Abstract
The power of visual analysis of Enterprise Architecture (EA) model tends to diminish when we deal with large and complex viewpoints. However, we still can extract useful information from them. In this work, we apply ideas from the Design Structure Matrix (DSM) theory to derive enterprise architecture viewpoints from primary models already available at organizations. In order to do so, we propose four derivation mechanisms. Subsequently, we apply network analysis metrics to those new viewpoints in three empirical datasets. With our suggested approach, it was possible to derive cross-layer viewpoints, amplifying the analysis possibilities for experts through the creation of “non-standard/implicit” visualizations of the enterprise architecture which have proven useful to those experts.

Keywords
Enterprise architecture, analysis, design structure matrix, data, empirical.

Introduction
After thirty years with Zachman’s framework (Zachman, 1987) and more recent releases of modeling languages such as Archimate (The Open Group, 2017), viewpoints – a well-established concept from software architecture (Steen, 2004) – represent a core concept in enterprise architecture (EA) modeling. Diagrams and other visual representations have proven to be appropriate means to create EA views that express the EA from the perspective of specific concerns and stakeholders (Steen, 2004). They are also an essential result of any EA management initiative (Grigoriev and Kudryav, 2013).

However, their power of representation tends to diminish when modeling complex systems. In some situations, performing a visual analysis alone is a serious limitation that precludes working with a large amount of data (long lists or dense diagrams) (Grigoriev and Kudryavtsev, 2013). Matrix-based methods seem to be an alternative in modeling and analysis. Indeed, the very nature of enterprise architecting requires a lot of cross mappings (goals-processes, processes-applications, etc.), which is supported by most EAM tools (Grigoriev and Kudryav, 2013). To that end, a matrix-based method known as the Design Structure Matrix (DSM) is being applied in system engineering, system of systems, project management, and product engineering as a way to analyze complex systems. Yet, despite the similarities between EA and those fields of endeavor, few studies have explored this modeling approach in an EA context.

Thinking of EA as an intuitive matrix, rows, columns, and cells can represent EA layers and components. Once we have that EA matrix, and using data derivation concepts initially applied by van Buuren et al. (2004) and Matthews et al. (2014), artificial viewpoints can be derived providing new analytic perspectives and new questions for stakeholders. To explore that, we elaborate on the work of van Buuren...
et al. (2004), first expanding the possibilities for data derivation by taking well-known EA viewpoints as inputs and then applying network analysis techniques commonly used in DSM research to provide new analysis data. Thus, in this paper, we look at EA viewpoints (matrices) as EA DSMs (or, equivalently, EA networks) to propose a meta-model of derived viewpoints (EADV-MM) and perform EA analysis as a way to answer the following questions: How can we derive a set of EA viewpoints using indirect relations among their components? (RQ1) Which questions might arise based on those new viewpoints? (RQ2) The main contribution of this study will be conceptual: the EADV-MM and deriving operators, approaching a still lacking, concrete way of using DSM principles in EA analysis. We also go further to demonstrate the applicability of a subset of the derived viewpoints in three empirical datasets using network analysis metrics.

This paper is structured as follows: we introduce key concepts such as EA, viewpoints and network analysis in the next section. Thereafter, we present our research design. In the “The Derivation operators and the EADV meta-model” section that follows, we discuss our derivation mechanisms. An “Empirical analysis” section illustrates through three examples how the derived EA viewpoints can be applied in practice. Finally, we draw our conclusions.

Key concepts

Enterprise Architecture

EA is defined in a variety of ways. We initially adopt the definition proposed in the literature review of Schütz et al. (2013), which is also supported by TOGAF (2011): EA is a system formed by four subsystems, namely Business, Data/Information, Application, and Infrastructure (or Technology) Architecture. After conducting a recent survey of EA analysis, Barbosa (2016) identified the need to include an additional “value” layer (based on the ARMOR language proposed by Quartel et al., 2009). Therefore we consider those five layers (value, business, information, application, technology).

