Journal of Theoretical and Applied Information Technology
30th June 2019. Vol.97. No 12
© 2005 – ongoing JATIT & LLS
ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

ANALYSIS OF ENTERPRISE ARCHITECTURE COMPLEXITY: MODELING AND DIAGNOSIS HETEROGENEITY

[1]JIHANE LAKHROUIT, [2]KARIM BAÏNA

[1]Doctor, Laboratory of Innovation in Management and Engineering for Enterprise (LIMIE)
Higher Institute of Engineering and Business, ISGA
Marrakech, Morocco
[1]University Mohammed V, ENSIAS
Alqualsadi research team on Enterprise Architecture
Rabat, Morocco
[2]Doctor, University Mohammed V, ENSIAS
Alqualsadi research team on Enterprise Architecture
Rabat, Morocco
E-mail: [1]jihane.lakhrouit@gmail.com, [2]baina@ensias.ma

ABSTRACT

The management of heterogeneity is of particular importance today. This new problem is being tackled more and more by several works in various fields of research. Among these areas is Enterprise Architecture EA, some work has been limited to defining the various types of complexity measure, while others have focused on the representation or direct calculation of different measures. In our contribution we propose a complete methodology ranging from the discovery of measurements to the implementation of the prototype.
Keywords: *Enterprise Architecture; EA patterns; Analysis of Enterprise Architecture; Complexity; Heterogeneity.*

1. INTRODUCTION

Enterprise Architecture aims to identify the application and information heritage of an organization by developing a mapping of technical, application, functional and organizational infrastructures. This is usually the starting point before embarking on new computerization projects for a part of the IS, the implementation of a global IS governance approach or the evaluation of this architecture.

The idea of documenting the enterprise architecture dates back to the 1980s when Zachman (1987) published the first EA framework. Thereby, the goal of enterprise architecture is—as indicated by its name—to describe the architecture (system structure) of an entire enterprise.

Complexity is considered one of the most critical issues to deal with because of the constraints and difficulties that surround it, many companies seem to consider it as a general source of problems, it is held responsible for the rise in coordination efforts, operating costs, and also increased efforts to make changes, which significantly hinders the agility of the information system. For that we chose to project the complexity and especially the heterogeneity on the domain of the enterprise architecture. Our objective is to propose an evaluating methodology for guiding designers and architects in evaluating and improving the EA models.

The goal of this paper is to (1) present the enterprise architecture component regarding agility and complexity measurement, (2) identify formula of complexity and especially the dimension of heterogeneity, and (3) apply metrics to enterprise architecture components and relationships.

The paper is structured as follows: the second section describes the state of the art of our research, the third section details how to calculate the heterogeneity of enterprise architecture, the fourth section presents our proposed approach and presents some of our results, the fifth section presents the diagnostic method and the numeric results of the proposed method and finally, the last section is dedicated to conclude our paper.

The case study which is used in this paper is an abbreviated version of a study under development in a bank. This is done to give a more

Journal of Theoretical and Applied Information Technology
30th June 2019. Vol.97. No 12
© 2005 – ongoing JATIT & LLS
ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

comprehensive presentation of how the method can be used and to demonstrate the efficacy of our approach.

2. STATE OF THE ART

In the following section, we present our literature review related to our research. We define the complexity and its dimensions and the existing methods in the literature which discussed and developed a method of evaluating enterprise architecture.

2.1 Complexity – definition and dimensions

According to Davis and LeBlanc [1] the complexity of application architecture is "number of its components or elements, kind or type of elements and structure of the relationship between elements". On the infrastructure architecture level defined complexity as "The complexity can be defined here as the dramatic increase in the number and heterogeneity of included components, relations, and their dynamic and unexpected interactions in IT solutions" [4], another definition proposed by [3] covers all aspects of complexity "The complexity can be defined on the basis of the number and variety of components and interactions plus the rate of change of these". From the different definitions cited we can notice that the complexity is a fuzzy term, because different stakeholders have generally different views and conceptions of complexity term.

From these definitions, we have proposed a global definition: "The complexity of architecture is the description of its structure and quantification of the numbers and heterogeneity of components and relations between them over the time".

The figure 1 shows the four dimensions of the complexity of enterprise architecture (Number of components and relations, the component heterogeneity, the relation Heterogeneity and Rate and impact of change), although the number of components and relationships can be determined by simply counting the respective elements, heterogeneity, calculating change rates and the architectural structure must be calculated using formulas and measures that we must clarify

