

Uncovering the Runtime Enterprise Architecture of a Large Distributed Organisation

A Process Mining-Oriented Approach

Robert van Langerak, Jan Martijn E.M. van der Werf^(✉),
and Sjaak Brinkkemper

Department of Information and Computing Science, Utrecht University, Utrecht,
Netherlands

{r.p.vanlangerak,j.m.e.m.vanderwerf,s.brinkkemper}@uu.nl

Abstract. Process mining mainly focuses on analyzing a single process that runs through an organization. Often organisations consist of multiple departments that need to work together to deliver a process. ArchiMate introduced the Business Process Cooperation Viewpoint for this. However, such models tend to focus on modeling design time, and not the runtime behavior. Additionally, many approaches exist to analyze multiple departments in isolation, or the social network they form, but the cooperation between processes received little attention.

In this paper we take a different approach by analyzing the runtime execution data to create a new visualization technique to uncover cooperation between departments by means of the Runtime Enterprise Architecture using process mining techniques. By means of a real-life case study at a large logistic organization, we apply the presented approach.

Keywords: Process mining · Enterprise architecture · Data analytics · Business analytics · Runtime enterprise architecture

1 Introduction

In larger organisations, departments work jointly to deliver the services or products of that organisation. Capturing this cooperation is part of the domain of Enterprise Architecture (EA) [29]. EA consists of principles, methods, and models to design and realize an enterprise's organisational structure, business processes, information systems, and infrastructure. An important aspect within organisations is the cooperation between different departments in the overall processes of the organisation. ArchiMate [13] introduced the Business Process Co-operation Viewpoint (BPC) to model this explicitly. These models mainly focus on the design of cooperation: which processes and departments within an organisation are allowed to communicate. The runtime behaviour, i.e., whether and when communication occurs, and the possible execution orders are typically left out.

Process mining [2] offers many opportunities to assist the enterprise architect in uncovering the runtime behaviour of their EA. In process mining, many algorithms exist to discover processes (e.g. [5, 8, 10, 16]), to check for conformance (e.g. [3, 7, 22]), and to enhance process models (e.g. [11, 25]). Although the process is viewed from different perspectives [17], such as the case, process and resource perspective, the organisational perspective [23] has been given little attention. Approaches like PM² [24] and Process Diagnostics [6] focus on the overall process within an organisation, rather than focusing on how the different departments within the organisation contribute to deliver its service. Consequently, for larger organisations where multiple departments cooperate to deliver their services, current process mining techniques are hard to apply.

As a running example, consider the insurance company InsComp. The organisation has three departments: the Policy Department, the Claim Department and the Financial Administration. InsComp delivers two services: the issuing of policies and the handling of claims, where the former is the responsibility of the Policy Department, and the latter of the Claim Department. The Claim Department sometimes asks the Policy Department to check a policy. Once a claim is approved, the Financial Administration is instructed to compensate the claim. As InsComp has a problem with the Claim Handling service, they want to obtain insights into the cooperation and functioning of the different departments and locations.

In this paper, we want to close the gap between the static descriptions created in EA and the runtime environment in which all these processes have been implemented. We do this by addressing the research question: *How can the analysis of runtime execution data facilitate the visualisation of the actual business process cooperation in enterprise architecture?* To pursue this research, we applied the objective-centered approach in Design Science Research [20], to design and evaluate visualisation techniques of runtime execution data in the field of enterprise architecture.

Based on the successful application of process mining in other fields [2, 4, 6, 10], we propose to apply process mining techniques for the analysis of questions about the quality of the actual different departments and their cooperation. We do this by introducing the Runtime Enterprise Architecture (REA) of an organisation, which uses the runtime operation data from the processes operated within the organisation. This allows us to create new visualisations to uncover the involvement of departments, their cooperation, and their relative achievements in the process.

In the remainder of this paper we make the following contributions:

- Incorporation of the runtime behaviour of an organisation into the Business Process Co-operation Viewpoint of Enterprise Architecture (Sect. 2);
- Visualisation techniques to uncover the Runtime Business Process Co-operation View of an organisation (Sect. 3); and
- Showing the applicability and possibilities of the techniques through a case study in a large parcel distributor in the Netherlands (Sect. 4).

2 Runtime Enterprise Architectures

To model the different departments within an organisation and how these cooperate to deliver the services of an organisation, ArchiMate 3.0 [13] introduced the Business Process Co-operation Viewpoint (BPC) [15]. This viewpoint shows the relation between the business processes and their surroundings, and can be used to create a high-level design of business processes within their context and to provide insight into their dependencies [15]. The BPC viewpoint of our example organisation InsComp is shown in Fig. 1.

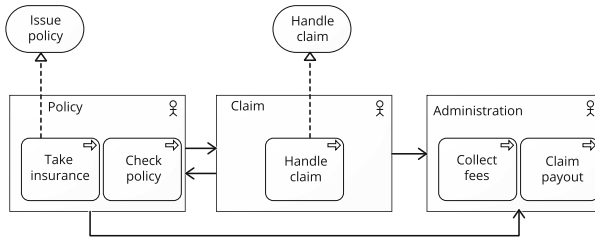


Fig. 1. Business process co-operation viewpoint

The BPC viewpoint reflects the allowed cooperations at design time. Whether in real life this blue print is always followed is a complete different question. With the logging capabilities of current Process-Aware Information Systems (PAISs), it is possible to record reality in the form of audit trails or event logs [26]. These event logs are input for process mining.

