

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/300207754>

Lightweight Metrics for Enterprise Architecture Analysis

Conference Paper · June 2015

DOI: 10.1007/978-3-319-26762-3_11

CITATIONS

3

READS

2,036

2 authors:



Prince Singh

Eindhoven University of Technology

14 PUBLICATIONS 64 CITATIONS

[SEE PROFILE](#)



Marten van Sinderen

University of Twente

117 PUBLICATIONS 759 CITATIONS

[SEE PROFILE](#)

Some of the authors of this publication are also working on these related projects:



IDS connector store and interoperability simulator for SMEs [View project](#)



Telelearning [View project](#)

Lightweight Metrics for Enterprise Architecture Analysis

Prince M. Singh^(✉) and Marten J. van Sinderen

University of Twente, Enschede, The Netherlands
{p.m.singh, m.j.vansinderen}@utwente.nl

Abstract. The role of an Enterprise Architecture model is not limited to a graphical representation of an organization and its dynamics. Rather, it is also a tool for analysis and rational decision making. If firms do not use their enterprise architecture model to aid decision making then they run the risk of underutilizing its true potential. This paper proposes seven easily computable metrics to measure criticality and impact of any element in an Enterprise Architecture model. These metrics will aid managers in decision making and are suitable for board room discussions.

Keywords: Enterprise architecture · Analysis · ArchiMate · Decision making · Degree · Connectedness · Influence · Criticality · Impact factor

1 Introduction

Enterprise Architecture (EA) has received a lot of attention from researchers over the last decade [6, 9, 10, 13] and it has developed into *the* most accepted tool for graphical representation of different domains (i.e. organizational, business, technology, application etc.) in a firm. Its use has been demonstrated for portfolio analysis, cost benefit calculation, relating business goals to IT infrastructure and requirement analysis [13, 14]. A major reason for this wide adoption of EA is its ability to give a holistic view of the current state of an enterprise and its internal and external dynamics [13]. Owing to these features, EA models are an obvious starting point to aid decision making in an enterprise [2, 3, 6, 7, 9, 14]. An EA model has to be analyzed to obtain relevant information to aid in decision making. The nature of such an analysis will depend on the decisions to be made. In this paper, we present lightweight metrics to aid decision makers in answering the following questions:

- how critical an organisational asset (process/function/service/hardware) is?
- whether one asset is more critical than the other?
- is a certain asset easily replaceable?
- how will the business be affected if an asset is removed or is unavailable?

These are typical questions which are often raised in board room discussions. They can only be answered if we know *how critical* a certain asset is for *value creation* [14, 15]. Since every asset of a firm can be modeled as an *element* in an EA model, by measuring the criticality of *that* element in a EA model, we can make an estimation of

the *criticality* of *that* asset in real life. In this paper we propose simple metrics to measure the *criticality* of an EA element. We don't propose a *works-for-all* analysis approach and neither do we propose to sideline existing analysis approaches. On the contrary, we propose that different analysis approaches (aimed at answering the same questions), including those presented in this paper, should be used together. Yet, these approaches must not be cumbersome or too complicated. The contribution of our approach are metrics to aid decision making, harnessing the information contained in an EA and bringing EA to the board room (with minimum effort).

The rest of the paper is organized as follows. Existing EA approaches and their classification are discussed in Sect. 2 followed by Sect. 3 in which we present our metrics. Usage of the metrics is demonstrated with an example case in Sect. 4. We discuss our results and limitations in Sects. 5 and 6 respectively. The paper is concluded in Sect. 7.

2 Background

There exist various EA analysis approaches. In Subsect. 2.1 we discuss some of these approaches, their classification and their applicability. In Subsect. 2.2 we briefly clarify why we used ArchiMate EA modeling language as a base for developing our metrics.

2.1 Previous EA Analysis Techniques

Two important classifications of EA analysis approaches are discussed here. The first classification, shown in Table 1, proposed by [3], classifies EA analysis approaches based on aspect of analysis (Body of Analysis), the time phase of the analysis (Time Reference), the technique used for analysis (Analysis Technique), property to be analyzed (Analysis Concern) and whether the analysis module accesses itself (Self Referentiality). In the second classification, proposed by [11], shown in Table 2, EA analysis approaches are classified based on analysis type.

