Comparing Business Processes to Determine the Feasibility of Configurable Models: A Case Study

J.J.C.L. Vogelaar, H.M.W. Verbeek, B. Luka, and Wil M.P van der Aalst

Technische Universiteit Eindhoven
Department of Mathematics and Computer Science
P.O. Box 513, 5600 MB Eindhoven, The Netherlands
{h.m.w.verbeek,w.m.p.v.d.aalst}@tue.nl

Abstract. Organizations are looking for ways to collaborate in the area of process management. Common practice until now is the (partial) standardization of processes. This has the main disadvantage that most organizations are forced to adapt their processes to adhere to the standard. In this paper we analyze and compare the actual processes of ten Dutch municipalities. Configurable process models provide a potential solution for the limitations of classical standardization processes as they contain all the behavior of individual models, while only needing one model. The question rises where the limits are though. It is obvious that one configurable model containing all models that exist is undesirable. But are company-wide configurable models feasible? And how about cross-organizational configurable models, should all partners be considered or just certain ones? In this paper we apply a similarity metric on individual models to determine means of answering questions in this area. This way we propose a new means of determining beforehand whether configurable models are feasible. Using the selected metric we can identify more desirable partners and processes before computing configurable process models.

Keywords: process configuration, YAWL, CoSeLoG, model merging.

1 Introduction

The results in this paper are based on 80 process models retrieved for 8 different business processes from 10 Dutch municipalities. This was done within the context of the CoSeLoG project [1,5]. This project aims to create a system for handling various types of permits, taxes, certificates, and licenses. Although municipalities are similar in that they have to provide the same set of business processes (services) to their citizens, their process models are typically different. Within the constraints of national laws and regulations, municipalities can differentiate because of differences in size, demographics, problems, and policies. Supported by the system to be developed within CoSeLoG, individual municipalities can make use of the process support services simultaneously, even though their process models differ. To realize this, *configurable process models* are used.

Configurable process models form a relatively young research topic [7,9,10,3]. A configurable process model can be seen as a union of several process models into

F. Daniel et al. (Eds.): BPM 2011 Workshops, Part II, LNBIP 100, pp. 50–61, 2012.

one. While combining different process models, duplication of elements is avoided by matching and merging them together. The elements that occur in only a selection of the individual process models are made configurable. These elements are then able to be set or configured. In effect, such an element can be chosen to be included or excluded. When for all configurable elements such a setting is made, the resulting process model is called a configuration. This configuration could then correspond to one of the individual process models for example.

Configurable process models offer several benefits. One of the benefits is that there is only one process model that needs to be maintained, instead of the several individual ones. This is especially helpful in case a law changes or is introduced, and thus all municipalities have to change their business processes, and hence their process models. In the case of a configurable process model this would only incur a single change. When we lift this idea up to the level of services (like in the CoSeLoG project [1,5]), we also only need to maintain one information system, which can be used by multiple municipalities.

Configurable process models are not always a good solution however. In some cases they will yield better results than in others. Two process models that are quite similar are likely to be better suited for inclusion in a configurable process model than two completely different and independent process models. For this reason, this paper strives to provide answers to the following three questions:

1. *Which business process is the best starting point for developing a configurable process model?* That is, given a municipality and a set of process models for every municipality and every business process, for which business process is the configurable process model (containing all process models for that business process) the less complex?
2. *Which other municipality is the best candidate to develop configurable models with?* That is, given a municipality and a set of process models for every municipality and every business process, for which other municipality are the configurable process models (containing the process models for both municipalities) the less complex?
3. *Which clusters of municipalities would best work together, using a common configurable model?* That is, given a business process and a set of process models for every municipality and every business process, for which clustering of municipalities are the configurable process models (containing all process models for the municipalities in a cluster) the less complex?

The remainder of this paper is structured as follows. Section 2 introduces the 80 process models and background information about these process models. Section 3 makes various comparisons to produce answers to the proposed questions. Finally, Section 4 concludes the paper. For additional details, we refer the interested reader to [13], which is the technical report which underlies this paper.