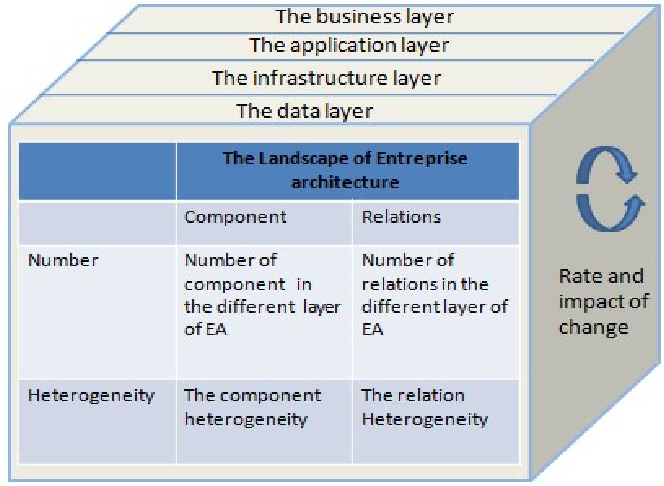

Figure 1: The dimensions of enterprise architecture complexity [5]

In this paper, we will discuss only the dimension of enterprise architecture heterogeneity (components and relations).

2.2 Methods for measuring complexity

•

This section presents two types of approaches, the first for presenting and calculating only the complexity of one layer of enterprise architecture EA and the second type are more holistic and propose generic approaches for one or more layers of enterprise architecture.

- **Analysis approaches specific to one layer for evaluating complexity of enterprise architecture**

In the literature, the methods of measuring the complexity capture very different aspects of this concept. Two of the most interesting and most relevant aspects affecting the development effort are: The size which refers to the size of the program (eg lines of code..) Or functional size (eg function points) and the complexity of the software structure interpreted through the complexity of the design or structure of the code structure [6][7].

Mocker [8] identified the complexity of the application layer as the age of applications and the number of functional requirements defined for each application. Based on the available literature, he identified four different measures to quantify the complexity: the interdependence, the diversity in technologies, and the heterogeneity of standards and of technologies. The method proposed by Widjaja[9] revolves around two axes : the

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

heterogeneity of a landscape as a statistical property that can be measured by statistical indices. The measurement to quantify the heterogeneity is the entropy which is proposed by Jacquemin and Berry [10] and a generic mathematical model to quantify the heterogeneity in computer landscape.

- **Holistic approaches for evaluating complexity of enterprise architecture**

During the analysis of the identified contributions only few methods were presented to quantify complexity and the existing methods merely cover parts of an EA, not the EA as a whole. Often the application is so specific that it is not possible to transfer the method to other dimensions of an EA.

In the paper [9] it discussed the metrics for EAs and application landscapes are introduced as decision support techniques based on analysis of structural dependencies. The approach emphasizes on operational risk, failure propagation and availability, based on a practitioner survey. In order to explicate the structural dependencies analyzed in the paper, an information model with derived attributes is used. An EA level application example is also given in the paper [9] with visual analysis of ex post information about failure propagation to compare different project proposals for the evolution of the application landscape. Thus, the project portfolio management process is supported.

Lagerström et al. [13] proposed to use an approach pervasive in the software architecture discipline—Design Structure Matrix—to visualize the hidden structure of an AL and thereby identify spots of increased complexity. Schuetz et al. [11] introduce a metric to quantify the structural complexity of an IT landscape, which is also applicable to application landscape. The proposed approach of Schutz [11] revolves around the conceptualization of the complexity of EA by adopting the concept of the system to the context of EA. This approach presented a holistic conceptualization of complexity but don't apply it in the different layers of EA.

3. DEFINITION OF HETEROGENEITY

Heterogeneity is one of the most important dimensions of complexity. To be able to analyze it, we must first define the existing measures and then discuss their applicability.

3.1 Definition of heterogeneity

Measure	Additional Features	Change in classification	Effect of small sets of elements	Effect of proportional changes
The Herfindahl index	Yes	Yes	Yes, but with marginal impact	No
The Horvath index	Yes	No	Yes, but with marginal impact	No
The measurement of entropy	Yes	Yes	Yes	No

Heterogeneity is defined as the diversity of elements or relationships of a system according to its characteristics. More precisely, in computer science, the heterogeneity of a computer landscape is a statistical property that presents the diversity of the types of elements that compose it [9], taking as an example the heterogeneity of database management systems (DBMS). This heterogeneity can be understood as a frequency distribution and can be expressed in graphical form as shown in the following figure.

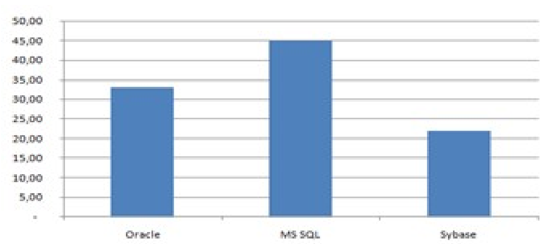

Figure 2: The number of instances per DBMS.

The figure above shows the number of instances per DBMS. Each of the bars represents an instance of the DBMS.

3.2 Definition of heterogeneity measurements

In the literature the most widely used method for measuring heterogeneity is the use of concentration measurements, however there are three measures: Herfindahl-Hirschman index, Horvath index and entropy measure.