Table 1. Classification EA analysis approaches [3]

Body of Analysis	Structure	Behavior Statistics	Dynamic Behavior
Time Reference	Ex-post		Ex-ante
Analysis Technique	Expert based	Rule based	Indicator based
Analysis Concern	Functional		Non functional
Self Referentiality	none	Single-level	Multi-level

In [2] a recent selection of 6 EA analysis approaches which are consistently referred to in literature is given. Below, we examine 4 of these approaches which align to our goal of measuring the criticality of EA elements.

Table 2. Classification based on analysis type [11]

Analysis Type	Example
Dependency	Assessing how would a business service be affected if any component it uses crashes, put offline or removed.
Coverage	Which application or processes contribute to a service or product.
Interface	Processes, functions or applications which have more than one relationship.
Heterogeneity	Finding out if there are different version of the same software being used in the firm
Complexity	Analyzing incoming and outgoing relationships between architecture elements.
Compliance	To determine whether the components comply with agreements, norms or standards.
Cost Analysis	Calculation of cost and maintenance of EA components.
Benefit Analysis	Determining the contribution of an EA element in realizing goals of the enterprise.

EA Analysis Using XML [5]. In this paper a methodology is proposed for static and dynamic analysis of an EA model primarily using XML (Extensible Markup Language) and RML (Rule Markup Language). A predefined XML vocabulary is used to first encode an EA model in XML for static analysis and then enriched by RML rules for dynamic analysis. This allows simulations to be run over architecture instances. The authors are of the view that by this methodology EA architects can understand the EA model better.

EA Analysis Using EID [9]. The authors investigate many languages to support the EA analysis for better decision making. They finally propose an extended influence diagram (EID) as the most suitable technique since it allows handling decision uncertainty at varied abstraction levels. The influence diagram can be seen as an algorithm which is run on an instance of the EA model (which serves as the data for the algorithm). EID can be used by the decision makers to decide between two or more alternative organizational states.

EA Analysis Using AHP [4]. A six-step method (having further sub-steps) involving Analytical Hierarchy Process (AHP) and Fuzzy AHP is presented in this paper. The motive is to evaluate different EA models (based on a quality attribute) and choose the one which best fits a given scenario. The attribute or property to be measured is broken down into multiple criteria and sub-criteria.

EA Dependency Analysis Using Fault Trees and Bayesian Networks (BN) [6]. Bayesian networks and fault trees are used in [6] to evaluate the feasibility of different scenarios. The authors propose an extension to an existing EA metamodel for application of their approach. For each scenario a new BN has to be made.

Some other EA analysis approaches include an EA analysis method based on Decision Design graphs [12] which is used to make a metamodel for capturing decision

relationship dependencies. The authors clearly mention the considerable effort which is involved in this approach. [7] proposes a four-step decision support system based on mapping EA elements and relationships to a Bayesian Belief Network (BBN). In [10] four separate metamodels are combined into one and then used for assessing application usage, system availability, service response time and data accuracy.

All these approaches address criticality of an EA element in their own way, yet there are some implementation roadblocks. Although these approaches claim and demonstrate sufficient accuracy of results yet they are either too mathematical and require special skills (e.g. [4, 9]), demand ample amount of effort (e.g. [4, 12]), use an intermediate mapping step (on which an EA is mapped) (e.g. [5–7, 10]), ignore the need to include human inputs or are just too cumbersome (e.g. [4, 5, 9]). Approaches which are too mathematical or demand significant manual effort usually don't have an associated tool to assist (or automate) the analysis, such that the underlying calculations are hidden. Moreover, if the EA model to be analyzed is mapped to a meta-model, a decision tree, a Bayesian network, an ontology, etc. then there is a risk of losing critical information contained in the EA model which can compromise the correctness of results. Ideally, an EA analysis approach should have *three* necessary features. *Firstly*, it should provide accurate and consistent results. *Secondly*, it should involve minimum of time, effort and cost. *Finally*, it should be easy to explain to high level managers. We are of the view that if the reasoning behind an analysis approach is too complex then its results would be *less* prone to acceptance and critique. Although the existing approaches promise consistent results yet they rarely stress upon making the analysis lightweight, fast and simple.

2.2 ArchiMate® EA Modeling Language

Our EA analysis metrics are based on ArchiMate® which is adopted by the Open Group® as a standard language for Enterprise Architecture modeling. It is widely used by academics and practitioners in the EA domain [15]. Since most of the EA analysis approaches use ArchiMate for evaluation, it will be easier for researchers to compare the results of our approach.