2 YAWL Models

We collected 80 YAWL[8] models in total. These YAWL models were retrieved from the ten municipalities, which are partners in the CoSeLoG project: Bergeijk, Bladel, Coevorden, Eersel, Emmen, Gemert-Bakel, Hellendoorn, Oirschot, Reusel-de Mierden and Zwolle. In the remainder of this paper we will refer to these municipalities as Mun_A to Mun_J (these are randomly ordered).

For every municipality, we retrieved the YAWL models for the same eight business processes, which are run by any Dutch municipality. Hence, our process model collection is composed of eight sub-collections consisting of ten YAWL models each. The YAWL models were retrieved through interviews by us and validated by the municipalities afterwards.

The eight business processes covered are:

1. The processing of an application for a receipt from the people registration (3 variants):
   (a) When a customer applies through the internet: GBA_1.
   (b) When a customer applies in person at the town hall: GBA_2.
   (c) When a customer applies through a written letter: GBA_3.
2. The method of dealing with the report of a problem in a public area of the municipality: MOR.
3. The processing of an application for a building permit (2 parts):
   (a) The preceding process to prepare for the formal procedure: $WABO_1$.
   (b) The formal procedure: $WABO_2$.
4. The processing of an application for social services: WMO.
5. The handling of objections raised against the taxation of a house: WOZ.

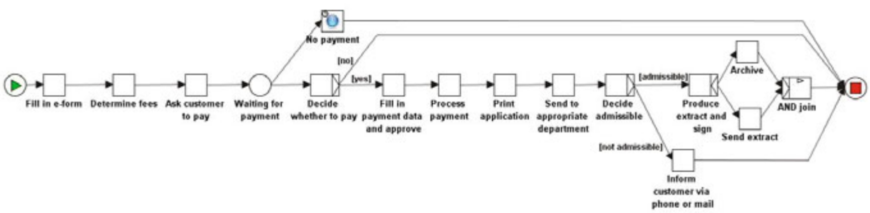

Fig. 1. GBA_1 YAWL model for Mun_E

To give an indication of the variety and similarity between the different YAWL models some examples are shown. Figure 1 shows the GBA_1 YAWL model of Mun_E, whereas Figure 2 shows the GBA_1 YAWL model of Mun_G. The YAWL models of these two municipalities are quite similar. Nevertheless, there are some differences. Recall that GBA_1 is about the application for a certain document through the internet. The difference between the two municipalities is that Mun_E handles the payment through the internet (so before working on the document), while Mun_G handles it manually after having sent the document. However, the main steps to create the document are the same. This explains why the general flow of both models is about the same, with exception of the payment-centered elements.

People can apply for this document through different means too. Figure 3 shows the GBA_2 YAWL model for Mun_E. This model seems to contain more tasks than either of the GBA_1 models. This makes sense, since more communication takes place during

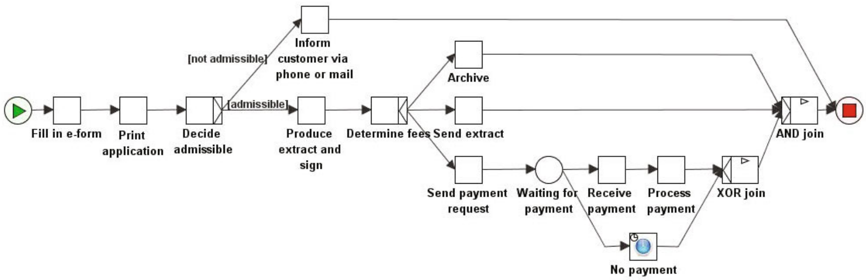

Fig. 2. GBA_1 YAWL model for Mun_G

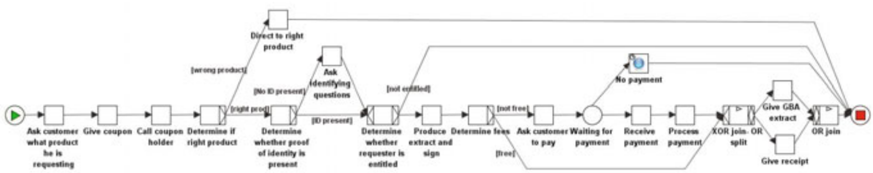

Fig. 3. GBA_2 YAWL model for Mun_E

the application. The employee at the town hall needs to gain the necessary information from the customer. In the internet case, the customer had already entered the information himself in the form, because otherwise the application could not be sent digitally. As the YAWL model still describes a way to produce the same document, it is to be expected that GBA_2 models are somewhat similar to GBA_1 models. Indeed, the general flow remains approximately the same, although some tasks have been inserted. This is especially the case in the leftmost part of the model, which is the part where in the internet case the customer has already given all information prior to sending the digital form. In the model shown in Figure 3 the employee asks the customer for information in this same area. This extra interaction also means more tasks (and choices) in the YAWL model.

