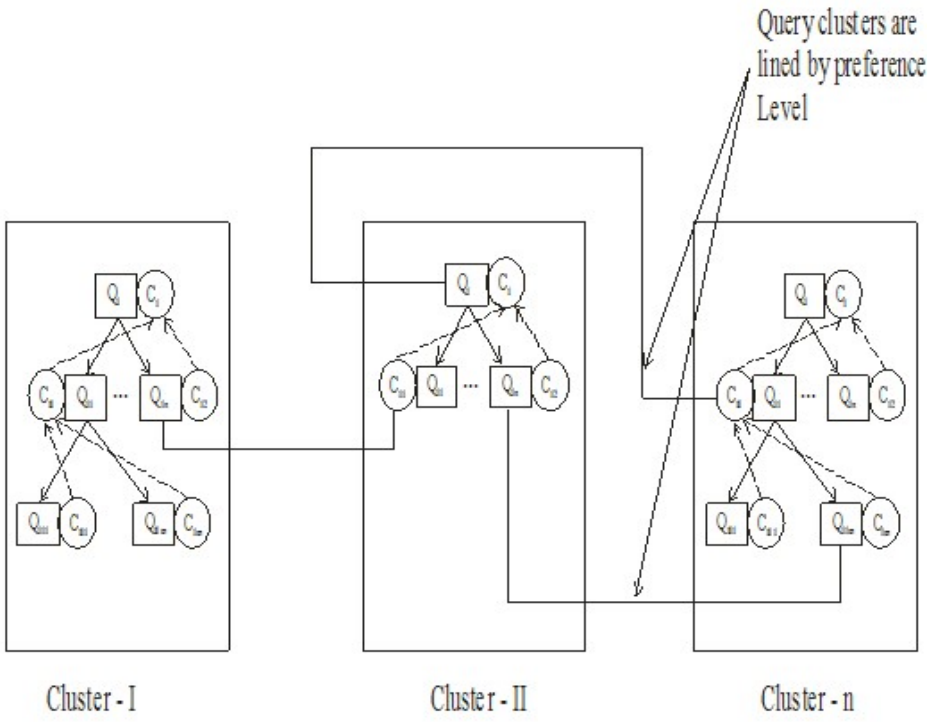


Fig. 7: Cluster formation of Queries with OCL Logic

V. Conclusion

The authors in this paper have proposed a multi-agent system in order to deal with the problem of tremendous amount of querying on e-commerce websites. Amazon is taken as the case study. The research shows the step by step derivation of software agents by using the O-MaSE methodology. The Goal, Role and the Agent diagrams are constructed and functionality defined [XIX], [XX], [XXI]. Customer preferences are extracted from the user interface for clustering queries through AP algorithm. These clusters are given priority based on their size. The prioritized clusters are passed through the OCL logic intervention process where the query results are obtained and finally displayed to the end user.

References

I. Aßmann, U., Bartho, A., Bürger, C., Cech, S., Demuth, B., Heidenreich, F., Johannes, J., Karol, S., Polowinski, J., Reimann, J., Schroeter, J., Seifert, M., Thiele, M., Wende, C., Wilke, C.: Dropsbox: the dresden open software toolbox. Software & Systems Modeling 13(1) (2014) 133–169

II. Balsters, H.: Modelling database views with derived classes in the UML/OCL-framework. In: UML2003-The Unified Modeling Language. Modeling Languages and Applications. Springer (2003) 295–309

III. Beckert, B., Keller, U., Schmitt, P.H.: Translating the object constraint language into first-order predicate logic. In: Proceedings of VERIFY, Workshop at Federated Logic Conferences (FLoC). (2002)

IV. Case, Denise, and Scott DeLoach. "Applying an o-mase compliant process to develop a holonicmultiagent system for the evaluation of intelligent power distribution systems." International Workshop on Engineering Multi-Agent Systems. Springer, Berlin, Heidelberg, 2013.

V. Clavel, M., Egea, M., de Dios, M.A.G.: Checking unsatisfiability for OCL constraints. In: Proceedings of the Workshop The Pragmatics of OCL and Other Textual Specification Languages. Volume 24., ECEASST (2009)

VI. DeLoach, Scott A. "O-MaSE: an extensible methodology for multi-agent systems." Agent-Oriented Software Engineering. Springer, Berlin, Heidelberg, 2014.

