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 W.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.

Key words: 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, 6]. 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 [8, 12, 13, 3]. A configurable process model can be seen as a union of several process models into 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, 6]), 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 discusses the techniques used in this paper to answer the proposed questions. Section 3 then introduces the 80 process models and background information about these process models. Section 4 makes various comparisons to produce answers to the proposed questions. Finally, Section 5 concludes the paper.

2 Preliminaries

2.1 YAWL

This paper presents several business processes modeled in YAWL (Yet Another Workflow Language) [9]. YAWL allows for the basic components that are present in the process models obtained from the municipalities. It is a workflow language developed by the YAWL Foundation and based on the Workflow Patterns [4]. Figure 1 shows an annotated example YAWL model.

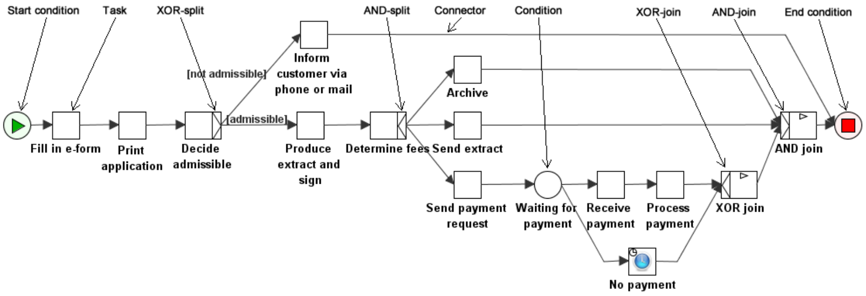

Fig. 1: An annotated example YAWL model

A YAWL model basically consists of conditions (circles), tasks (rectangles) and connectors (arrows). The connectors indicate the flow of control in a YAWL model, where each undecorated task can only have one incoming and one outgoing connector. The YAWL model in Figure 1 should be read from left tot right. The element furthest to the left is the start condition, which corresponds to the start of the process. The end of the process is located all the way to the right. A YAWL model can only have one start and one end condition.

A task can be a normal task (like “Fill in e-form”), or act as a branching node (like “Decide admissible”) in the process model. If the latter is the case, then the task has a decorator to indicate whether it is an AND-join (or -split), an XOR-join (or -split), or an OR-join (or -split). XOR-splits (like “Decide admissible”) introduce choice branches where one of the offered choices can be followed, whereas XOR-joins (like “XOR join”) merge alternative flows. AND-splits (like “Determine fees”) introduce parallel branches, whereas AND-joins (like “AND join”) merge parallel branches. OR-splits (not present) introduce a (non-empty) subset of parallel branches, whereas OR-joins (not present) merge a subset of those branches by waiting until the remaining branches are dead. Conditions (like “Waiting for payment”) can have multiple incoming or outgoing connectors. This can be seen as an XOR-split/join, with the subtle difference that this is an implicit choice [4]. It is also possible to give a task some extra meaning which is indicated by its decorations. A clock (like “No payment”) indicates that it is a timed task, which executes after some timer expires. A small triangle (like “XOR join”) indicates that it is an automatic task, which are mostly needed for routing purposes.

2.2 EPC models

Although the process models are presented as YAWL models, the metrics used in this paper are typically defined on EPC (Event-driven Process Chain) models [10, 11, 16]. For this reason, we also introduce EPC models.

An EPC model typically consists of functions (rectangles), events (hexagons), connectors (circles), and edges (arrows). Roughly spoken, EPC functions correspond to YAWL tasks, EPC events correspond to YAWL conditions, EPC connectors correspond to YAWL task decorations, and EPC edges correspond to YAWL connectors. In an EPC model, only connectors are allowed to have multiple input edges and/or multiple output edges.

The conversion from a YAWL model to an EPC model is straightforward:

- A YAWL task is converted into an EPC fragment containing of a join connector, an event, a function, a split connector, and a series of three connecting edges, where the YAWL task decorators determine the type of the EPC connectors.
- A regular YAWL condition is converted into an XOR-join connector, an XOR-split connector, and a connecting edge.
- The YAWL input condition is converted into an event, a (dummy) function, an XOR-split connector, and a series of two connecting edge, whereas the YAWL output condition is converted into an XOR-join connector, an event, and a connecting edge.
- A YAWL connector is converted into an edge.

Superfluous connectors and a possible dummy function at the start of the EPC model will be removed in a post-processing step. Figure 2 shows the annotated example YAWL model of Figure 1 converted into an EPC model.

2.3 Creating configurable models

For creating a configurable model from two different process models we use the approach as described in [8]. This approach has been implemented in the “EPC merge” plug-in of the “ProM 5.2” toolkit [18, 17]. However, given the fact that we had a specific set of process models to work with, we tailored this plug-in to our needs.

When running the “EPC merge” plug-in on two EPC models, the user needs to specify which functions of one EPC model match which functions of the second, and the same for events. To help the user with this task, the plug-in offers a default match which is based on the String-edit distance (SED) metric on the names of the functions (events): The function (event) with the smallest SED value will be selected by default as a match. However, in our set of YAWL models tasks were considered to be identical if their names were identical. On the EPC level, this corresponds to the requirement that function and event names should be identical modulo some trailing underscore and number, which are added by the YAWL editor automatically. As a result, two functions named “Fill_in_e-form_11” and “Fill_in_e-form_36” should be considered to be identical. Furthermore, we sometimes needed to duplicate a YAWL task, while the YAWL editor does not allow for duplicate names. In such a case, we simply added a number to the end of the task name. For example, “Fill in e-form” would become “Fill in e-form1”. The matching algorithm takes these trailing number also into account, and

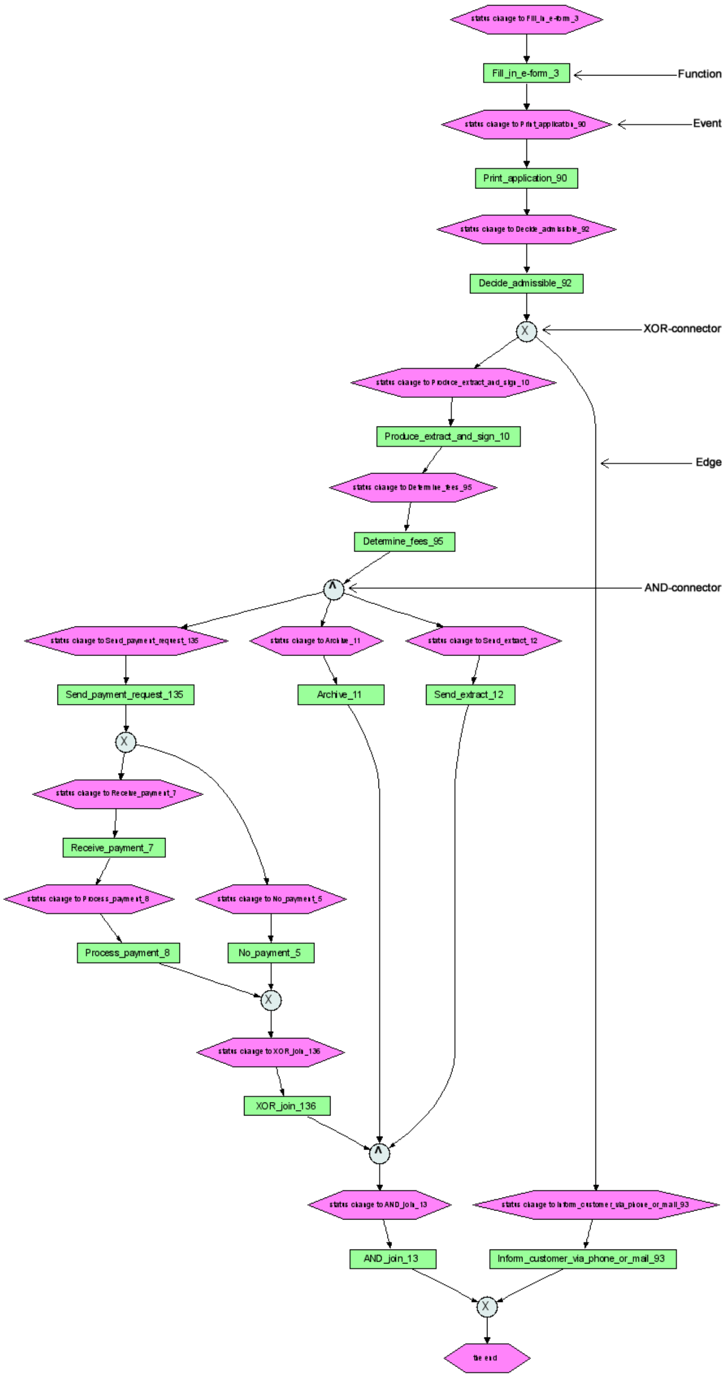

Fig. 2: The annotated example YAWL model converted into an EPC

is able to match "Fill_in_e-form_11" with "Fill_in_e-form1_36". As some minor typos could be present in the names of the YAWL models, we decided to allow for a single typo. As a result, the SED value between two matching names was allowed to be at most one. Hence, "Fill_in_eform_11" would be matched with "Fill_in_e-form1_36". Finally, there was no reason to match different joins and/or splits in the models, as there was no guarantee that a correct match could be found for these dummy functions and dummy events. As a result, we decided to remove any match from a function or events that was named like "AND_join_11", "status_change_to_XOR_split1_36" etc.

2.4 Graph-edit distance similarity

This paper strives to give an answer to a couple of questions about models. To answer these questions, the models need to be compared to each other. There has been extensive research into the comparison of models on different levels and in different modeling languages [7, 19, 22]. In this paper we limit ourselves to using the *Graph-Edit Distance (GED) similarity* metric and the *Structural process similarity (SPS)* metric, which were introduced in [7].

The GED metric is a structural metric based on the *minimal number of graph-edit operations* needed to transform one graph into an other, taking node deletion, node insertion, node substitution, edge deletion, edge insertion into account. Let $M : (N_1 \nrightarrow N_2)$ be the partial injective mapping that induces the GED between two process models and let sn be the set of all inserted and deleted nodes, se be the set of all inserted and deleted edges and let $Sim(n, m)$ be a function that assigns a similarity score to a pair of nodes. As shown in [7], a similarity metric is gained from the graph-edit distance metric by calculating:

$$sim_{GED}(G_1, G_2) = 1 - \frac{snv + sev + sbv}{3},$$

where:

$$\begin{aligned} snv &= \frac{|sn|}{|N_1| + |N_2|}; \\ sev &= \frac{|se|}{|E_{1|} + |E_2|}; \\ sbv &= \frac{2 \cdot \Sigma_{(n,m) \in M} 1 - Sim(n, m)}{|N_1| + |N_2| - |sn|}. \end{aligned} \tag{1}$$

The "graph similarity" plug-in of ProM 5.2 was used (with default settings) to compare the different YAWL models to each other on the EPC level, that is, we first convert both YAWL models to EPC models as described earlier, and compare the resulting EPC models instead.

2.5 Structural process similarity

The SPS metric also considers the EPC to be plain labeled graphs, but uses a combination of:

Syntactic similarity, which considers only the syntax of the labels,
Semantic similarity, which abstracts from the syntax and looks at the semantics of the words within the labels, and
Contextual similarity, which considers not only the labels of the elements themselves, but also the context (surrounding nodes) in which these elements occur.

These metrics determine the similarity score between pairs of elements in the two models. The overall metric has been implemented in the *Process Similarity* tool, which is part of the *Synergia* toolset. For any two EPCs that are provided as input, the Process Similarity tool calculates their SPS similarity, which is a decimal value between 0 and 1, where 1 means that the processes are identical.

2.6 Control-flow complexity (CFC)

Aside from the comparison between models, the paper also strives to give complexity measures of individual models [15]. One of the metrics used is the control-flow complexity (CFC) as introduced in [5]:

$$CFC(G_{EPC}) = \sum_{n \in N_S} CFC(n)$$

where $G_{EPC} = (N_F \cup N_E \cup N_C, E)$ is the corresponding EPC model with functions N_F, events N_E, connectors N_C, and edges E, and N_S is the set of split nodes ($N_S \subseteq N_C$). For a split node $n \in N_S$ with fan out k (number of output arcs):

$$CFC(n) = \begin{cases} 1 & \text{if } n \text{ is an AND-split;} \\ k & \text{if } n \text{ is an XOR-split;} \\ 2^k & \text{if } n \text{ is an OR-split.} \end{cases}$$

The “EPC complexity analysis” plug-in of ProM 5.2 was used to determine the CFC metric. Again, we first convert the YAWL model at hand to an EPC model, and determine the CFC of the resulting EPC model instead. The CFC metric of the YAWL model as shown by Figure 1 yields $2 + 2 + 1 = 5$, as in the resulting EPC model (see Figure 2) both XOR-split connectors have CFC value 2 and the AND-split connector has CFC value 1.

2.7 Density

Another complexity metric used in this paper, is the density metric as discussed in [15]. In general, for a graph $G = (N, E)$ with nodes N and edges E, this metric corresponds to the number of actual arcs divided by the maximal number of possible arcs, which can be computed as

$$Density(G) = \frac{|E|}{|N| \cdot (|N| - 1)}$$

However, for an EPC model $G_{EPC} = (N_F \cup N_E \cup N_C, E)$ with functions N_F, events N_E, connectors N_C, and edges E we know that functions and events do not allow for multiple input and/or output edges. Therefore, for computing the density metric we take only the connectors into account by using

$$Density(G_{EPC}) = \frac{|E| - |N_F| - |N_E|}{|N_C| \cdot (|N_C| - 1)}$$

This metric is computed with the help of "EPC complexity analysis" plug-in of ProM 5.2, and in a similar way. However, the density metric as returned by this plug-in does not correspond to the density metric as defined in [15]. Instead, it corresponds to the density metric as defined in [14]. Luckily, from the former density metric we could quite easily compute the latter density metric. The density metric of the YAWL model shown by Figure 1 yields $\frac{38-14-15}{6 \cdot 5} = 0.3$.

2.8 Cross-connectivity (CC)

A third density metric is the cross-connectivity metric (CC) as defined in [20]. This metric computes the maximal weights for any path between every two nodes, and divides this by the number of paths between every two nodes. The weight of a path equals the product of the weight of the nodes on this path, where:

- the weight of an XOR connector equals $\frac{1}{d}$ (where d is the degree of the node, that is, the total number of input and output arcs of the node),
- the weight of an OR connector equals $\frac{1}{2^d+1} + \frac{2^d-2}{2^d-1} \cdot \frac{1}{d}$, and
- the weight of every other node (functions, events, AND connectors) equals 1.

In contrast to the other two complexity metrics, which are assumed to be better if lower, the CC metric is assumed to be better if higher.

This metric is computed as well by the "EPC complexity analysis" plug-in of ProM 5.2. However, the computation by this plug-in for computing this metric suffers from two problems: it runs out of space, and it runs out of time. The first problem was solved by a rearrangement of the algorithm, whereas the second problem was tackled by imposing a weight threshold to any path under consideration: A path will only be extended if its current weight exceeds this threshold. The CC metric of the YAWL model as shown by Figure 1 yields approx. 0.1169.

2.9 k-means clustering

k-means clustering is a standard technique to partition a data set into k clusters. First, k initial cluster centers are determined (randomly) and each data element is assigned to the closest of these centers. The center of each cluster is recomputed (take the average of all its data elements) and the data elements are again assigned to the closest of these centers. This is repeated several times to find k cluster centers with minimal distances to elements corresponding to these centers. We will use k-means clustering to find processes and municipalities that are most similar, and we will use "Weka 3.6.5" to do this clustering with the following parameters:

```
Scheme:weka.clusterers.SimpleKMeans -N 3 -A "weka.core.EuclideanDistance
-R first-last" -I 500 -S 10
```

that is, find 3 clusters, use Euclidian distance, do 500 iterations, and use 10 as the initial seed.

3 YAWL models

We collected 80 YAWL 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).

Five of the mentioned municipalities started working together in 2010. They share a service center, which provides most of the IT-support the municipalities need. They also share a social services provider. The remaining five municipalities also work together in the IT-area, but to a lesser extent: They make use of a commonly developed software system (hosted individually). This system is meant to handle the front-end of all participating municipalities in a similar way, and gets expanded to provide comprehensive workflow support. Needless to say, both these groups of municipalities could greatly benefit from the use of configurable models as they have to deliver the same set of services.

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.

To give an indication of the variety and similarity between the different YAWL models some examples are shown. Figure 3 shows the GBA_1 YAWL model of Mun_E, whereas Figure 1 showed 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

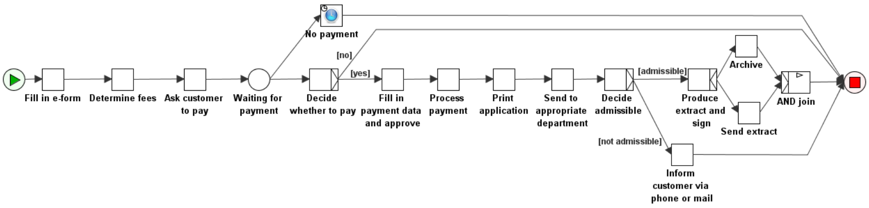

Fig. 3: GBA_1 YAWL model for Mun_E

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.

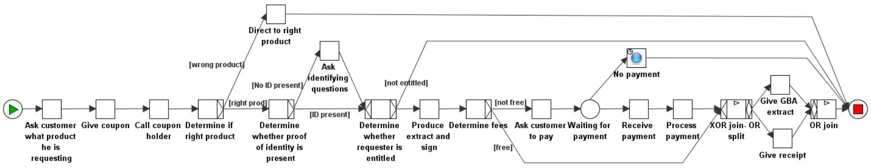

Fig. 4: GBA_2 YAWL model for Mun_E

People can apply for this document through different means too. Figure 4 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 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 4 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 5 shows the WOZ YAWL model for Mun_E, which is clearly different from the three GBA models. The WOZ model shown in Figure 5 is more time-consuming. Customers need to be heard and their objections need to be assessed thoroughly. Next, 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

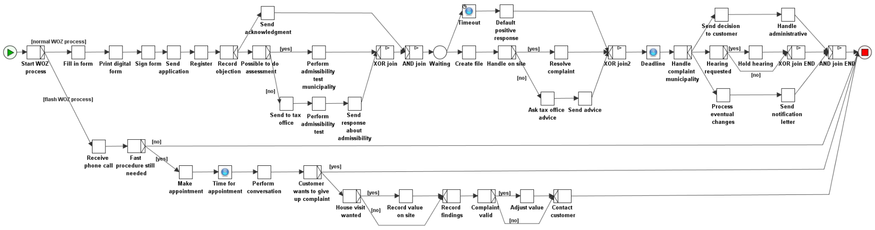

Fig. 5: WOZ YAWL model for Mun_E

Table 1: Complexity metrics *GBA1* process

	Mun_A	Mun_B	Mun_C	Mun_D	Mun_E	Mun_F	Mun_G	Mun_H	Mun_I	Mun_J
CFC	6	5	4	5	7	5	5	6	5	3
Density	0.350	0.400	0.667	0.350	0.300	0.350	0.300	0.350	0.350	0.417
CC	0.078	0.205	0.172	0.167	0.108	0.180	0.117	0.078	0.180	0.184

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), 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.

4 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 the complexity metrics as discussed earlier to all YAWL models. Second, we compare the models using the *GED* similarity metric as described in [7]. Third and last, we answer the three questions as proposed earlier using these metrics.

4.1 Complexity

For every YAWL model, we calculated the CFC, density, and CC metric to get an indication of its complexity. The results can be found in Appendix B. As an example,

Table 2: 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

Table 3: *GED* similarities *GBA1* Process

	Mun_A	Mun_B	Mun_C	Mun_D	Mun_E	Mun_F	Mun_G	Mun_H	Mun_I	Mun_J
Mun_A	1.000	0.837	0.817	0.883	0.845	0.803	0.667	1.000	0.942	0.698
Mun_B	0.837	1.000	0.772	0.915	0.841	0.842	0.708	0.837	0.896	0.769
Mun_C	0.817	0.772	1.000	0.807	0.799	0.798	0.665	0.817	0.798	0.664
Mun_D	0.883	0.915	0.807	1.000	0.884	0.891	0.719	0.883	0.950	0.801
Mun_E	0.845	0.841	0.799	0.884	1.000	0.851	0.732	0.845	0.908	0.858
Mun_F	0.803	0.842	0.798	0.891	0.851	1.000	0.711	0.803	0.879	0.793
Mun_G	0.667	0.708	0.665	0.719	0.732	0.711	1.000	0.667	0.717	0.723
Mun_H	1.000	0.837	0.817	0.883	0.845	0.803	0.667	1.000	0.942	0.698
Mun_I	0.942	0.896	0.798	0.950	0.908	0.879	0.717	0.942	1.000	0.793
Mun_J	0.698	0.769	0.664	0.801	0.858	0.793	0.723	0.698	0.793	1.000

Table 1 shows the complexity metrics for all GBA_1 models. Figure 6 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 6 ($CC(G) = 0.4611 \cdot CFC(G)^{-0.851}$ 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 2 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.

4.2 Similarity

For every pair of YAWL models from the same sub-collection, we calculated the GED and SPS metric to get an indication of their similarity. The results can be found in Appendix C. As an example, Table 3 shows the *GED* similarity metrics for the GBA_1 YAWL models. In the table, the minimum is 0.664 and the maximum element (excluding the main diagonal) is 1.000. Figure 7 shows the relation between the GED

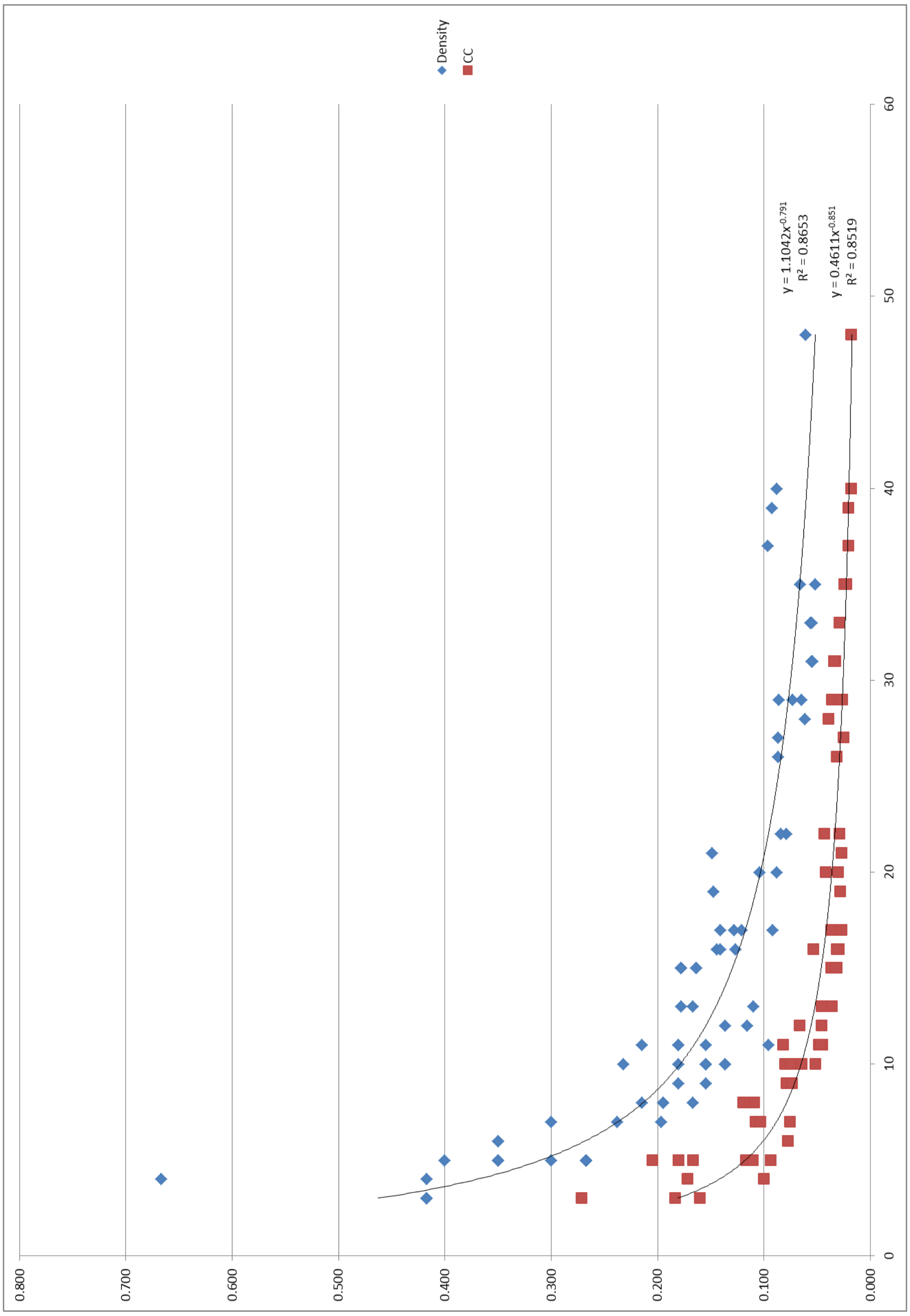

Fig. 6: Comparison of the CFC metric with the CC and Density metrics.