These three measures have been used and applied in a number of areas, for example, by the US authorities in calculating market concentration [21] or measuring the power of companies[14], in the field of geography to calculate industry concentration, in software engineering [16] [17] And in many different economic sectors. The table 1 shows the three concentration measurements with their respective formulas.

Table 1: The formulas of heterogeneity measurement in the literature.

Name of measure	Measurement formula
The Herfindahl Index	$HER = \sum_{i=1}^{n} p_i^2$
The Horvath Index	$HOR = p_1 + \sum_{i=2}^{n} p_i^2 (2 - p_i)$
The measurement of entropy	$H = EM = -\sum_{i=1}^{n} p_i \ln(p_i)$

Based on the state of the art [9][11][22], a set of potential criteria were identified to validate heterogeneity measurements as explained in the table 2.
Each of these requirements is applied to the three measurements (Herfindahl Index, Horvath Index and Entropy Measurement) in the table 5.

Table 2: Validation of heterogeneity measures.

From the study in the table 2, where we verified the conditions necessary to validate the heterogeneity measurements, we note that: 1- The change in classification impact the result of the Horvath index 2 -the effect of the addition of small sets of elements has a marginal impact on the value of the Herfindahl index. If the two distributions 50%, 50% and 50%, 48%, 2% are calculated there was only a marginal impact which is 0.02 (0.5, 0.48), to avoid these problems of value we will use a concentration ratio which verifies amply these properties which is : the measure of entropy.
If the measure of entropy is used to compare the two distributions already calculated by the Herfindahl index 50%, 50% and 50%, 48%, 2%, the values obtained have a more significant difference, namely, 0.69 and 0.78

In this section we have explored the various measures existing in the literature and according to the validity criteria applied to the measurements and we have chosen the entropy of Shannon. Note that the entropy measure can be used to quantify the heterogeneity of the components as well as the heterogeneity of the relations of the system. In the next section, we will apply this measure to calculate the heterogeneity in the enterprise architecture based on the enterprise architecture management pattern approach.

4. OUR PROPOSAL PATTERNS FOR EVALUATING ENTERPRISE ARCHITECTURE HETEROGENEITY

4.1 Presentation of enterprise architecture management pattern

The EAM pattern language developed by Buckl [22][23][24] distinguishes between four different types of patterns:

- M-Patterns: Methodologies define steps to be taken in order to address given objectives. These objectives are addressed by procedures defined by the methodology. Others refer to them as Process Patterns.
- V-Patterns: Viewpoints provide the languages used by methodologies. A viewpoint proposes a way to present data stored according to one or more information.
- I-Patterns: Information models represent underlying models for the data visualized in one or more viewpoints. An information model pattern conveys an information model fragment including the definitions and descriptions of the used information objects.

To improve readability, the comparability and structure of enterprise architecture management EAM pattern, all of the patterns follow the extension of the formalism proposed and discussed in the paper [5].

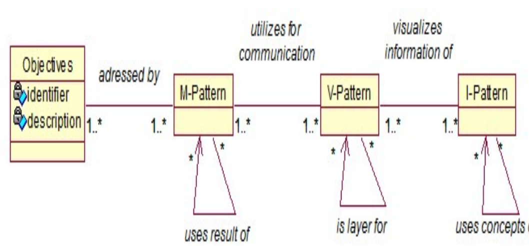

Figure 3: The conceptual model underlying the EAM pattern language.

Journal of Theoretical and Applied Information Technology
30th June 2019. Vol.97. No 12
© 2005 – ongoing JATIT & LLS

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

4.2 I-patterns for analyzing enterprise architecture heterogeneity

Based on the information pattern I-50 presented on the paper [5] we present three types of concepts in which we apply the measure of entropy. Concept 1 represents only the heterogeneity of a single component of the enterprise architecture, concept 2 represents the relationship between two components and calculates heterogeneity with respect the relation and the concept 3 is an exceptional case from concept 2 it presents a relationship path that connects several components. These concepts are summarized in the Table 3 below.
The I- pattern I-52 presents the measurements in the Table 3. The measurements are illustrated and numbered from 1 to 8 in the diagram. The most important concepts are: ProcessPerComponent to model the Heterogeneity Measure of Process Implementation by component, ServicePerComponent to model the Heterogeneity of database services by component and ProcessPerOrgUnit to analyze process heterogeneity by organizational unit. The 8 applications of heterogeneity, which we propose, are illustrated by numbers in the diagram; each application is represented by an algorithm. In this paper we will present only the most important.