Key in process mining is that each event is related to a process instance of some business process. Process mining focuses on analyzing a single process, whereas in an organisation many different processes run intertwined. For this we define the *Runtime Enterprise Architecture* (REA) as the set of structures and metrics to capture and analyze the runtime behaviour of that organisation based on its Enterprise Architecture. In this paper, we focus on the dynamic behaviour of the BPC viewpoint, the Runtime Business Process Co-operation View (RBPC). As for process mining in general, this approach relies on the assumption that for each event it is known to which activity (and thus process) it belongs.

2.1 Meta-Model of the Runtime Business Process Co-operation View

The conceptual model that maps the relevant event log concepts to the concepts of the BPC viewpoint is shown in Fig. 2. On the left, the relevant elements of the BPC viewpoint are depicted. The gray elements are the default elements of ArchiMate. *Organisation* and *Department* are specializations of the ArchiMate

element *Business Actor*. An *Organisation* has a hierarchical structure of *Departments*, and delivers some *Business Service*. A *Business Service* is implemented by one or more *Business Processes*. *Activities* in a *Business Process* form *Cooperation*. A *Cooperation* is always initiated by an *Activity* (relation *from*) and concluded by an *Activity* (relation *to*).

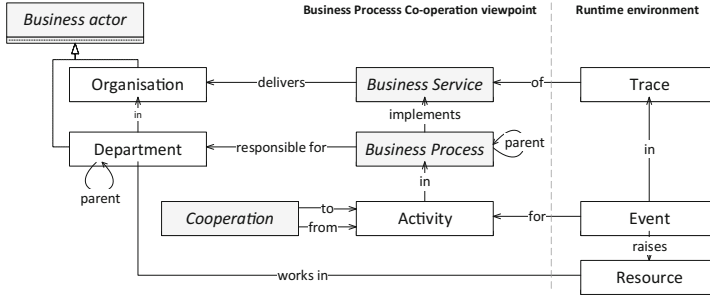


Fig. 2. Conceptual model of runtime business architectures

At runtime, *Business Services* are instantiated, resulting in *Traces*, that flow through the organisation. For a *Trace*, *Events* are raised by executing *Activities*. Possibly, the *Resource* is recorded as well. Notice that in many organisations, traces are identified by some global identifier that is used throughout the business service or organisation.

2.2 Runtime Business Process Co-operation View

At design time, different cooperations can be modelled in the BPC view. The definition of the concepts and relations are directly derived from the conceptual model in Fig. 2. Let c be a cooperation. From the conceptual model, we can define the following types of cooperations.

Intra-process A cooperation occurs within the same process, i.e.

$$in(from(c)) = in(to(c));$$

Inter-process A cooperation occurs between two different processes, i.e.,

$$in(from(c)) \neq in(to(c));$$

Intra-departmental A cooperation within the same department (possibly between different processes), i.e.,

$$responsible_for(in(from(c))) = responsible_for(in(to(c)))$$

Inter-departmental A cooperation between different departments, i.e.,

$$responsible_for(in(from(c))) \neq responsible_for(in(to(c)))$$

In this model, a cooperation can be both inter-process and intra-departmental at the same time, if the cooperation is between two business processes for which the same department is responsible.

3 Uncovering Cooperations

To obtain insight in the cooperations within an organisation, we first discuss how to discover cooperations. Next, we present a new visualisation technique for cooperations, the Runtime Business Process Co-operation View that visualises the runtime behaviour of an organisation, rather than only focusing on the design time, as is current practice in EA.

3.1 Discovering Cooperations Using Process Mining

Several techniques have been proposed in process mining to analyze both inter and intra organisations [1], such as social network analysis [23], artifact-centric techniques [21] and feature discovery [28]. Social Network Analysis (SNA) focuses on identifying nodes and their relationships [19]. A social network consists of nodes and a set of relationships or links. In [23], the authors use event logs to generate a social network of the resources within the event log. In process analysis this derived SNA can be used to identify resources in a network, and to show how these resources interact. Additionally, SNA can be used to study patterns within an organisations network and enabling organisations to use these patterns to create competitive advantages [14].

Whereas process mining relies on the assumption that each process instance belongs to the same business process, the artifact-centric approach assumes that the process instances are manipulated by artifacts [18], and tries to discover the processes and interactions of these artifacts [9, 21]. Artifact-centric mining is used to discover a process by using the artifacts that are present in the process and is therefore often used to create better process models for real life or physical processes [12].

Recent research focuses on the use of process discovery techniques to construct functional architectures [28] by relating software execution data to features. In this way, it is possible to discover the communication protocols between features from the behavioural profile [27].