Figure 4 shows the WOZ YAWL model for Mun_E, which is clearly different from the three GBA models. The WOZ model shown in Figure 4 is more time-consuming. Customers need to be heard and their objections need to be assessed thoroughly. Next,

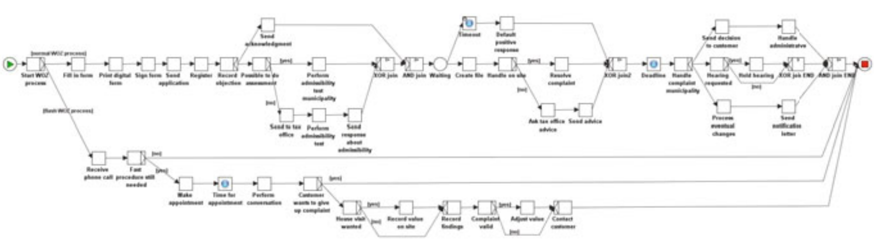

Fig. 4. WOZ YAWL model for Mun_E

the grounds for the objections need to be investigated, sometimes even leading to a house visit. After all the checking and decision making has taken place, the decision needs to be communicated to the customer, several weeks or months later. The WOZ models are quite a bit different from the GBA models, where information basically needs to be retrieved and documented.

The remainder of this paper presents a case study of the 80 YAWL models (which can found in Appendix A of [13]), and compares them within their own sub-collections. This way, we show that the YAWL models for the municipalities are indeed different, but not so different that it justifies the separate implementation and maintenance of ten separate software systems.

3 Comparison

This section compares all YAWL models from each of the sub-collections. As certain models are more similar than others, we want to give an indication on which processes are very similar, and which are more different. This similarity we will use as an indication of which models have more or less complexity when merged into a configurable model. The higher the similarity between models, the lower we expect the complexity to be for the configurable models. Making a configurable model for equivalent models (similarity score 1.0) approximately results in the same model again (additional complexity approx. 0.0), since no new functionality needs to be added to any of the original models.

First, we apply a combination of three known complexity metrics to all YAWL models. Second, we compare the models using a combination of two known similarity metrics. Third and last, we answer the three questions as proposed earlier using these metrics.

3.1 Complexity

For every YAWL model, we calculated the CFC [4], density [11], and CC metric [12] (see also [13] for details) to get an indication of its complexity. The complete results can be found in Appendix B of [13]. Figure 5 shows the relation between the CFC metric and the other two complexity metrics. Clearly, these relations are quite strong: The higher the CFC metric, the lower the other two metrics. Although this is to be expected for the CC metric, this is quite unexpected for the density metric. Like the CFC metric, the density metric was assumed to go up when complexity goes up, hence the trend should be that the density metric should go up when the CFC metric goes up. Obviously, this is not the case. As a result, for the remainder of this paper we will assume that the density metric goes down when complexity goes up.

Based on the strong relations as suggested in Figure 5 ($CC(G) = 0.4611 \cdot CFC(G)^{-0.851}$ and $density(G) = 1.1042 \cdot CFC(G)^{-0.791}$) we can now transform the other two complexity metrics to the scale of the CFC metric. As a result, we can take the rounded average over the resulting three metrics and get a unified complexity metric. Table 1 shows the average complexity metrics for all business processes. As this table shows, the processes $WABO_2$ and WMO are the most complex, and GBA_1 and $WABO_1$ the least complex.