VII. Demuth, B., Hussmann, H.: Using UML/OCL constraints for relational database design. In: «UML»99 - The Unified Modeling Language. Springer (1999) 598–613

VIII. Egea, M., Dania, C., Clavel, M.: MySQL4OCL: A stored procedure-based MySQL code generator for OCL. Electronic Communications of the EASST 36 (2010)

IX. Egea, M., Dania, C.: Sql-pl4ocl: an automatic code generator from ocl to sql procedural language. Software & Systems Modeling (May 2017)

X. Franconi, E., Mosca, A., Oriol, X., Rull, G., Teniente, E.: Logic foundations of the ocl modelling language. In: European Workshop on Logics in Artificial Intelligence, Springer (2014) 657–664

XI. Kant, Gaurav, et al. "Legal Semantic Web-A Recommendation System." International Journal of Applied Information Systems (IJAIS) 7 (2014).

XII. Li, Peixin, et al. "Dynamic equivalent modeling of two-staged photovoltaic power station clusters based on dynamic affinity propagation clustering algorithm." International Journal of Electrical Power & Energy Systems 95 (2018): 463-475.

XIII. Mandel, L., Cengarle, M.V.: On the expressive power of the object constraint language OCL. In: FM'99 — Formal Methods. Volume 1708 of Lecture Notes in Computer Science. Springer Berlin Heidelberg (1999) 854–874

XIV. Oriol, X., Teniente, E., Tort, A.: Computing repairs for constraint violations in uml/ocl conceptual schemas. Data & Knowledge Engineering 99 (2015) 39–58

XV. Queralt, A., Teniente, E.: Verification and validation of UML conceptual schemas with OCL constraints. ACM Trans. Softw. Eng. Methodol. 21(2) (2012) 13

XVI. Saadatpour, Mohsen, et al. "Priority-based Clustering in Weighted Graph Streams." Journal of Information Science & Engineering 34.2 (2018).

XVII. Shang, Ronghua, et al. "A multiobjective evolutionary algorithm to find community structures based on affinity propagation." Physica A: Statistical Mechanics and its Applications 453 (2016): 203-227.

XVIII. Shankhdhar, Gaurav Kant, and ManujDarbari. "Introducing Two Level Verification Model for Reduction of Uncertainty of Message Exchange in Inter Agent Communication in Organizational-Multi-Agent Systems Engineering, O-MaSE." IOSR Journal of Computer Engineering (IOSR-JCE) IOSR Journal of Computer Engineering (IOSR-JCE) 19 (2017): 08-18.

XIX. Shankhdhar, Gaurav Kant, and ManujDarbari. "Building custom, adaptive and heterogeneous multi-agent systems for semantic information retrieval using organizational-multi-agent systems engineering, O-MaSE." 2016 2nd International Conference on Advances in Computing, Communication, & Automation (ICACCA)(Fall). IEEE.

XX. Shankhdhar, Gaurav Kant, et al. "Fuzzy Approach to Select Most Suitable Conflict Resolution Strategy in Multi-Agent System." 2019 International Conference on Cutting-edge Technologies in Engineering (ICon-CuTE). IEEE, 2019.

XXI. Shankhdhar, Gaurav Kant, et al. "Implementation of Validation of Requirements in Agent Development by means of Ontology" International Journal of Computer Sciences and Engineering 6 (7), 2018

XXII. Shikha Singh, ManujDarbari, "Logical Intervention in the Form of Work Breakdown Strategy using Object Constraint Language for e-Commerce Application" , International Journal of Advanced Computer Science and Applications, vol.11, No.3, pp-266-271, 2020.

XXIII. Shikha Singh, ManujDarbari, "Ontological Representation of the UML/OCL Models and Their Verifications", International Journal of Future Generation Communication and Networking, Vol.13, No.1, pp.940-951, 2020.

XXIV. Sun, Leilei, et al. "Fast affinity propagation clustering based on incomplete similarity matrix." Knowledge and Information Systems 51.3 (2017): 941-963.

XXV. Wei, Zexian, et al. "A novel intelligent method for bearing fault diagnosis based on affinity propagation clustering and adaptive feature selection." Knowledge-Based Systems 116 (2017): 1-12.