Enterprise Architecture, Views, and Viewpoints

There is a plethora of frameworks and models to integrate the diverse architecture descriptions, as shown in Lankhorst et al. (2004), so we also advocate an approach by which architects and other stakeholders can define their own views and viewpoints of the EA. Two initial concepts need to be defined: a “view” is a representation of an entire system from the perspective of a related set of concerns, while a “viewpoint” establishes the purposes and audience for a view and the techniques for its creation and analysis (Lankhorst et al., 2013). Viewpoints are at the heart of several EA frameworks, such as Zachman (1987), ArchiMate (The Open Group, 2017), and TOGAF (The Open Group 2011). Steen et al. (2004) provide a comprehensive discussion of EA viewpoints.

Viewpoints and DSMs

Each viewpoint may be modeled as a DSM, which is a square matrix with identical row and column labels that provides a simple and compact visual representation of a complex system (Browning, 2001). The presence or absence of a relation between each pair of components is represented by a “1” or “0” (a binary DSM), as Figure 1 shows.

![Figure 1: Graphical representation of a binary DSM](image)

The values contained in the cells may have diverse semantics, such as the frequency of data exchange between two applications. The DSM and domain-mapping matrix (DMM) are the two central elements of
a multiple domain matrix (MDM). A DMM is a special kind of DSM that maps the relations between exactly two different domains (in the EA context, a domain is equivalent to a layer). An example can be found at the bottom of Figure 2, displaying a matrix cell with a blue square (which could represent an application component) and a green triangle (which could represent a business process). An MDM (the entire Figure 2 is a MDM) extends the capabilities of the DMM by integrating multiple domains (or multiple EA layers, e.g., business, application, and technology, as represented by triangles, squares, and circles in Figure 2) together with different kinds of relations (e.g., “deployed on”, “connected with”, “depends on”, as represented by dashed, thicker, and continuous lines in Figure 2) among components from those layers.

Figure 2: Representation of MDM’s concepts – (adapted from Furtmeier and Tommelein, 2010).

**DSMs and EA**

Performing a visual analysis alone with EA models is a serious limitation; it precludes working with a large amount of data. The system representation afforded by DSMs has led to their increasing use in a variety of contexts, including product development, project management, systems engineering, and organization design (Browning, 2001). Following the intuitive idea of seeing EA as a system that can be represented as matrices, this paper explores the applicability of DSMs to model EA-derived viewpoints (based on indirect dependencies) and their potential to offer new analytic insights.

**Related works**

Bartolomei (2007) examined frameworks for system engineering modeling and presented an improved framework called Engineering Systems Multiple-Domain Matrix (ES-MDM), composed of six domains (system drivers, stakeholders, objectives, functions, objects, and activities). Similar to the model of Hollauer et al. (2015), each row/column pair of the ES-MDM represents a “view” of the system. Kreimeyer (2009) worked with MDMs and focused on the process architecture, using DSMs and structural metrics as a management toolbox to find structural weak spots (points of improvement). Furtmeier and Tommelein (2010) explored the application of MDMs as a mapping process, similar to what our aim in this paper. However, they focused on a lean design context, while EA is our target.

According to Vakkuri (2013), the DSM seems to be a good choice for EA analysis for two reasons. First, past EA frameworks (DoD, 2010; The Open Group, 2011) propose using square matrices in modeling connections between system components. Second, many types of analysis approaches have been developed for the DSM. However, there are few studies in the literature in which the DSM is used explicitly in the EA context, with the exception of Lagerström et al. (2013a, 2013b), Vakkuri (2013), Baldwin et al. (2013), and Santana et al. (2016). Lankhorst et al. (2013) discuss the importance of transformation models to convert models from any language to EA concepts and, thus, generate viewpoints in a tool-supported environment. Likewise, Grigoriev and Kudryav (2013) emphasize the
potential of using matrices to model the EA and propose a modeling tool to integrate business process engineering with EA. However, those authors do no actual analysis in their paper. Regarding the use of indirect relations in DSMs and EA contexts, Baldwin et al. (2013) define an EA DSM to analyze the impact of indirect changes and identify the structural architectural arrangements with the “hidden structure method.” Matthews et al. (2014) consider multimodal dependencies among components to form a dependency graph to deduce indirect dependencies between applications and IT projects. Another interesting approach in this direction is presented in van Buuren et al. (2004), who define a composition operator to generate stakeholder-oriented viewpoints. In this paper, we are inspired by Bartolomei (2007), Furtmeier and Tommelein (2010), and van Buuren et al. (2004) to present a set of essential EA layers and components, their relations, and their deriving operators to create new EA viewpoints.