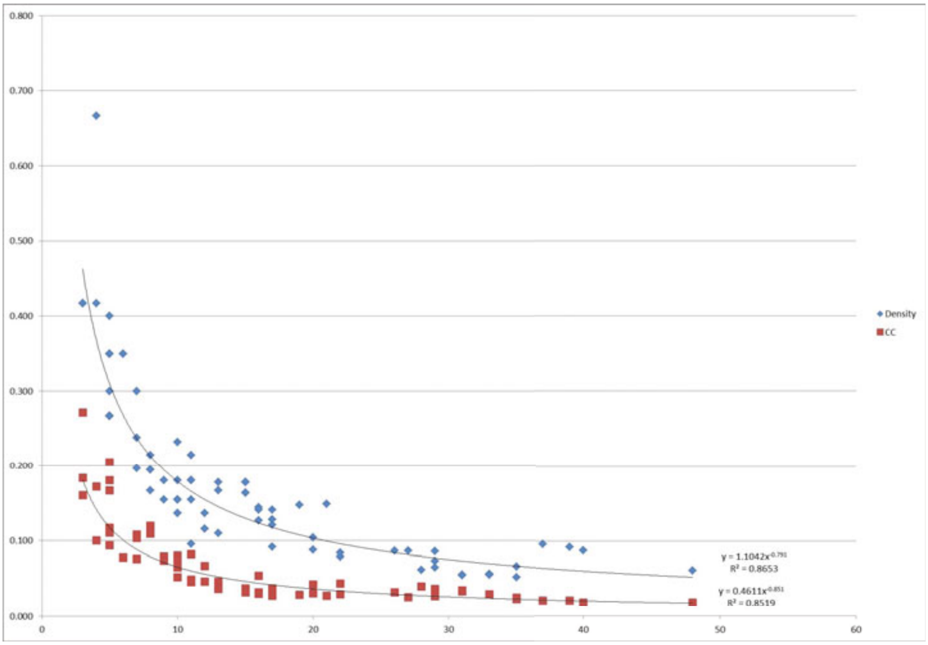

Fig. 5. Comparison of the CFC metric with the CC and Density metrics

Table 1. Comparison of the business processes on the complexity metrics

	GBA_1	GBA_2	GBA_3	MOR	$WABO_1$	$WABO_2$	WMO	WOZ
CFC	5.100	14.400	9.800	15.400	4.700	29.800	33.800	12.000
Density	0.383	0.165	0.170	0.159	0.305	0.061	0.080	0.132
CC	0.147	0.038	0.088	0.035	0.119	0.034	0.024	0.064
Unified	5	15	9	17	5	30	33	13

3.2 Similarity

For every pair of YAWL models from the same sub-collection, we calculated the GED and SPS metric [6] (see also [13] for details) to get an indication of their similarity. The complete results can be found in Appendix C of [13]. Figure 6 shows the relation between the GED and the SPS metric. Although the relation between these metrics ($SPS(G_1, G_2) = 2.0509 \cdot GED(G_1, G_2) - 1.082$) is a bit less strong as the relation between the complexity metrics, we consider this relation to be strong enough to unify both metrics into a single, unified, metric. This unified similarity metric uses the scale of the SPS metric, as the range of this scale is wider than the scale of the GED metric. Table 2 shows the averages over the values for the different similarity metrics for each of the processes. From this table, we conclude that the GBA_2 models are most similar to each other, while the MOR models are least similar.