3 Metrics for Enterprise Architecture Analysis

Previous research has shown that ArchiMate models can be successfully transformed to graphs for analysis purpose [8, 14, 15]. Elements of the model have been treated as *nodes* of the graph and the relationships between the elements as *edges*. Concepts from graph theory like *in-degree* and *out-degree* have already been used in [8] for calculation of workload and performance measures. An EA model can also be traversed just like a graph as shown in [14, 15]. Our metrics build upon these previous results.

3.1 Metrics for Criticality Based on Relationships

Degree. Based on the number of relationships which an element has, we define the *degree* of that element. The degree of an element measures its connectedness with other

elements. The more relationships an element has, the higher its degree. Consider an example of a typical relationship between ArchiMate elements shown in Fig. 1. There exists a *used by* relationship between a Business Service and an Application service.



Fig. 1. An ArchiMate relationship



Fig. 2. Reading an ArchiMate relationship

Depending on the element, there are two views on this relationship (Fig. 2), (a) Business service *uses* (and thus, depends on) the Application service and (b) Application service *is used by* (and thus, supports) the Business service. Adopting this approach for all relationships, the *In-degree* of element E is defined as the number of elements on which E depends and the *Out-degree* is defined as the number of elements which depend on E . Table 3 shows for each ArchiMate relationship which view is used for calculating the in-degree and out-degree. A high value for in-degree is an indication that the element is *highly dependent* on other elements. The out degree of an element is a measure of *criticality* of that element. Therefore, an element having a high out-degree has high chances of being a *key resource of the firm*.

Table 3. ArchiMate relationships and their weights

Relationships	for calculating <i>In-degree</i>	for calculating <i>Out-degree</i>	Weight
Composition	Is composed of	Composes	8
Aggregation	Is an aggregation of	Aggregates	7
Assignment	Is performed by	Performs	6
Realization	Is realized by	Realizes	5
Use	Uses	Is used by	4
Accesses	Reads/Writes	Is Read/Written	3
Trigger	Is triggered by	Triggers	2
Flow	In flow (receive data)	Outflow (sends data)	1

Relationship Score. Two elements having the same out-degree can still be much different in their *criticality*, owing to the types of relationships involved. To incorporate the type of relationship in measuring the criticality of an element we use the concept of relationship *weight*. Each relationship in ArchiMate can be assigned a weight [14, 15] (Table 3) which is a measure of the coupling and thus the dependency between two elements [15]. The *Relationship Score (RS)* of any element E , is the weighted mean of all out-degree relationships

$$RS(E) = \frac{1}{n} \sum_{k=1}^n (W_k) \tag{1}$$

where, the summation is over all out-degree relationships of E , n = total number of out-degree relationships of E and W_k = weight of relationship k . As an example, we calculate the RS of 3 elements from Fig. 3 (see Tables 4 and 5).

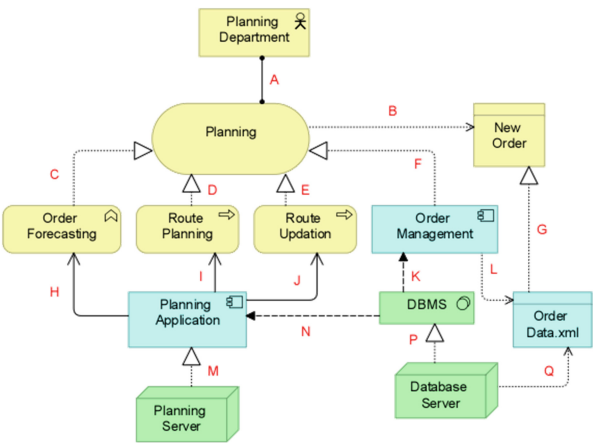


Fig. 3. An example EA snippet

Table 4. Relationship in Fig. 3 and their weights

Relationships	L	K	F	P	N	M	I	J	H
Weights	3	1	5	5	1	5	4	4	4

Table 5. Calculation of Relationship Score

Elements	In-degree relationship	In-degree	Out-degree relationship	Out degree	n	$\sum_{k=1}^n (W_k)$	RS
Order Management	L, K	2	F	1	1	5	5
DBMS	P	1	K, N	2	2	2	1
Planning Application	M, N	2	I, J, H	3	3	12	4