Each of these process mining techniques can be used to enhance the existing BPC viewpoint by updating the *Cooperations* from the event log. Next step is to visualise and quantify the cooperations found at runtime.

3.2 Visualising the Runtime Business Process Cooperation

At runtime, many different metrics are available about the business processes and their cooperations. The current visualisation of the BPC viewpoint in ArchiMate only focuses on depicting the EA at design time. Consequently, we require new visualisations to provide useful insights in the organisation.

In this paper, we introduce the Runtime Business Process Co-operation View (RBPC), which is an interactive representation of process execution data. As an example, a RBPC of InsComp is depicted in Fig. 3. The view combines a chord diagram and a sunburst diagram. Chord diagrams have been developed to visualise large sets of arcs by bundling all arcs between two nodes. Similarly, a

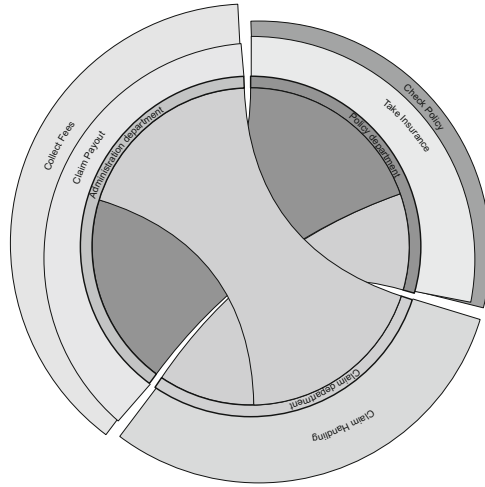


Fig. 3. Runtime business process co-operation view

sunburst diagram provides a visualisation for the frequency of elements: the size of each element is proportional to its frequency.

The view consists of two circles. The inner circle is a chord diagram that represents the cooperation between the different departments. Each part represents a department. The length of each part is determined by the centrality of the node in the social network, which is a combination of the number of cooperations each part initiated and concluded, and the size of the flows represent the volumes traveling between the nodes. The colour of the flow is determined by the node that initiates most cooperations. The outer circle is a sunburst that indicates the percentage of process instances handled by the departments. In case the departments run multiple processes, each process is depicted as a layer, with a height proportional to the number of process instances handled by that process. In this way, the view provides a high level overview of the processes, and their cooperation, and at the same time insights in how frequent these processes have been executed.

Additionally, we can colour the outer circle with other metrics, such as the overall duration of the cooperations, or the conformance of the different processes within the department.

For the running example InsComp, an example RBPC is depicted in Fig. 3. The three parts represent the three departments, and their processes as layers in outer circle. The colour scale represents the duration time of the process. From its size in the diagram, we directly see that the Financial Department plays a large role in the organisation. Also, we see that a third of the cooperations with the Policy Department are with the Claim Department. Most cooperations of this department are with the Financial Administration. From this view, we can conclude from the flows that a third of the communication of the Policy

Department comes from the Claim Department, and from the colouring we conclude that the “Check Policy” process in the Policy Department is a bottleneck, and that in the Financial Administration the duration for the “Claim Payout” process is above average, which would explain why the Claim Handling service of InsComp requires attention.

4 Validation of the Runtime Enterprise Architecture

We aim to validate the proposed Runtime Business Process Co-operation View with a non-trivial case study in a large logistics company. The selected case organisation is one of the largest mail and parcel distributors in the Netherlands, referred to as SendIT. In 2014, the organisation addressed 2,705 million mail items and 142 million parcels. SendIT has 18 distribution centers throughout the Netherlands for the distribution of parcels. Each center represents a specific area for delivery and is responsible for that part of the distribution process. Consequently, there are 18 instances of the same departmental processes, and cooperations between all 18 distribution centers. Each center has its own facilities to record the process execution data. Each of the instances can be analyzed and compared using this data. Currently, the organisation lacks proper visualisations of the performance of the different centers, and their cooperation in the different processes. In this case study, we use the data to compare the different distribution centers on process execution and performance, and to discover the different cooperations between departmental processes.

4.1 Distribution Process and Scan Trails

Each distribution center is responsible for three main processes: sorting shifts (B), sorting routes (J), and delivery (I). The intended happy flow of the parcel delivery process is depicted in Fig. 4.

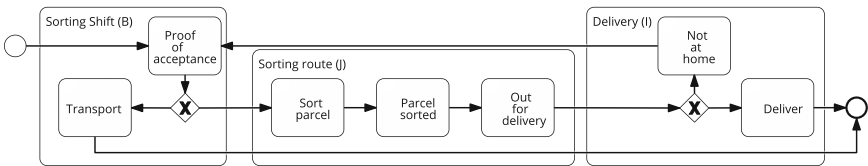


Fig. 4. Intended process model of the happy flow of the parcel delivery at SendIT

During the night, process B is executed at the centers. Each parcel is scanned, and based on the postal code, the center decides to either sort itself, or to transport it to a different center. In the morning, once all parcels of the other centers are received, each parcel is scanned again, and passed to the sorting route process (J), that determines the parcels each delivery man has to handle. Once

all packages are sorted to the different routes, the delivery men start the delivery process (*I*). In case a parcel cannot be delivered at an address, it returns to the sorting shift.