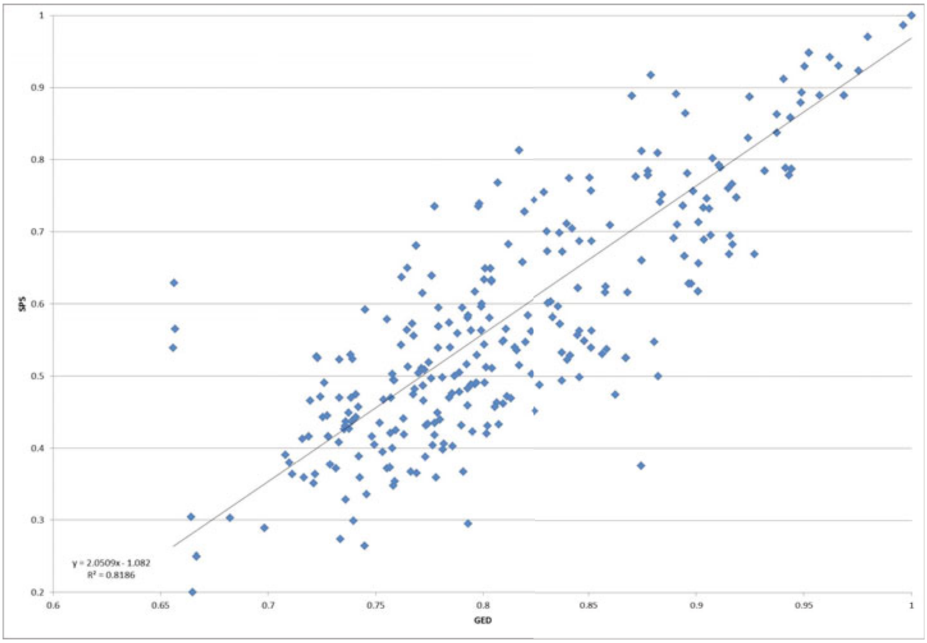

Fig. 6. Comparison of the GED metric with the SPS metric

Table 2. Average similarity values

	GBA_1	GBA_2	GBA_3	MOR	$WABO_1$	$WABO_2$	WMO	WOZ
GED	0.829	0.916	0.828	0.797	0.871	0.891	0.830	0.820
SPS	0.646	0.759	0.632	0.556	0.774	0.725	0.546	0.615
Unified	0.632	0.778	0.624	0.554	0.739	0.735	0.583	0.607

Recall that a configurable process model "contains" all individual process models. Whenever one wants to use the configurable model as an executable model, it needs to be configured by selecting which parts should be left out. The more divergent the individuals are, the more complex the resulting configurable process model needs to be to accommodate all the individuals. So, the more similar models are, the easier to construct and maintain the configurable model will most likely be.

The similarity value for the GBA_1 models for Mun_A and Mun_H equals 1.0. Merging these models into a configurable model, yields an equivalent model, which we find not so interesting. Taking a look at another high similarity value in the table, we construct the configurable *GBA1* model for Mun_D and Mun_I. The complexity metrics for the configurable model yield 7 (CFC), 0.238 (density), 0.091 (CC), and 7 (unified). Similarly we construct a configurable model for the two least similar models: Mun_G and Mun_F. The resulting complexity values are 34 (CFC), 0.108 (density), 0.026 (CC),

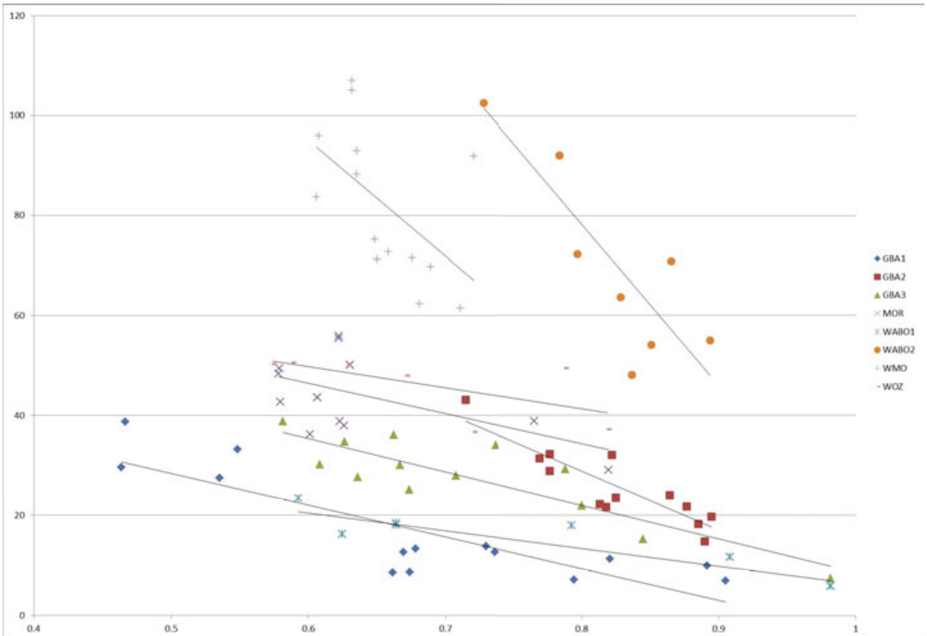

Fig. 7. Unified similarity vs. unified complexity for 100 pairs of models

and 28 (unified). These results are in line with our expectations, as the former metrics are all better than the latter.