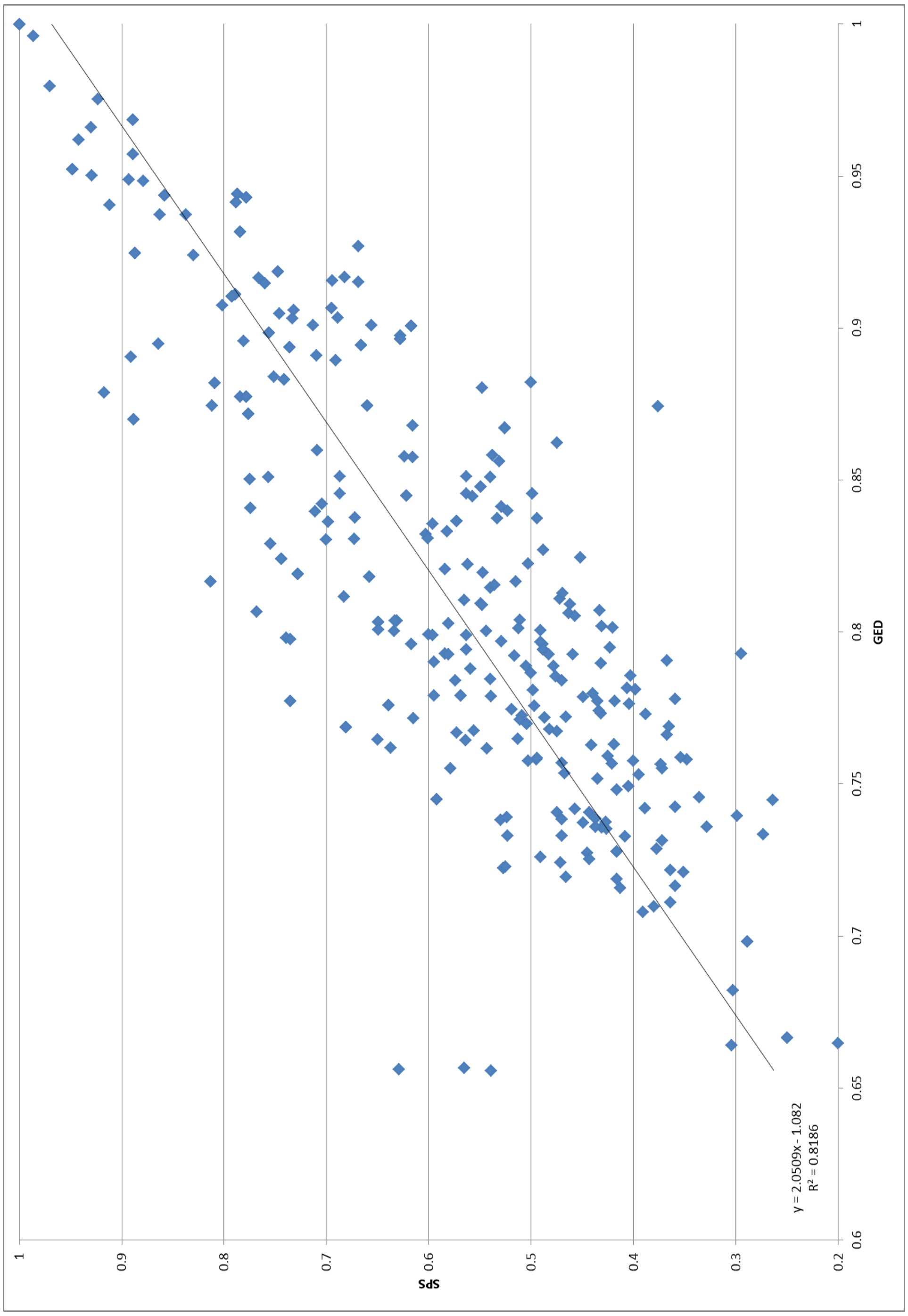

Fig. 7: Comparison of the GED metric with the SPS metric.

Table 4: 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

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 4 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.

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.

As shown in Table 3, 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), 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 8 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.

4.3 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

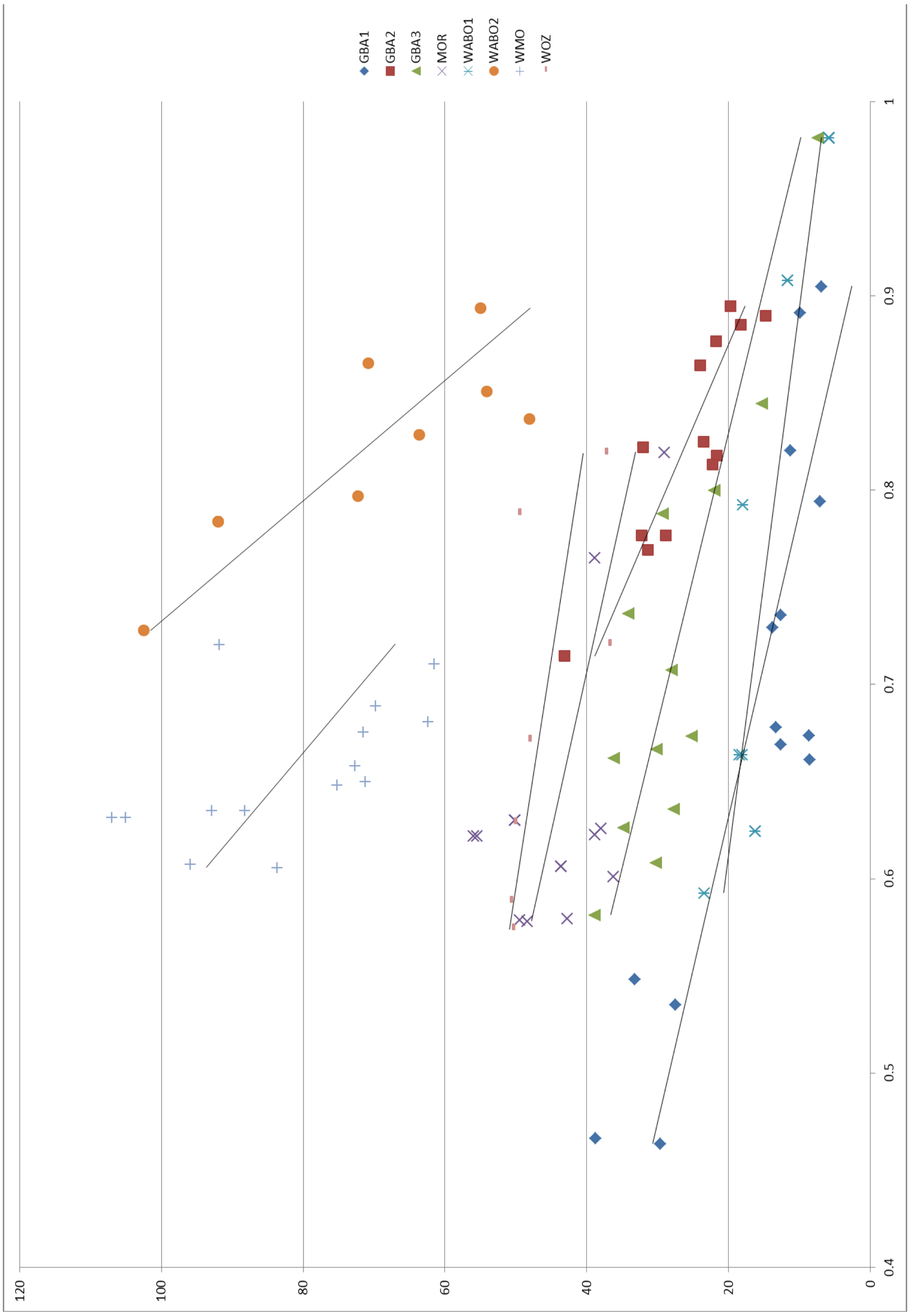

Fig. 8: Unified similarity vs. unified complexity for 100 pairs of models.

Table 5: 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 6: Comparing $WABO_2$ and WMO for Mun_D

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

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 5 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 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 6 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.

Table 7: 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 8: Comparing Mun_H and Mun_A for Mun_D

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

From Table 5 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.

4.4 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 7 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 8 shows the results. Clearly, the average 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 7 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.

4.5 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). Let's for example take Table 3. This table contains the similarity metrics for the *GBA1* processes.e 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 $Mun_A + Mun_C + Mun_H$.

To further illustrate the correlation between the similarity and the complexity of a configurable model, we present Table 9, which shows the complexity metrics for the configurable models for the clusters obtained from the k-means clustering approach, and the metrics for the configurable models for the clusters in 10 random clusterings. Note that for sake of brevity we have simply used A for Mun_A etc. Observe that the complexity metrics for the suggested clustering are better than the metrics for any of the randomly selected clusters.

Table 10 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 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.

5 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 [5], the density metric from [15], and the cross-connectivity (CC) metric from [20] 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 ($WABO_2$

Table 9: Comparing *GBA1* clusterings

	Per cluster	Average over clusters
BDEFI	17	15
GJ	15	
ACH	12	
AF	13	15
G	5	
BCDEHIJ	28	
AJ	15	28
BDGH	48	
CEFI	21	
EIJ	11	23
ACFH	21	
BDG	36	
CEFI	21	26
BJ	12	
ADGH	46	
E	6	27
CFHJ	26	
ABDGI	48	
ABCF	27	26
DEIJ	12	
GH	39	
F	4	26
BCDH	25	
AEGIJ	49	
CEFIJ	25	24
BG	35	
ADH	13	
AEGJ	49	25
CH	12	
BDFI	14	
BCDGI	50	27
FH	13	
AEJ	18	

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 [7]. Based on the combined similarity of two process

Table 10: 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

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.

The algorithm to compute the CC metric in the "EPC complexity analysis" plug-in of ProM 5.2 was unable to cope with larger process models: It frequently ran out of space, and out of time. Furthermore, the density metric as computed by this plug-in does not correspond to the density metric as defined in [15]. Instead, it corresponds to the metric as defined in [14]. Finally, the label matching as used by the "EPC merge" plug-in of ProM 5.2 (that was used to obtain a configurable model of two process models) was not tailored towards our needs. As a result, we would have to change the label match by hand, which is extremely error-prone (especially if one has to do this many times) and would require us to remember the match for sake of reference. For these reasons, a new, tailored, version of ProM 5.2 has been build that solves the problem with the CC metric and provides us with a tailored and good match. This version can be downloaded from `http://www.win.tue.nl/coselog/files/ProM-CoSeLoG-20110802.zip`. The problem with the density metric has not been solved by this version, but the density metric as defined in [15] can be computed quite easily from the other metrics the "EPC complexity analysis" plug-in provides.

The merging of models A and B possibly differs from the merging of models B and A. As a result the order in which the merger is applied, can be important for the complexity of the resulting configurable model. Therefore, we would like to look into this issue and determine which order of merging is more suitable for a configurable process, and whether the GED metric could play a role in this. In parallel, we also use *cross-organizational process mining* [1, 2] to compare the actual processes of the municipalities involved in CoSeLoG.

References

1. W.M.P. van der Aalst. Configurable Services in the Cloud: Supporting Variability While Enabling Cross-Organizational Process Mining. In *International Conference on Cooperative Information Systems (CoopIS 2010)*, volume 6426 of *Lecture Notes in Computer Science*, pages 8–25. Springer-Verlag, 2010.
2. W.M.P. van der Aalst. *Process Mining: Discovery, Conformance and Enhancement of Business Processes*. Springer-Verlag, 2011.
3. W.M.P. van der Aalst, M. Dumas, F. Gottschalk, A.H.M. ter Hofstede, M. La Rosa, and J. Mendling. Preserving Correctness During Business Process Model Configuration. *Formal Aspects of Computing*, 22:459–482, May 2010.
4. W.M.P. van der Aalst, A.H.M. ter Hofstede, B. Kiepuszewski, and A.P. Barros. Workflow Patterns. *Distributed and Parallel Databases*, 14(1):5–51, 2003.
5. J. Cardoso. How to Measure the Control-flow Complexity of Web Processes and Workflows. 2005.
6. CoSeLoG. Configurable Services for Local Governments (CoSeLoG) Project Home Page. `www.win.tue.nl/coselog`.
7. R. Dijkman, M. Dumas, B. F. van Dongen, R. Krik, and J. Mendling. Similarity of Business Process Models: Metrics and Evaluation. *Information Systems*, 36(2):498–516, April 2011.
8. F. Gottschalk. *Configurable Process Models*. PhD thesis, Eindhoven University of Technology, The Netherlands, December 2009.
9. A. Hofstede, W.M.P. van der Aalst, M. Adams, and N. Russell. *Modern Business Process Automation: YAWL and its Support Environment*. Springer-Verlag, 2009.
10. G. Keller, M. Nüttgens, and A.W. Scheer. Semantische Processmodellierung auf der Grundlage Ereignisgesteuerter Processketten (EPK). Veröffentlichungen des Instituts für Wirtschaftsinformatik, Heft 89 (in German), University of Saarland, Saarbrücken, 1992.
11. G. Keller and T. Teufel. *SAP R/3 Process Oriented Implementation*. Addison-Wesley, Reading MA, 1998.
12. M. La Rosa. *Managing Variability in Process-Aware Information Systems*. PhD thesis, Queensland University of Technology, Brisbane, Australia, April 2009.
13. M. La Rosa, M. Dumas, A.H.M. ter Hofstede, and J. Mendling. Configurable Multi-perspective Business Process Models. *Information Systems*, 36(2):313–340, 2011.
14. J. Mendling. Testing Density as a Complexity Metric for EPCs. In *German EPC Workshop on Density of Process Models*, 2006.
15. J. Mendling, G. Neumann, and W.M.P. van der Aalst. Understanding the Occurrence of Errors in Process Models Based on Metrics. In *CoopIS 2007*, volume 4803 of *Lecture Notes in Computer Science*, pages 113–130. Springer-Verlag, 2007.
16. A.W. Scheer. *Business Process Engineering, Reference Models for Industrial Enterprises*. Springer-Verlag, Berlin, 1994.
17. W. M. P. van der Aalst, B. F. van Dongen, C. Gnther, A. Rozinat, H. M. W. Verbeek, and A. J. M. M. Weijters. Prom: The process mining toolkit, September 2009.

18. B. F. van Dongen, A. K. Alves de Medeiros, H. M. W. Verbeek, A. J. M. M. Weijters, and W. M. P. van der Aalst. The prom framework: A new era in process mining tool support. In G. Ciardo and P. Darondeau, editors, *Application and Theory of Petri nets 2005*, volume 3536 of *Lecture Notes in computer Science*, pages 444–454, Miami, Florida, June 2005. Springer, Berlin, Germany.
19. B.F. van Dongen, R. Dijkman, and J. Mendling. Measuring Similarity Between Business Process Models. In *Proceedings of the 20th international conference on Advanced Information Systems Engineering*, CAiSE '08, pages 450–464. Springer-Verlag, 2008.
20. I. Vanderfeesten, H. Reijers, J. Mendling, W. van der Aalst, and J. Cardoso. On a Quest for Good Process Models: The Cross-Connectivity Metric. In *Advanced Information Systems Engineering*, pages 480–494. Springer, 2008.
21. J. Vogelaar, B. Luka, and H. Verbeek. Comparing Business Processes to Determine the Feasibility of Configurable Models: A Case Study. Technical report, Eindhoven University of Technology, 2011.
22. M. Weidlich, A. Polyvyanyy, N. Desai, and J. Mendling. Process Compliance Measurement Based on Behavioural Profiles. In *Proceedings of the 22nd international conference on Advanced information systems engineering*, CAiSE'10, pages 499–514. Springer-Verlag, 2010.