Absolute Relationship (AR). ArchiMate elements which are not directly connected by a relationship might still be indirectly connected [1]. Here we are interested in an indirect out-degree relationship between a source element and a target element, which exists if there is a path to traverse from source to target via out-degree relationships $r_1, r_2, r_3, \dots, r_n$. AR is the normalized weighted sum of all such out-degree relationships

$$AR(a, b) = \frac{1}{n} \sum_{k=1}^n \frac{1}{k} (W_k) \quad (2)$$

where, a is the target element, b is the source element, n is the number of out-degree relationships between b and a in a single path and W_k = weight of the relationship r_k .

The summation adds the effect of all the relationships from b and a . The division by k (inside the summation) gives each successive relationship in the path, a lesser weightage than the previous one. The division by n (outside the summation) normalizes the result of the summation. Had there been a direct relationship between a and b , its weight would have been $AR(a, b)$. There can be more than one path from b to a , resulting in more than one value of AR. In such a case, AR with higher value is chosen. This is done to pick the strongest relationship between a and b . If there does not exist any path from b to a , it implies that a and b are independent of each other and $AR(a, b)$ equal to 0. Taking the example of Fig. 3, we calculate the AR (*Planning, Planning Server*) and AR (*Planning, Database Server*), Table 6. Based on the AR values we can deduce that the Planning Server is more *tightly* coupled to Planning service than the Database Server.

Table 6. AR calculation

Planning Server → Planning		Database Server → Planning	
1 st path	M→H→C	1 st path	Q→G→B
2 nd path	M→I→D	2 nd path	P→K→F
3 rd path	M→J→E	Paths are different	
All paths are equal		Q→G→B	Q→G→B
n = 3		n=3	n=3
$\sum_{k=1}^3 \frac{1}{k} (W_k) = 8.67$		$\sum_{k=1}^3 \frac{1}{k} (W_k) = 6.5$	$\sum_{k=1}^3 \frac{1}{k} (W_k) = 7.16$
		$AR_1 = 2.16$	$AR_2 = 2.38$
$AR (\text{Planning, Planning Server}) = 2.89$		$AR (\text{Planning, Database Server}) = 2.38$	

3.2 Metrics for Criticality Based on Element Type

Till now we have not included the type of an element for determining its criticality. As an example, Quality Manager (modeled as a *Role* in ArchiMate) in most cases, will be more important than, say, a Display Terminal (modeled as *Node* in ArchiMate). This suggests that, to claim that an element is more critical than another element solely

based on relationships is not be sufficient. An element having an out-degree 2 can be more critical than an element having an out-degree 4.

Automatic Weight (atw). Although ArchiMate specification [1] do not assign a value to each element based on its type, yet, it does classify elements into active, behavior and passive elements. Elements are also classified based on layers i.e. Business, Application and Technology layer. This leads to a table of elements divided in 9 blocks as shown in Table 7. Behavior elements (e.g. business service, application service, infrastructure service etc.) are the most important in a firm because they *create value* and are the building blocks for *value creation* [15]. Passive elements (e.g. data object and representation) are less important. In between the behavior and passive elements lie the active elements (e.g. Actor, Application Component etc.).

Table 7. Automatic weights to ArchiMate elements

	Passive	Behavior	Active
Business	3	9	6
Application	2	8	5
Technology	1	7	4

Assigned Weight (asw). IT architects and process managers owing to their experience can make well informed and rational estimations over the criticality of elements in an EA model. Their knowledge and experience is impossible to automate. So, in addition to automatic weight, the IT-architect/process manager can assign an additional weight to element, called the *assigned weight*.

One way to do this is to use a Likert Scale between 1 and 5, where 5 denotes the *most important* elements and 1 denotes the *least important* element. *Weight* of an element is thus defined as the weighted mean of its automatic weight and assigned weight. We are of the view that assigned weight should be given more importance than automatic weight. Therefore, we multiply the assigned weight by factor t , called the *preference factor*.

$$W(E) = \frac{1}{(t+1)} (atw + t \times asw) \quad (3)$$

3.3 Metrics for Criticality Based on Relationship and Element Type

Importance of an Element. To get an even better measure of the criticality of an element, E , metrics based on its relationships and its type can be combined. The weight of an element E , $W(E)$ and its relationship score $RS(E)$ can be used to obtain its *importance*, $I(E)$. (See Eq. 4).