To confirm these relation between similarity on the one hand and complexity on the other, we have selected 100 pairs of models (each pair from the same sub-collection), have merged every pair, and have computed the complexity metrics of the resulting model. Figure 7 shows the results: When similarity goes down, complexity tends to go up.

Based on the illustrated correlations, we assume that the unified similarity metric gives a good indication for the unified complexity of the resulting configurable model. Therefore, we use this metric to answer the three questions stated in the introduction.

3.3 Question 1: Which Business Process Is the Best Starting Point for Developing a Configurable Process Model?

To answer this question we select a specific business process P and compute the average similarity between the YAWL model of process P in a selected municipality and all models of P in other municipalities. Take for example Mun_D. For the GBA_1 process, the average value for Mun_D (that is, average distance to other municipalities) is $\frac{0.735+0.777+0.670+0.741+0.818+0.430+0.735+0.898+0.526}{9} = 0.703$. Table 3 shows the averages for each municipality and each business process. In this table we can see that for Mun_D the $WABO_2$ process scores highest, followed by $WABO_1$ and GBA_1. Note

Table 3. Average similarity values per model

	Mun_A	Mun_B	Mun_C	Mun_D	Mun_E	Mun_F	Mun_G	Mun_H	Mun_I	Mun_J
GBA_1	0.631	0.612	0.560	0.703	0.645	0.641	*0.354*	0.631	0.715	*0.442*
GBA_2	**0.766**	**0.821**	0.667	0.602	0.807	**0.771**	0.751	**0.821**	0.725	**0.821**
GBA_3	0.530	0.513	0.486	0.607	0.550	0.587	0.678	0.551	0.678	0.664
MOR	*0.496*	0.548	0.501	0.482	0.585	*0.488*	0.573	*0.468*	*0.430*	0.491
$WABO_1$	0.501	0.483	0.602	0.776	**0.818**	0.662	**0.818**	0.818	**0.818**	0.818
$WABO_2$	0.646	*0.419*	**0.730**	**0.800**	0.746	0.741	0.800	0.800	0.750	0.644
WMO	0.621	0.539	0.543	*0.426*	*0.491*	0.503	0.496	0.625	0.615	0.522
WOZ	0.507	0.448	*0.447*	0.601	0.562	0.616	0.600	0.651	0.657	0.561

Table 4. Comparing $WABO_2$ and WMO for Mun_D

	Mun_A	Mun_B	Mun_C	Mun_E	Mun_F	Mun_G	Mun_H	Mun_I	Mun_J	Average
$WABO_2$	92	72	71	51	55	32	32	34	64	56
WMO	105	112	84	95	78	85	102	102	82	94

that for ease of reference, we have highlighted the best (bold) and worst (italics) similarity scores per municipality. So, from the viewpoint of Mun_D, these three are the best candidates for making a configurable model. In a similar way we can determine such best candidates for any of the municipalities.

We now construct configurable models for the $WABO_2$ model for Mun_D and each of the other municipalities and take the average complexity metrics for these. We do the same for the WMO model. Table 4 shows the results. Although the complexities of the $WABO_2$ models (30) and the WMO models (33) are quite similar, it is clear that merging the latter yields worse scores on all complexity metrics than merging the former yields. Therefore, we conclude that the better similarity between the $WABO_2$ models resulted in a less-complex configurable model, while the worse similarity between the MOR models resulted in a more-complex configurable model.

From Table 3 we can also conclude that the GBA_2, $WABO_1$, and $WABO_2$ processes are, in general, good candidates to start a configurable approach with, as they turn out best for 5, 3, and 2 municipalities.

3.4 Question 2: Which Other Municipality Is the Best Candidate to Develop Configurable Models With?

The second question is not so much about which process suits the municipality best, but which other municipality. To compute this, we take the average similarity over all models for every other municipality. Table 5 shows the results for all municipalities. Again, we have highlighted the best match. This table shows that Mun_H and Mun_I are most similar to Mun_D. Apparently, these municipalities are best suited to start working with Mun_D on an overall configurable approach.