A YAWL models

A.1 YAWL models for the GBA_1 process

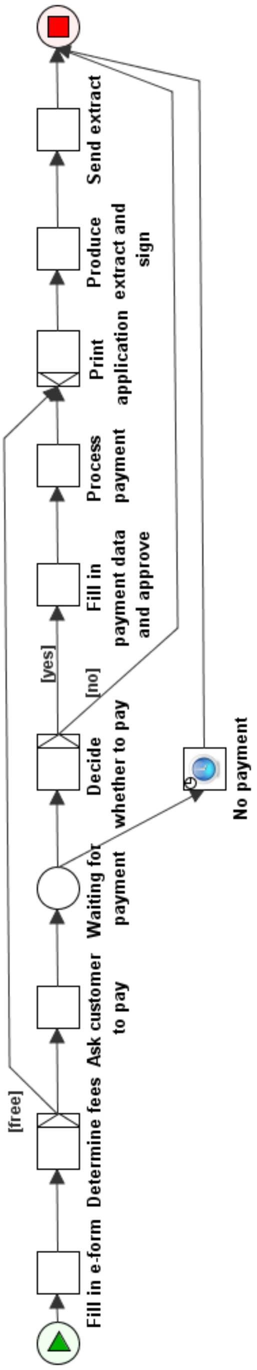

Fig. 9: GBA_1 YAWL model for Mun_A

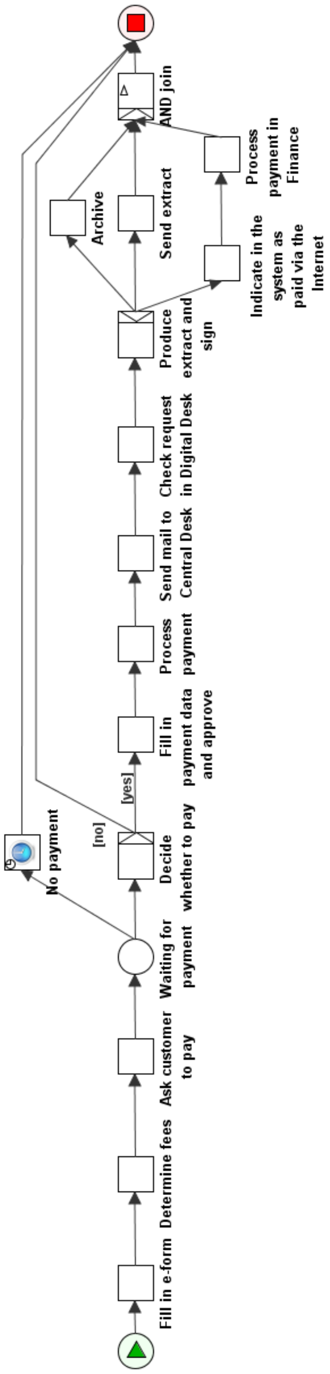

Fig. 10: GBA_1 YAWL model for Mun_B

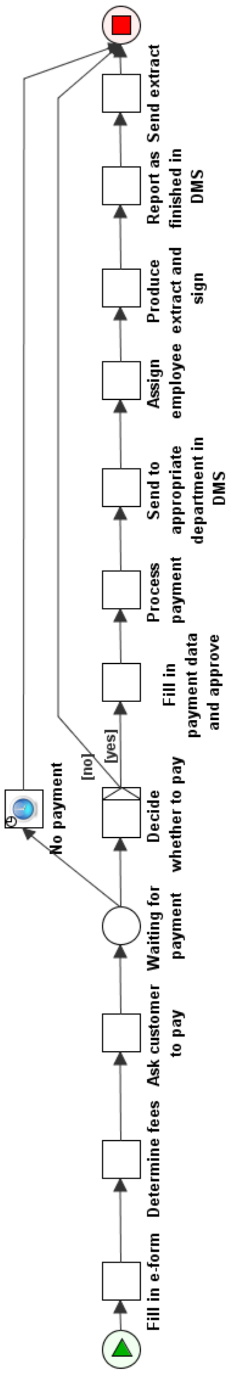

Fig. 11: GBA_1 YAWL model for Mun_C

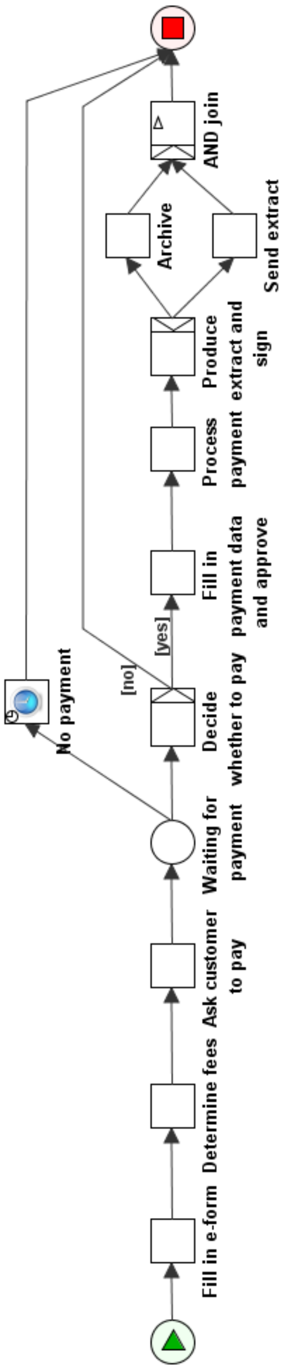

Fig. 12: GBA_1 YAWL model for Mun_D

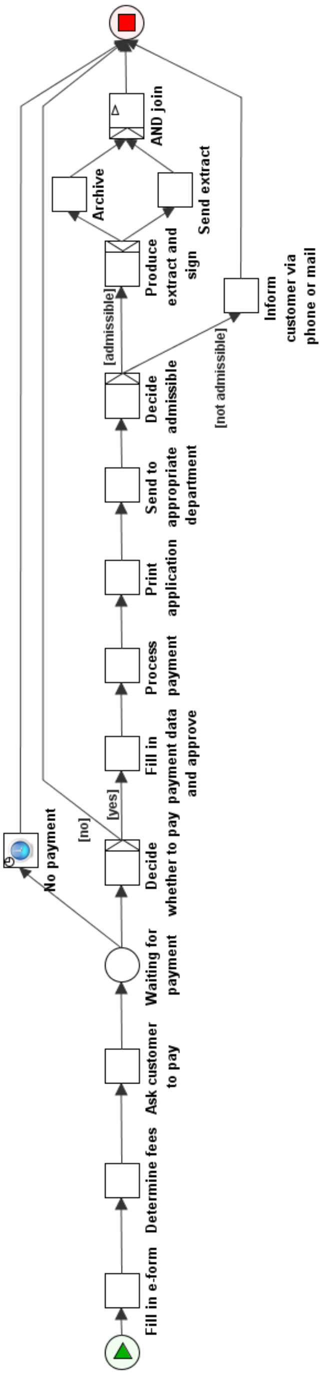

Fig. 13: GBA_1 YAWL model for Mun_E

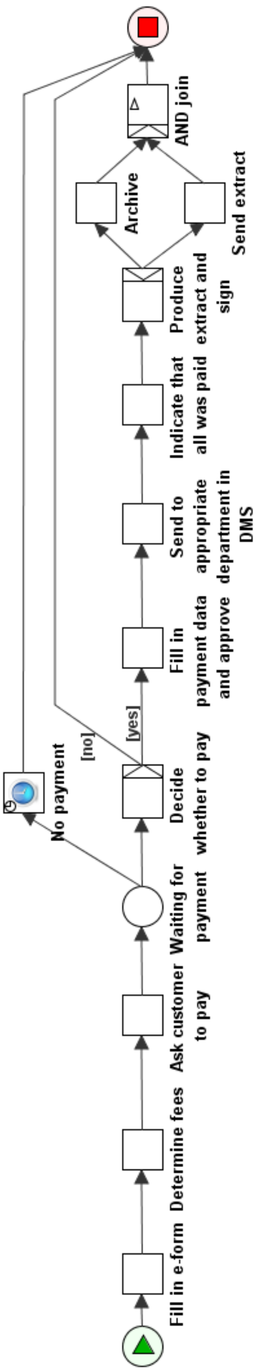

Fig. 14: GBA_1 YAWL model for Mun_F

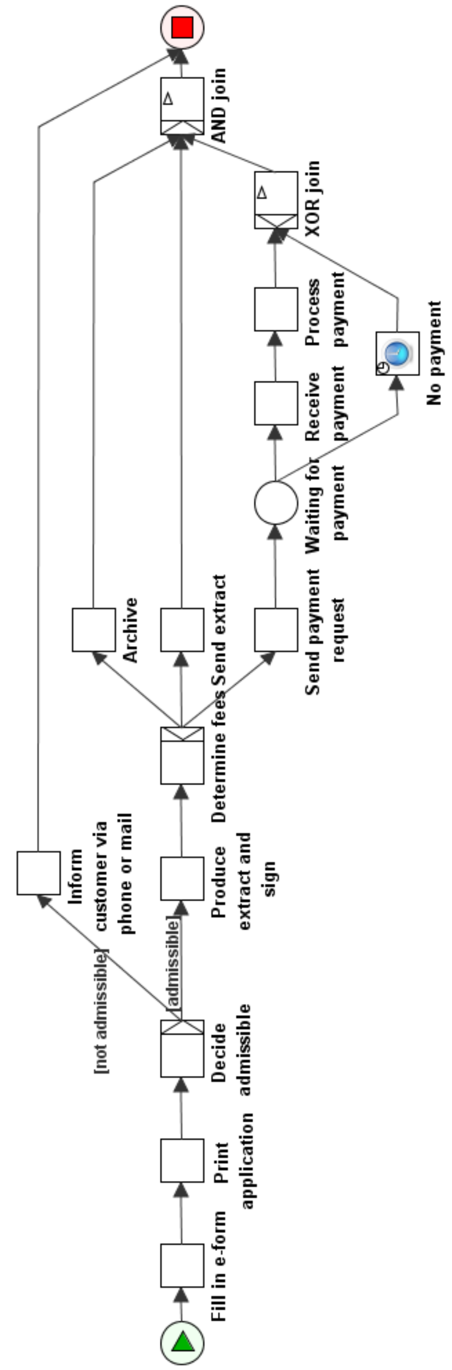

Fig. 15: GBA_1 YAWL model for Mun_G

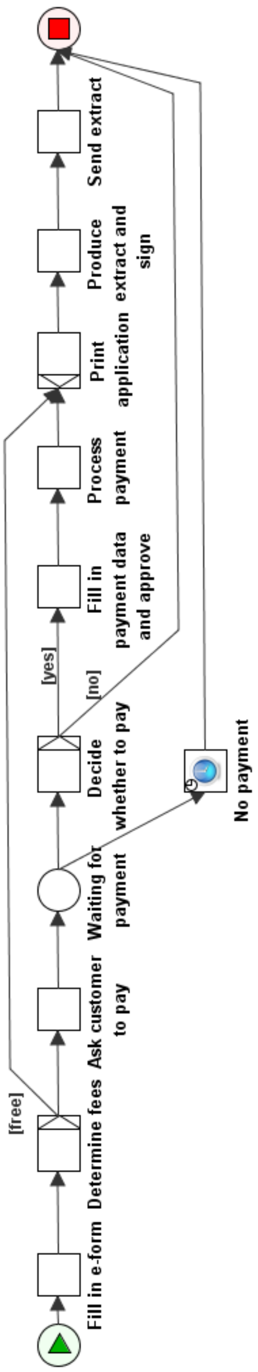

Fig. 16: GBA_1 YAWL model for Mun_H

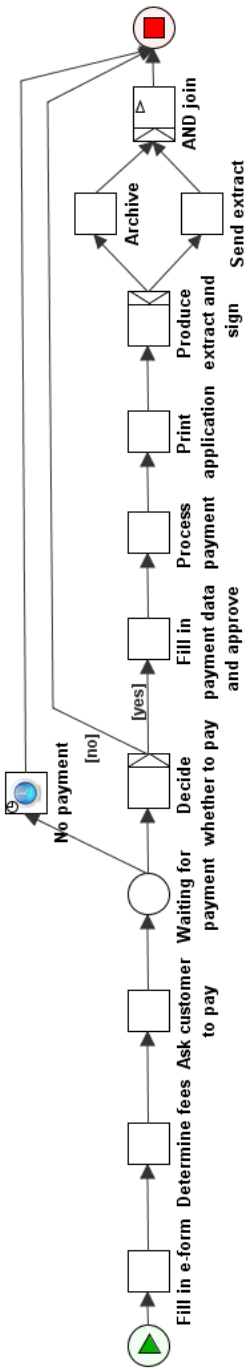

Fig. 17: GBA_1 YAWL model for Mun_I

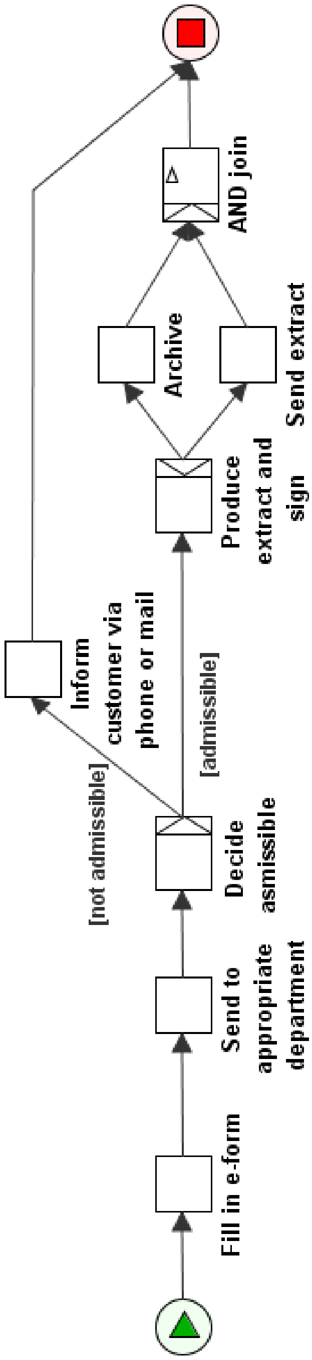

Fig. 18: GBA_1 YAWL model for Mun_J

A.2 YAWL models for the GBA_2 process

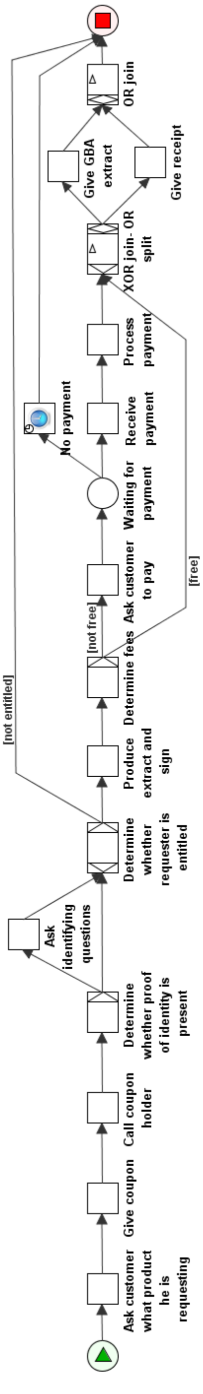

Fig. 19: GBA_2 YAWL model for Mun_A

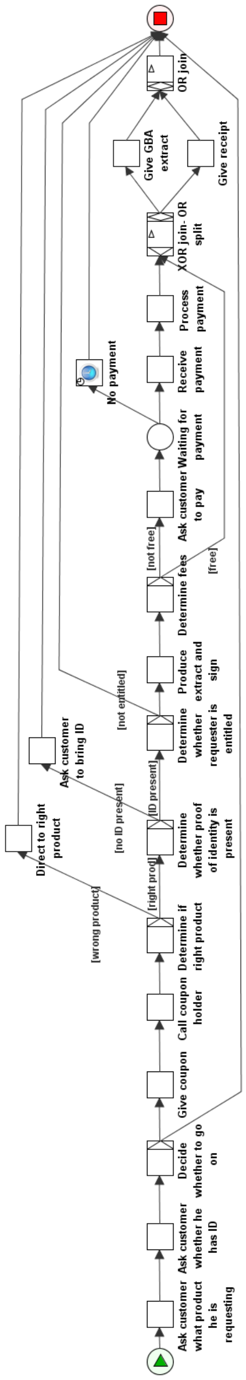

Fig. 20: GBA_2 YAWL model for Mun_B

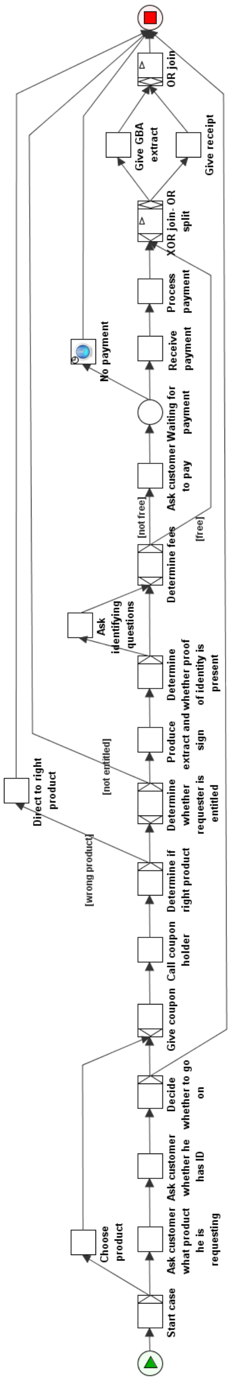

Fig. 21: GBA_2 YAWL model for Mun_C

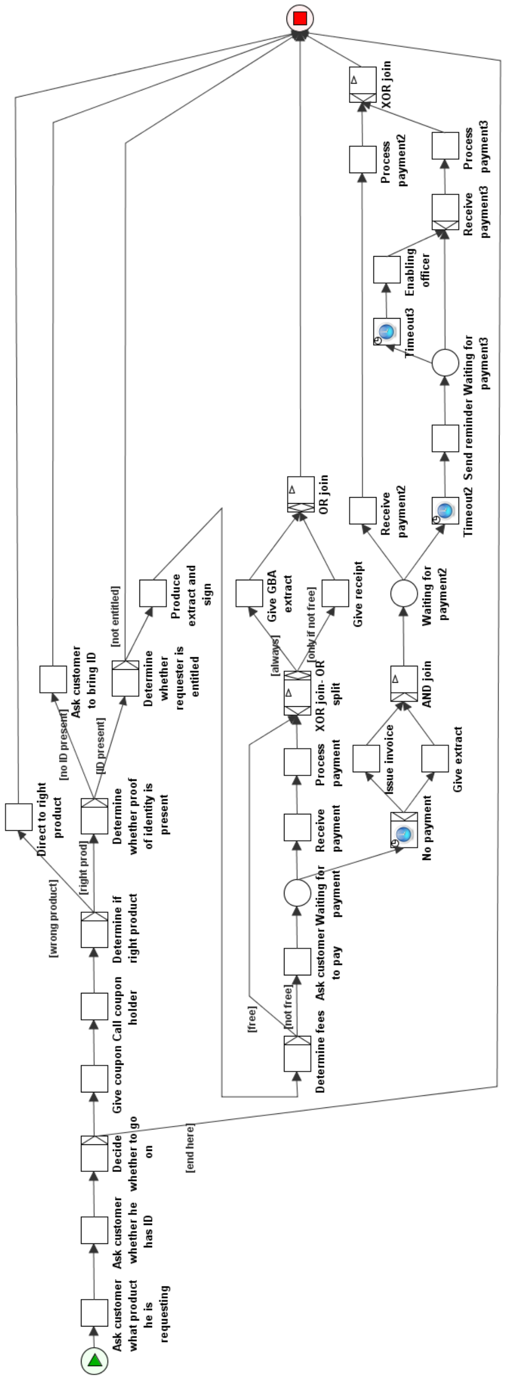

Fig. 22: GBA_2 YAWL model for Mun_D

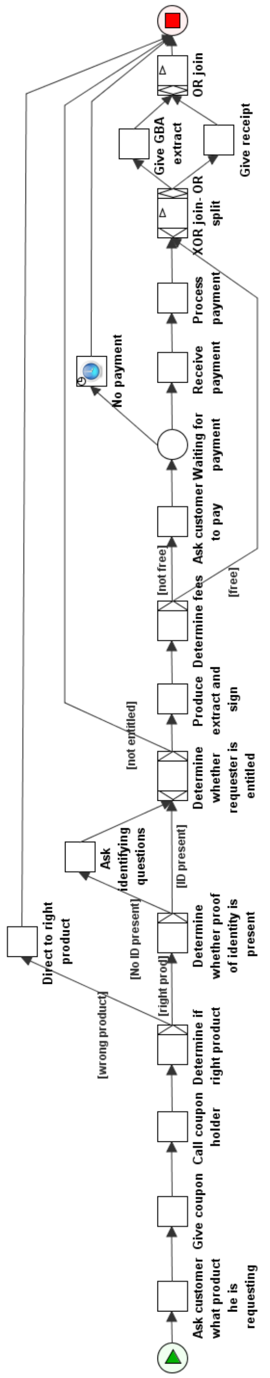

Fig. 23: GBA_2 YAWL model for Mun_E

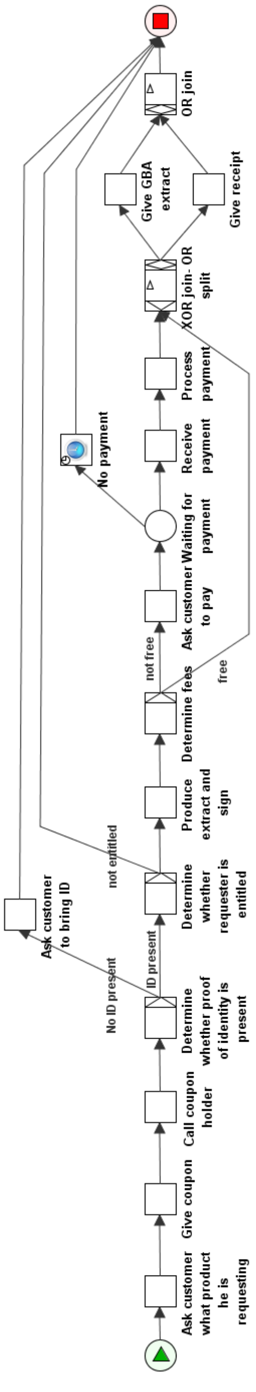

Fig. 24: GBA_2 YAWL model for Mun_F

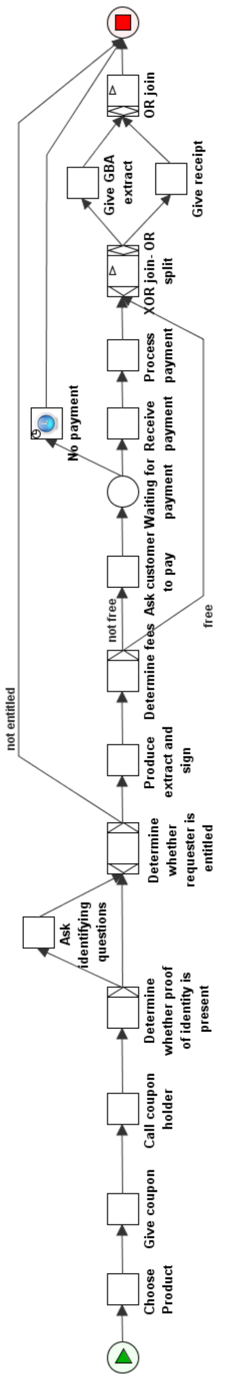

Fig. 25: GBA_2 YAWL model for Mun_G

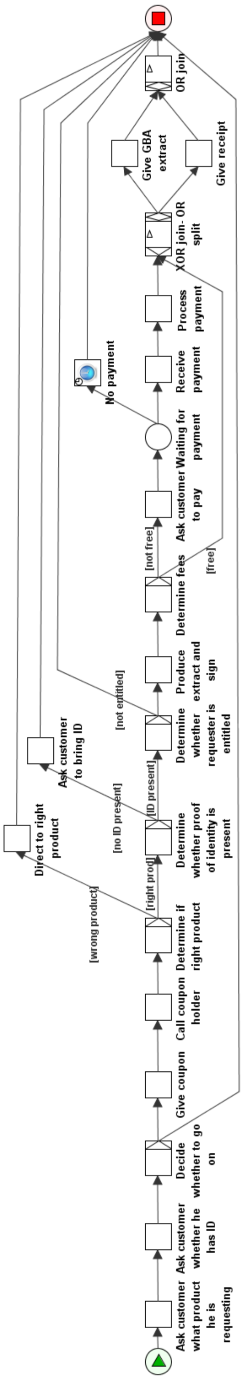

Fig. 26: GBA_2 YAWL model for Mun_H

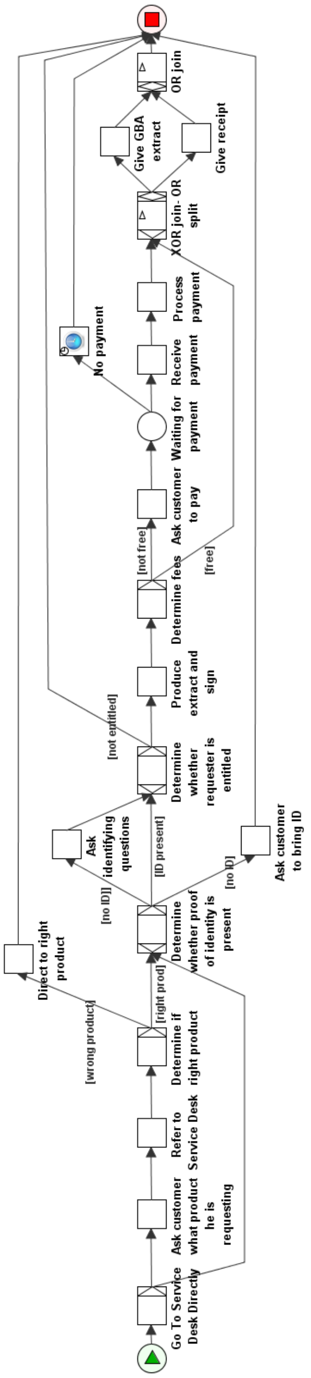

Fig. 27: GBA_2 YAWL model for Mun_I

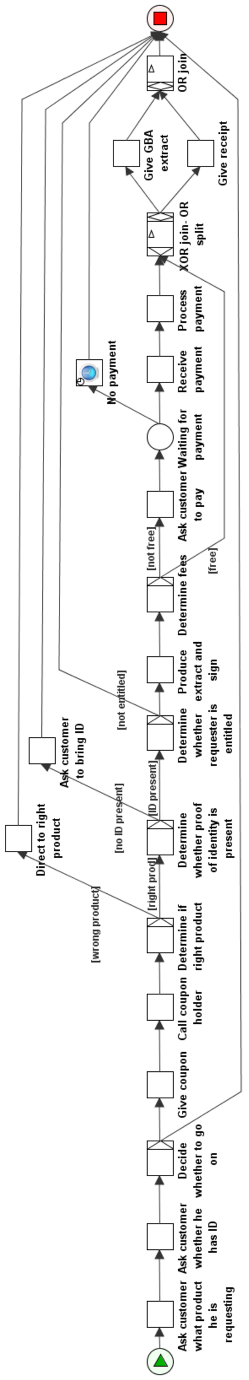

Fig. 28: GBA_2 YAWL model for Mun_J

A.3 YAWL models for the GBA_3 process

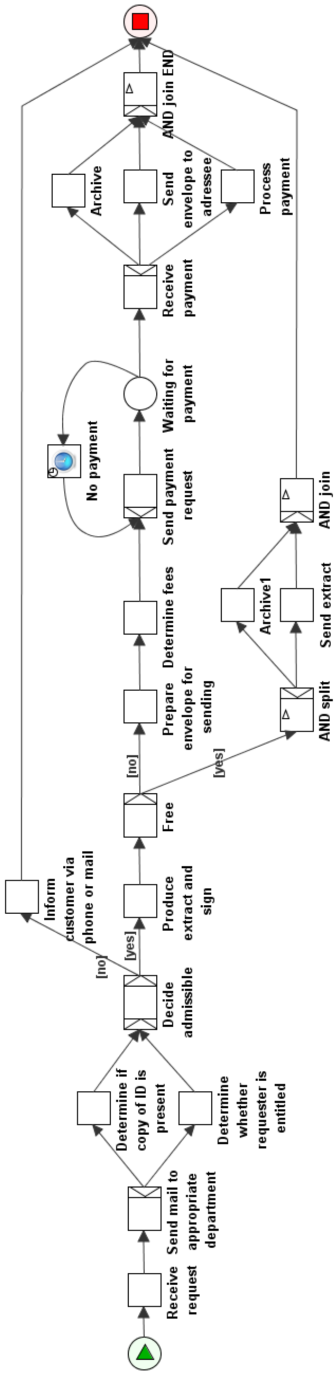

Fig. 29: GBA_3 YAWL model for Mun_A

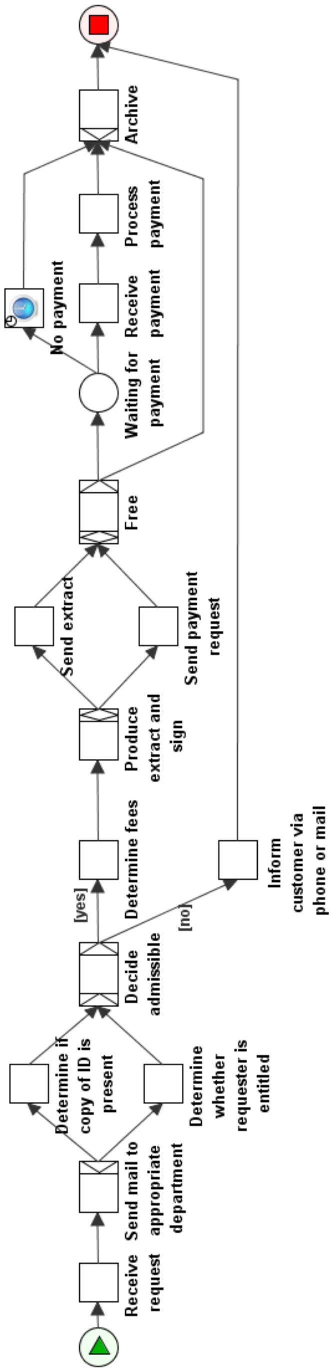

Fig. 30: GBA_3 YAWL model for Mun_B

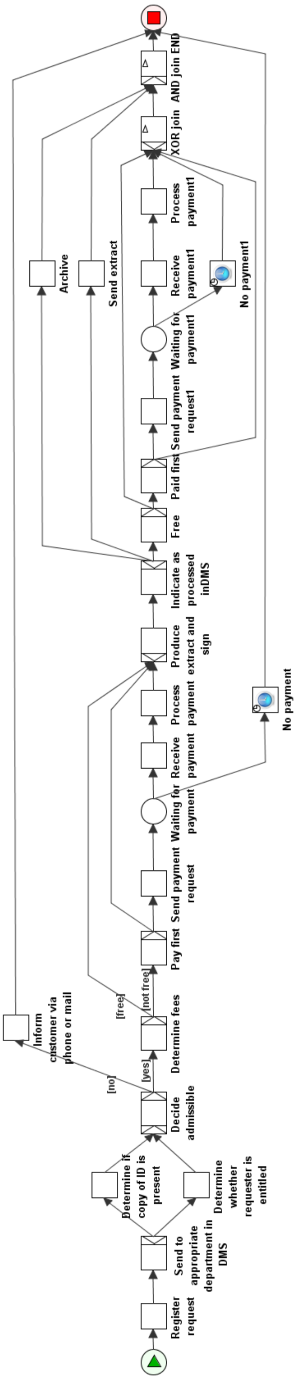

Fig. 31: GBA_3 YAWL model for Mun_C

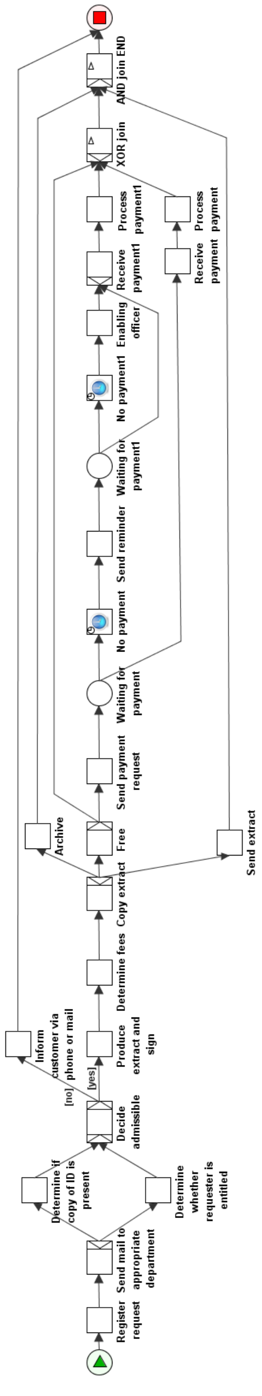

Fig. 32: GBA_3 YAWL model for Mun_D

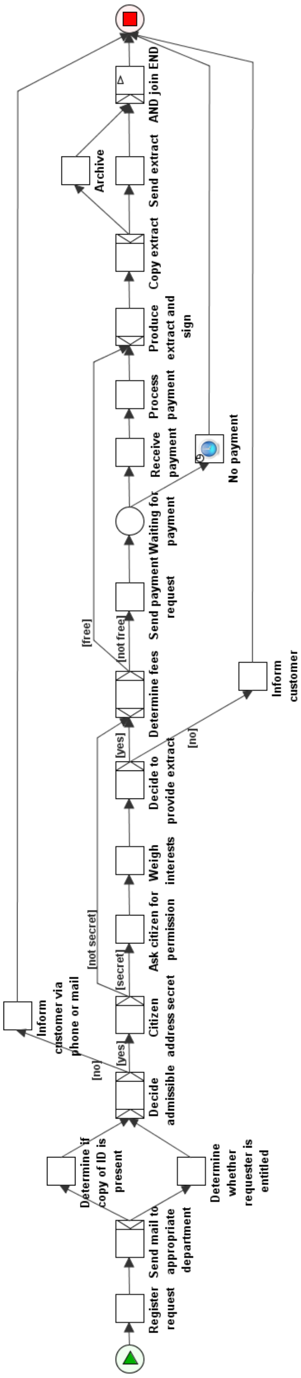

Fig. 33: GBA_3 YAWL model for Mun_E

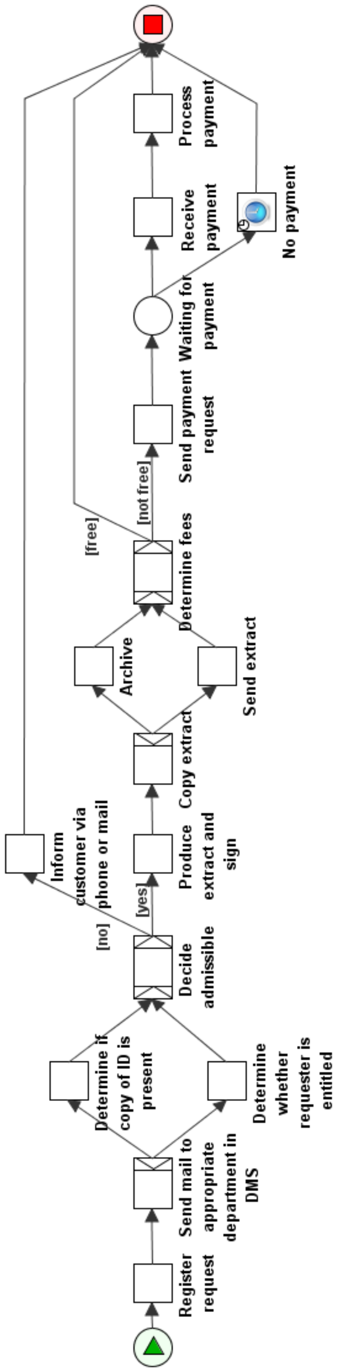

Fig. 34: GBA_3 YAWL model for Mun_F

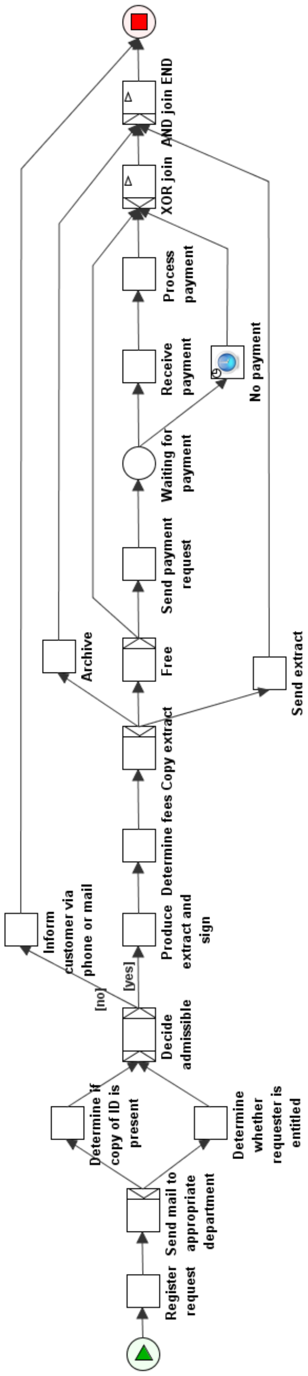

Fig. 35: GBA_3 YAWL model for Mun_G

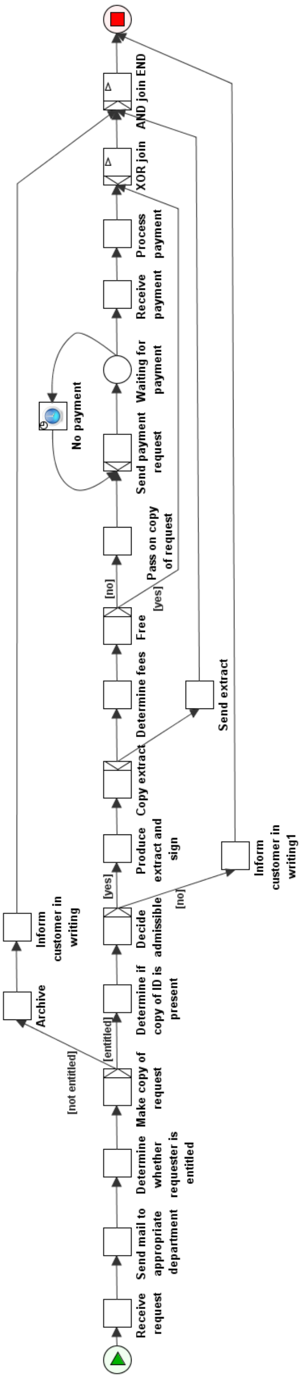

Fig. 36: GBA_3 YAWL model for Mun_H

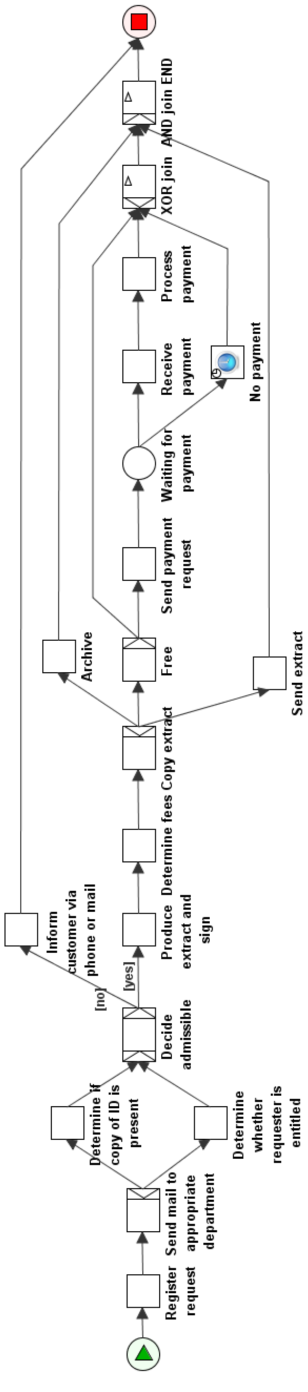

Fig. 37: GBA_3 YAWL model for Mun_I

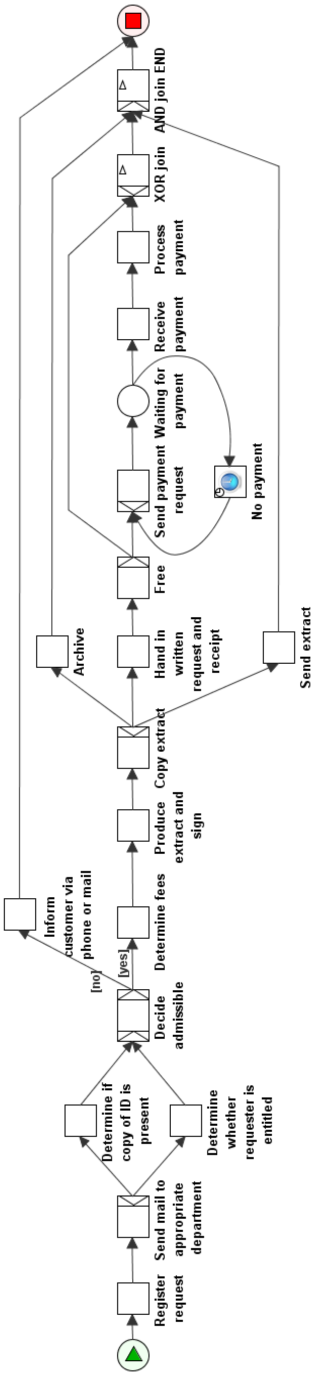

Fig. 38: GBA_3 YAWL model for Mun_J

A.4 YAWL models for the MOR process

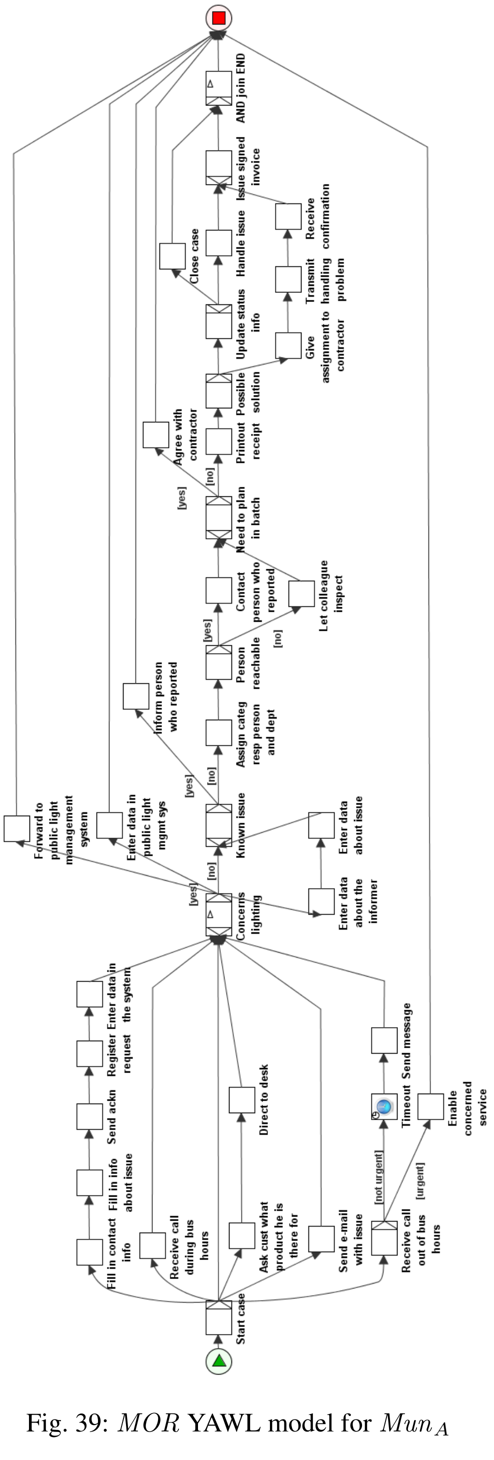

Fig. 39: MOR YAWL model for Mun_A

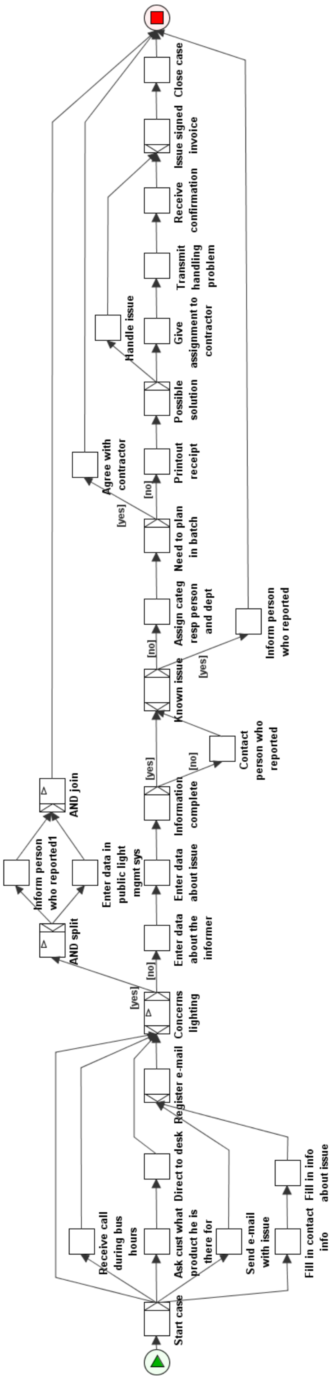

Fig. 40: MOR YAWL model for Mun_B

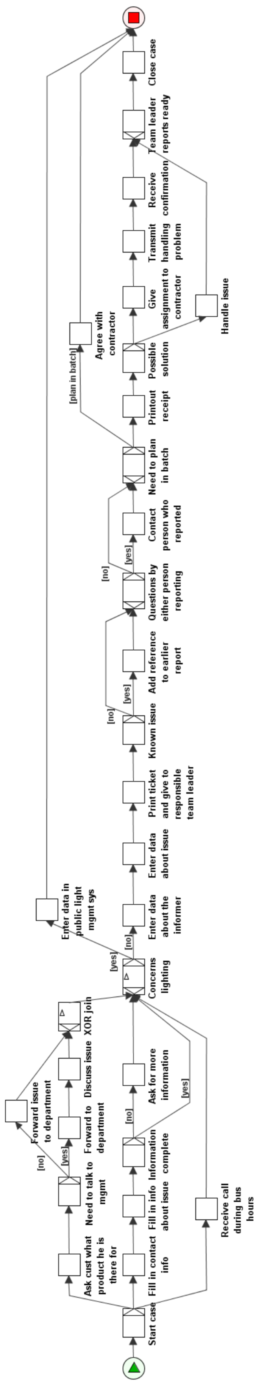

Fig. 41: MOR YAWL model for Mun_C

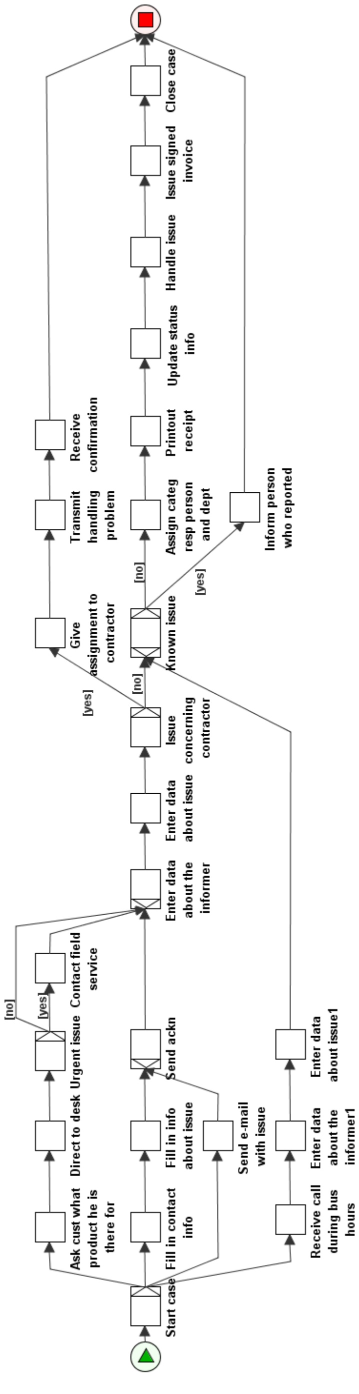

Fig. 42: MOR YAWL model for Mun_D

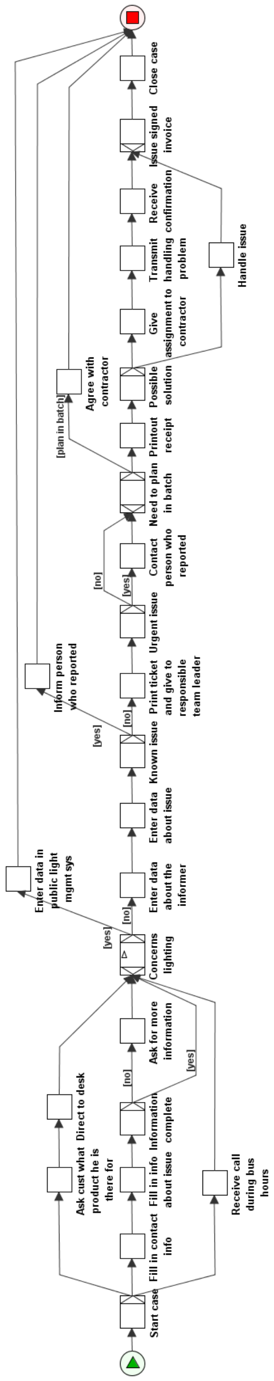

Fig. 43: MOR YAWL model for Mun_E

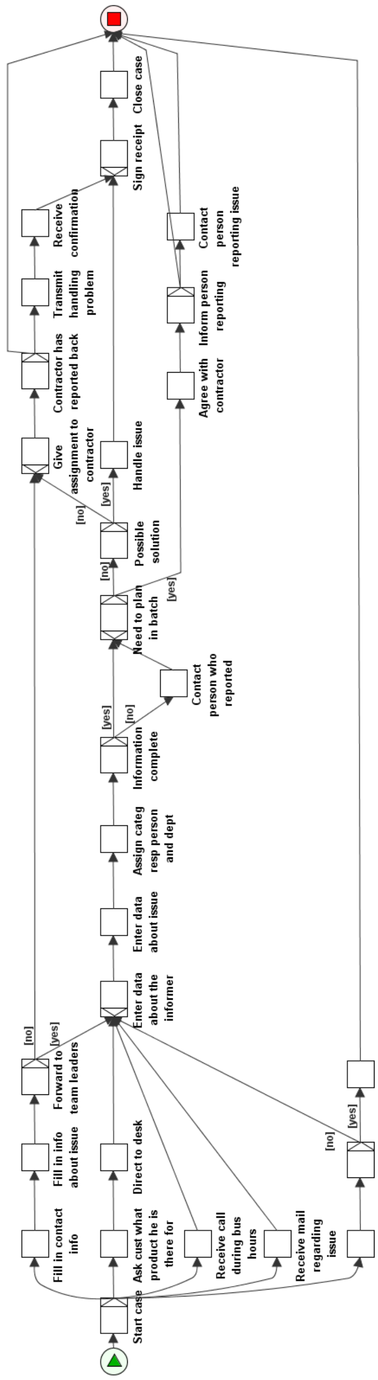

Fig. 44: MOR YAWL model for Mun_F

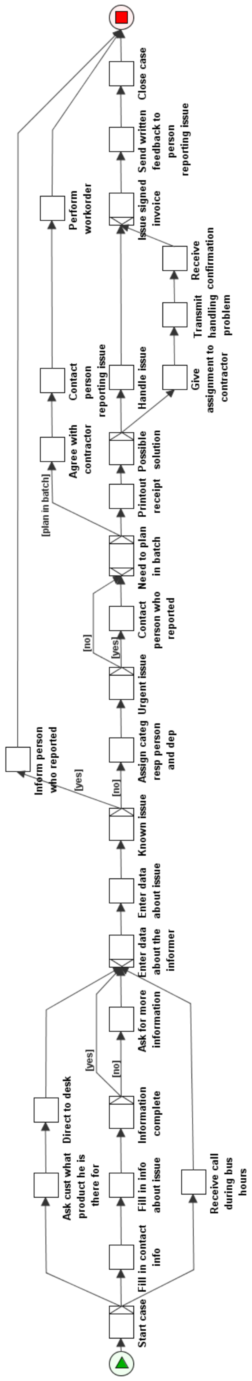

Fig. 45: MOR YAWL model for Mun_G

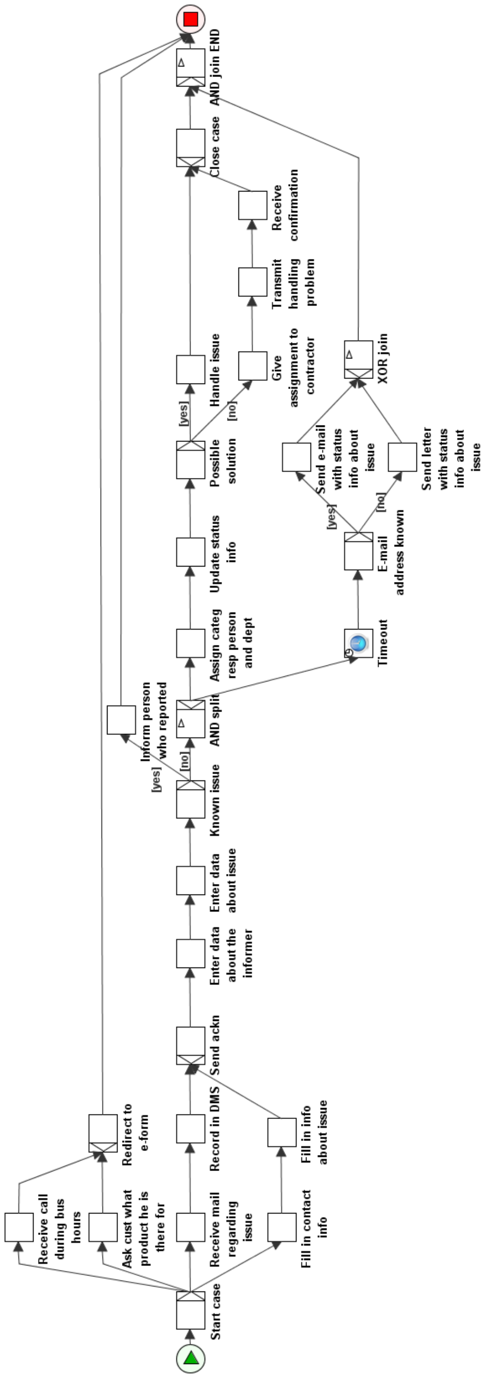

Fig. 46: MOR YAWL model for Mun_H

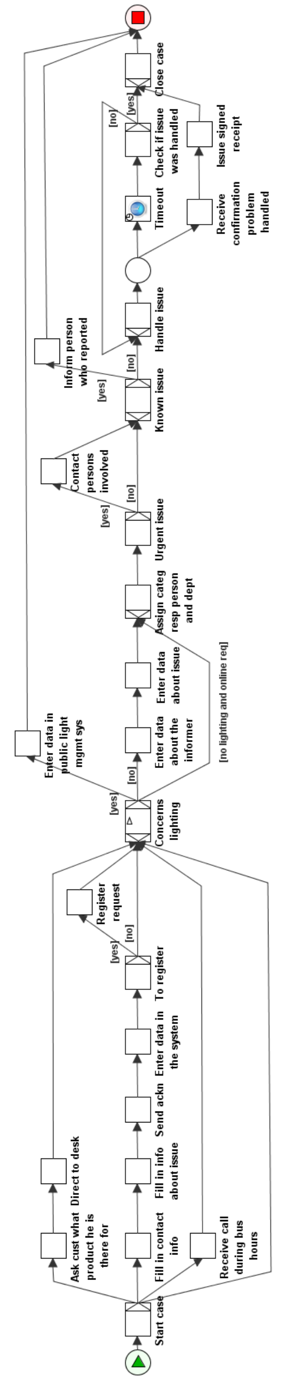

Fig. 47: MOR YAWL model for Mun_I

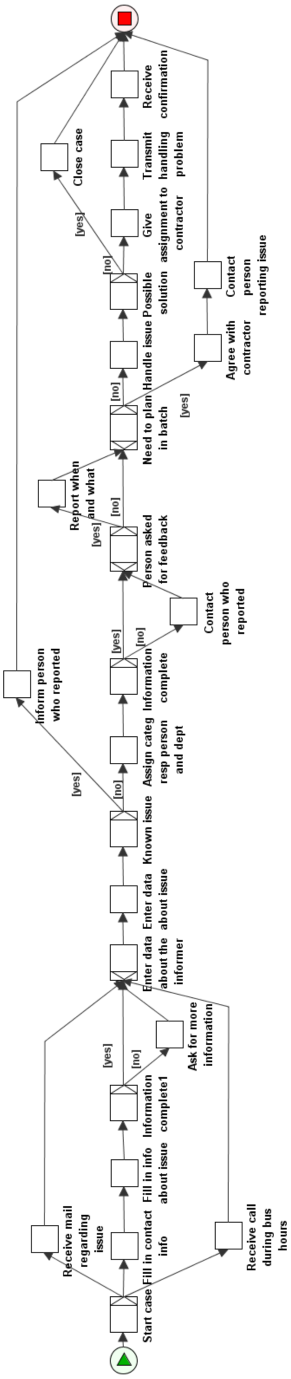

Fig. 48: MOR YAWL model for Mun_J