**Research Design**

According to the taxonomy of Wohlin and Aurum (2015), this work can be classified as exploratory and descriptive applied research. For our research, we have adopted the design science research method (Hevner et al., 2008). Our method, described in Table 1, parallels Peffers et al. (2007):

<table>
<thead>
<tr>
<th>DSR phase</th>
<th>Research contextualization</th>
</tr>
</thead>
<tbody>
<tr>
<td>Identify problem</td>
<td>Organizations have big IT landscapes, which makes it difficult to perform a holistic visual analysis with EA viewpoints.</td>
</tr>
<tr>
<td>Define solution objectives</td>
<td>Increase the analysis capabilities of EA viewpoints by proposing newly derived ones.</td>
</tr>
<tr>
<td>Design and development</td>
<td>Model EA viewpoints as DSMs to benefit from the related analysis techniques such as network analysis, which are our knowledge base (Hevner et al., 2008).</td>
</tr>
<tr>
<td>Demonstration</td>
<td>Describe a proof-of-concept demonstration of a subset of the proposed artifact.</td>
</tr>
<tr>
<td>Evaluation</td>
<td>Evaluate our artifact using expert opinions and instantiation.</td>
</tr>
<tr>
<td>Communication</td>
<td>We make the research exposition in this paper (to the academic audience) and the analysis report for the experts (management audience).</td>
</tr>
</tbody>
</table>

**Table 1: DSR phases and contextualization (Hevner et al., 2008)**

Based on the EA layers and their potential combinations introduced, we have an immediate set of primary EA viewpoint candidates available for our research. Although not original ones, those viewpoints are used in combination with network analysis. Since the application of network analysis with viewpoints modeled as networks has also been addressed to some extent in the literature (e.g., Mathews et al., 2014), we do not claim originality here either. However, we use derived data differently in our analysis. We identify EA derived data, that is, the data artificially generated based on primary data transformed by derivation operators, which in turn generates EA-derived viewpoints. We refer to the set of derived viewpoints as the EADV-MM. Our hypothesis (H1) is that those new viewpoints, in combination with network metrics, provide new insights that can be used by experts. The EADV-MM and their respective operators combined with the network analysis metrics thus constitute our proposed artifacts.

We collected EA data from three different organizations as described in Table 2.

**Table 2: Datasets description**

The first operates in the media industry and employs several thousand people at its headquarters in Germany and about twenty international sites. The second is a large multi-industry player also headquartered in Germany and operating in a few other European countries. The third is an automotive company with multi-billion dollar revenues and more than 40,000 employees. While the EA management
function of the first organization was established several years ago, the second organization’s EA management function was about only two years old at the time of data collection. The third organization has no officially established EA management function at all; however, we found a high level of control over process documentation at the company. In Table 2, we refer to the resulting datasets from these three organizations as Dataset01, Dataset02, and Dataset03, respectively (due to anonymity reasons, we cannot be more specific and, e.g., provide network figures with named nodes in this document). We built two of the datasets manually from raw documents such PDFs and datasheets and interviews with experts. One dataset was exported directly from an EA modeling tool. We applied visual analysis (Ramos et al., 2014) and network analysis metrics such as in and out degrees, betweenness, closeness, and eigenvector centralities (Scott, 1992) to investigate the derived EA viewpoints.