Parcels are represented by unique barcodes. In each step of the process, the parcels are scanned, which results in adding a new scan value to its respective barcode. A scan value has its own definition and possible consequences for the further distribution of this parcel. In total there are more than 150 possible scan values. Each scan value consists of a label indicating the process (value), and the activity in that process (reason). Examples of possible scan values include entering a distribution center, with scan value *B1*, ‘Proof of Acceptance’, placing a parcel on a conveyer (*J1*), out for delivery (*J5*), and delivered (*I1*).

Table 1. Scan trail of a single package representing the most frequent happy flow

Barcode	Date	Time	Value	Reason	Description
1B1671337	10/02/2016	00:49:25	B	1	Proof of acceptance
1B1671337	10/02/2016	09:34:25	J	1	Sorted
1B1671337	10/02/2016	09:34:26	J	40	Sorted on route
1B1671337	10/02/2016	10:06:03	J	5	Out for delivery
1B1671337	10/02/2016	12:28:45	I	1	Delivered

The scan trail of a parcel is a sequence of all its scan values and their occurrence. Each scan value always consists of a character and a number, together with a timestamp. Several happy flows exist for these trails: flows where nothing went wrong and the parcel was delivered. The most frequent happy flow is depicted in Table 1. For each parcel a trail can be exported from the different PAISs of the distribution centers at SendIT.

4.2 Data Selection and Extraction

SendIT handles roughly 30.000 parcels per distribution center per day, resulting in approximately 160M events per month. Consequently, we had no option than to take a random sample from this data set. The data selected for this case study covers the whole month February in 2016. For each center, a dataset was created with at most 500 parcels per day. For these parcels, the scan trails were extracted and combined into a large dataset for a distribution center. The total number of events per center is depicted in Table 2. As a last step, all datasets were combined into a single dataset for analysis. This resulting dataset contains 136.575 scan trails.

Each dataset had to be prepared before it can be analyzed. An excerpt of the trail is depicted in Table 1. For example, the date and time values had to be merged into a single timestamp, as this is required by the different process

Table 2. Events per distribution center (DC). In total, the dataset contains 1.555.492 events divided over 136.575 scan trails

DC	# Events
1	111.001
2	84.309
3	85.935
4	103.887
5	72.345
6	90.116
7	81.362
8	124.722
9	66.348
10	79.234
11	74.715
12	83.476
13	85.049
14	87.188
15	80.159
16	77.148
17	88.714
18	79.784

Table 3. Structure of an event in the event logs after conversion. The Scan letter and number together form the activity name to which the event is related. The Barcode is the instance identifier.

Attribute	Example
Barcode (ID)	1B1671337
Timestamp	10/02/2016 00:49:25
Scan letter	B
Scan number	01
Combinedscan	B01
Center	LOC 3

mining tools. The Value and Reason attributes in the scan trail are merged to create the activity name for each event. Both values were added to the event log, to be able to analyze the event log on different levels of abstraction, as the Value and Reason represent the business process, and the corresponding activity, respectively. The final structure of an event is shown in Table 3.

4.3 Analysis

For the analysis of the datasets, the open source software ProM [26] is used. To exclude parcels that are not yet delivered, we filtered the dataset by removing all scan trails that do not contain an activity with an I-value. Next, multiple analyses have been executed with ProM to identify the structures and flows in the dataset. To create an overview of the entire process, the organisational process is visualised first. Next, a generic departmental process model is created from the most occurring traces in the dataset. Lastly, the subprocesses of the departmental processes are identified. By combining these models using Archimate, a static enterprise architecture is created.

The organisational model is created using the Social network miner plugin [23]. The result is depicted in Fig. 5. It is a complete graph, i.e., all distribution centers send parcels to each other.

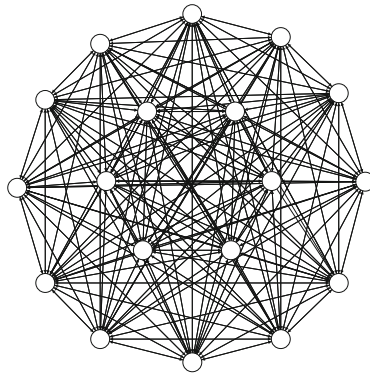


Fig. 5. Organisational model as mined with the social network miner of ProM

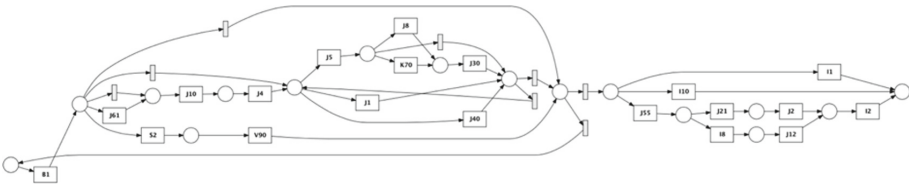


Fig. 6. 10 most occurring traces

Departmental Models. As a complete process model representing all 136.575 scan trails returns a spaghetti-like model, we decided to apply Occam’s razor, and as a first step [6], created a new dataset containing only the 10 most occurring traces of the 18 distribution centers. The ten most occurring traces cover together almost 68% of all scan trails. Applying the Inductive Visual Miner [16] resulted

in the Petri net as depicted in Fig. 6. The process starts with a B1 event, then the sorting process is started (J-valued events), after which the parcel is delivered (I-valued events). From the same dataset containing the 10 most frequent scan trails, we discovered for each of the processes a separate process model. The J process is depicted in Fig. 7.