A.5 YAWL models for the $WABO_1$ process

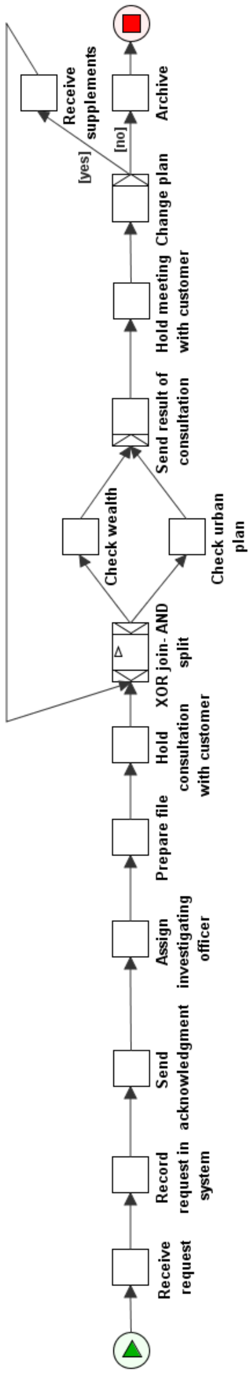

Fig. 49: $WABO_1$ YAWL model for Mun_A

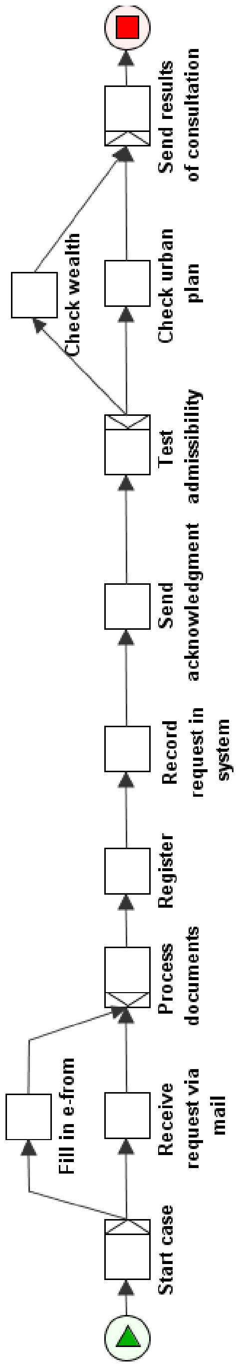

Fig. 50: $WABO_1$ YAWL model for Mun_B

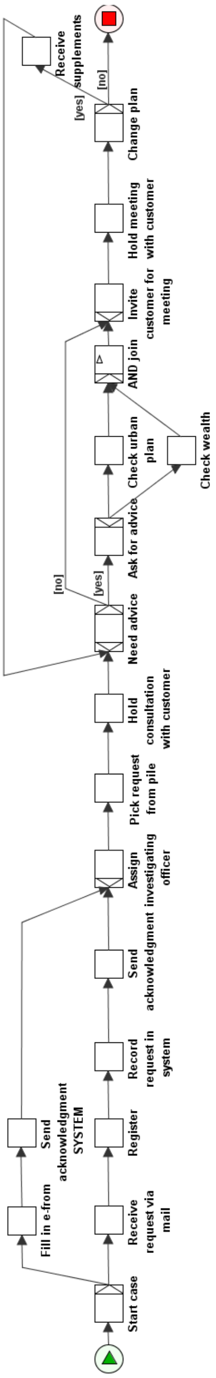

Fig. 51: $WABO_1$ YAWL model for Mun_C

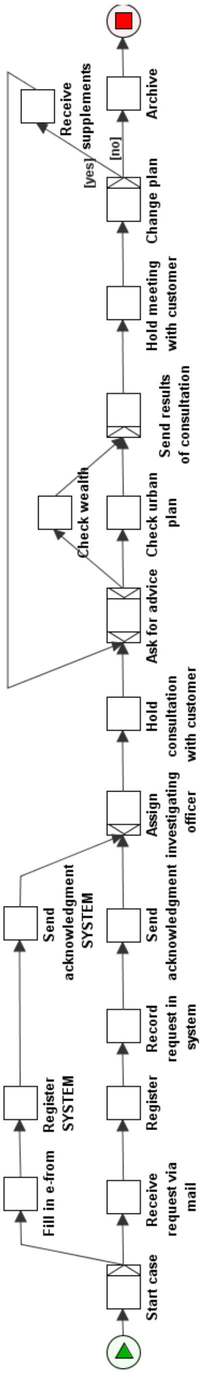

Fig. 52: $WABO_1$ YAWL model for Mun_D

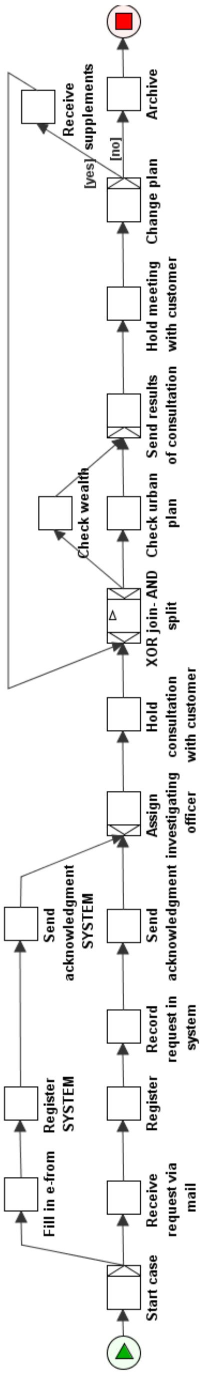

Fig. 53: $WABO_1$ YAWL model for Mun_E

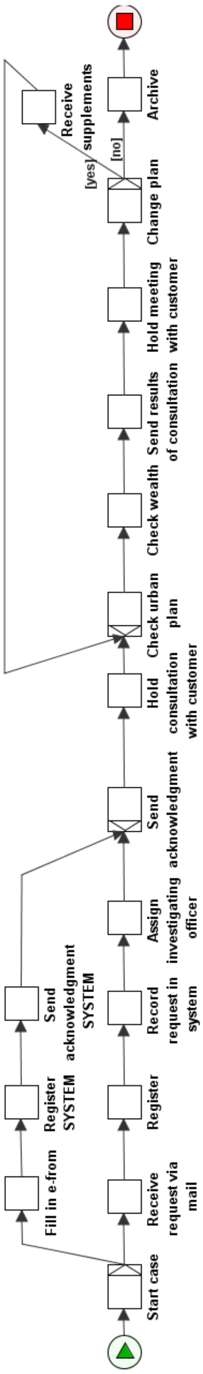

Fig. 54: $WABO_1$ YAWL model for Mun_F

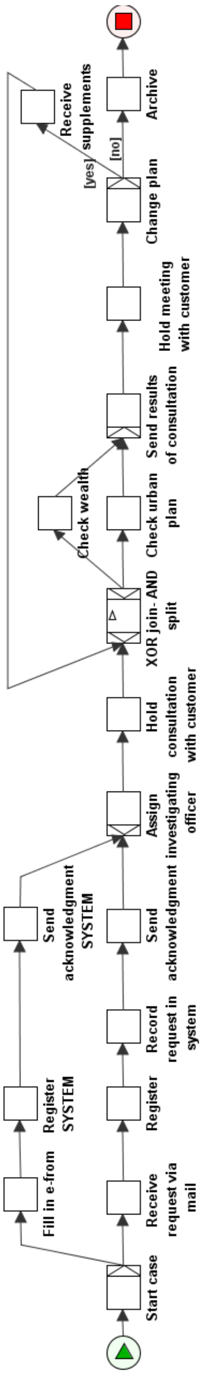

Fig. 55: $WABO_1$ YAWL model for Mun_G

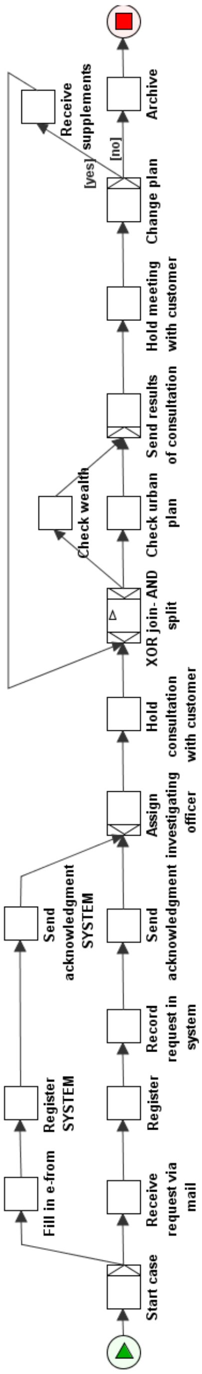

Fig. 56: $WABO_1$ YAWL model for Mun_H

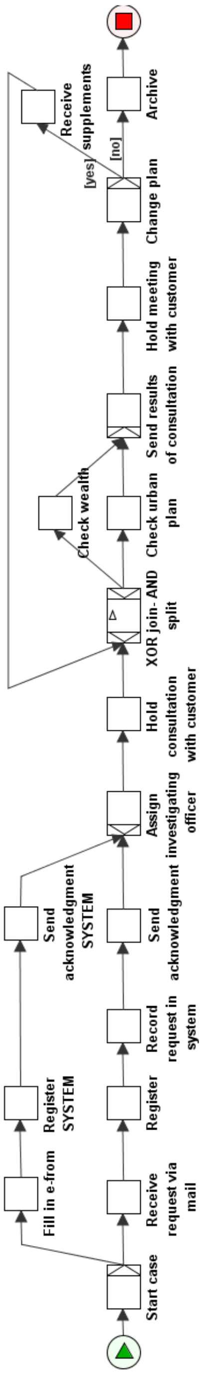

Fig. 57: $WABO_1$ YAWL model for Mun_I

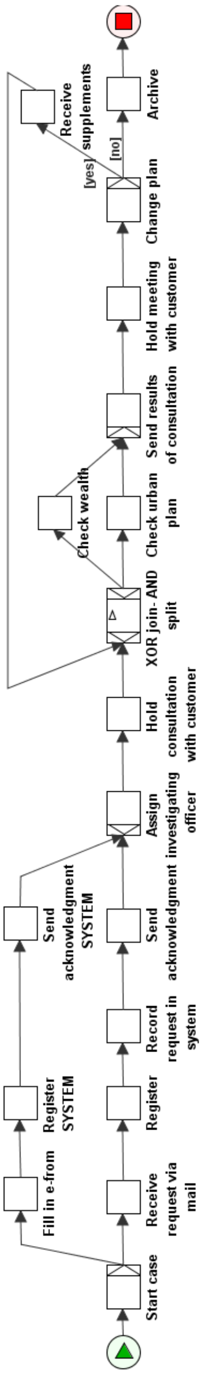

Fig. 58: $WABO_1$ YAWL model for Mun_J

A.6 YAWL models for the $WABO_2$ process

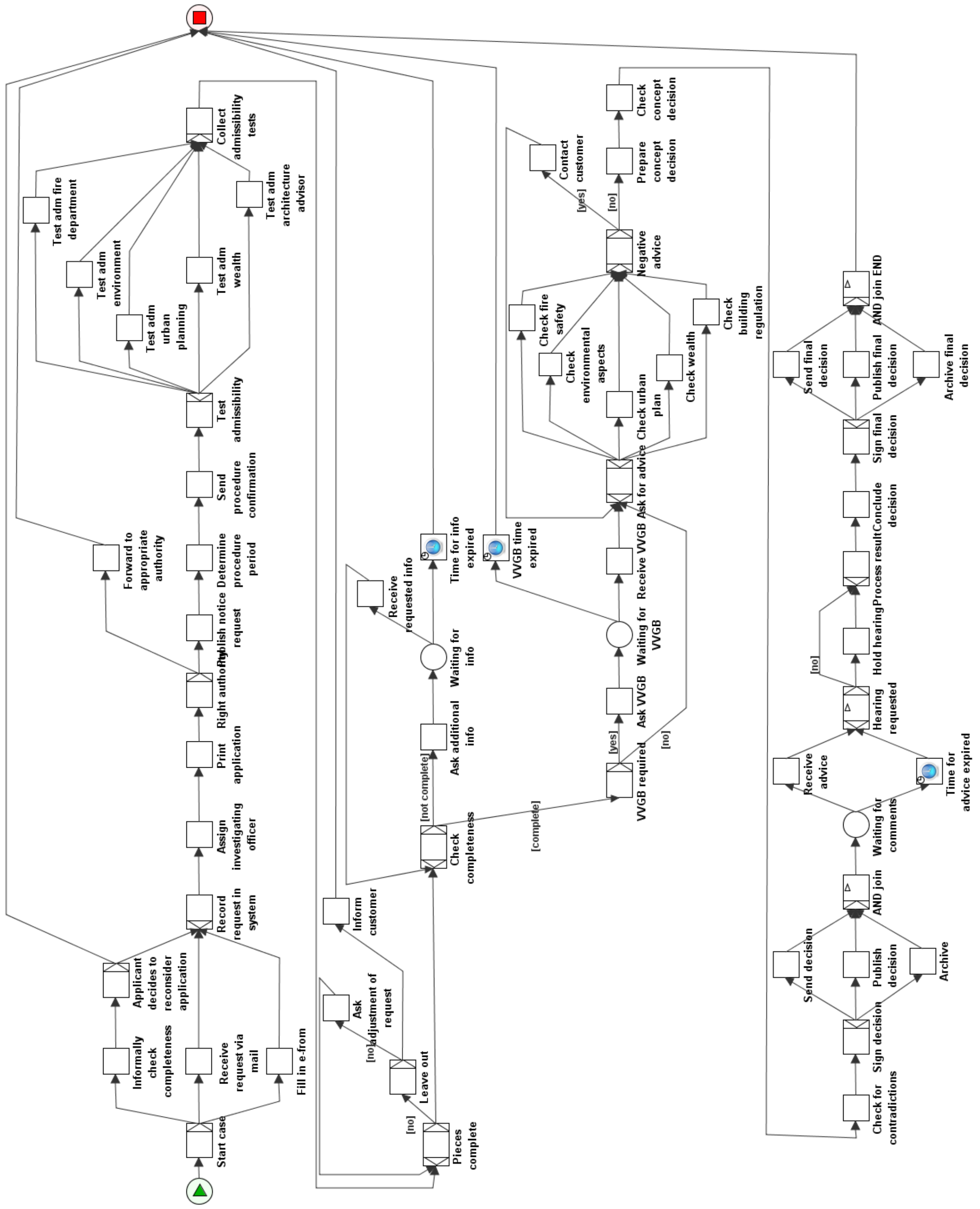

Fig. 59: $WABO_2$ YAWL model for Mun_A

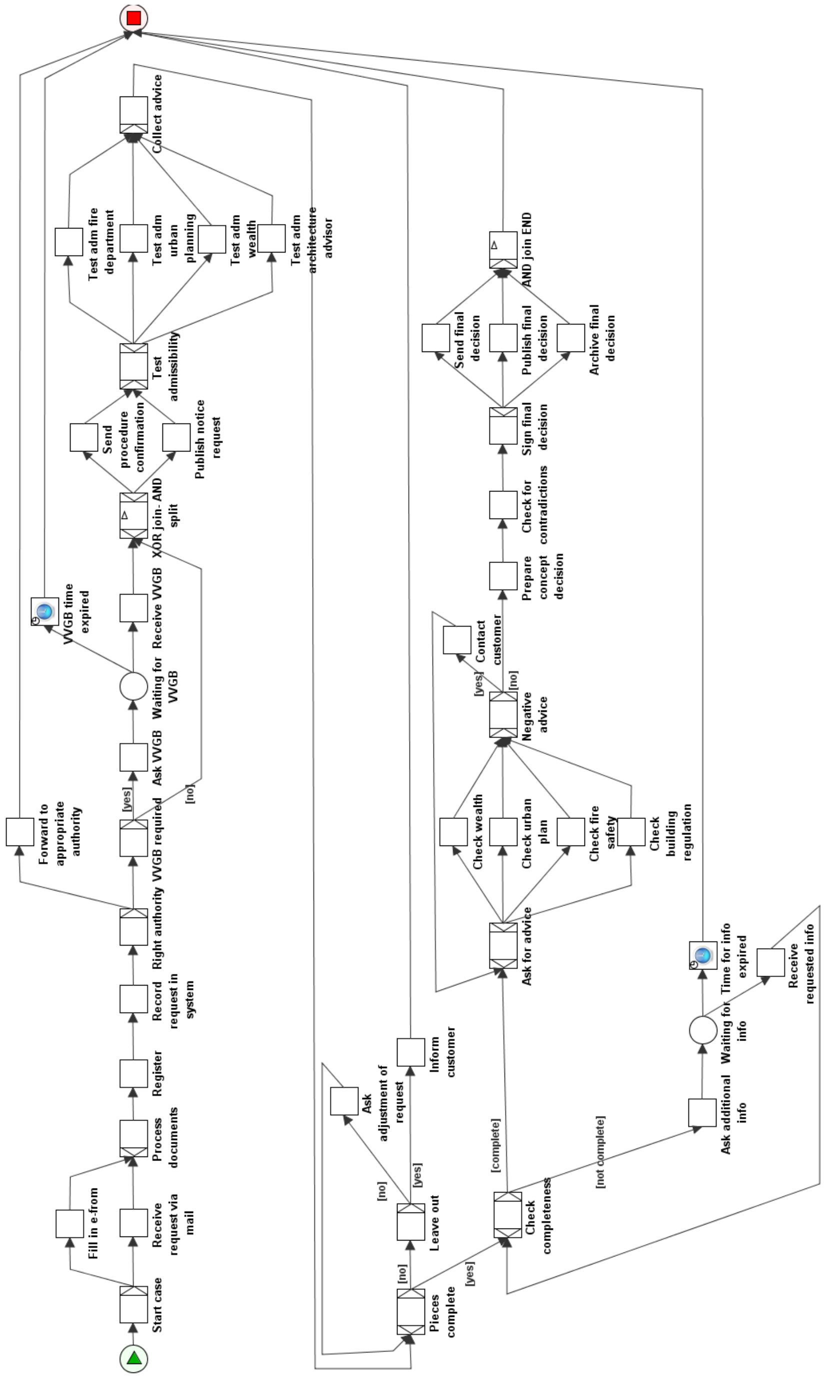

Fig. 60: $WABO_2$ YAWL model for Mun_B

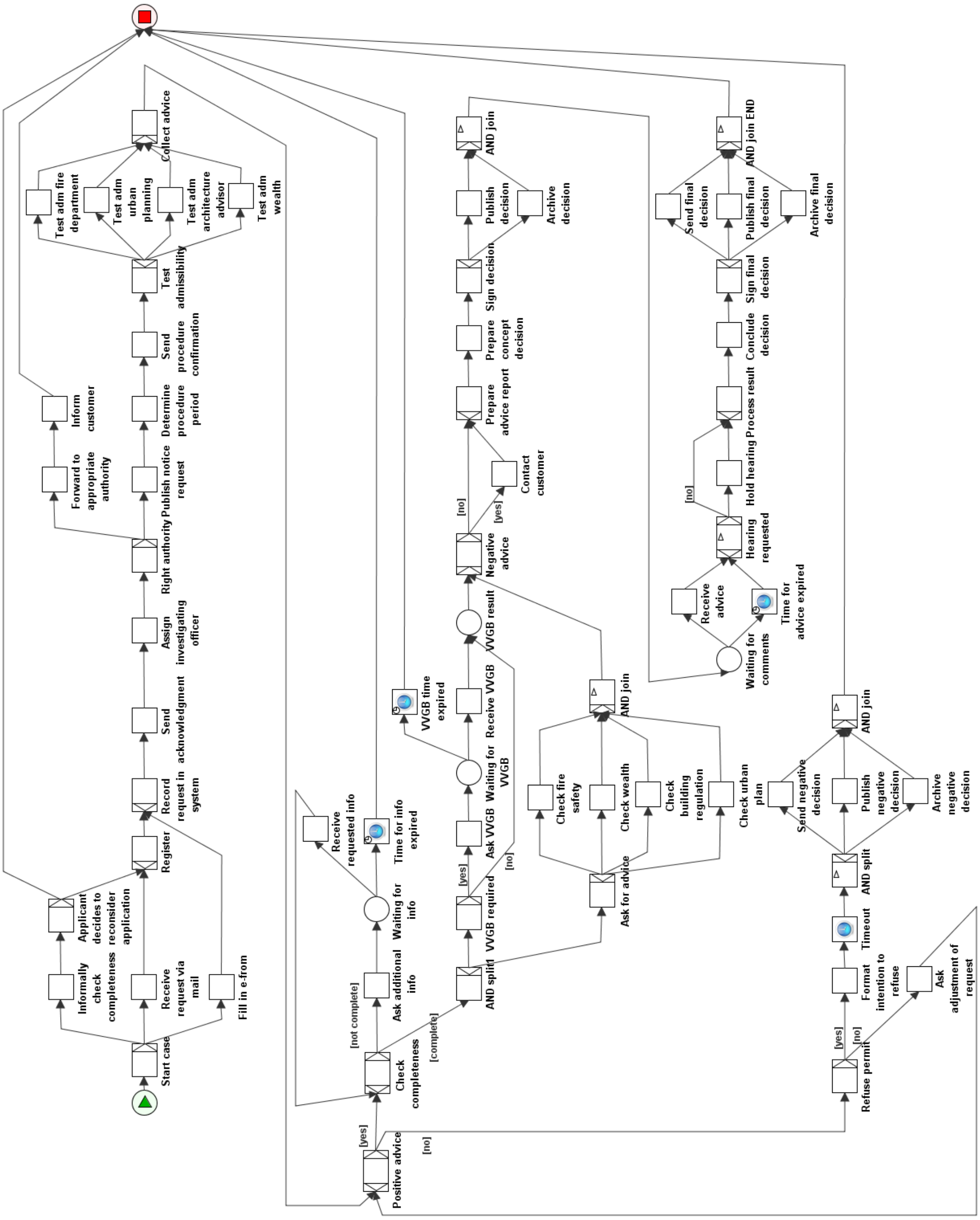

Fig. 61: $WABO_2$ YAWL model for Mun_C

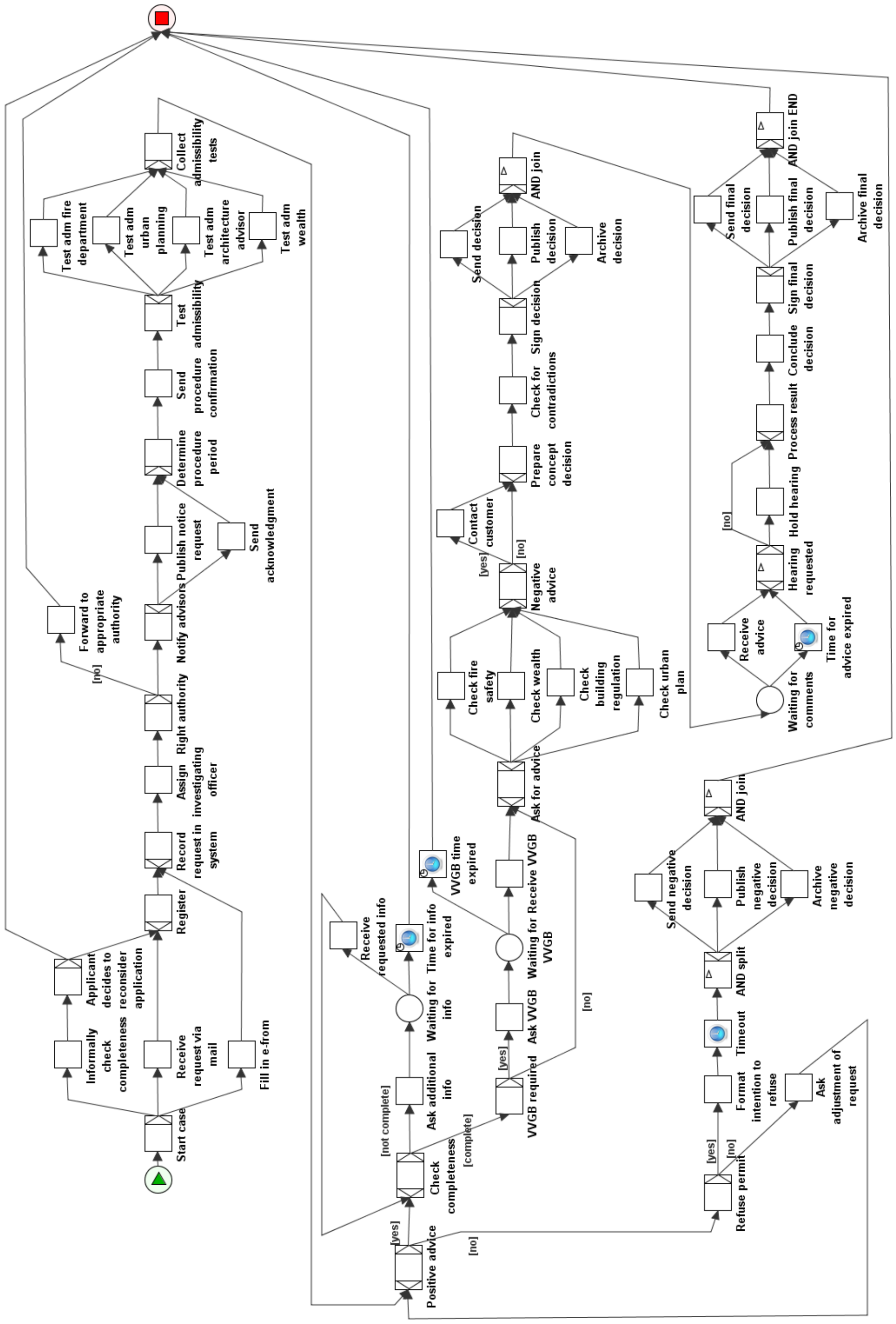

Fig. 62: $WABO_2$ YAWL model for Mun_D

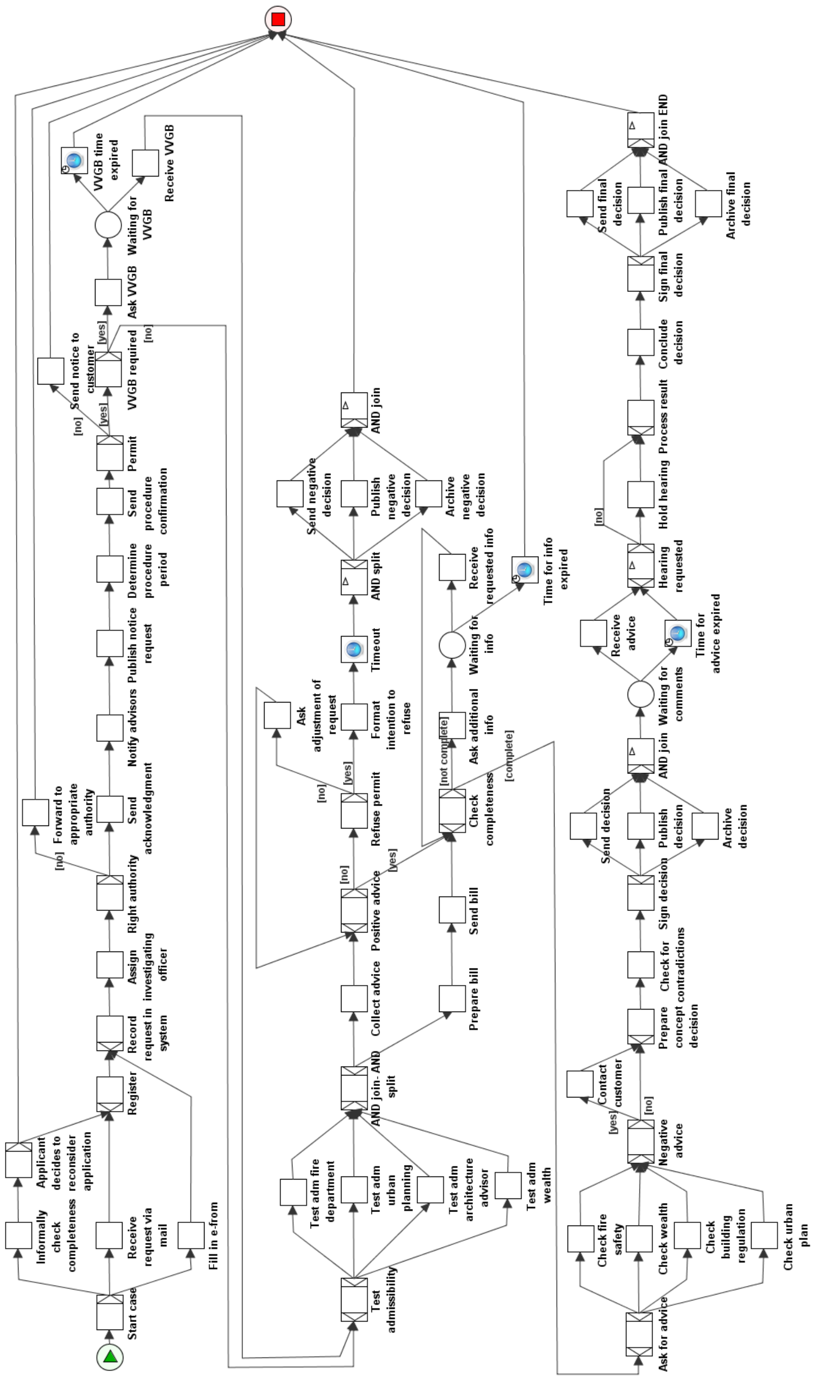

Fig. 63: $WABO_2$ YAWL model for Mun_E

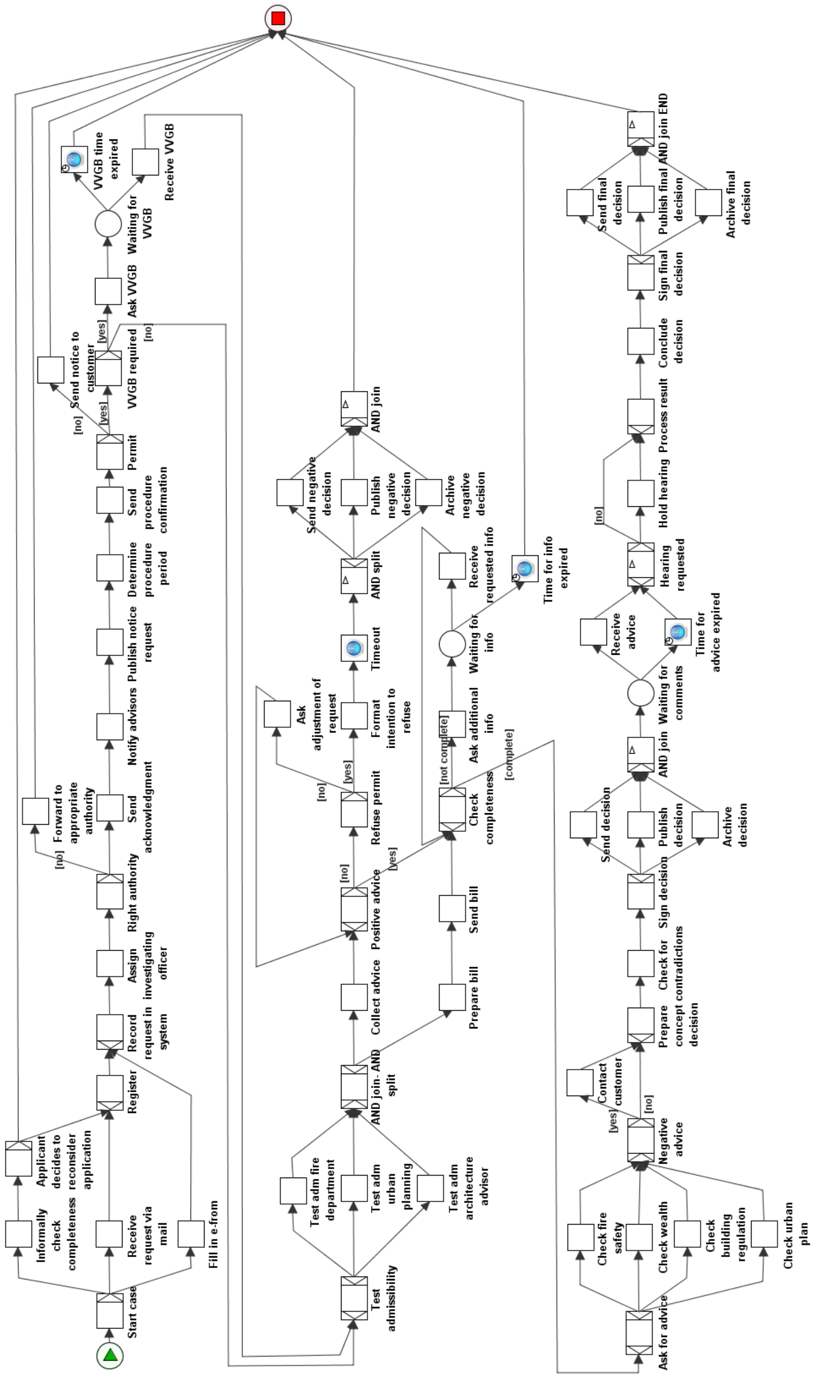

Fig. 64: $WABO_2$ YAWL model for Mun_F

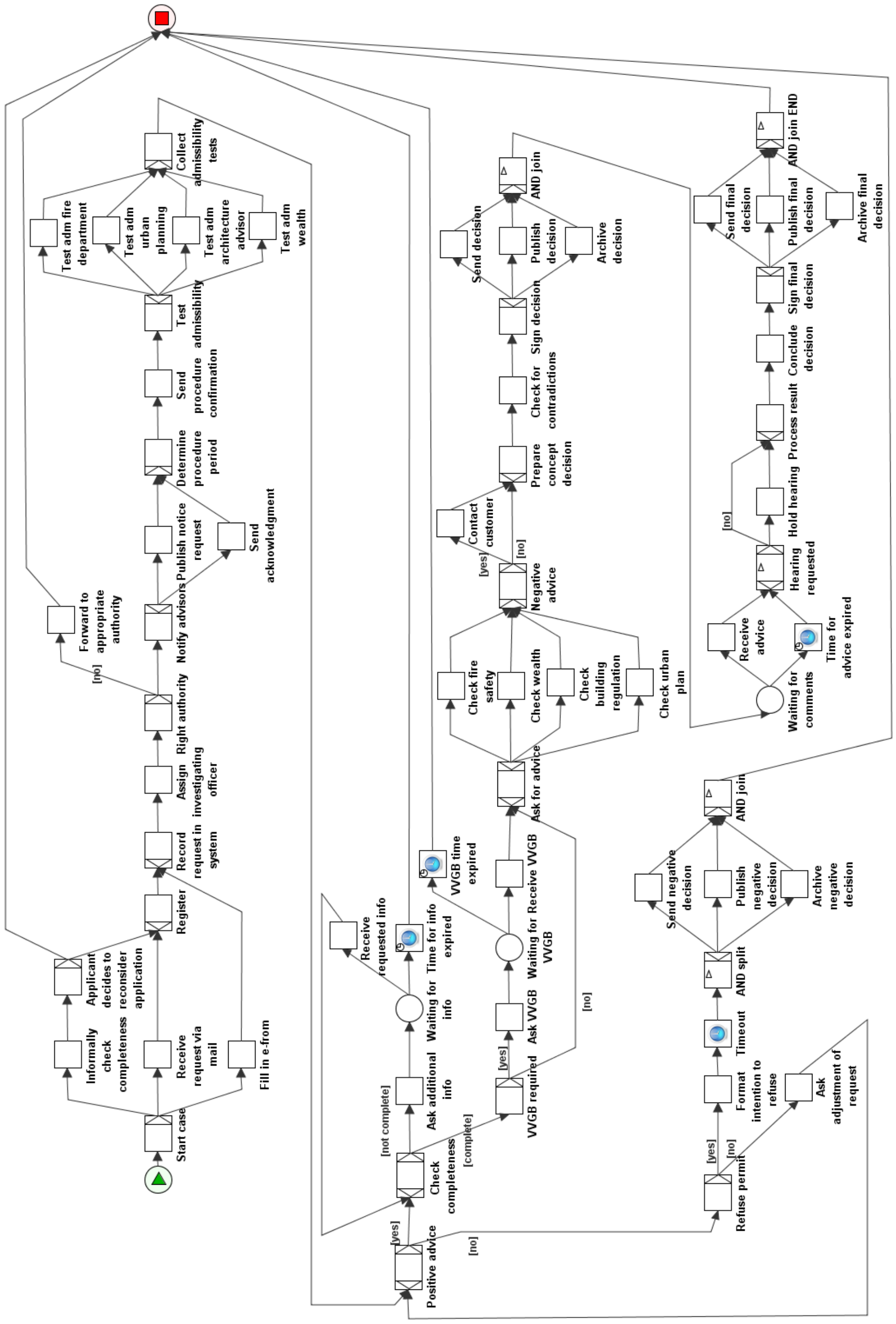

Fig. 65: $WABO_2$ YAWL model for Mun_G

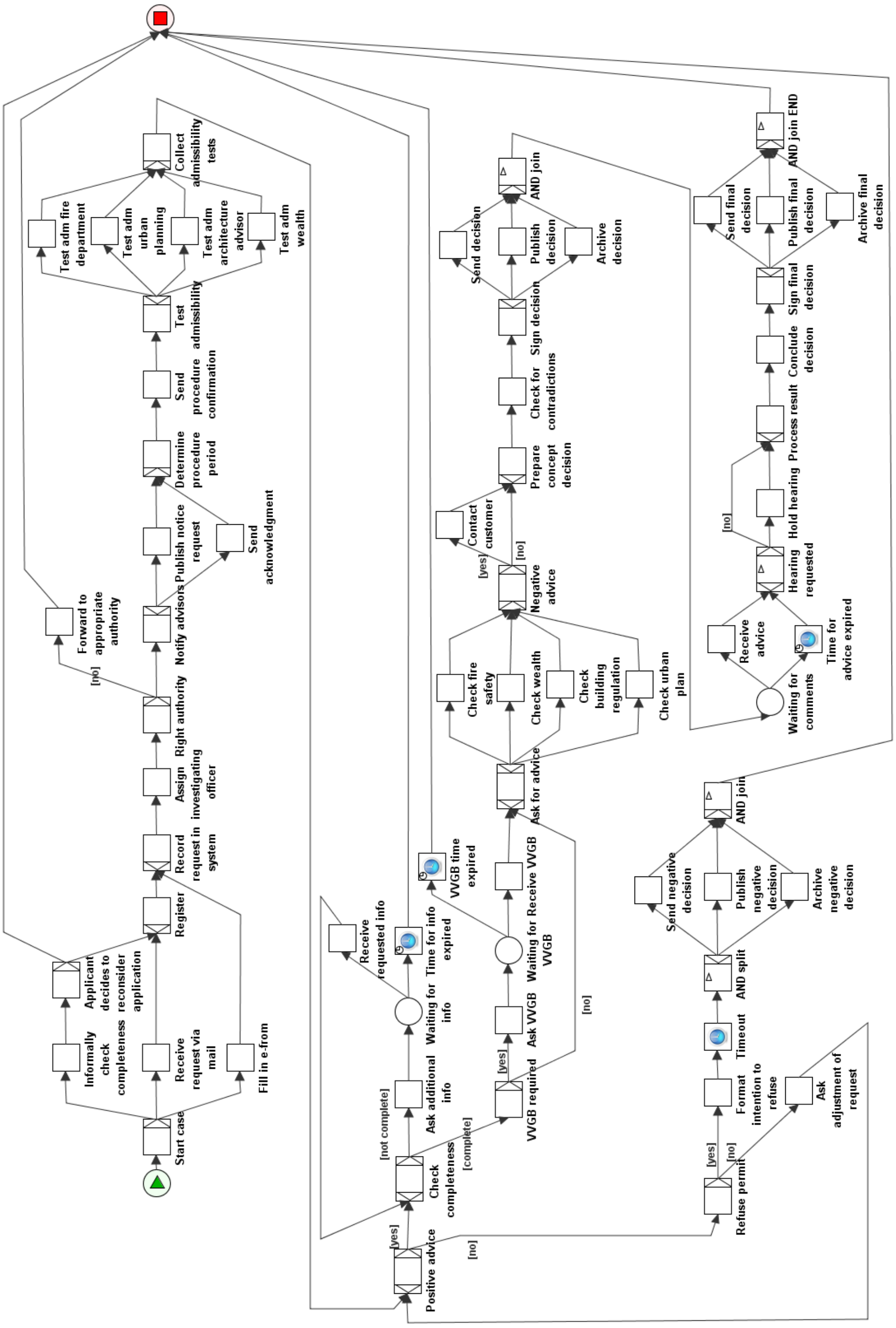

Fig. 66: $WABO_2$ YAWL model for Mun_H

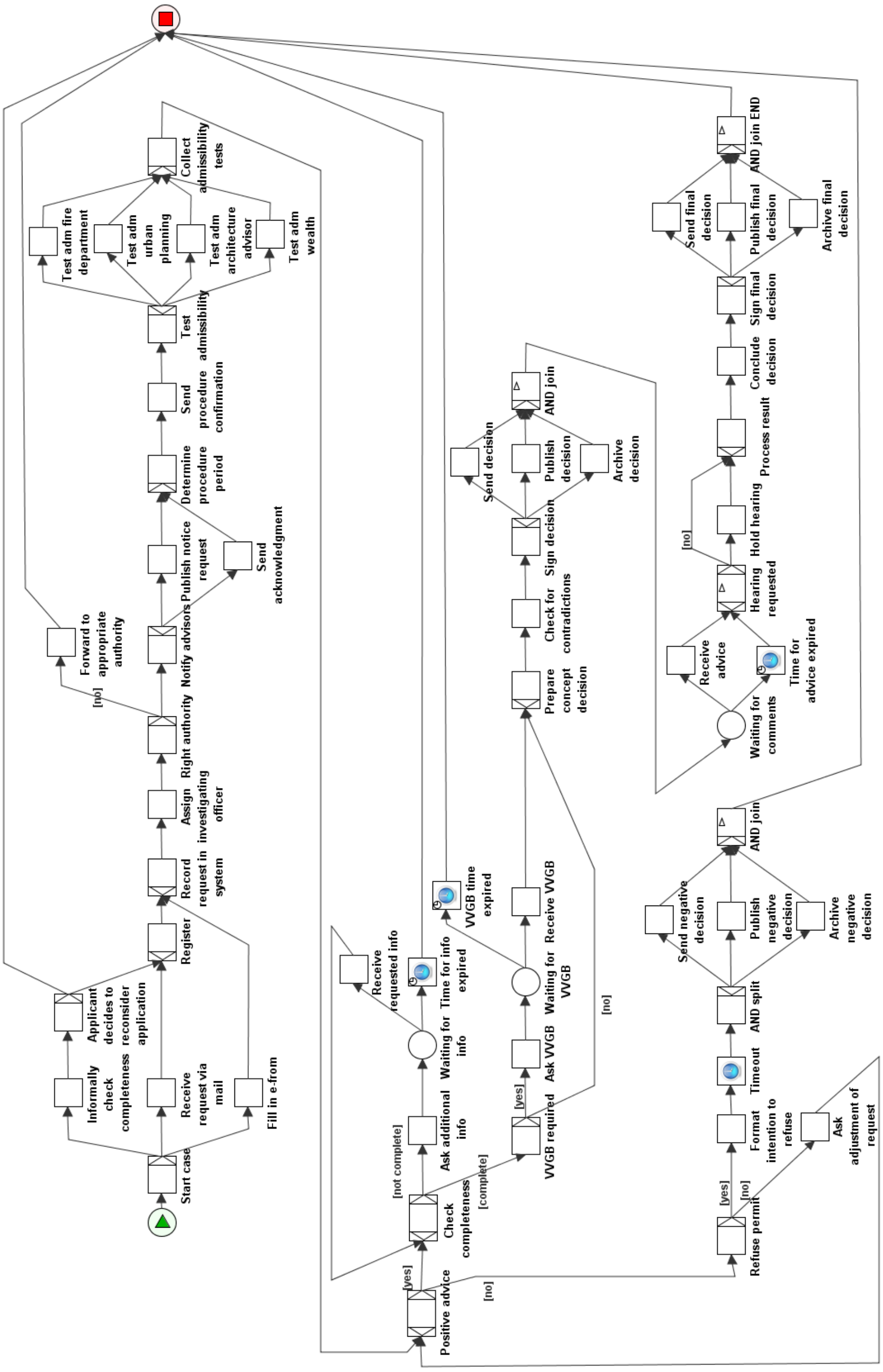

Fig. 67: $WABO_2$ YAWL model for Mun_I

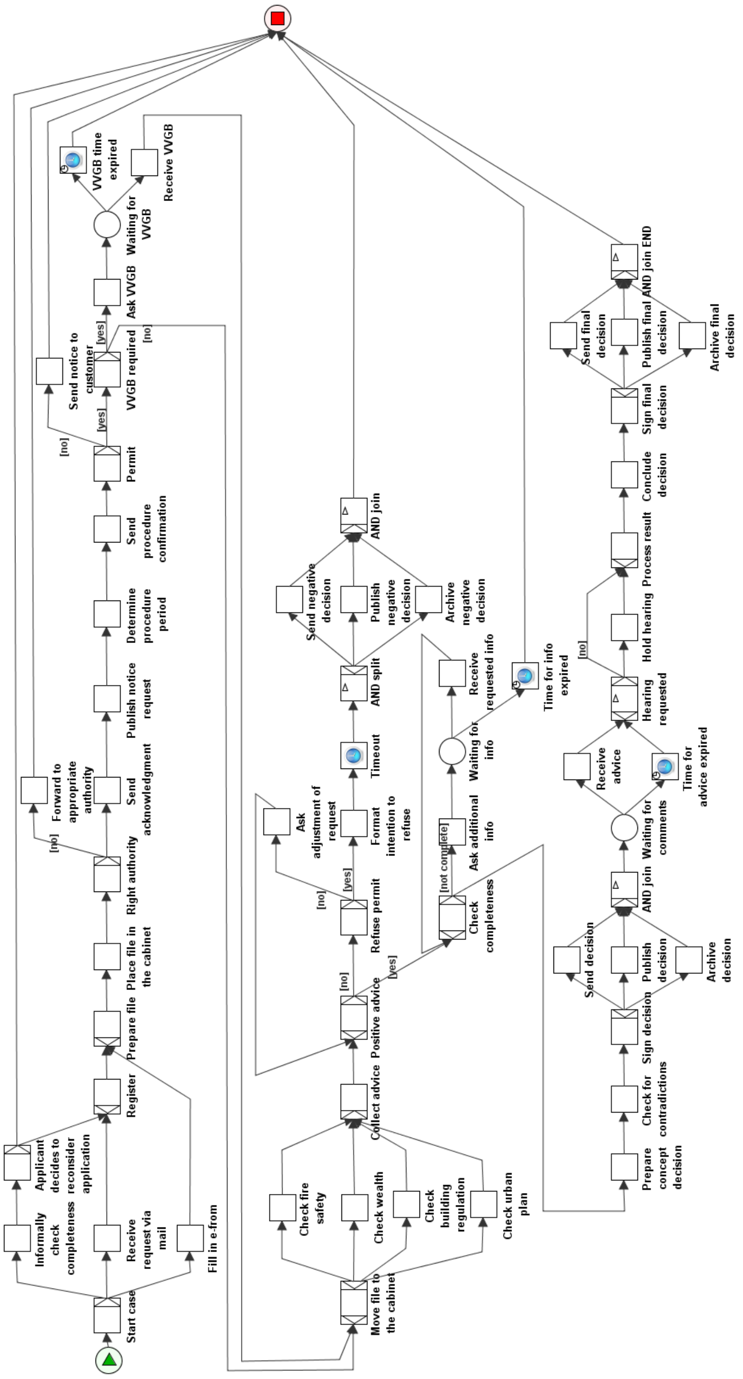

Fig. 68: $WABO_2$ YAWL model for Mun_J

A.7 YAWL models for the WMO process

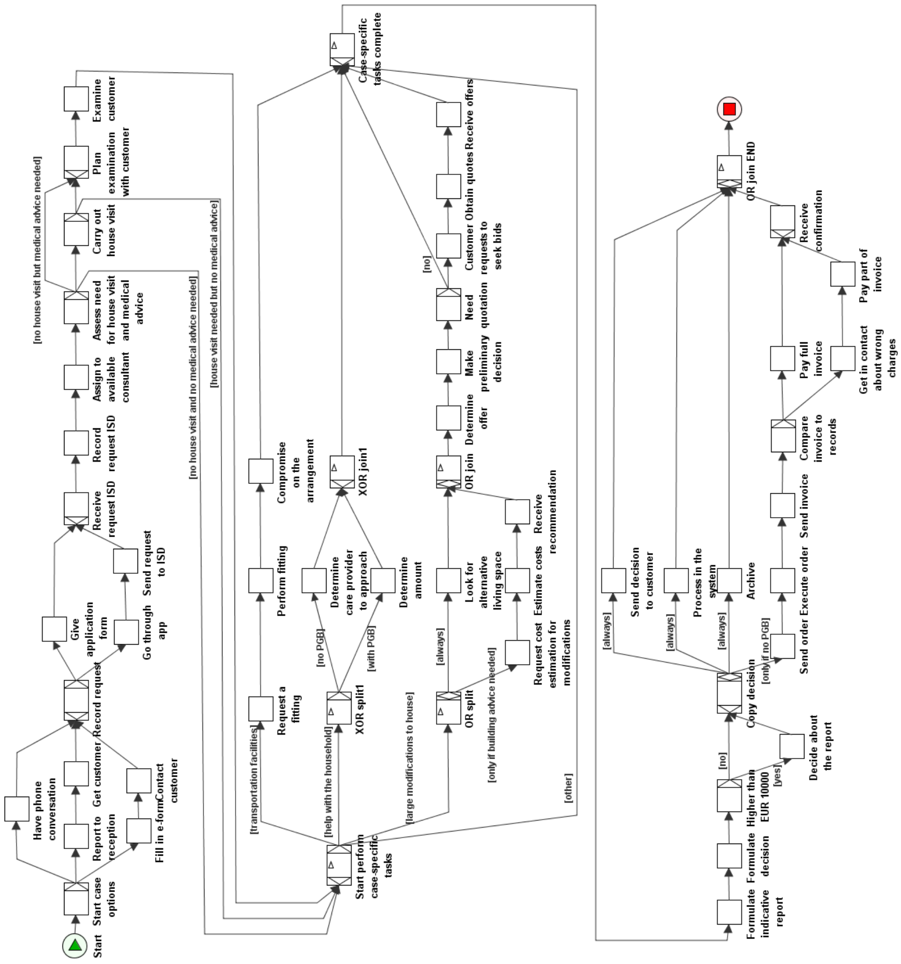

Fig. 69: WMO YAWL model for Mun_A

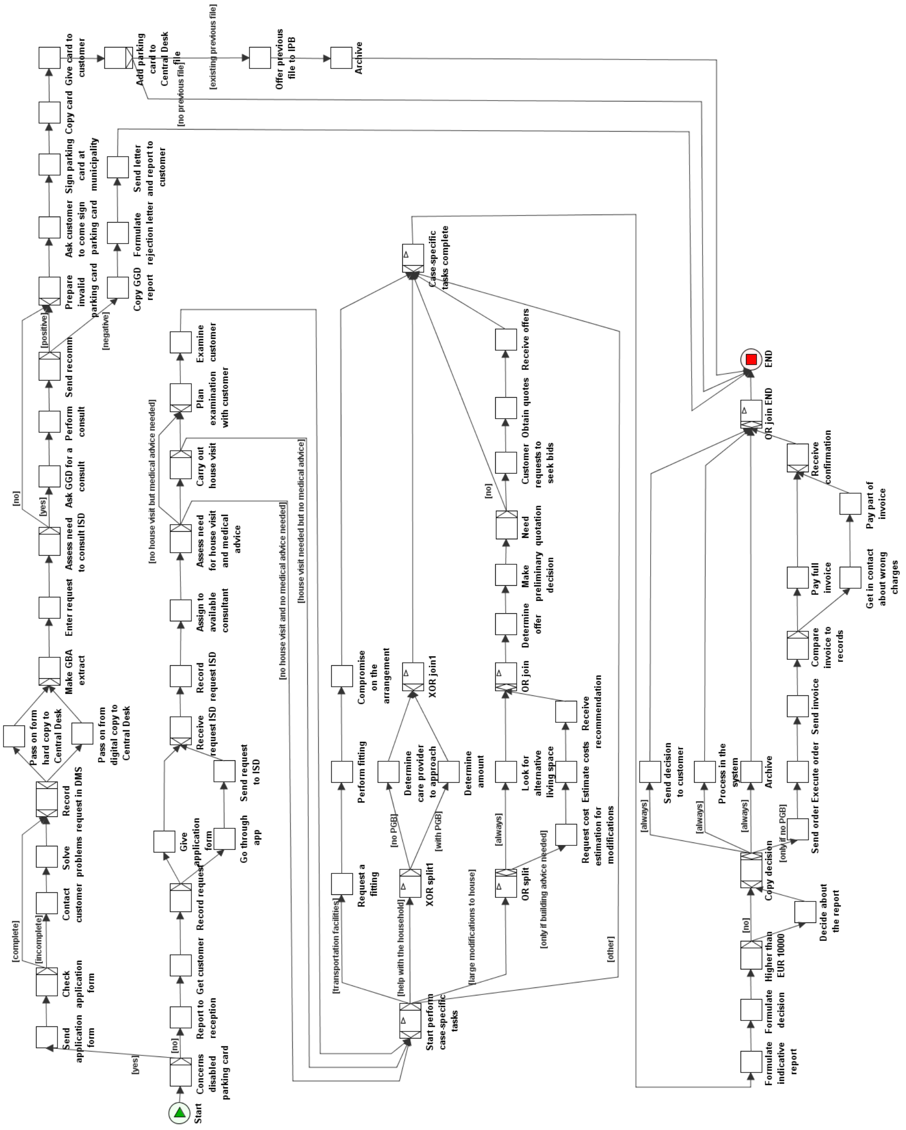

Fig. 70: WMO YAWL model for Mun_B

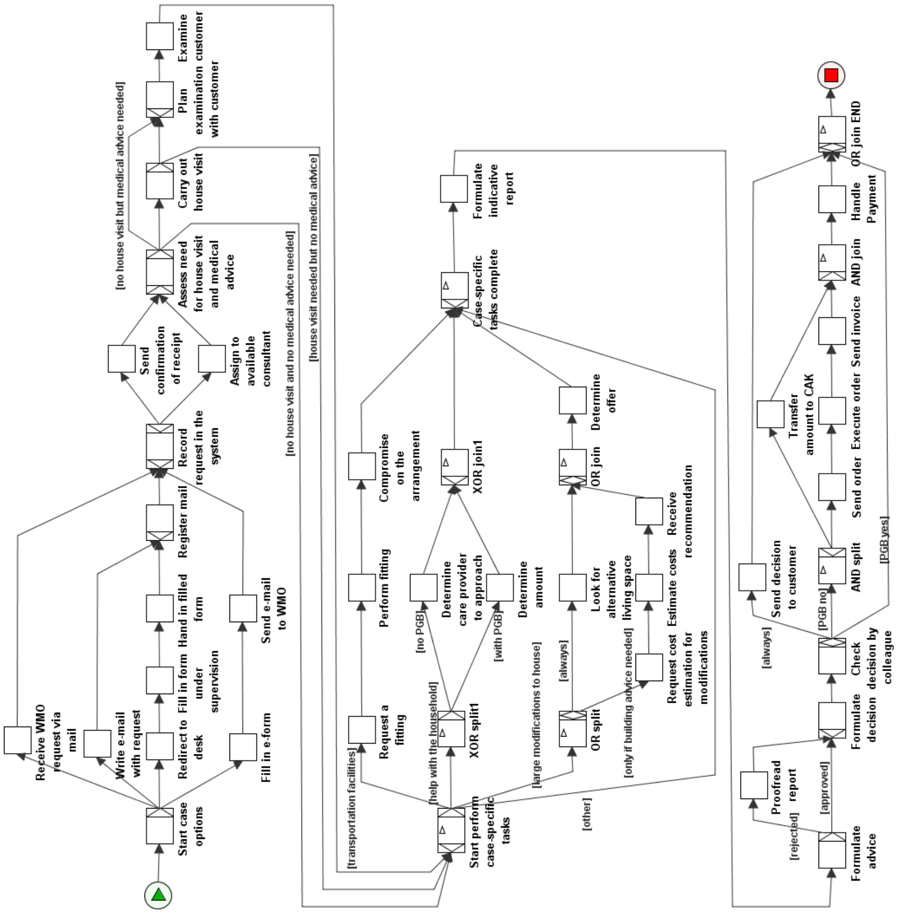

Fig. 71: WMO YAWL model for Mun_C

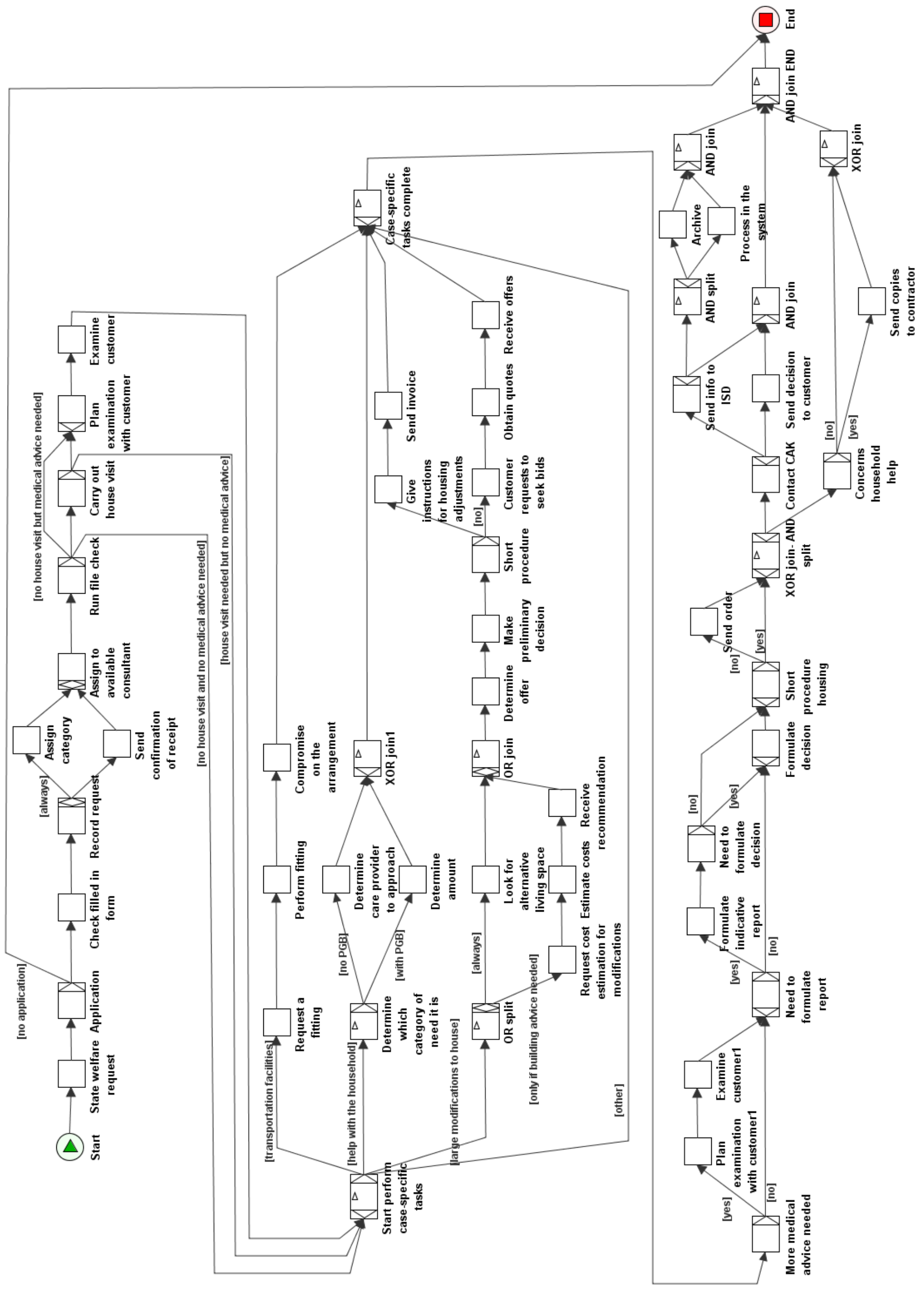

Fig. 72: WMO YAWL model for Mun_D

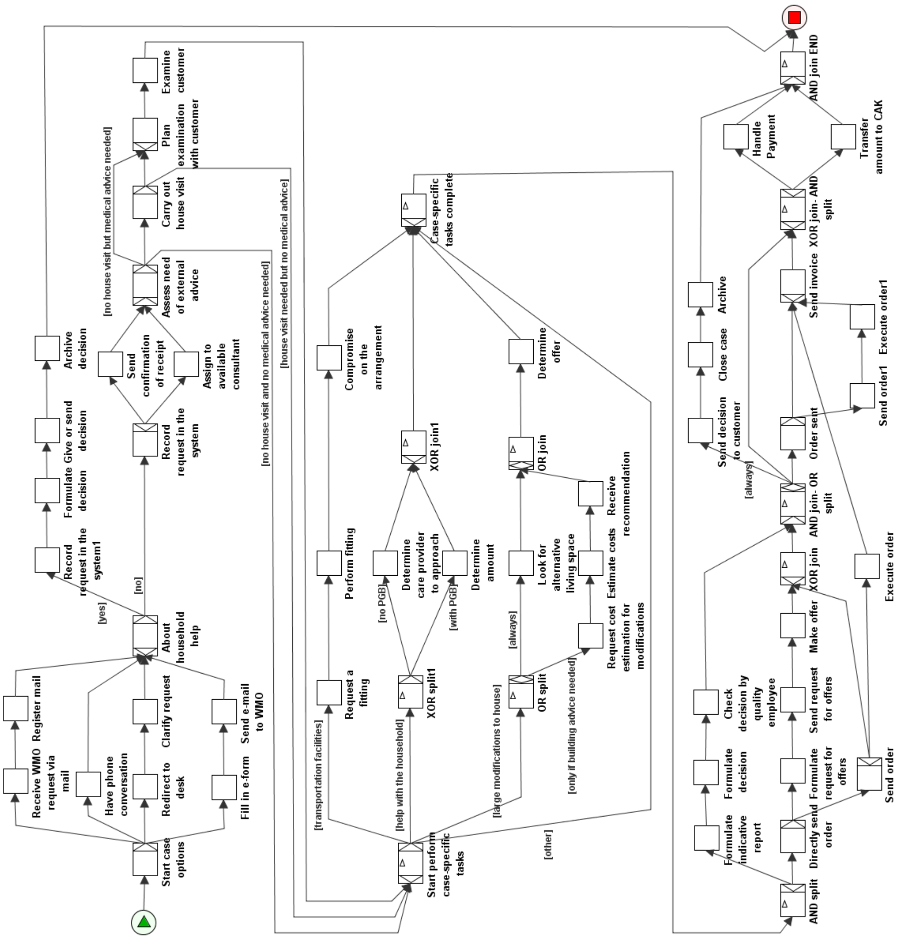

Fig. 73: WMO YAWL model for Mun_E

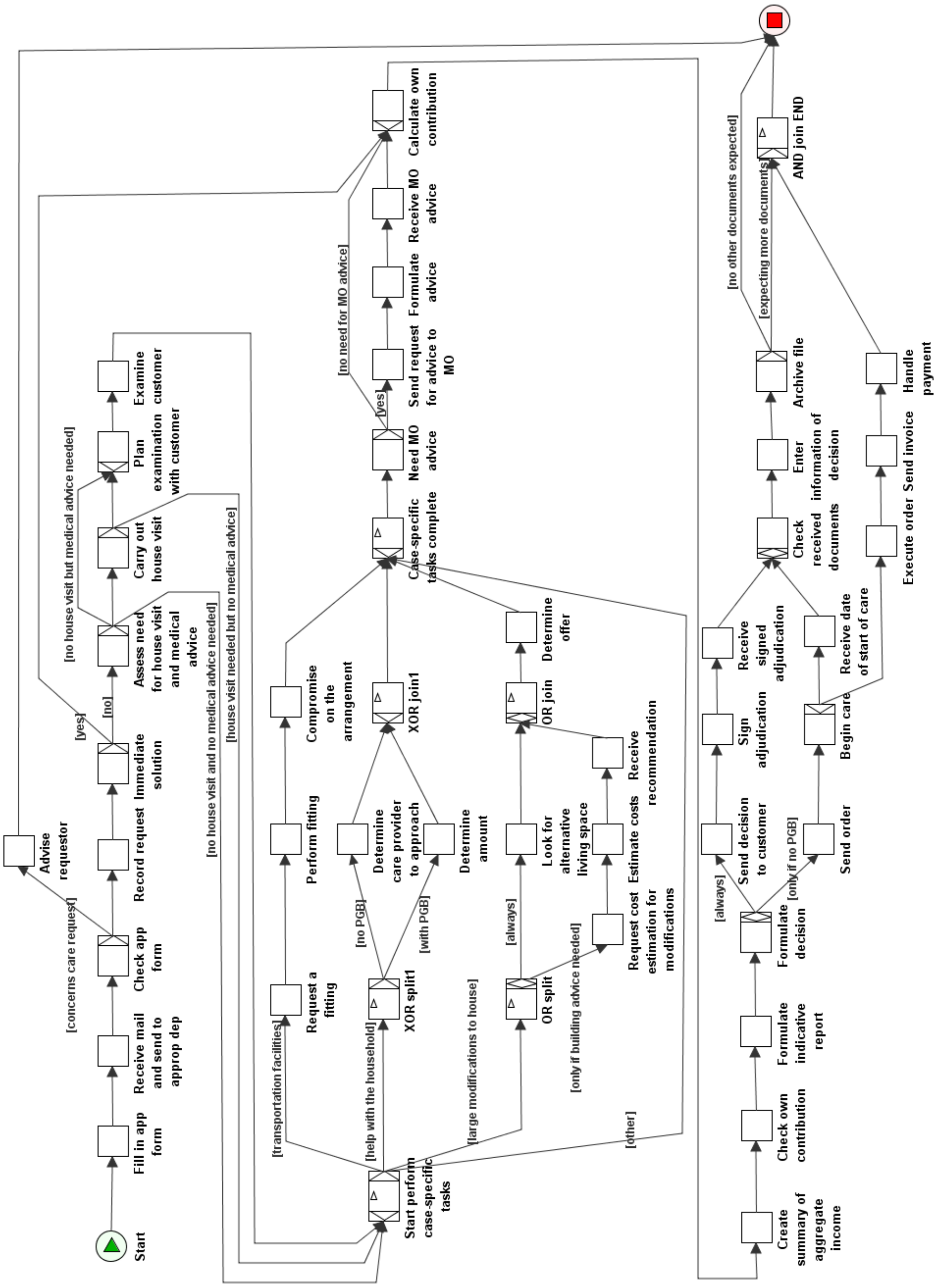

Fig. 74: WMO YAWL model for Mun_F

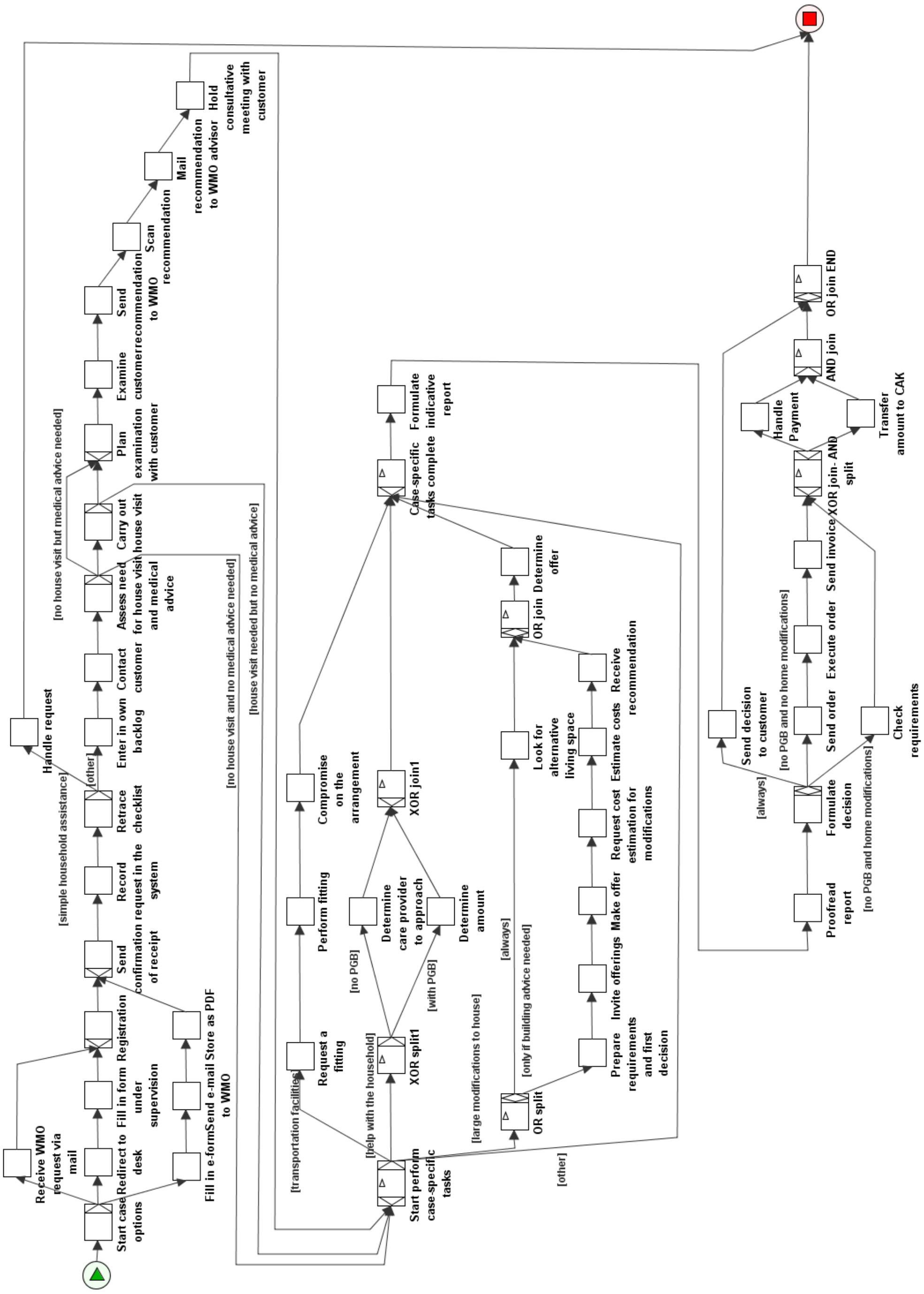

Fig. 75: WMO YAWL model for Mun_G

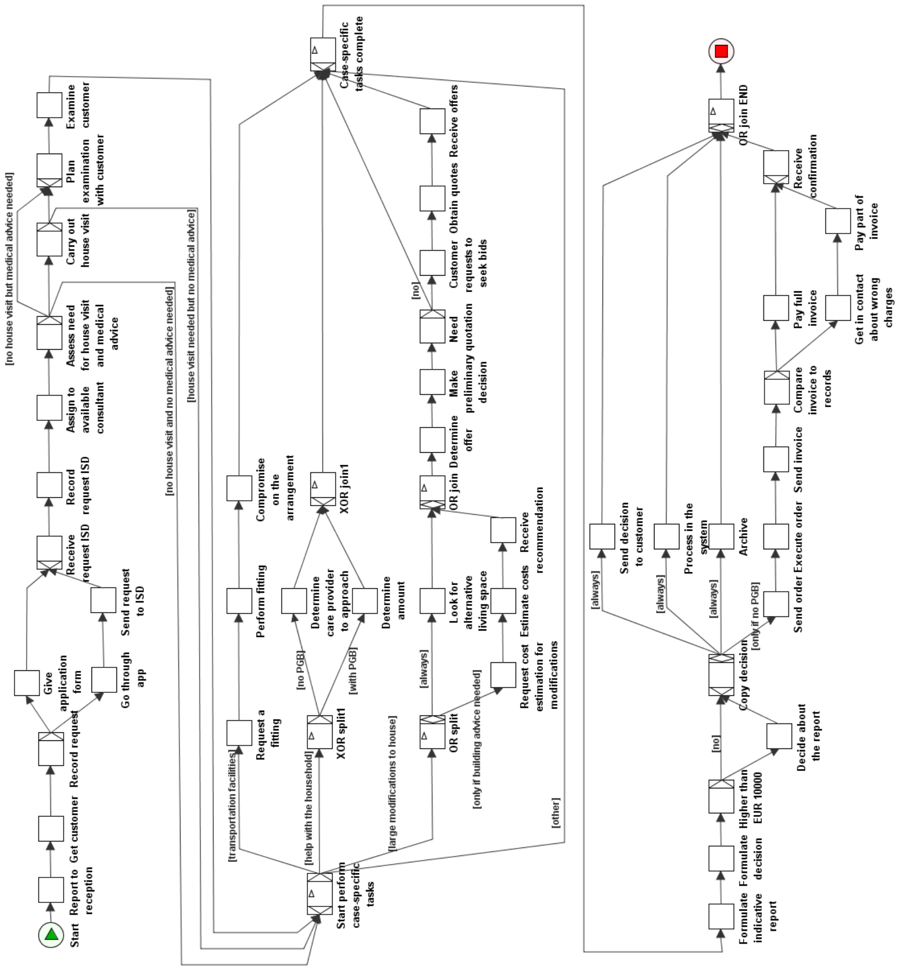

Fig. 76: WMO YAWL model for Mun_H

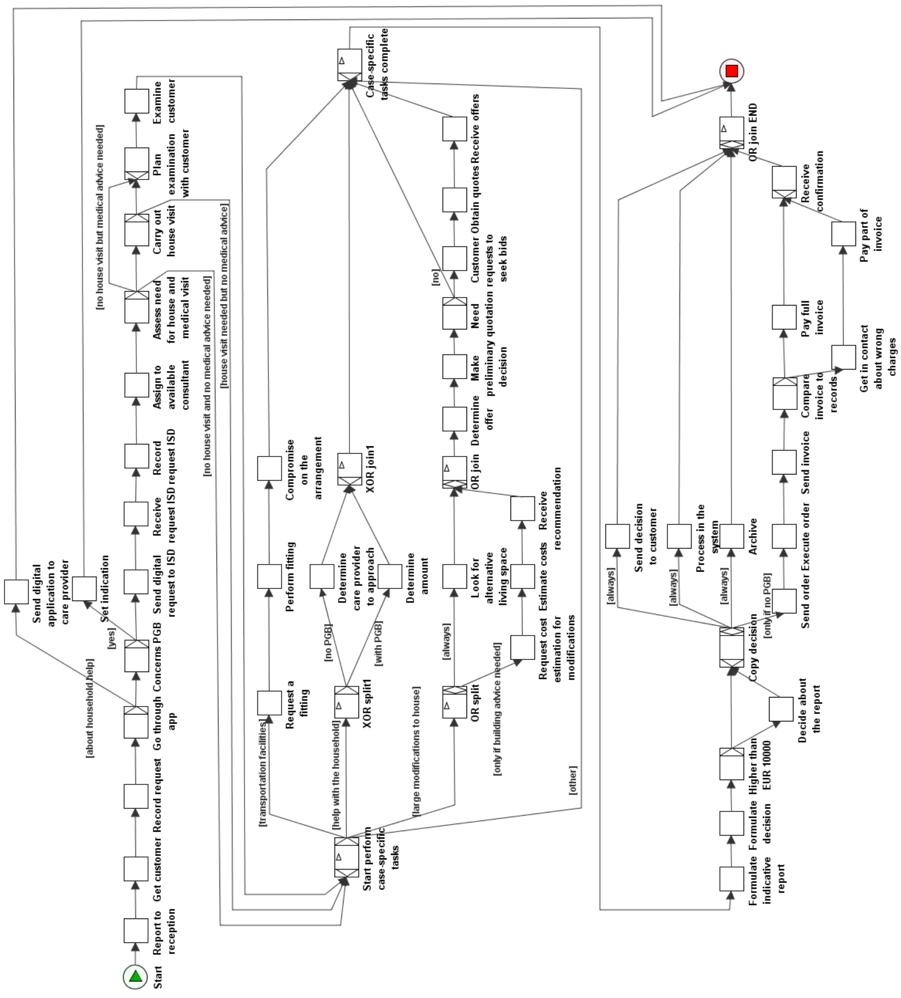

Fig. 77: WMO YAWL model for Mun_I

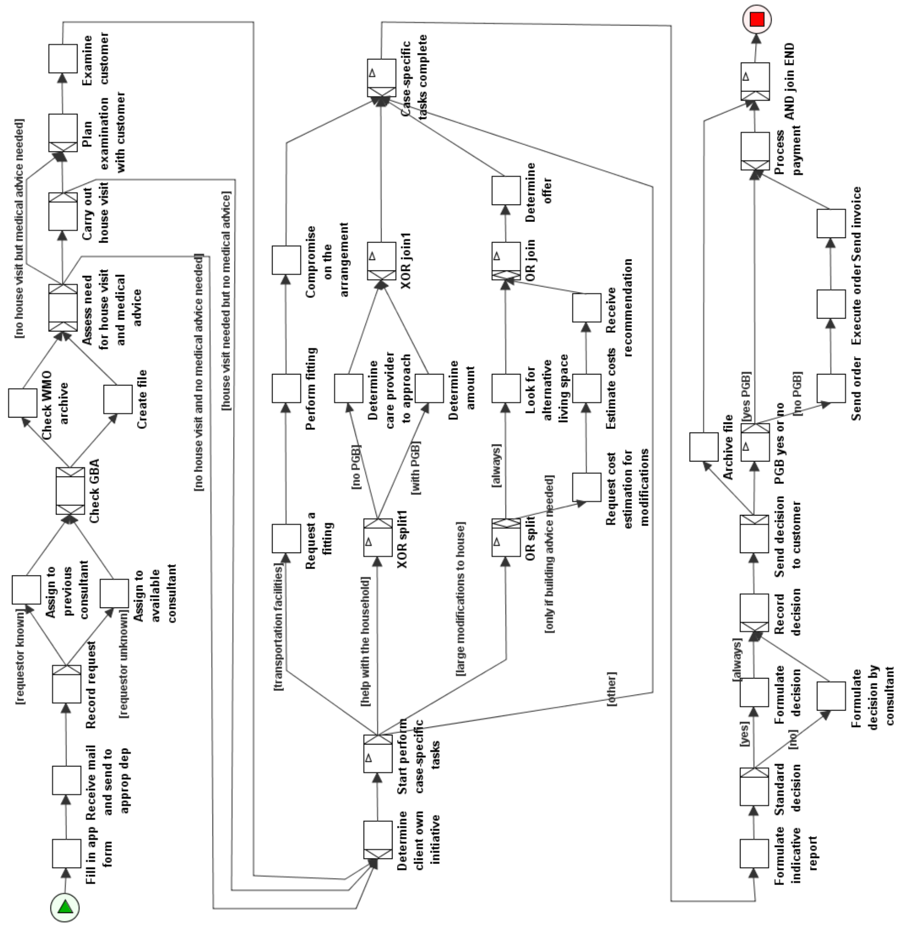

Fig. 78: WMO YAWL model for Mun_J

A.8 YAWL models for the WOZ process

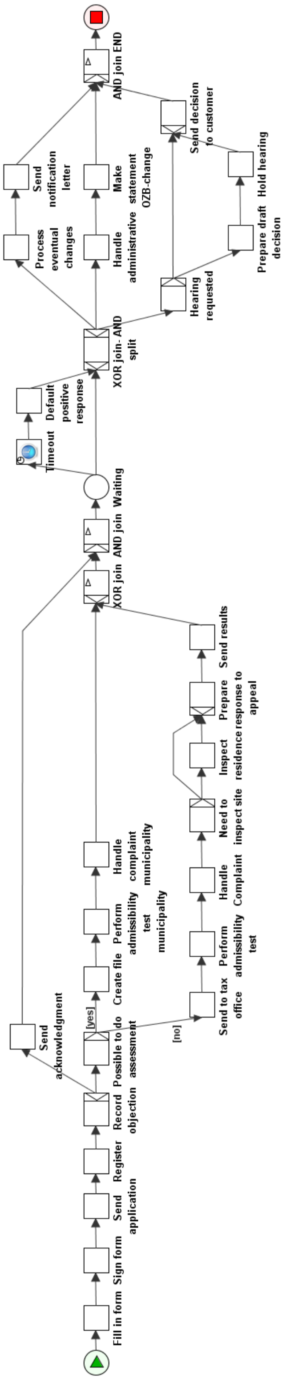

Fig. 79: WOZ YAWL model for Mun_A

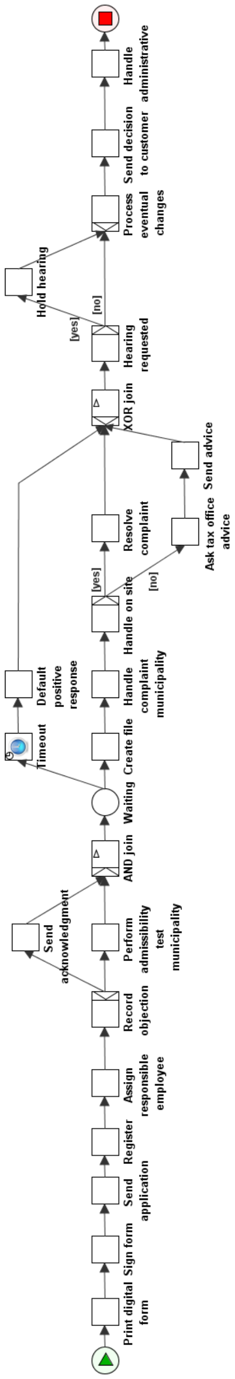

Fig. 80: WOZ YAWL model for Mun_B

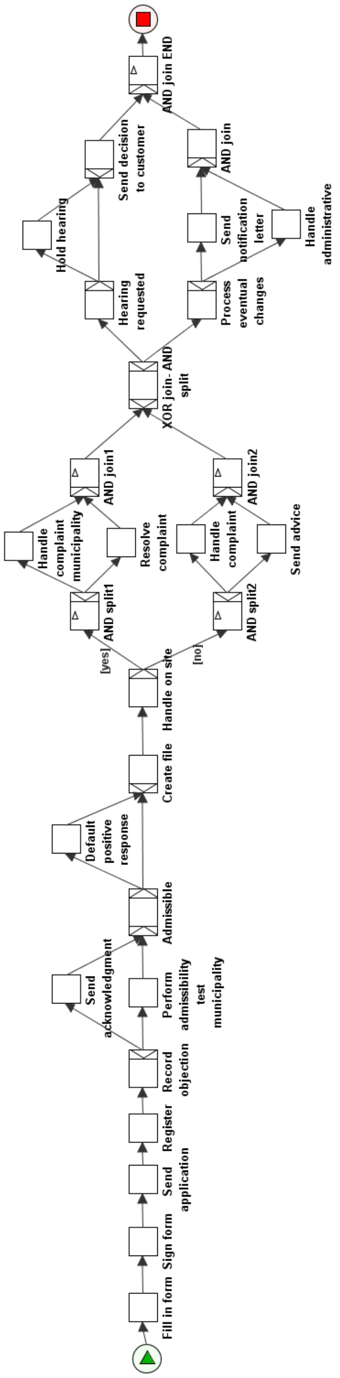

Fig. 81: WOZ YAWL model for Mun_C

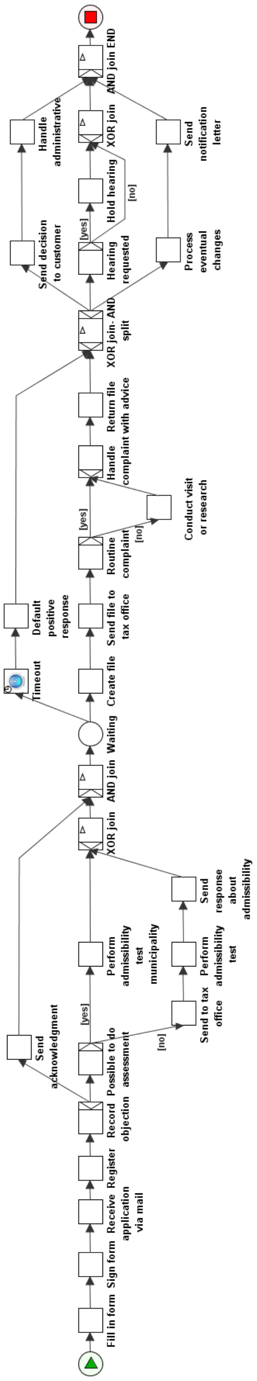

Fig. 82: WOZ YAWL model for Mun_D

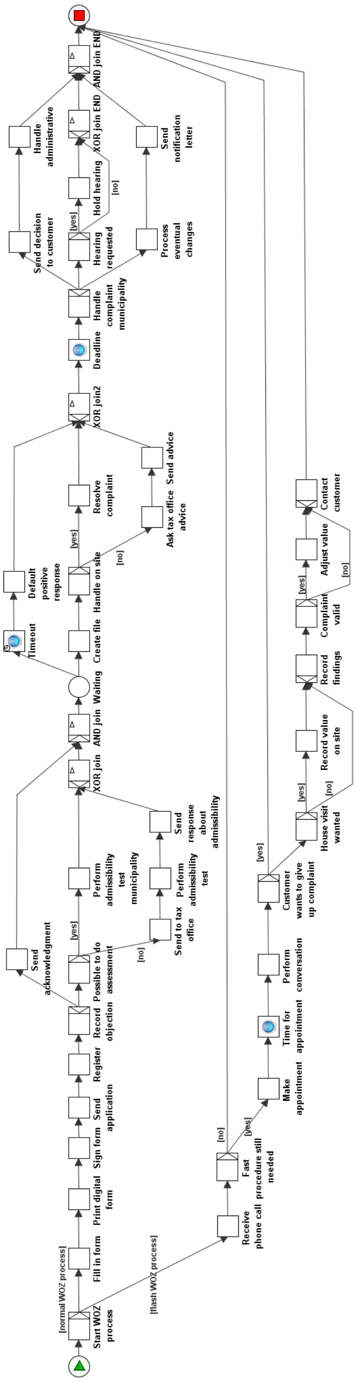

Fig. 83: WOZ YAWL model for Mun_E

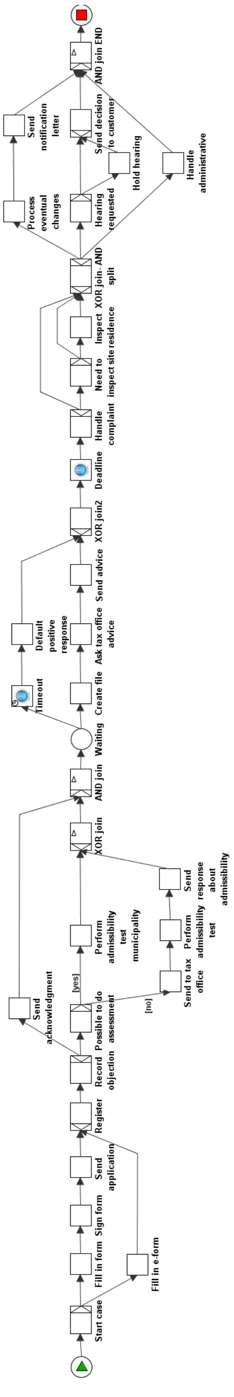

Fig. 84: WOZ YAWL model for Mun_F

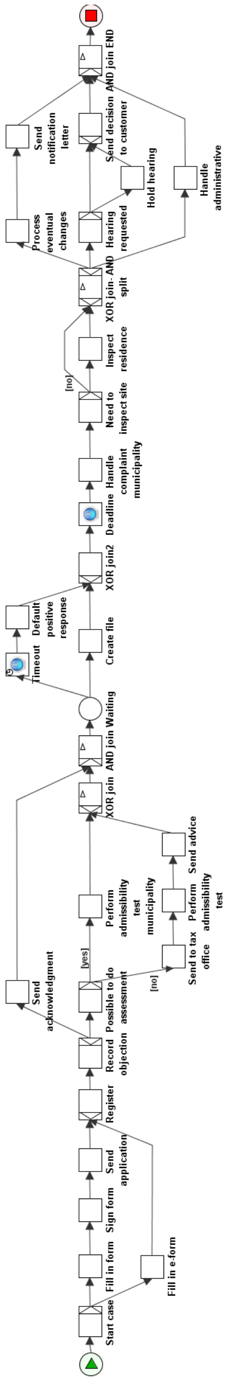

Fig. 85: WOZ YAWL model for Mun_G

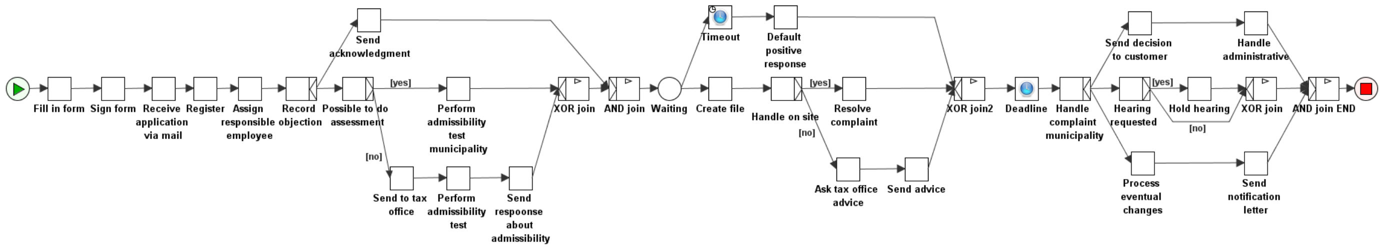

Fig. 86: WOZ YAWL model for Mun_H

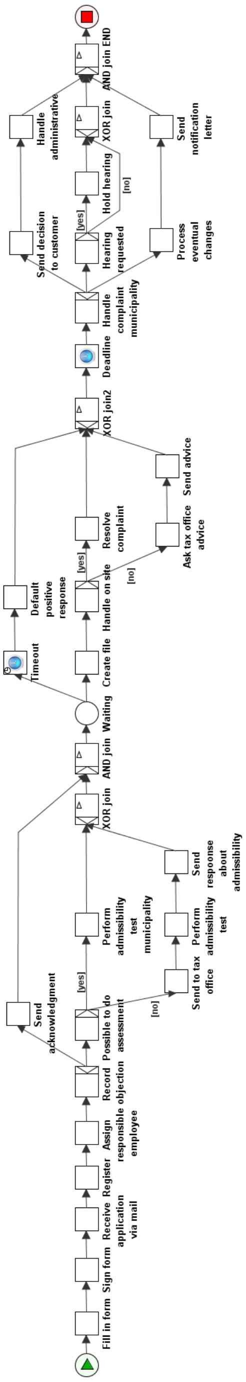

Fig. 87: WOZ YAWL model for Mun_I

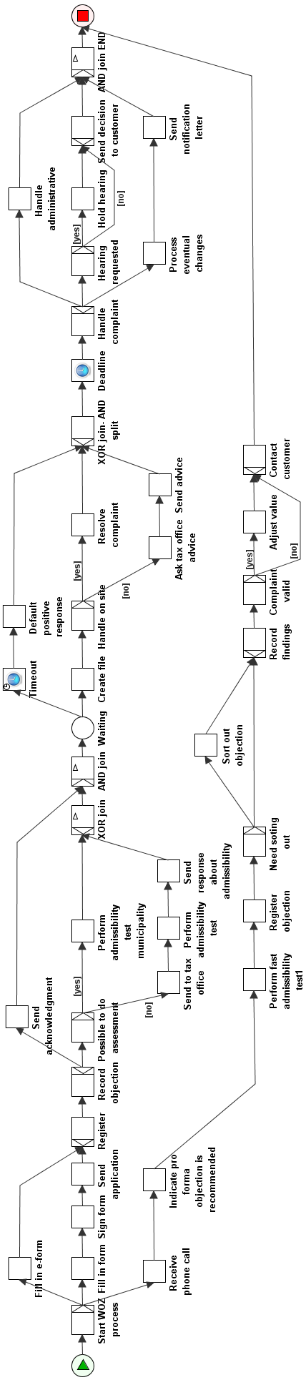