The Derivation Operators and the EADV Meta-Model

The derivation of relations is studied in network theory as co-affiliation networks derived from affiliation networks, incidence matrices, or bi-partite data (Borgatti and Halgin, 2011). Analogously in DSM theory, derived DSMs can be generated based on DMMs (Lindeman et al., 2009). The operating mechanisms are depicted in Figure 3 and explained in the following.

**Figure 3: The four derivation operators (adapted from Lindemann et al, 2009).**

*Derivation by affiliation.* This operator uses the classical concept of co-affiliation networks (Bogatti and Halgin, 2011) as Figure 3 (a) depicts. In this case, the components “App1” and “App2” appear connected in the derived network AppxApp since they are co-affiliated with the component “P1” in the original network (AppxProc). This could, for example, represent applications that belong to the same line of business.

*Derivation by intra-domain relation or “transfer relation.”* To illustrate this, the dashed line in Figure 3 (b) is created between “Bu1” and “Bu2” based on the continuous lines that connect the component “Bu1” with component “App1” and “Bu2” with “App2” (both are inter-layers relations), on the one hand, and the line that connects applications “App1” and “App2” (intra-layer relation within the application layer), on the other hand. With this operator, we “transfer” or “project” the existing relation from one layer (Application) to another (Business).

*Derivation by attribute.* In this case, to generate a derived network we use more information than only the component and its relations. For instance, we may have a unimodal network “TechTec” and its component attribute “vendor,” as Figure 3 (c) depicts. We decide to abstract the technology component and represent it only by its attribute “vendor.” In this case, rather than considering a relation between “T1” (e.g., an application server) and “T2” (e.g., a database server), we are looking for a relation between two technology vendors (e.g., Microsoft and Oracle). This operator may derive networks that might be unimodal or bimodal according to the primary networks used as input, as Table 3 shows.

*Derivation by transitivity.* Based on the concept of transitivity, this operator has its function illustrated in Figure 3 (d). Van Bureen et al. (2004) explored the same concept as “composition of relations.”

Table 3 summarizes the four derivation operators and their respective transformation.
Deriving Viewpoints for Enterprise Architecture Analysis

Table 3: Operators used to derive EA network data

<table>
<thead>
<tr>
<th>Type of original network (input)</th>
<th>Type of derivation</th>
<th>Operator symbol</th>
<th>Example</th>
<th>Derived network (output)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 Unimodal</td>
<td>Derivation by attribute</td>
<td>$\Phi$</td>
<td>$\Phi$(Tec $\times$ Tec)</td>
<td>Unimodal derived network (e.g., VendorxVendor)</td>
</tr>
<tr>
<td>2 Bimodal</td>
<td>Derivation by attribute</td>
<td>$\Phi$</td>
<td>$\Phi$(Tec $\times$ Tec)</td>
<td>Bimodal derived network (e.g., AppxVen)</td>
</tr>
<tr>
<td>3 Bimodal</td>
<td>Derivation by affiliation</td>
<td>$\simeq$</td>
<td>$\simeq$(AppxBu)</td>
<td>Bimodal derived network (e.g., BuxBu or AppxApp)</td>
</tr>
<tr>
<td>4 Bimodal and Unimodal</td>
<td>Derivation by intra-layer relation or transfer relation.</td>
<td>$\uparrow$</td>
<td>$\uparrow$(AppxBu, AppxApp)</td>
<td>Bimodal derived network (e.g., BuxBu or AppxApp)</td>
</tr>
<tr>
<td>5 Multimodal</td>
<td>Derivation by transitivity or composition</td>
<td>$\equiv$</td>
<td>$\equiv$(TecxAppxBu)</td>
<td>Bimodal Derived network (e.g., TecxBu)</td>
</tr>
</tbody>
</table>