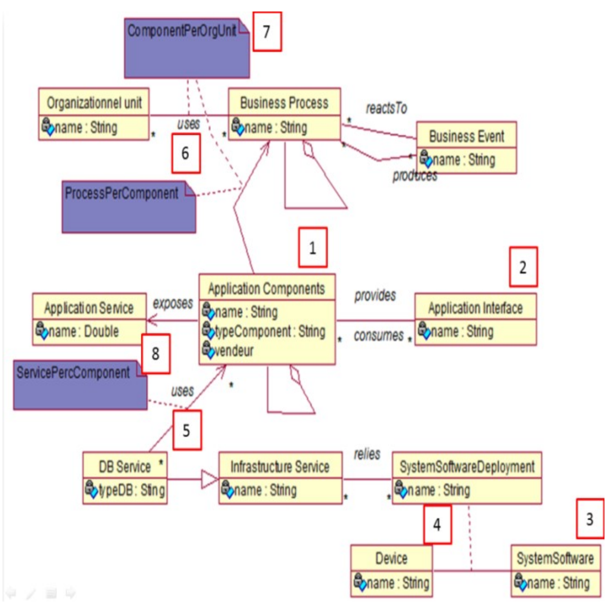

Figure 4: The I-Pattern diagram "Analysis of Heterogeneity" I-52.

Name: calculating the heterogeneity of databases
Variables: DB: all Databases Service
Map instance = map <String instanceType, Integer numberInstance>
Double sum
Double percentage
Double heterogeneity
Integer numberInstance
Create a map = instance that has a String for the type of database and an integer for the number of instances
For any db in DB
If instance contains instanceType = db.type
numberInstance = instance.get (db.type)
Increment numberInstance number by 1
instance.get (db.type) .SetValue (nombreInstance)
If not
Add a new entry in the map as db.type key and value 1
Instance.add (db.type, 1)
End if
End For
For i ranging from 0 to N = DB.size ()
// divide numberInstance by N
Double percentage = Instance.get (i) .getValue () / N
sum = sum + percentage * log (percentage)
endfor
heterogenity = -sum

Table 3: The application of heterogeneity measures.

Concept Type	Concept of Heterogeneity	Number of instances	The Heterogeneity of the Concept
Type 1	Application Components	Number of Application Components	Concentrations of applications by vendor or type (developed, purchased and adapted, purchased).
	Application Interface	Number of Interfaces	Concentrations of the types of interfaces.
	Computer	Number of Computers	Computer Concentrations by Type
	Operating System	Number of Operating System	Operating System Concentrations by Type
	Database	Number of databases	Database Concentrations by Type
Type 2/3	Implemented Processes	Number of Implemented Processes	Concentration of implemented processes by component
	Using application components	Number of components used by organizational units	Concentration of processes by organizational unit.
	Using Databases	Number of database instances used.	Concentration of databases by component

```
Name: calculation of process concentration by component
Variables: BS: all business processes
CP: the application components
 Map instances = map <String componentType,
Integer processNumber>
Double sum
Double percentage
Integer Comp
Double heterogeneity
Create a map = instance: its key is a String for the
application components and an integer for the
number of processes
For all cp in CP do
For all r in cp.relations do
   If (r.target = bs) then
count = count + 1
         If instances contains componentType =
cp.name
For any instance in the instances map
If (instance.composingType == cp.name)
Increment the number numberProcess by 1
End if
endfor
          If not
Add a new entry in the map with the key cp.name
and value 1
          End if
    If not
Do nothing and move on to the next relationship
   End if
End For
End For
For any instance in the instances map
// Divide instance.numberProcess by count
Double percentage = instance. numberProcess /
comp
sum = sum + percentage * ln (percentage)
endfor
heterogenity = -sum
return heterogenite
```

Journal of Theoretical and Applied Information Technology
30th June 2019. Vol.97. No 12
© 2005 – ongoing JATIT & LLS

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Name: calculation of the heterogeneity of operating systems and computers
Variables: SSD: All Instances of Operating Systems Deployed SystemSoftwareDeployment
Map instancesSE = map <String instanceType, Integer numberInstance>
Map instancesComputer = map <String instanceType, Integer numberInstance>
Double sum, sumDorDI
Double percentageSE, percentage ORDI
Double heterogeneity ORDI, heterogenite
Integer numberInstanceSE, numberInstanceORDI

Create a map = instanceSE that has a String for the OS type and an integer for the number of instances
Create a map = instanceComputer that has a String for the computer type and an integer for the number of instances
For all ssd in SSD
If instanceSE contains instanceType = ssd.systemSoftware
numberInstanceSE = instanceSE.get (ssd.systemSoftware)
Increment the number numberInstanceSE by 1
instanceSE.get (ssd.systemSoftware) .SetValue (nombreInstanceSE)
If not
Add a new entry in the instanceSE map as ssd.systemSoftware key and value 1
InstanceSE.add (ssd.systemSoftware, 1)
End if

If the computer instance contains instanceType = ssd.device
numberInstanceORDI = instanceComputer.get (ssd.device)
Increment the numberComputer instance by 1
instanceSE.get (ssd.device) .SetValue (nombreInstanceORDI)
If not
Add a new entry in the computer instance instance as ssd.device key and the value 1
instanceOrdinateur.add (ssd.device, 1)
End if
End For

For i ranging from 0 to N = SSD.size ()
// divide numberInstance by N
Double percentage = instancesSE.get (i) .getValue () / N
sum = sum + percentage * log (percentage)
endfor
heterogeniteSE = -som
sum = 0