Impact Factor. Using *Absolute Relationship (AR)* between two elements (a,b) and the weight of element b , we can define *impact factor (IF)* which predicts the impact on a if b is removed. Different *IF*'s can be compared to assess the impact of two or more elements over a . (See Eq. 5).

$$I(E) = [W(E) \times RS(E)] \tag{4}$$

$$IF(a,b) = [W(b) \times AR(a,b)] \tag{5}$$

4 Application of Metrics for EA Analysis

Case Description: Figure 4 shows an EA snippet of a Newspaper Publisher [14]. To boost sales, it provides gifts to customers who purchase a monthly subscription. The business service *Customer Acquisition* performs the task of acquiring new customers and the business service, *Provide Gift to Customer* performs the task of obtaining, managing and sending the gifts to the customers. *Financial Service*, is responsible for managing the accounts. The application and infrastructure elements which support these services are also shown in Fig. 4. The IT department has suggested many changes in the IT infrastructure to higher level management. One of the proposed changes is to combine the *Web Server* and the *Database Server* to reduce maintenance cost and free up hardware. The management is considering to implement this proposal but before that it wants some questions to be answered. **Q1:** Which is/are the most critical element (s) for the business service *Provide Gifts to Customer*. **Q2:** Is, *Provide Gifts to Customer*, affected by a change in the servers? **Q3:** If the server(s) are down during migration, which of them would affect *Provide Gift to Customer* most?

4.1 Most Critical Element(s)

Concentrating on the right side of the model in Fig. 4, we see that 3 elements, namely, *Inventory Requirement Status*, *Logistic Application* and *Inventory List of Advertising Gift* have out-degree of 2 each. All other elements have a maximum out degree of 1. To further investigate the criticality of each of these three elements we calculate their *Relationship Score (RS)* as shown in Table 8.

Table 8. Calculation of RS

Elements	n	$\sum_{k=1}^n (w_k)$	RS
Inventory Requirement Status	2	8	4
Logistic Application	2	8	4
Inventory List of Advertising Gifts	2	6	3

RS shows that *Inventory Requirement Status* and *Logistic Application* have same criticality. Let us now calculate the *weight* and *importance* of these elements to

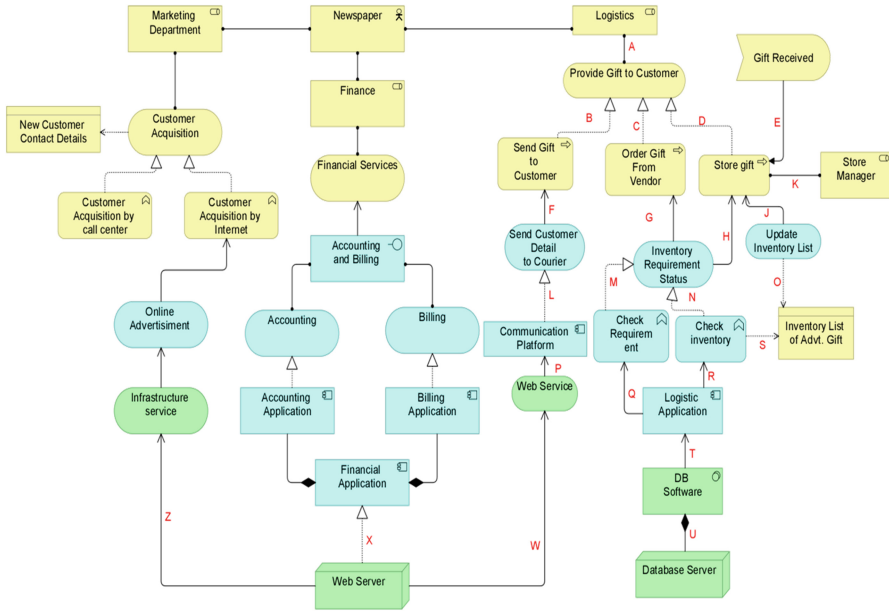


Fig. 4. An Example EA Model in ArchiMate

investigate their criticality further. For the purpose of this example, assume the assigned weights of Inventory Requirement Status and Logistic Application are 2 and 4 respectively and preference factor (t) is 2. Their automatic weights will be the same (Table 7), i.e. 8. Therefore,

$$\begin{aligned}
 W_{LogisticApplication} &= \frac{1}{(2+1)} (8 + 2 \times 4) = 5.33 \text{ and} \\
 I(LogisticApplication) &= 21.32 \\
 W_{InventoryReq.Status} &= \frac{1}{(2+1)} (8 + 2 \times 2) = 4 \text{ and} \\
 I(InventoryReq.Status) &= 16
 \end{aligned}$$