Table 4: The EADV-MM partial instantiation

<table>
<thead>
<tr>
<th>EA Layers</th>
<th>Business</th>
<th>Application</th>
<th>Technology</th>
</tr>
</thead>
<tbody>
<tr>
<td>Value</td>
<td>GoalxGoal from $\uparrow$(GoalxBp, BpxBp)</td>
<td>BpxBp from $\simeq$(GoalxBp)</td>
<td>BuxBu from $\uparrow$(BuxAppxTec),TecxTec</td>
</tr>
<tr>
<td>Business–BUS</td>
<td>BusxBus from (=BuxBp)</td>
<td>BuxBus from (=BuxApp)</td>
<td>$\uparrow$(=BuxAppxTec),TecxTec</td>
</tr>
<tr>
<td>Information-INF</td>
<td>BoxBo from (=BoxApp)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Application-APP</td>
<td>TecxTec from $\uparrow$(TecxApp, AppxApp)</td>
<td>TecxVendor from(Tecx$\Phi$Tec)</td>
<td></td>
</tr>
<tr>
<td>Technology-TEC</td>
<td>AppxVendor from (Appx$\Phi$Tec)</td>
<td>VendorxVendor from ($\Phi$Tec $\times$ $\Phi$Tec)</td>
<td></td>
</tr>
</tbody>
</table>

4.2 A meta-model for derived EA viewpoints

The EADV-MM was created in the form of an MDM. The layers and components necessary for the creation of the meta-model were selected based on the five EA layers discussed earlier. As we said before, previous works have already dealt with primary EA viewpoints; however, our work goes further by proposing derivation mechanisms applicable to those viewpoints and by discussing how to perform EA structural analysis with these new “data sources.” We group some possibilities of EA-derived viewpoints in Table 4, which represents our EADV-MM instantiation.

Empirical analysis

Our analysis began using the derived EADV-MM to brainstorm various analysis scenarios during two discussion sessions of around one hour each, held with EA practitioners responsible for the three datasets presented earlier. The choice of which derived viewpoints should be used was made considering the data available for our research. We divide the derived viewpoints analysis according to the datasets:

- Dataset01

Business objects (BOs) are information entities processed by applications during business process execution. The first organization provided us with information about the BOs defined and applications operating on those BOs. At first, we knew of no associations among the BOs and thus wanted to check
how the entities were related, considering the access by the same application(s). The derived BoxBo has 127 BOs and 1440 relations with the following semantic: if two or more BOs are supported by the same application, then they must appear connected in the derived BoxBo network (derivation by affiliation).

At the group level analysis (Santana et al., 2017), the expert can analyze visually the represented level of encapsulation/modularization of applications as per their operations on BOs. In other words, in the ideal encapsulation the network topology should appear rather clustered, indicating a high level of agility in terms of potential changes in the application landscape. Figure 4 (a) depicts the BoxBo, produced by the “circular layout for groups” algorithm from the ORA tool\(^1\) (23 isolated BOs were removed).

Some entities will, of course, be manipulated by more than one application. In this case, at the component level, the betweenness centrality becomes an important indicator of components that act as “brokers” between clusters. The blue components represent the top-10 betweenness values. We asked the EA experts of the given company to comment on the results; they indicated that the highlighted components also represent important BOs that require special architectural attention. Continuing with the Dataset01, we now apply the affiliation operator in the AppxTec network, which has 183 components. The idea behind the design of this derived network is that each time we have a set of technologies used to make an application work, those technologies together constitute some sort of local ecosystem that can be reinforced by looking at all technologies across all applications in the network. In the end, we can build a visual map of the technologies and check how they relate to each other. Analyzing Figure 4 (b) at the network level, it is possible to identify a very dense network with a core-periphery architecture in which a bunch of technologies shape the core, including MS SQL Server, MS Windows Server, and C# (blue nodes), for example. Clusters can only be found in the periphery.