For i from 0 to N = SSD.size ()
// divide numberInstance by N
Double percentage = computerInstance.get (i) .getValue () / N
sum = sum + percentage * log (percentage)
endfor
heterogeniteORDI = -sum
Return heterogeniteORDI

•

The indicators	Heterogenity AS-IS architecture	Heterogenity TO-BE architecture
Application Components	0,68	0,57
Application Interface	0.71	0,71
Ordinateur	0,50	0,50
System Software	0,30	0,30
Base de données	0,50	0,70
Processus Implémenté	0,38	0,54
Utilisation des composants applicatifs	0,45	0,73
Utilisation des bases de données	0,69	0,82

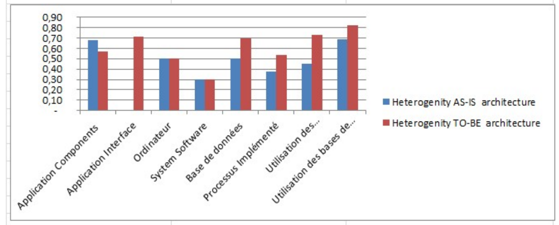

Figure 5: The comparison between enterprise architecture heterogeneity measurements between the AS-IS and TO BE architectures.

5. DIAGNOSIS AND CASE STUDY

5.1 Diagnosis

Since the beginning of this paper we have discussed and proposed algorithms and measures to quantify the heterogeneity of the enterprise architecture, the next step is to diagnose these results by interpreting the different measures, because decision-makers have to make decisions and corrective actions to improve the situation of complexity.
Section 3 shows that Shannon's entropy which is the most appropriate measure for measuring heterogeneity within the enterprise architecture context. The measure of entropy takes its minimum value if all the elements share a single characteristic, and reaches its maximum value when a distribution of the different characteristics at an equal percentage.
The interpretation of the entropy measurement is facilitated by the equivalent entropy measurement EM_A = EXP (EM): the number which designates the equivalent number of characteristics with equal distribution.
The figure 6 presents two examples of heterogeneity interpretation. The first distribution has a heterogeneity value equal to EM = 0.7 which corresponds to an equivalent distribution equal to EM_A = 2 or the actual distribution is equal to 4 components, which is explained by a high concentration on one component, indeed the component having a high concentration is the Oracle database. The second distribution has a heterogeneity value equal to EM = 1.39 which corresponds to an equivalent distribution equal to EM_A = 4 and indeed there are 4 components

(Oracle, DB2, SQL Server and MySql), which is explained by an equivalent distribution between the components.

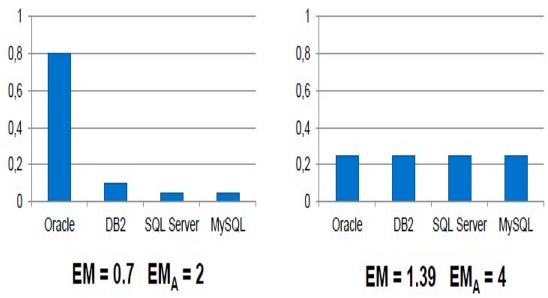

Figure 6: The original EM distribution and the equivalent EMA equivalent distribution.

5.2 Case Study

The company we studied is a bank that plays a key role in the market. In this case study, the company wanted to assess the complexity and especially the diversity of its landscape to justify a recent increase in IT budget costs and to identify the potential for reducing complexity. In order to provide senior managers with all the artifacts presented, we discussed with them all the metrics proposed and defined in the "Analysis of Heterogeneity of Enterprise Architecture" pattern. I-52

- **Data Collection and Enterprise Architecture Modeling**

For this approach, the initial set was 8 measurements divided into two types as shown in Table 4, after completing the process of choosing the measurements using the AHP method explained in the paper [2], the architects decided to apply the four measures presented in the Table 4 below.

Table 4: The Measures Chosen By The Architects By AHP Method.

Type	Concept of heterogeneity	Number of instance	Heterogeneity of the concept
Type 1	Operating System	Number of Operating System	Operating System Concentrations by Type
	Database	Number of databases	Database Concentrations by Type
Type 2/3	Implemented Processes	Number of Implemented Processes	Concentration of implemented processes by component
	Using Databases	Number of database instances used.	Concentration of databases by component

- **Diagnosis and decision making for the heterogeneity of the enterprise architecture components**

After studying the recovered data, the implementation seems possible. We apply the different algorithms proposed in the I-Pattern. The "Operating Systems" and "Database Management Systems" complexity were chosen as the first step; afterwards we have modeled and then measured "Process Implemented" and "Database Utilization". The result of collecting data on the number of operating systems and database instances is 15 operating systems with 1146 instances and 10 databases with 41 instances. As shown in the figures below.