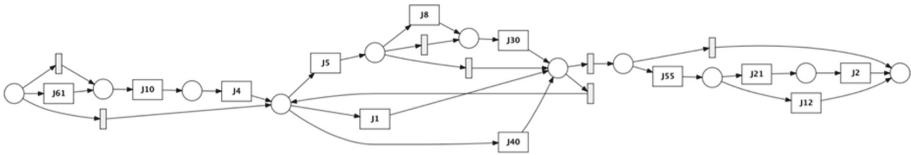


Fig. 7. Model of process J in isolation

To check the degree of conformance of the processes of the different distribution centers the logs are replayed through the petri net using the “Replay for conformance checking plugin” in ProM. This results in a fitness value per center (Table 4), indicating how well a model represents a log, on a scale from 0 (no fitness) to 1 (complete fitness). Overall, the mined process shows a high fitness (average: 0.945), indicating that our razor of only taking the 10 most occurring traces approximates the overall process very well. Additionally, we used Disco¹ to analyze the median duration time for each center. All values range between 14,3 and 23,3 h, the average is 17,4 h between first scan and delivery.

Enterprise Architecture. The Enterprise Architecture comprises the different processes, and abstracts from the detailed activities. To discover how the different processes cooperate, we decided to create an additional organisational model in which the hand-over of work is analysed between the different processes by taking the Value as resource. This resulted in the model depicted in Fig. 8(a). Based on this organisational model, it is possible to create the BPC viewpoint of the static EA of SendIT, shown in Fig. 8(b).

Runtime Business Process Co-operation View. One of the main drivers of SendIT is to compare the runtime behaviour of the different centers, and the amount of parcels that is transported between the centers. For this, we created two separate RBPC views for SendIT. Both RBPC views use the location changes of the parcels in the chord diagram, and the respective number of scan trails handled in the center for the length of the sunburst. For the colouring schema, the former is based on the median duration of scan trails, the latter is based on the fitness of the sub processes at each center.

To define the chord diagram of both RBPC views, we first analyzed the mined social network (Fig. 5), where the location changes have been defined as

¹ <https://fluxicon.com/disco/>.

Table 4. Fitness values and median durations in hours per distribution center (DC), calculated with the Replay for Conformance Plugin of ProM on the model depicted in Fig. 6. The fitness is a score between 0 and 1, and indicates how well the instances adhere to the given process model. The average fitness is 0.945.

DC	Fitness	Dur. (h)
1	0.911	15.8
2	0.956	16.0
3	0.960	14.3
4	0.916	18.6
5	0.964	17.9
6	0.870	15.9
7	0.957	15.8
8	0.943	16.8
9	0.954	15.1
10	0.941	19.1
11	0.954	19.6
12	0.955	23.3
13	0.968	19.7
14	0.954	18.7
15	0.952	17.6
16	0.949	15.1
17	0.948	16.0
18	0.949	17.6

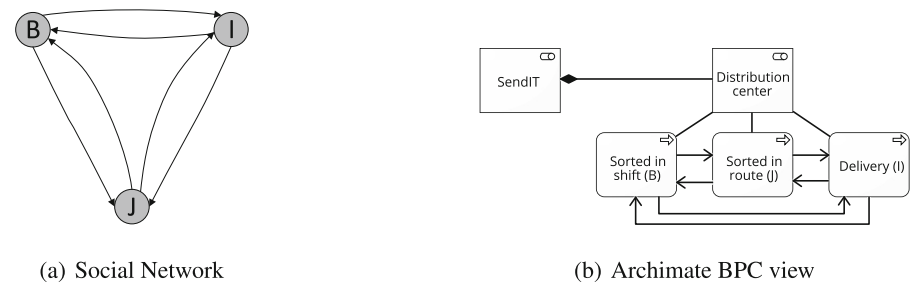


Fig. 8. The discovered BPC view of SendIT generated from the social network miner of ProM

the hand-over of work between the centers [23]. As a first step, their respective frequencies have been analyzed. As each parcel is always at exactly one location, we counted the number of consecutive event pairs with different locations.

As a next step, these numbers have been normalized using the total number of location changes. In this way, location changes become relative to each other, and all add up to 100%. Based on this data, the chord diagram is constructed.

To determine the length of each center in the RBPC views, i.e., the number of scan trails handled by the center, we analyzed for each center the ratio between parcels with a B scan value, i.e., the number of parcels that arrive, with the number of parcels that have an I scan value, i.e., the number of parcels delivered by the center. If this ratio is high, most parcels that arrive at the center are distributed in the region, i.e., the center handles many parcels, whereas if the ratio is low, most parcels are transported to different centers, thus the center handles few parcels.