A1: Based on the values of *Importance* we conclude that *Logistic Application* is the most critical element, followed by *Inventory Req. Status* for the business service *Provide Advertising Gift*.

4.2 Impact of the Servers

There is no direct relationship between *Provide Advt. Gift to Customer* and the servers. We compute AR for both servers to measure the indirect relationships (Table 9).

A2: Based on the calculation of *AR* we can conclude that each server has a relationship with the service, having weights 2.36 and 1.93.

If we assume the assigned weights of both the servers to be the same (say 3) and t as 2, their weights turn out to be 3.33 each. Thus,

Table 9. Calculation of AR

Database Server → Provide gift to customer		Web Server → Provide gift to Customer	
1 st path	U→T→Q→M→G→C	Path	W→P→L→F→B
2 nd path	U→T→Q→M→H→D		
3 rd path	U→T→R→N→G→C		
4 th path	U→T→R→N→H→D		
All paths are equal (same order of relationships)			
n = 6 $\sum_{k=1}^6 \frac{1}{k} (W_k) = 14.21$		n=5 $\sum_{k=1}^5 \frac{1}{k} (W_k) = 9.67$	
AR (Provide Gift., Database Server) = 2.36		AR (Provide Gift., Web Server) = 1.93	

$$I.F(Providegifttocustomer, DatabaseServer) = 3.33 \times 2.36 = 7.85$$
$$I.F(Providegifttocustomer, WebServer) = 3.33 \times 1.93 = 6.42$$

A3: Comparing the value of *I.F*, the Database server, impacts the service 20 % more as compared to the web server.

5 Discussion

We have presented and demonstrated the use of lightweight metrics for criticality and impact of EA elements in an ArchiMate model. These metrics will aid managers in decision making. In Table 10 we use the classification proposed by [3] (Table 1) to classify our approach. Moreover, based on Table 2, our approach falls under the following analysis types: dependency, coverage, interface, complexity and benefit analysis. Due to space constraints the justification for the classification under each of these analysis types is not discussed further. Figure 5 shows a diagrammatic representation of the metrics.

Table 10. Classification of our approach according to [3]

Body of Analysis	Structure		
Time Reference	Ex-post	Ex-Ante	
Analysis Technique	Expert based		Indicator based
Analysis Concern			Non functional
Self Referentiality	none		

6 Limitation and Future Works

The metrics based deduction depends on the correctness of the EA model. Therefore, if the EA model is incorrect it might lead to incorrect metric values leading to incorrect deductions. This is called *empirical uncertainty*. We have used an example case of

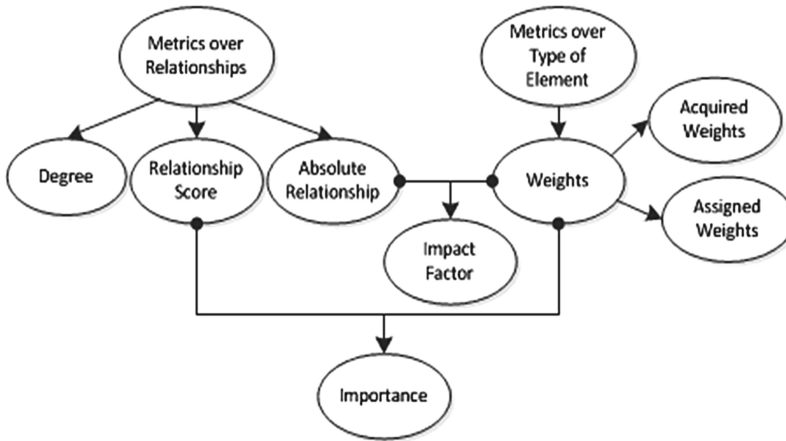


Fig. 5. Diagrammatic representation of the metrics

ArchiMate, but application of the metrics over different EA modeling techniques is yet to be investigated. Our research also opens many areas for future research. *Firstly*, the metrics can be incorporated into existing EA modeling tools e.g. BiZZDesign Architect®, such that the values for the metrics could be directly obtained from the tool. *Secondly*, combination of the metrics with other analysis method has to be investigated. We would like to re-iterate that our metrics might be used in conjunctions with other analysis techniques. This is an ongoing research. *Lastly*, interviews with practitioners in the field of EA are needed to evaluate the applicability and correctness of the metrics.