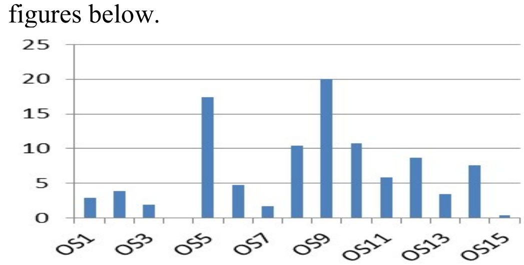

Figure 7: Percentages of distribution of instances of operating system.

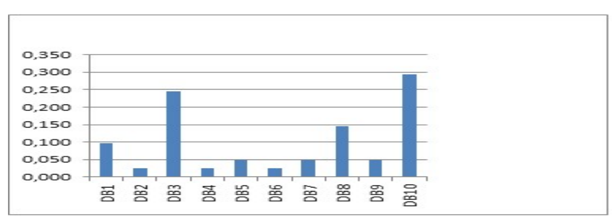

Figure 8: Percentages of distribution of instances of databases.

The concentration of the databases is EM = 1,93 which corresponds to EM_A = 6,8 and at the level of

Journal of Theoretical and Applied Information Technology
30th June 2019. Vol.97. No 12
© 2005 – ongoing JATIT & LLS
ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

our architecture we have 10 databases, these results show us that there is a wide choice of type of DB that exceeds the desired average which can be problematic in terms of the cost of maintenance and purchase licenses and also the level of variety of technical skills desired to manage these different databases.
The concentration of the operating systems SE is EM = 2.34 which corresponds to EM_A = 10.40 and at the level of our architecture we have 15 operating systems, these results also show us that there is a wide choice of OS type that exceeds the desired average, which can be problematic in terms of increasing the cost of maintenance and purchasing licenses.
For the "Implemented Processes" and "Database Utilization" measures, the company used 100 application components that support 60 business processes. After performing the concentration measurement we noticed that on the one hand there are several applications that were obsolete and that could be replaced by new existing applications as they did not implement any current operational process, on the other hand we noticed from the value of the equivalent entropy that there is an overload at the level of the implementation of the important processes in two applications, to solve this problem we proposed to apply the scenario of decrease of the number and relationships and also the consolidation scenario.
.After detailing these different metrics, in a meeting with the corporate architects, they all agreed that these metrics are valuable for their work. As a result, the company decided to implement these measures and perform calculations every six months to see trends over time.

6. PROTOTYPE

The application architecture is divided into three layers: an information management or backup layer that stores data from a model or from existing source files in a data warehouse, a reporting layer that presents the results Heterogeneity measures in graphical form and an interaction layer that offers the possibility of modeling the desired points of view.

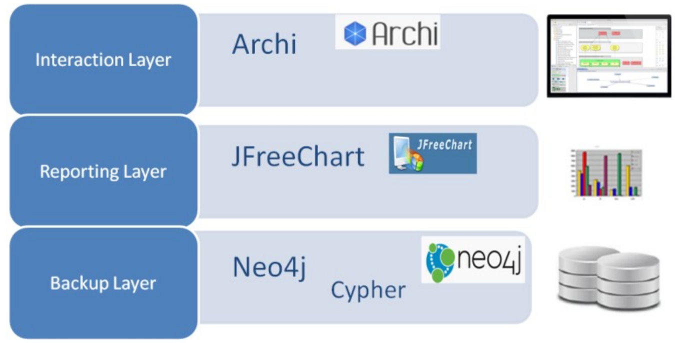

Figure 9: The Three Layers Of Prototype.

The interaction layer represents the applications that will allow decision makers to model the views of the enterprise architecture and enrich it with existing data. The modeling editor is as shown in the following figure.

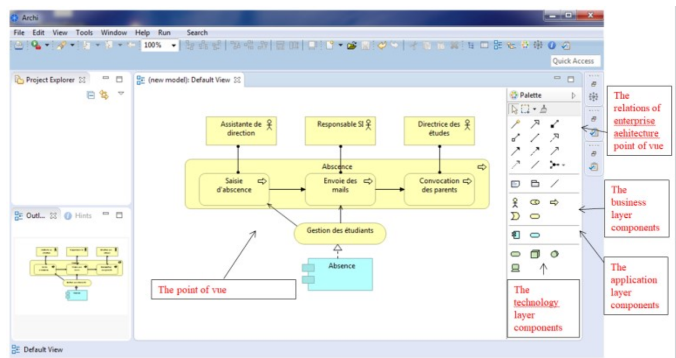

Figure 10: The Modeling Interface.

The illustrated tool represents the first step which is the modeling of the enterprise architecture by graphically describing the elements and existing relations, it is an ArchiMate point of view modeled by the Archi interface. It consists of an element set of each layer.