Fig. 88: WOZ YAWL model for Mun_J

B Complexity results

B.1 *GBA1* process

	Mun_A	Mun_B	Mun_C	Mun_D	Mun_E	Mun_F	Mun_G	Mun_H	Mun_I	Mun_J
CFC	6	5	4	5	7	5	5	6	5	3
Density	0.350	0.400	0.667	0.350	0.300	0.350	0.300	0.350	0.350	0.417
CC	0.078	0.205	0.172	0.167	0.108	0.180	0.117	0.078	0.180	0.184

B.2 *GBA2* process

	Mun_A	Mun_B	Mun_C	Mun_D	Mun_E	Mun_F	Mun_G	Mun_H	Mun_I	Mun_J
CFC	11	15	17	20	13	11	11	15	16	15
Density	0.181	0.178	0.128	0.104	0.167	0.214	0.181	0.178	0.144	0.178
CC	0.045	0.037	0.030	0.030	0.039	0.048	0.045	0.037	0.030	0.037

B.3 *GBA3* process

	Mun_A	Mun_B	Mun_C	Mun_D	Mun_E	Mun_F	Mun_G	Mun_H	Mun_I	Mun_J
CFC	9	10	16	10	12	8	8	9	8	8
Density	0.155	0.181	0.126	0.155	0.136	0.214	0.194	0.181	0.194	0.167
CC	0.079	0.075	0.054	0.080	0.067	0.120	0.113	0.074	0.113	0.109

B.4 *MOR* process

	Mun_A	Mun_B	Mun_C	Mun_D	Mun_E	Mun_F	Mun_G	Mun_H	Mun_I	Mun_J
CFC	21	16	17	10	15	19	13	11	17	15
Density	0.148	0.141	0.121	0.232	0.164	0.147	0.178	0.155	0.141	0.164
CC	0.027	0.032	0.027	0.051	0.032	0.028	0.036	0.047	0.032	0.032

B.5 *WABO1* process

	Mun_A	Mun_B	Mun_C	Mun_D	Mun_E	Mun_F	Mun_G	Mun_H	Mun_I	Mun_J
CFC	3	3	7	5	5	4	5	5	5	5
Density	0.417	0.417	0.196	0.267	0.267	0.417	0.267	0.267	0.267	0.267
CC	0.160	0.271	0.076	0.110	0.094	0.100	0.094	0.094	0.094	0.094

B.6 *WABO2* process

	Mun_A	Mun_B	Mun_C	Mun_D	Mun_E	Mun_F	Mun_G	Mun_H	Mun_I	Mun_J
CFC	29	22	31	31	33	33	31	31	28	29
Density	0.073	0.079	0.054	0.055	0.056	0.055	0.055	0.055	0.062	0.065
CC	0.036	0.043	0.033	0.034	0.029	0.029	0.034	0.034	0.039	0.030

B.7 *WMO* process

	Mun_A	Mun_B	Mun_C	Mun_D	Mun_E	Mun_F	Mun_G	Mun_H	Mun_I	Mun_J
CFC	40	48	29	35	35	26	27	37	39	22
Density	0.088	0.060	0.086	0.051	0.066	0.087	0.087	0.096	0.092	0.084
CC	0.018	0.018	0.026	0.025	0.022	0.031	0.025	0.021	0.021	0.029

B.8 *WOZ* process

	Mun_A	Mun_B	Mun_C	Mun_D	Mun_E	Mun_F	Mun_G	Mun_H	Mun_I	Mun_J
CFC	10	7	11	10	20	13	12	10	10	17
Density	0.136	0.238	0.096	0.136	0.088	0.110	0.115	0.155	0.155	0.092
CC	0.067	0.103	0.082	0.064	0.042	0.045	0.046	0.075	0.075	0.037

C Similarity results

C.1 GBA_1 process

GED	Mun_A	Mun_B	Mun_C	Mun_D	Mun_E	Mun_F	Mun_G	Mun_H	Mun_I	Mun_J
Mun_A	1.000	0.837	0.817	0.883	0.845	0.803	0.667	1.000	0.942	0.698
Mun_B	0.837	1.000	0.772	0.915	0.841	0.842	0.708	0.837	0.896	0.769
Mun_C	0.817	0.772	1.000	0.807	0.799	0.798	0.665	0.817	0.798	0.664
Mun_D	0.883	0.915	0.807	1.000	0.884	0.891	0.719	0.883	0.950	0.801
Mun_E	0.845	0.841	0.799	0.884	1.000	0.851	0.732	0.845	0.908	0.858
Mun_F	0.803	0.842	0.798	0.891	0.851	1.000	0.711	0.803	0.879	0.793
Mun_G	0.667	0.708	0.665	0.719	0.732	0.711	1.000	0.667	0.717	0.723
Mun_H	1.000	0.837	0.817	0.883	0.845	0.803	0.667	1.000	0.942	0.698
Mun_I	0.942	0.896	0.798	0.950	0.908	0.879	0.717	0.942	1.000	0.793
Mun_J	0.698	0.769	0.664	0.801	0.858	0.793	0.723	0.698	0.793	1.000

SPS	Mun_A	Mun_B	Mun_C	Mun_D	Mun_E	Mun_F	Mun_G	Mun_H	Mun_I	Mun_J
Mun_A	1.000	0.573	0.813	0.741	0.622	0.649	0.250	1.000	0.788	0.289
Mun_B	0.573	1.000	0.615	0.760	0.774	0.704	0.391	0.573	0.781	0.365
Mun_C	0.813	0.615	1.000	0.768	0.600	0.739	0.200	0.813	0.735	0.304
Mun_D	0.741	0.760	0.768	1.000	0.751	0.891	0.466	0.741	0.929	0.491
Mun_E	0.622	0.774	0.600	0.751	1.000	0.757	0.372	0.622	0.802	0.538
Mun_F	0.649	0.704	0.739	0.891	0.757	1.000	0.364	0.649	0.917	0.483
Mun_G	0.250	0.391	0.200	0.466	0.372	0.364	1.000	0.250	0.359	0.525