![Figure 4: (a) BoxBo derived network and highlighted betweenness outliers; (b) TecxTec derived network and highlighted eigenvector outliers.](http://www.casos.cs.cmu.edu/projects/ora/)

On the right part of the network, one cluster contains the J2EE component (a black node) linked to other technologies such as TopLink, GlassFish, Oracle Database, and Solaris. This association is a technology stack regularly found in the market. However, the fact that the overall network is only weakly clustered pointed out that reusable stacks in this company are either not defined or highly overlapping, which provided a valuable insight to the EA experts. We also apply the eigenvector centrality to identify the top-10 most structurally well-connected components (technologies that are used a lot and are connected with others frequently used). Among these components are z/OS, ADABAS, MS Windows Server, C#, and .NET, for example; they are shown in green in Figure 4 (b), providing additional insights into where technologies may best be replaced in order to reduce costs, maintenance efforts, etc.

- Dataset02:

\(^1\) http://www.casos.cs.cmu.edu/projects/ora/
From a TecxTec network (composed of 169 components), we derived the VendorxVendor network by applying the derivation by attribute operator ($\Phi$). Figure 5 depicts the resulting network:

![VendorxVendor derived network and eigenvector outliers highlighted](image)

**Figure 5 – VendorxVendor derived network and eigenvector outliers highlighted**

With this network, we explore the vendors’ relationships, which we consider important information for technology migration strategies pursued by the company. With this sort of a vendor map, one can monitor and evaluate external technology discontinuities and updates (e.g., market withdrawals, acquisitions) that could impose technology changes upon the organization. The VendorxVendor network has 27 components. After applying the eigenvector centrality, we show the top-10 eigenvector values (highlighted as blue nodes in Figure 5), indicating dependencies on specific vendors. According to the EA experts we interviewed, the result was generally expected for vendors. Nevertheless, one observation was made regarding the IBM node, which was not listed as a top node in the network, thus signaling the importance of complementing the structural analysis results with expert knowledge.

- Dataset03

In the given organization, business units (Bu) execute different workloads depending on the associated process phase. We consider two phases in our analysis – PROD (phase I) and KONZ (phase II) – and pose this operational question: Can we identify key structural Bu considering the interaction in different process phases? Thus, our analysis concern here is to find the key Bu that might be important when it comes to involving the right people (Bu) in the decision-making process regarding business process changes. As the input, we took the BuxBp network and applied the transfer operator ($\Upsilon$) to generate a derived BuxBu network. With this derivation method, if two processes BP1 and BP2 are connected, an artificial connection is created between their respective Bu in the BuxBu network. The hypothesis H2 formulated by the experts is then broken down as follows: H2.1: In Phase I, the focus of the project management unit (Bu1) should be fairly continuous as they manage all activities; H2.2: In Phase II, the focus will be more on the technology people, with a ramp up to production and logistics and possibly to purchasing; H2.3: Overall, in Phase II, design engineers will be fairly central, as they function as a sort of information hub around which all technical concept design is focused. From Table 5, we notice that Bu1, Bu5, Bu2, and Bu7 also appear in different rankings of network measures.

<table>
<thead>
<tr>
<th>TOP Out-degree Bus</th>
<th>TOP eigenvector Bus</th>
<th>Most recurrent Bus</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bu5: Product management</td>
<td>Bu1: Project management</td>
<td>Bu1: Project management</td>
</tr>
<tr>
<td>Bu1: Project management</td>
<td>Bu5: Product management</td>
<td>Bu1: Project management</td>
</tr>
<tr>
<td>Bu7: Total integration</td>
<td>Bu2: Controlling</td>
<td>Bu2: Controlling</td>
</tr>
<tr>
<td>Bu3: Quality</td>
<td>Bu7: Total integration</td>
<td>Bu5: Prod. Management</td>
</tr>
<tr>
<td>Bu2: Control</td>
<td>Bu13: Production preparation</td>
<td>Bu7: Total integration</td>
</tr>
</tbody>
</table>

**Table 5. Network analysis at the component level for BuxBu Prod (Phase I)**

For H2.2, we obtained the following results: Bu13, responsible for production aspects, became imperative in Phase II (detected by high in-degree centrality components and eigenvector centrality); Purchasing (Bu15) had importance detected by high in-degree centrality and also was among the most recurrent
outliers; Validation and integration (Bu7, Bu10, Bu3) aspects were a focus in Phase II, detected by eigenvector centrality, most recurrent outliers, and out and in-degree centralities. Although identified by the experts a priori, logistics did not appear as a focus in Phase II. In conclusion, we found H2.3 to be partially supported. For hypothesis H2.3, integration, validation, and preparation for production activities were the main ones in Phase II. So, H2.3 was also supported.