The description of the AE is stored in two Comma-separated values CSV files. To manage this metadata, we have developed a desktop application java, illustrated in the Figure, which allows us to manage this metadata, to apply the heterogeneity measurement algorithms and to visualize the output graphs.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

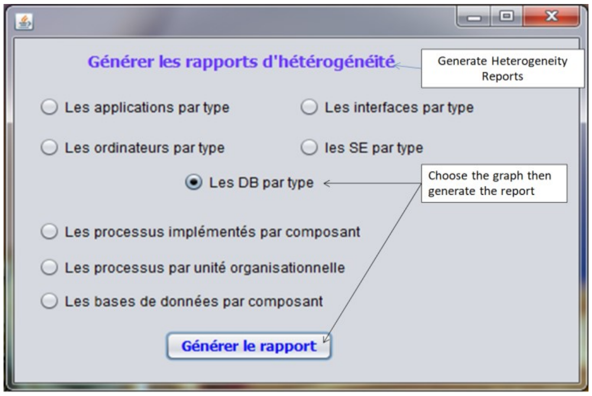

Figure 11: The Interface To Generate The Heterogeneity Graphs.

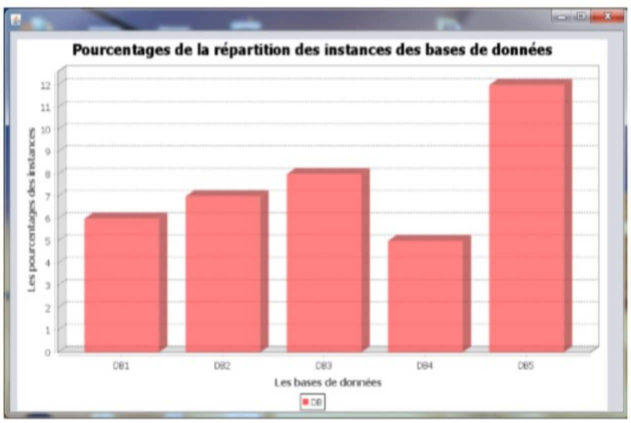

Figure 12: The Report Generated For The Distribution Of Databases Instances.

7. CONCLUSION

Nowadays, enterprise architecture (EA) has garnered considerable attention from both practitioners and academics in the fields of information systems and business management. Enterprise architecture (EA) is an approach to managing the complexity of an organization's structures, information technology (IT) and business environment, and facilitating the integration of strategy, personnel, business and IT towards a common goal through the production and use of structural models providing a holistic view of the organization. In this paper, we present a complete pattern based methodology for evaluating the heterogeneity of enterprise architecture. Our objective is to propose an evaluating methodology for guiding designers and architects in evaluating and improving the EA models.

Table 5: *Application of criteria for the three measures.*

Condition	Description of the application	The Herfindahl index $\sum_{i=1}^{n} p_i^2$	The Horvath index $p_1 + \sum_{i=2}^{n} p_i^2(2-p_i)$	The measurement of Entropy $-\sum_{i=1}^{n} p_i \ln(p_i)$
Additional Features	A company that uses a single DBMS A and introduces a second DBMS B. This leads to a change in the distribution (eg Case 1 {A: 100%} Case 2 {A: 80%, B: 20%}) Which evidently causes a higher level of heterogeneity.	Case 1 : 1 Case 2 : 0,68	Case 1 : 1 Case 2 : 0,88	Case 1 : 0 Case 2 :0,50
Change in classification	Replacing an application that uses DBMS B by an application that uses a DBMS A Case 1 {A: 80%; B: 20%} Case 2 {A: 90%; B: 10%} leads to lower heterogeneity	Case1 : 0,68 Case 2 : 0,82	Case 1 : 0,88 Case 2 : 0,92	Case 1 :0,50 Case 2 :0,32
Effect of small sets of elements	The introduction of a third DBMS - even just used by an application - leads to significant heterogeneity. Case 1 ({A: 50%, B: 50%} Case 2 {A: 50%, B: 48%, C: 2%})	Case 1 :0,50 Case 2 :0,48	Case 1 :0,62 Case 2 :0,60	Case 1 :0,69 Case 2 :0,78
Effect of proportional changes	The increase in the number of instances for each of the DBMS has no impact because the absolute number of elements does not have an impact on the heterogeneity.100 instances for DBMS A and B lead to The same heterogeneity as 300 cases of A and B. Case 1 {A: 50%, B: 50%} Case 2 {A: 50%, B: 50%})	Case 1 :0,5 Case 2 :0,5	Case 1 :0,62 Case 2 :0,62	Case 1 :0,69 Case 2 :0,69