Mun_H	1.000	0.573	0.813	0.741	0.622	0.649	0.250	1.000	0.788	0.289
Mun_I	0.788	0.781	0.735	0.929	0.802	0.917	0.359	0.788	1.000	0.459
Mun_J	0.289	0.365	0.304	0.491	0.538	0.483	0.525	0.289	0.459	1.000

C.2 GBA_2 process

GED	Mun_A	Mun_B	Mun_C	Mun_D	Mun_E	Mun_F	Mun_G	Mun_H	Mun_I	Mun_J
Mun_A	1.000	0.898	0.867	0.811	0.957	0.962	0.980	0.898	0.891	0.898
Mun_B	0.898	1.000	0.919	0.897	0.944	0.932	0.894	1.000	0.911	1.000
Mun_C	0.867	0.919	1.000	0.840	0.898	0.863	0.867	0.919	0.845	0.919
Mun_D	0.811	0.897	0.840	1.000	0.851	0.838	0.806	0.897	0.827	0.897
Mun_E	0.957	0.944	0.898	0.851	1.000	0.938	0.937	0.944	0.924	0.944
Mun_F	0.962	0.932	0.863	0.838	0.938	1.000	0.941	0.932	0.901	0.932
Mun_G	0.980	0.894	0.867	0.806	0.937	0.941	1.000	0.894	0.890	0.894
Mun_H	0.898	1.000	0.919	0.897	0.944	0.932	0.894	1.000	0.911	1.000
Mun_I	0.891	0.911	0.845	0.827	0.924	0.901	0.890	0.911	1.000	0.911
Mun_J	0.898	1.000	0.919	0.897	0.944	0.932	0.894	1.000	0.911	1.000

SPS	Mun_A	Mun_B	Mun_C	Mun_D	Mun_E	Mun_F	Mun_G	Mun_H	Mun_I	Mun_J
Mun_A	1.000	0.756	0.526	0.472	0.889	0.942	0.970	0.756	0.710	0.756
Mun_B	0.756	1.000	0.747	0.628	0.858	0.784	0.736	1.000	0.792	1.000
Mun_C	0.526	0.747	1.000	0.523	0.628	0.475	0.526	0.747	0.557	0.747
Mun_D	0.472	0.628	0.523	1.000	0.540	0.494	0.463	0.628	0.488	0.628
Mun_E	0.889	0.858	0.628	0.540	1.000	0.837	0.863	0.858	0.830	0.858
Mun_F	0.942	0.784	0.475	0.494	0.837	1.000	0.912	0.784	0.713	0.784
Mun_G	0.970	0.736	0.526	0.463	0.863	0.912	1.000	0.736	0.691	0.736
Mun_H	0.756	1.000	0.747	0.628	0.858	0.784	0.736	1.000	0.792	1.000
Mun_I	0.710	0.792	0.557	0.488	0.830	0.713	0.691	0.792	1.000	0.792
Mun_J	0.756	1.000	0.747	0.628	0.858	0.784	0.736	1.000	0.792	1.000

C.3 GBA_3 process

GED	Mun_A	Mun_B	Mun_C	Mun_D	Mun_E	Mun_F	Mun_G	Mun_H	Mun_I	Mun_J
Mun_A	1.000	0.758	0.735	0.762	0.788	0.796	0.779	0.765	0.779	0.793
Mun_B	0.758	1.000	0.749	0.759	0.782	0.779	0.776	0.741	0.776	0.801
Mun_C	0.735	0.749	1.000	0.764	0.799	0.793	0.770	0.733	0.770	0.804
Mun_D	0.762	0.759	0.764	1.000	0.823	0.841	0.911	0.762	0.911	0.837
Mun_E	0.788	0.782	0.799	0.823	1.000	0.874	0.848	0.786	0.848	0.882
Mun_F	0.796	0.779	0.793	0.841	0.874	1.000	0.875	0.793	0.875	0.868
Mun_G	0.779	0.776	0.770	0.911	0.848	0.875	1.000	0.777	1.000	0.870
Mun_H	0.765	0.741	0.733	0.762	0.786	0.793	0.777	1.000	0.777	0.829
Mun_I	0.779	0.776	0.770	0.911	0.848	0.875	1.000	0.777	1.000	0.870
Mun_J	0.793	0.801	0.804	0.837	0.882	0.868	0.870	0.829	0.870	1.000

SPS	Mun_A	Mun_B	Mun_C	Mun_D	Mun_E	Mun_F	Mun_G	Mun_H	Mun_I	Mun_J
Mun_A	1.000	0.495	0.426	0.543	0.559	0.617	0.569	0.650	0.569	0.581
Mun_B	0.495	1.000	0.405	0.494	0.406	0.595	0.639	0.475	0.639	0.634
Mun_C	0.426	0.405	1.000	0.564	0.563	0.295	0.504	0.408	0.504	0.633
Mun_D	0.543	0.494	0.564	1.000	0.503	0.529	0.789	0.637	0.789	0.698
Mun_E	0.559	0.406	0.563	0.503	1.000	0.376	0.549	0.403	0.549	0.500
Mun_F	0.617	0.595	0.295	0.529	0.376	1.000	0.660	0.584	0.660	0.616
Mun_G	0.569	0.639	0.504	0.789	0.549	0.660	1.000	0.735	1.000	0.888
Mun_H	0.650	0.475	0.408	0.637	0.403	0.584	0.735	1.000	0.735	0.755
Mun_I	0.569	0.639	0.504	0.789	0.549	0.660	1.000	0.735	1.000	0.888
Mun_J	0.581	0.634	0.633	0.698	0.500	0.616	0.888	0.755	0.888	1.000

C.4 MOR process

GED	Mun_A	Mun_B	Mun_C	Mun_D	Mun_E	Mun_F	Mun_G	Mun_H	Mun_I	Mun_J
Mun_A	1.000	0.832	0.773	0.763	0.837	0.738	0.801	0.743	0.766	0.757
Mun_B	0.832	1.000	0.785	0.790	0.858	0.767	0.820	0.755	0.774	0.778