7 Conclusion

EA analysis is important for decision making yet it was found that most EA analysis in literature are too arduous. We have presented intuitive, simple and lightweight metrics for EA analysis. These metrics are based on basic properties of elements in an ArchiMate EA model. They are neither too complex nor use an intermediate mapping tool. These metrics will aid the decision making in companies by measuring the criticality and impact of assets. By the use of these metrics EA models can become more relevant for board room discussion and trigger greater understanding among managers. Only then the true potential of EA's can be harnessed.

References

1. The Open Group.: ArchiMate 2.0 Specifications. Van Haren Publishing, Zaltbommel (2012)
2. Antunes, G., Caetano, A., Borbinha, J.: Enterprise architecture model analysis using description logic. In: EDOCW, pp. 237–244. IEEE, Ulm (2014)

3. Buckl, S., Matthes, F., Schweda, C.M.: Classifying enterprise architecture analysis approaches. In: Poler, R., van Sinderen, M., Sanchis, R. (eds.) IWEI 2009. LNBIP, vol. 38, pp. 66–79. Springer, Heidelberg (2009)
4. Davoudi, M.R., Sheikhvand, K.: An approach towards enterprise architecture analysis using AHP and fuzzy AHP. *Int. J. Mach. Learn. Comput.* **2**, 46–51 (2012)
5. De Boer, F.S., Bonsangue, M.M., Jacob, J., Stam, A., van der Torre, L.: Enterprise architecture analysis with XML. In: HICSS. IEEE, Hawaii (2005)
6. Franke, U., Flores, W.R., Johnson, P.: Enterprise architecture dependency analysis using fault trees and bayesian networks. In: *Spring Simulation Multiconference* (2009)
7. Holschke, O., Närman, P., Flores, W.R., Eriksson, E., Schönherr, M.: Using enterprise architecture models and Bayesian belief networks for failure impact analysis. In: Feuerlicht, G., Lamersdorf, W. (eds.) ICSOC 2008. LNCS, vol. 5472, pp. 339–350. Springer, Heidelberg (2009)
8. Iacob, M.E., Jonkers, H.: Quantitative analysis of enterprise architectures. In: Konstantas, D., Bourrières, J.-P., Léonard, M., Boudjlida, N. (eds.) *Interoperability of Enterprise Software and Applications*, pp. 239–252. Springer, London (2006)
9. Johnson, P., Lagerstrom, R., Narman, P., Simonsson, M.: Extended influence diagrams for enterprise architecture analysis. In: EDOC. IEEE (2006)
10. Narman, P., Buschle, M., Ekstedt, M.: An enterprise architecture framework for multi attribute information systems analysis. *Softw. Syst. Model.* **13**, 1086–1116 (2014)
11. Niemann, K.D.: *From Enterprise Architecture to IT Governance*. Vieweg & Sohn Verlag, Wiesbaden (2006)
12. Plataniotis, G., de Kinderen, S., Proper, H.A.: Relating decisions in enterprise architecture using decision design graphs. In: *17th IEEE International Enterprise Distributed Object Computing Conference*, pp. 139–146. IEEE Computer Society (2013)
13. Sasa, A., Krisper, M.: Enterprise architecture patterns for business process support analysis. *J. Syst. Softw.* **84**, 1480–1506 (2011)
14. Singh, P.M.: *Integrating Business Value in Enterprise Architecture Modeling and Analysis*. University of Twente, Enschede (2013). <http://essay.utwente.nl/63695/>
15. Singh, P.M., Iacob, M.E., Jonkers, H., van Sinderen, M.J.: Modeling value creation with enterprise architecture. In: *ICEIS 2014-16th International Conference on Enterprise Information Systems*, pp. 343–351. SCITEPRESS, Lisbon (2014)