Next, for each center we determine the size of the internal process by normalizing the amount of parcels each process handles with the ratio determined for that center. For example, if a center has 50% of B scan values, 25% of J scan values, and 25% of I scan values, parts J and I will be similar in size, and the size of B has the size of J and I combined. As each center has three processes, sorting shifts (B), sorting routes (J), and delivery (I), this results in three rings for each center in the RBPC views.

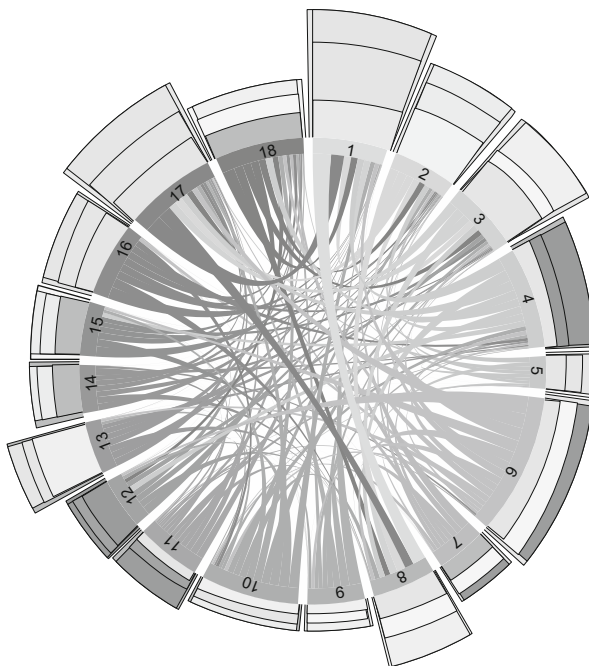
These two steps create the basis for the RBPC views using the d3 javascript framework². The first RBPC view uses the median duration, as depicted in Table 4, for colouring its elements. The lowest median duration is coloured green, the highest is coloured red. Each center is assigned a gradient colour relative to the highest and lowest median durations, resulting in the view depicted in Fig. 9(a).

Analyzing the two RBPC views, we directly observe that location 6 sends out many more parcels than that it handles, as its size is large, while its length is relatively low. Location 1 handles most parcels, as it has the largest length. Another observation in Fig. 9(a) is that at 5 centers (i.e., 28%) the delivery process have a longer than average median duration (red), and that at 3 centers (i.e., 17%) the process takes shorter than average. Only at one location, the sorting to shifts process (B) takes much longer than average. Similarly, analyzing Fig. 9(b), we observe that all centers conform the B process, whereas the other two processes have more deviations. Only location 1 and 4 show outlying fitness values for the I and J processes.

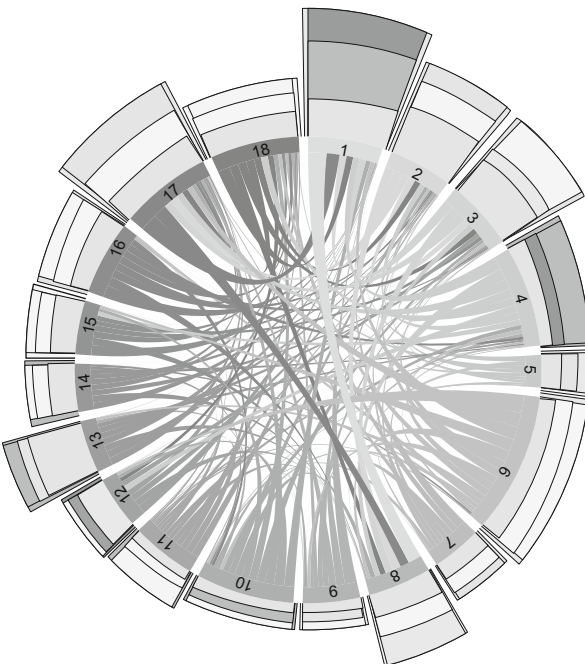
4.4 Expert Validation

To validate the results and different visualisations of the RBPC, we presented the results to two stakeholders of SendIT, being a process manager and a process data analyst, who were not involved in executing the case study. The goal of the interviews was to identify the perceived usefulness of the design, and to obtain possible improvements and additional feedback. The technical details of the design were discussed, as well as the possible business applications of the solution. Additionally, the feasibility within SendIT was part of the discussion.

² <https://d3js.org/>.



(a) Median duration



(b) Fitness

Fig. 9. Two runtime business process co-operation views of SendIT. Each part in the diagram corresponds to a department (1–18). Each layer in the department represents the processes *B*, *I* and *J* (from inner to outer layer)

The validation started with discussing the standard process mining models, which are also presented in this section. The stakeholders mentioned that these models were “interesting to see, but are not really useful in day to day practice”. There is no real chance for active management using only the petri nets. However, they could be useful to get some “low level process insight”.