We calculated the average complexity of the configurable models for Mun_D and Mun_H and for Mun_D and Mun_A. Table 6 shows the results. Clearly, the average

Table 5. Average similarity values per municipality

	Mun_A	Mun_B	Mun_C	Mun_D	Mun_E	Mun_F	Mun_G	Mun_H	Mun_I	Mun_J
Mun_A		0.556	0.546	0.555	0.598	0.585	0.591	0.682	0.644	0.527
Mun_B	0.556		*0.508*	*0.538*	*0.559*	*0.547*	*0.512*	*0.595*	0.591	*0.525*
Mun_C	0.546	*0.508*		0.580	0.617	0.552	0.575	0.604	*0.569*	0.552
Mun_D	0.555	0.538	0.580		0.638	0.630	0.642	0.702	0.717	0.619
Mun_E	0.598	0.559	**0.617**	0.638		0.672	**0.692**	0.679	0.705	**0.696**
Mun_F	0.585	0.547	0.552	0.630	0.672		0.675	0.651	0.671	0.651
Mun_G	0.591	0.512	0.575	0.642	0.692	**0.675**		0.656	0.687	0.672
Mun_H	**0.682**	**0.595**	0.604	0.702	0.679	0.651	0.656		**0.801**	0.664
Mun_I	0.644	0.591	0.569	**0.717**	**0.705**	0.671	0.687	**0.801**		0.677
Mun_J	*0.527*	0.525	0.552	0.619	0.696	0.651	0.672	0.663	0.676	

Table 6. Comparing Mun_H and Mun_A for Mun_D

	GBA_1	GBA_2	GBA_3	MOR	$WABO_1$	$WABO_2$	WMO	WOZ	Average
Mun_H	13	29	47	41	12	32	102	26	38
Mun_A	13	38	34	55	16	92	105	42	49

complexity scores when merging Mun_D with Mun_H are better than the scores when merging Mun_D with Mun_A. This is in line with our expectations. Also note that only for the GBA_3 process a configurable model with Mun_A might be preferred over a configurable model with Mun_H.

From Table 5 we can also conclude that Mun_I and Mun_E are preferred partners for configurable models, as Mun_I are the preferred partner for 3 of the municipalities.

3.5 Question 3: Which Clusters of Municipalities Would Best Work Together, Using a Common Configurable Model?

The third question is a bit trickier to answer, but this can also be accomplished with the computed metrics. To answer this question, we only need to consider the values in one of the comparison tables (see Appendix C of [13]). We now want to see which clusters of municipalities could best work together in using configurable models. There are different ways to approach this problem. One of the approaches is using the k-means clustering algorithm [2]. Applying this algorithm to the mentioned metrics, we obtain the clusters $Mun_B + Mun_D + Mun_E + Mun_F + Mun_I$, $Mun_G + Mun_J$, and $Mun_A + Mun_C + Mun_H$.

Table 7 shows the complexity for all processes, where cluster k is the cluster as selected by the k-means clustering technique and cluster 1 up to 10 are 10 randomly selected clusters per process (see Appendix E of [13] for the cluster details). This table clearly shows that the clusters as obtained by the k-means clustering technique are quite good. Only in the case of the GBA_3 and $WABO_1$ processes, we found a better clustering, and in case of the latter process the gain is only marginal.

Table 7. Comparing clusters on CC

Cluster	GBA_1	GBA_2	GBA_3	MOR	$WABO_1$	$WABO_2$	WMO	WOZ
k	**15**	**25**	48	**50**	19	**76**	**101**	**59**
1	**15**	29	*54*	75	26	92	117	75
2	*28*	32	47	67	21	95	116	74
3	23	33	52	73	*27*	88	115	*88*
4	26	32	**45**	*81*	24	87	103	76
5	27	32	49	69	**18**	84	130	85
6	26	30	46	77	*27*	*100*	113	80
7	26	*34*	48	66	*27*	90	121	82
8	24	33	50	71	22	92	107	82
9	25	32	**45**	77	24	92	128	80
10	27	31	51	76	26	77	*133*	77
Average	24	31	49	71	24	88	117	78

4 Conclusion

First of all, in this paper we have shown that similarity can be used to predict the complexity of a configurable model. In principle, the more similar two process models are, the less complex the resulting configurable model will be.