Table 6 summarizes the viewpoints analyzed, targeted concerns (thus answering RQ2), and respective analysis methods according to the taxonomy defined by the Santana et al. (2017).

<table>
<thead>
<tr>
<th>Derived network</th>
<th>Analysis concern</th>
<th>Operational question</th>
<th>Analysis methods</th>
</tr>
</thead>
<tbody>
<tr>
<td>Unimodal BoxBo derived from = (AppxBo)</td>
<td>Application’s modularization of Bo</td>
<td>Are related Bo’s manipulated by a specific and modularized set of applications?</td>
<td>Visual cluster analysis (cluster level); betweenness centrality (component level)</td>
</tr>
<tr>
<td>Unimodal TecxTec derived from = (AppxTec)</td>
<td>Technology ecosystem; key technologies</td>
<td>What are the key structural technologies that support the application architecture and how do they relate to each other?</td>
<td>Eigenvector centrality (component level); topological network analysis.</td>
</tr>
<tr>
<td>Unimodal VendorxVendor derived from ΦTecxTec</td>
<td>Vendors ecosystem; key vendors</td>
<td>What are the key structural vendors and what is the frequency of their associations in the technology ecosystems?</td>
<td>Visual analysis (network level); eigenvector centrality (component level).</td>
</tr>
<tr>
<td>Unimodal BuxBu from ↑ (BuxBp)</td>
<td>Evolution of the whole of Bus along the business process execution.</td>
<td>What are key structural Bus considering a specific business process interaction set?</td>
<td>Network analysis metrics</td>
</tr>
</tbody>
</table>

Table 6. Network analysis at the component level for BuxBu Prod (Phase I)

Conclusions

In this paper, we propose an EA-derived viewpoint meta-model (EADV-MM) and four derivation operators (thus answering RQ1). The operators transfer concepts already applied in network theory and DSM theories to the EA analysis context. With the approach suggested, it is possible to derive cross-layer viewpoints, amplifying the analysis possibilities for EA experts. In the cases presented, the relations for components of BoxBo, TecxTec, VendorxVendor and BuxBu networks, implicit in their original networks, were created to offer new analysis perspectives. We highlighted some individual components and emerged clusters that were evaluated by EA experts of each dataset. We claim that the derivation rules we present can be used for the creation of “non-standard/implicit” stakeholder-oriented visualizations. The case of the BuxBu network is a clear example of the usefulness of derived data: the company did not have the relations between the Bus mapped, but had only the relations between of Bus and the BPs in which the Bus take part and the network of BPs itself (BuxBp and BpxBp networks). Nevertheless, despite the absence of the primary data about the Bus, it was possible to confirm the experts’ perceptions and highlight key structural Bus during business processes execution. Less pragmatic but still intuitive, the derived TecxTec and related networks bring to the analysis the relations among technologies and vendors and may help in understanding their influence and evolution in the enterprise from a temporal point of view. Therefore, we claim that H1 and H2 were supported by our results.

Despite the high number of possible combinations of viewpoints allowed with the EADV-MM and operators, we explored only a subset of them; therefore, the list of derived viewpoints presented is not exhaustive. It is important to clarify that EA data modelers still must define the meaning of the derived relations of each particular derived viewpoint. In general, the experts need to analyze the potential of those derived components and relations further, making sense of them and validating their utility and implications for EA managerial reality. This problem should be approached by testing alternative viewpoints, measures, and visualization methods, and seeing which have the best “value” for the EA analysis (Sosa et al., 2007a). This can foster further research using the respective DSMs, together with network analysis, to support the development of EA analysis tools. The manual and time-consuming information collection process employed is a downside of our approach.

REFERENCES