The RBPC was perceived very useful. The idea and the ability to present real time process data to the “Operations Management Division” of the organisation was met with an enthusiastic response. This would lead to more proactive management of the distribution process throughout each day. Currently, most of the improvements actions that are undertaken are based on negative outliers of the past week. Using the RBPC on a real time basis would enable Operations management to pro-actively correct problems that occur in a distribution center.

Additionally, the RBPC could be used to bridge the gap between Operations and Organisation management. The management is not aware and not interested in low level process information. Their main objective is to fulfill the KPIs that were set for a certain period. Using the RBPC, it is possible to convert low level process data to high level process information. This information subsequently can be used to actively used when discussion the current execution and performance of the entire organisation.

Concluding, the RBPC was received positively. Some remarks were made how the RBPC could be adapted to better fit the organisation, but this is a matter of implementation that varies per organisation. The stakeholders mentioned that “in todays world the trick is to create something that can covert the abundance of data into information, so it can be used by someone who is not familiar with the data. This is what the RBPC does.”.

5 Conclusions and Future Work

Current Enterprise Architecture mainly focus on modeling an organisation design-time only. In this paper, we propose the Runtime Enterprise Architecture (REA) that enhances the EA of an organisation with runtime execution data. To visualise the cooperation between departments and their process within an organisation, we propose the Runtime Business Process Co-operation viewpoint that visualises the runtime cooperation between departments and the relative volumes and quality of the different processes at the departments. The visualisation combines the chord diagram for visualising cooperations with the sunburst visualisation for the volume of the processes. The colouring schema is used to depict the quality of the processes.

To illustrate the visualisation, we applied it on one a large logistics organisation to analyze parcel transportation between departments. Initial validation at the organisation shows the perceived usefulness of the visualisation technique. Another limitation of the case study is that although the proposed visualisation technique in itself is quite general, the case study organisation had no concurrency in their processes. Generalization of the validation results therefore require further experimentation.

Many different viewpoints exist in EA modeling. In this paper we focused mainly on the Business Process Co-operation Viewpoint, but we envision the proposed techniques to be extended to different viewpoints as well. As the proof of the pudding is in the eating, we plan on fully automating the visualisation technique in ProM to perform more in-depth case studies to explore further analysis and visualisation possibilities of the technique.

References

1. van der Aalst, W.M.P.: Intra- and inter-organizational process mining: discovering processes within and between organizations. In: Johannesson, P., Krogstie, J., Opdahl, A.L. (eds.) *PoEM 2011. LNBIP*, vol. 92, pp. 1–11. Springer, Heidelberg (2011). doi:[10.1007/978-3-642-24849-8_1](https://doi.org/10.1007/978-3-642-24849-8_1)
2. van der Aalst, W.M.P.: *Process Mining: Discovery, Conformance and Enhancement of Business Processes*. Springer, Heidelberg (2011)
3. van der Aalst, W.M.P., Adriansyah, A., van Dongen, B.F.: Replaying history on process models for conformance checking and performance analysis. *Wiley Interdisc. Rev. Data Mining Knowl. Discov.* **2**(2), 182–192 (2012)
4. van der Aalst, W.M.P., Reijers, H.A., Song, M.: Discovering social networks from event logs. *Comput. Support. Coop. Work* **14**(6), 549–593 (2005)
5. van der Aalst, W.M.P., Weijters, A.J.M.M., Maruster, L.: Workflow mining: discovering process models from event logs. *Knowl. Data Eng.* **16**(9), 1128–1142 (2004)
6. Bozkaya, M., Gabriels, J., van der Werf, J.M.E.M.: Process diagnostics: a method based on process mining. In: *eKNOW 2009*, pp. 22–27. IEEE Computer Society (2009)
7. Buijs, J.C.A.M., Reijers, H.A.: Comparing business process variants using models and event logs. In: Bider, I., Gaaloul, K., Krogstie, J., Nurcan, S., Proper, H.A., Schmidt, R., Soffer, P. (eds.) *BPMDs/EMMSAD -2014. LNBIP*, vol. 175, pp. 154–168. Springer, Heidelberg (2014). doi:[10.1007/978-3-662-43745-2_11](https://doi.org/10.1007/978-3-662-43745-2_11)
8. Buijs, J.C.A.M., van Dongen, B.F., van der Aalst, W.M.P.: Discovering and navigating a collection of process models using multiple quality dimensions. In: Lohmann, N., Song, M., Wohed, P. (eds.) *BPM 2013. LNBIP*, vol. 171, pp. 3–14. Springer, Cham (2014). doi:[10.1007/978-3-319-06257-0_1](https://doi.org/10.1007/978-3-319-06257-0_1)
9. Cohn, D., Hull, R.: Business artifacts: a data-centric approach to modeling business operations and processes. *IEEE Data Eng. Bull.* **32**(3), 3–9 (2009)
10. van Dongen, B.F., Alves de Medeiros, A.K., Wen, L.: Process mining: overview and outlook of petri net discovery algorithms. *Trans. Petri Nets Other Models Concurrency* **II**(5460), 225–242 (2009)
11. Fahland, D., van der Aalst, W.M.P.: Model repair - aligning process models to reality. *Inf. Syst.* **47**, 220–243 (2015)
12. Fahland, D., de Leoni, M., van Dongen, B.F., van der Aalst, W.M.P.: Behavioral conformance of artifact-centric process models. In: Abramowicz, W. (ed.) *BIS 2011. LNBIP*, vol. 87, pp. 37–49. Springer, Heidelberg (2011). doi:[10.1007/978-3-642-21863-7_4](https://doi.org/10.1007/978-3-642-21863-7_4)
13. The Open Group. Archimate 3.0 specification (2016). <http://pubs.opengroup.org/architecture/archimate3-doc/>
14. Kim, Y., Choi, T.Y., Yan, T., Dooley, K.: Structural investigation of supply networks: a social network analysis approach. *J. Oper. Manag.* **29**(3), 194–211 (2011)