Mun_C	0.773	0.785	1.000	0.739	0.860	0.737	0.804	0.739	0.740	0.739
Mun_D	0.763	0.790	0.739	1.000	0.796	0.741	0.789	0.758	0.742	0.754
Mun_E	0.837	0.858	0.860	0.796	1.000	0.767	0.895	0.770	0.781	0.775
Mun_F	0.738	0.767	0.737	0.741	0.767	1.000	0.779	0.733	0.733	0.803
Mun_G	0.801	0.820	0.804	0.789	0.895	0.779	1.000	0.768	0.738	0.812
Mun_H	0.743	0.755	0.739	0.758	0.770	0.733	0.768	1.000	0.729	0.757
Mun_I	0.766	0.774	0.740	0.742	0.781	0.733	0.738	0.729	1.000	0.721
Mun_J	0.757	0.778	0.739	0.754	0.775	0.803	0.812	0.757	0.721	1.000

SPS	Mun_A	Mun_B	Mun_C	Mun_D	Mun_E	Mun_F	Mun_G	Mun_H	Mun_I	Mun_J
Mun_A	1.000	0.603	0.509	0.419	0.533	0.530	0.544	0.359	0.367	0.421
Mun_B	0.603	1.000	0.540	0.595	0.624	0.573	0.547	0.579	0.434	0.418
Mun_C	0.509	0.540	1.000	0.470	0.709	0.449	0.631	0.437	0.299	0.524
Mun_D	0.419	0.595	0.470	1.000	0.489	0.443	0.478	0.503	0.457	0.467
Mun_E	0.533	0.624	0.709	0.489	1.000	0.475	0.864	0.504	0.498	0.519
Mun_F	0.530	0.573	0.449	0.443	0.475	1.000	0.539	0.523	0.470	0.581
Mun_G	0.544	0.547	0.631	0.478	0.864	0.539	1.000	0.556	0.427	0.683
Mun_H	0.359	0.579	0.437	0.503	0.504	0.523	0.556	1.000	0.377	0.470
Mun_I	0.367	0.434	0.299	0.457	0.498	0.470	0.427	0.377	1.000	0.351
Mun_J	0.421	0.418	0.524	0.467	0.519	0.581	0.683	0.470	0.351	1.000

C.5 $WABO_1$ process

GED	Mun_A	Mun_B	Mun_C	Mun_D	Mun_E	Mun_F	Mun_G	Mun_H	Mun_I	Mun_J
Mun_A	1.000	0.682	0.656	0.656	0.769	0.657	0.769	0.769	0.769	0.769
Mun_B	0.682	1.000	0.748	0.794	0.794	0.765	0.794	0.794	0.794	0.794
Mun_C	0.656	0.748	1.000	0.850	0.819	0.784	0.819	0.819	0.819	0.819
Mun_D	0.656	0.794	0.850	1.000	0.952	0.878	0.952	0.952	0.952	0.952
Mun_E	0.769	0.794	0.819	0.952	1.000	0.878	1.000	1.000	1.000	1.000
Mun_F	0.657	0.765	0.784	0.878	0.878	1.000	0.878	0.878	0.878	0.878
Mun_G	0.769	0.794	0.819	0.952	1.000	0.878	1.000	1.000	1.000	1.000
Mun_H	0.769	0.794	0.819	0.952	1.000	0.878	1.000	1.000	1.000	1.000
Mun_I	0.769	0.794	0.819	0.952	1.000	0.878	1.000	1.000	1.000	1.000
Mun_J	0.769	0.794	0.819	0.952	1.000	0.878	1.000	1.000	1.000	1.000

SPS	Mun_A	Mun_B	Mun_C	Mun_D	Mun_E	Mun_F	Mun_G	Mun_H	Mun_I	Mun_J
Mun_A	1.000	0.303	0.539	0.629	0.681	0.565	0.681	0.681	0.681	0.681
Mun_B	0.303	1.000	0.416	0.488	0.488	0.513	0.488	0.488	0.488	0.488
Mun_C	0.539	0.416	1.000	0.775	0.728	0.574	0.728	0.728	0.728	0.728
Mun_D	0.629	0.488	0.775	1.000	0.948	0.784	0.948	0.948	0.948	0.948
Mun_E	0.681	0.488	0.728	0.948	1.000	0.778	1.000	1.000	1.000	1.000
Mun_F	0.565	0.513	0.574	0.784	0.778	1.000	0.778	0.778	0.778	0.778
Mun_G	0.681	0.488	0.728	0.948	1.000	0.778	1.000	1.000	1.000	1.000
Mun_H	0.681	0.488	0.728	0.948	1.000	0.778	1.000	1.000	1.000	1.000
Mun_I	0.681	0.488	0.728	0.948	1.000	0.778	1.000	1.000	1.000	1.000
Mun_J	0.681	0.488	0.728	0.948	1.000	0.778	1.000	1.000	1.000	1.000

C.6 $WABO_2$ process

GED	Mun_A	Mun_B	Mun_C	Mun_D	Mun_E	Mun_F	Mun_G	Mun_H	Mun_I	Mun_J
Mun_A	1.000	0.787	0.830	0.872	0.836	0.833	0.872	0.872	0.840	0.789
Mun_B	0.787	1.000	0.728	0.728	0.724	0.725	0.728	0.728	0.736	0.734
Mun_C	0.830	0.728	1.000	0.925	0.905	0.903	0.925	0.925	0.917	0.880
Mun_D	0.872	0.728	0.925	1.000	0.944	0.943	1.000	1.000	0.969	0.901
Mun_E	0.836	0.724	0.905	0.944	1.000	0.996	0.944	0.944	0.917	0.904
Mun_F	0.833	0.725	0.903	0.943	0.996	1.000	0.943	0.943	0.915	0.907
Mun_G	0.872	0.728	0.925	1.000	0.944	0.943	1.000	1.000	0.969	0.901
Mun_H	0.872	0.728	0.925	1.000	0.944	0.943	1.000	1.000	0.969	0.901