We have used the control-flow complexity (CFC) metric from [4], the density metric from [11], and the cross-connectivity (CC) metric from [12] as complexity metrics. We have shown that these three metrics are quite related to each other. For example, when the CFC metric goes up, the density and CC go down. Based on this, we have been able to unify these metrics into a single complexity metric that uses the same scale as the CFC metric.

The complexity of the 80 YAWL models used in this paper ranged from simple (GBA_1 and $WABO_1$ processes, unified complexity approx. 5) to complex ($WABO_2$ and WMO processes, unified complexity approx. 30). The complexity of the configurable models we obtained were typically quite higher (up to approx. 450). This shows that complexity can get quickly out of control, and that we needs some way to predict the complexity of a configurable model beforehand.

To predict the complexity of a configurable model, we have used the GED metric and the SPS metric as defined in [6]. Based on the combined similarity of two process models a prediction can be made for the complexity of the resulting configurable model. By choosing to merge only similar process models, the complexity of the resulting configurable model is kept at bay.

We have shown that the CFC and unified metric of the configurable model are positively correlated with the similarity of its constituting process models, and that the density and CC metric are negatively correlated. The behavior of the density metric came as a surprise to us. The rationale behind this metric clearly states that a density and the likelihood of errors are positively correlated. As such, we expected a positive correlation between the density and the complexity. However, throughout our set of models we observed the trend that less-similar models yield less-dense configurable models, whereas the other complexity metrics behave as expected. As a result, we concluded that the density is negatively correlated with the complexity of models.

References

1. van der Aalst, W.M.P.: Configurable Services in the Cloud: Supporting Variability While Enabling Cross-Organizational Process Mining. In: Meersman, R., Dillon, T.S., Herrero, P. (eds.) OTM 2010, Part I. LNCS, vol. 6426, pp. 8–25. Springer, Heidelberg (2010)
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., Dumas, M., Gottschalk, F., ter Hofstede, A.H.M., La Rosa, M., Mendling, J.: Preserving Correctness During Business Process Model Configuration. Formal Aspects of Computing 22, 459–482 (2010)
4. Cardoso, J.: How to Measure the Control-flow Complexity of Web Processes and Workflows (2005)
5. CoSeLoG. Configurable Services for Local Governments (CoSeLoG) Project Home Page, `http://www.win.tue.nl/coselog`
6. Dijkman, R.M., Dumas, M., van Dongen, B.F., Käärik, R., Mendling, J.: Similarity of Business Process Models: Metrics and Evaluation. Information Systems 36(2), 498–516 (2011)
7. Gottschalk, F.: Configurable Process Models. PhD thesis, Eindhoven University of Technology, The Netherlands (December 2009)
8. Hofstede, A., van der Aalst, W.M.P., Adams, M., Russell, N.: Modern Business Process Automation: YAWL and its Support Environment. Springer, Heidelberg (2009)
9. La Rosa, M.: Managing Variability in Process-Aware Information Systems. PhD thesis, Queensland University of Technology, Brisbane, Australia (April 2009)
10. La Rosa, M., Dumas, M., ter Hofstede, A.H.M., Mendling, J.: Configurable Multi-perspective Business Process Models. Information Systems 36(2), 313–340 (2011)
11. Mendling, J.: Testing Density as a Complexity Metric for EPCs. In: German EPC Workshop on Density of Process Models (2006)
12. Vanderfeesten, I.T.P., Reijers, H.A., Mendling, J., van der Aalst, W.M.P., Cardoso, J.: On a Quest for Good Process Models: The Cross-Connectivity Metric. In: Bellahsène, Z., Léonard, M. (eds.) CAiSE 2008. LNCS, vol. 5074, pp. 480–494. Springer, Heidelberg (2008)
13. Vogelaar, J.J.C.L., Verbeek, H.M.W., Luka, B., van der Aalst, W.M.P.: Comparing Business Processes to Determine the Feasibility of Configurable Models: A Case Study. BPM Center Report BPM-11-17, BPMcenter.org (2011)