15. Lankhorst, M.: Enterprise Architecture at Work: Modeling, Communication and Analysis, vol. 36. Springer, Heidelberg (2013)
16. Leemans, S.J.J., Fahland, D., van der Aalst, W.M.P.: Discovering block-structured process models from event logs - a constructive approach. In: Colom, J.-M., Desel, J. (eds.) PETRI NETS 2013. LNCS, vol. 7927, pp. 311–329. Springer, Heidelberg (2013). doi:[10.1007/978-3-642-38697-8_17](https://doi.org/10.1007/978-3-642-38697-8_17)
17. Mannhardt, F., de Leoni, M., Reijers, H.A., van der Aalst, W.M.P.: Balanced multi-perspective checking of process conformance. *Computing* **98**(4), 407–437 (2016)
18. Nooijen, E.H.J., van Dongen, B.F., Fahland, D.: Automatic discovery of data-centric and artifact-centric processes. In: Rosa, M., Soffer, P. (eds.) BPM 2012. LNBIP, vol. 132, pp. 316–327. Springer, Heidelberg (2013). doi:[10.1007/978-3-642-36285-9_36](https://doi.org/10.1007/978-3-642-36285-9_36)
19. Otte, E., Rousseau, R.: Social network analysis: a powerful strategy, also for the information sciences. *J. Inf. Sci.* **28**(6), 441–453 (2002)
20. Peffers, K., Tuunanen, T., Rothenberger, M.A., Chatterjee, S.: A design science research methodology for information systems research. *J. Manag. Inf. Syst.* **24**(3), 45–77 (2007)
21. Popova, V., Fahland, D., Dumas, M.: Artifact Lifecycle Discovery. arXiv preprint [arXiv:1303.2554](https://arxiv.org/abs/1303.2554), pp. 1–27 (2013)
22. Rozinat, A., van der Aalst, W.M.P.: Conformance checking of processes based on monitoring real behavior. *Inf. Syst.* **33**(1), 64–95 (2008)
23. Song, M., van der Aalst, W.M.P.: Towards comprehensive support for organizational mining. *Decis. Support Syst.* **46**(1), 300–317 (2008)
24. van Eck, M.L., Lu, X., Leemans, S.J.J., Aalst, W.M.P.: PM2: a process mining project methodology. In: Zdravkovic, J., Kirikova, M., Johannesson, P. (eds.) CAiSE 2015. LNCS, vol. 9097, pp. 297–313. Springer, Cham (2015). doi:[10.1007/978-3-319-19069-3_19](https://doi.org/10.1007/978-3-319-19069-3_19)
25. Vázquez-Barreiros, B., van Zelst, S.J., Buijs, J.C.A.M., Lama, M., Mucientes, M.: Repairing alignments: striking the right nerve. In: Schmidt, R., Guédria, W., Bider, I., Guerreiro, S. (eds.) BPMDS/EMMSAD -2016. LNBIP, vol. 248, pp. 266–281. Springer, Cham (2016). doi:[10.1007/978-3-319-39429-9_17](https://doi.org/10.1007/978-3-319-39429-9_17)
26. Verbeek, H.M.W., Buijs, J.C.A.M., van Dongen, B.F., van der Aalst, W.M.P.: XES, XESame, and ProM 6. In: Soffer, P., Proper, E. (eds.) CAiSE Forum 2010. LNBIP, vol. 72, pp. 60–75. Springer, Heidelberg (2011). doi:[10.1007/978-3-642-17722-4_5](https://doi.org/10.1007/978-3-642-17722-4_5)
27. Weidlich, M., van der Werf, J.M.E.M.: On profiles and footprints – relational semantics for Petri nets. In: Haddad, S., van der Pomello, L. (eds.) PETRI NETS 2012. LNCS, vol. 7347, pp. 148–167. Springer, Heidelberg (2012). doi:[10.1007/978-3-642-31131-4_9](https://doi.org/10.1007/978-3-642-31131-4_9)
28. van der Werf, J.M.E.M., Kaats, E.: Discovery of functional architectures from event logs. In: International Workshop on Petri Nets and Software Engineering (PNSE 2015), pp. 227–243 (2015)
29. Winter, R., Sinz, E.J.: Enterprise architecture. *Inf. Syst. e-Bus. Manag.* **5**(4), 357–358 (2007)