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

REFERENCES

[1] Zachman, J. A, "A framework for information systems architecture". *IBM systems journal*, *26*(3), 1987,276-292.
[2] J.S. Davis,and R.J. LeBlanc, A Study of the Applicability of Complexity Measures. IEEE Transactions on Software Engineering, 14 (9), 1366.
[3] S.L Schneberger, S.L. and E.R McLean. The Complexity Cross: Implications for Practice. Communications of the ACM, 46 (9), 2003, 216-225
[4] Y.Y. Yusuf, E.O Adeleye,"A comparative study of lean and agile manufacturing with related survey of current practices in the UK", International Journal of Production Research 40 (17), 2002, pp 4545–4562.
[5] Lakhrouit, J., & Baïna, K. (2016). A pattern based methodology for analyzing enterprise architecture landscape. International Journal of Computer Science Issues (IJCSI), 13(6), 15.
[6] Arti Chhikara and R.S.Chhillar Analyzing the Complexity of Java Programs using Object-Oriented Software Metrics, International Journal of Computer Science Issues, Vol. 9, Issue 1, No 3, January 2012.
[7] F. Sabhia., Md. U. Ashraf., A Survey on Clustering Algorithms and Complexity Analysis, JCSI International Journal of Computer Science Issues, Volume 12, Issue 2, March 2015.
[8] Mocker M., 2009, "What is complex about 273 applications? Untangling application architecture complexity in a case of European investment banking". In: Proceedings of the 42nd Hawaii international conference on system sciences (HICSS), IEEE Computer Society, Washington, DC.
[9] Widjaja T., Kaiser J., Tepel D., Buxmann P., 2012, "Heterogeneity in IT landscapes and monopoly power of firms: a model to quantify heterogeneity". The international conference on information systems (ICIS), Orlando, FL.
[10] Jacquemin, A.P., and Berry, C.H. 1979. "Entropy Measure of Diversification and Corporate Growth," The Journal of Industrial Economics (27:4), pp 359-369.
[11] Schutz, T. Widjaja, J. Kaiser, 2013, "Complexity in enterprise architectures—conceptualization and introduction of a measure from a system theoretic perspective". the European conference on information systems (ECIS), Utrecht, The Netherlands.
[12] Saat, J., Aier, S., and Gleichauf, B. 2009. „Assessing the Complexity of Dynamics in Enterprise Architecture Planning-Lessons from Chaos Theory" in Proceedings of the 15th Americas Conference on Information Systems, San Francisco, California.
[13] Lagerstrom R., Baldwin CY., Maccormack AD., Aier S., 2013, "Visualizing and measuring enterprise application architecture: an exploratory telecom case". School working paper 13-103. Harvard Business, Cambridge, MA.
[14] Kwoka Jr, J.E. (1985). The Herfindahl Index in Theory and Practice. Antitrust Bull., 30, 915-947
[15] Garrison, C.B. and Paulson, A.S. (1973). "An Entropy Measure of the Geographic Concentration of Economic Activity". Economic Geography, 49 (4), 319-324.
[16] Berlinger, E. (1980). "An Information Theory Based Complexity Measure". Proceedings of the National Computer Conference. ACM.
[17] Davis, J.S. and LeBlanc, R.J. (1988). "A Study of the Applicability of Complexity Measures". IEEE Transactions on Software Engineering, 14 (9), 1366-1372.
[18] Gollop, F.M. and Monahan, J.L. (1991). A Generalized Index of Diversification: Trends in U.S. Manufactoring. The Review of Economics and Statistics, 73 (2), 318-330.
[19] Hall, A.D. and Hagen, R.E. (1969). "Definition of System, New York, Wiley".
[20] Hall, M., & Tideman (1967). Measures of Concentration. Journal of American Statistical Society, 62, 162-168.
[21] [U.S. Department of Justice and the Federal Trade Commission, 2010] U.S. Department of Justice and the Federal Trade Commission. (2010). Horizontal Merger Guidelines [Online]. Available: http://www.justice.gov/atr/public/guidelines/hmg-2010.html (Accessed 03.08.2012).
[22] [AW. Schneider, 2015, "Complexity and Diversity Management of Application Landscapes'', Software Engineering for Business Information Systems. Department of Informatics Technische Universität München, Germany.
[23] [Buckl et al., 2008] Buckl, s., ernst, a. M., matthes, f., and lankes, j. 2008b. "Enterprise architecture management pattern catalog".
[24] [Buckl et al., 2009a] Buckl S, Matthes F and Schweda CM. "Classifying Enterprise

© 2005 – ongoing JATIT & LLS
ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Architecture analysis approaches". In Poler R, van Sinderen M and Sanchis R (eds.) Enterprise Interoperability (Lecture Notes in Business Information Processing, vol. 38). New York: Springer-Verlag, 2009,pp. 66–79.

[25] [Buckl et al., 2009b] S. Buckl, A.M. Ernst, J. Lankes, K. Schneider, and C.M. Schweda. A Pattern based Approach for constructing Enterprise Architecture Management Information Models. In In Proceedings of the IEEE Enterprise Distributed Object Computing Conference, 2009. IEEE.

[26] Lakhrouit J., Baïna K., 2015 Evaluating enterprise architecture complexity using fuzzy AHP approach: Application to university information system. AICCSA 12th IEEE/ACS International Conference of Computer Systems and Applications, AICCSA 2015, Marrakech, Morocco, November 17-20, 2015. IEEE 2015, ISBN 978-1-5090-0478-2