Mun_I	0.840	0.736	0.917	0.969	0.917	0.915	0.969	0.969	1.000	0.906
Mun_J	0.789	0.734	0.880	0.901	0.904	0.907	0.901	0.901	0.906	1.000

SPS	Mun_A	Mun_B	Mun_C	Mun_D	Mun_E	Mun_F	Mun_G	Mun_H	Mun_I	Mun_J
Mun_A	1.000	0.500	0.700	0.776	0.596	0.582	0.776	0.776	0.711	0.505
Mun_B	0.500	1.000	0.445	0.416	0.471	0.443	0.416	0.416	0.328	0.273
Mun_C	0.700	0.445	1.000	0.887	0.746	0.733	0.887	0.887	0.766	0.548
Mun_D	0.776	0.416	0.887	1.000	0.787	0.778	1.000	1.000	0.889	0.617
Mun_E	0.596	0.471	0.746	0.787	1.000	0.986	0.787	0.787	0.682	0.689
Mun_F	0.582	0.443	0.733	0.778	0.986	1.000	0.778	0.778	0.669	0.695
Mun_G	0.776	0.416	0.887	1.000	0.787	0.778	1.000	1.000	0.889	0.617
Mun_H	0.776	0.416	0.887	1.000	0.787	0.778	1.000	1.000	0.889	0.617
Mun_I	0.711	0.328	0.766	0.889	0.682	0.669	0.889	0.889	1.000	0.732
Mun_J	0.505	0.273	0.548	0.617	0.689	0.695	0.617	0.617	0.732	1.000

C.7 WMO process

GED	Mun_A	Mun_B	Mun_C	Mun_D	Mun_E	Mun_F	Mun_G	Mun_H	Mun_I	Mun_J
Mun_A	1.000	0.916	0.817	0.773	0.776	0.802	0.795	0.976	0.949	0.801
Mun_B	0.916	1.000	0.791	0.755	0.758	0.778	0.773	0.927	0.901	0.777
Mun_C	0.817	0.791	1.000	0.758	0.831	0.816	0.858	0.825	0.813	0.821
Mun_D	0.773	0.755	0.758	1.000	0.746	0.759	0.745	0.779	0.778	0.757
Mun_E	0.776	0.758	0.831	0.746	1.000	0.785	0.804	0.781	0.780	0.784
Mun_F	0.802	0.778	0.816	0.759	0.785	1.000	0.807	0.809	0.797	0.811
Mun_G	0.795	0.773	0.858	0.745	0.804	0.807	1.000	0.802	0.790	0.805
Mun_H	0.976	0.927	0.825	0.779	0.781	0.809	0.802	1.000	0.966	0.809
Mun_I	0.949	0.901	0.813	0.778	0.780	0.797	0.790	0.966	1.000	0.797
Mun_J	0.801	0.777	0.821	0.757	0.784	0.811	0.805	0.809	0.797	1.000

SPS	Mun_A	Mun_B	Mun_C	Mun_D	Mun_E	Mun_F	Mun_G	Mun_H	Mun_I	Mun_J
Mun_A	1.000	0.694	0.515	0.432	0.497	0.431	0.423	0.923	0.893	0.512
Mun_B	0.694	1.000	0.367	0.372	0.400	0.359	0.388	0.669	0.656	0.404
Mun_C	0.515	0.367	1.000	0.348	0.601	0.536	0.616	0.452	0.469	0.584
Mun_D	0.432	0.372	0.348	1.000	0.336	0.354	0.264	0.449	0.435	0.373
Mun_E	0.497	0.400	0.601	0.336	1.000	0.476	0.511	0.398	0.440	0.470
Mun_F	0.431	0.359	0.536	0.354	0.476	1.000	0.433	0.462	0.491	0.565
Mun_G	0.423	0.388	0.616	0.264	0.511	0.433	1.000	0.420	0.432	0.457
Mun_H	0.923	0.669	0.452	0.449	0.398	0.462	0.420	1.000	0.930	0.548
Mun_I	0.893	0.656	0.469	0.435	0.440	0.491	0.432	0.930	1.000	0.529
Mun_J	0.512	0.404	0.584	0.373	0.470	0.565	0.457	0.548	0.529	1.000

C.8 WOZ process

GED	Mun_A	Mun_B	Mun_C	Mun_D	Mun_E	Mun_F	Mun_G	Mun_H	Mun_I	Mun_J
Mun_A	1.000	0.916	0.817	0.773	0.776	0.802	0.795	0.976	0.949	0.801
Mun_B	0.916	1.000	0.791	0.755	0.758	0.778	0.773	0.927	0.901	0.777
Mun_C	0.817	0.791	1.000	0.758	0.831	0.816	0.858	0.825	0.813	0.821
Mun_D	0.773	0.755	0.758	1.000	0.746	0.759	0.745	0.779	0.778	0.757
Mun_E	0.776	0.758	0.831	0.746	1.000	0.785	0.804	0.781	0.780	0.784
Mun_F	0.802	0.778	0.816	0.759	0.785	1.000	0.807	0.809	0.797	0.811
Mun_G	0.795	0.773	0.858	0.745	0.804	0.807	1.000	0.802	0.790	0.805
Mun_H	0.976	0.927	0.825	0.779	0.781	0.809	0.802	1.000	0.966	0.809
Mun_I	0.949	0.901	0.813	0.778	0.780	0.797	0.790	0.966	1.000	0.797
Mun_J	0.801	0.777	0.821	0.757	0.784	0.811	0.805	0.809	0.797	1.000

SPS	Mun_A	Mun_B	Mun_C	Mun_D	Mun_E	Mun_F	Mun_G	Mun_H	Mun_I	Mun_J
Mun_A	1.000	0.389	0.511	0.549	0.395	0.499	0.531	0.487	0.487	0.466
Mun_B	0.389	1.000	0.491	0.431	0.413	0.435	0.425	0.592	0.592	0.416
Mun_C	0.511	0.491	1.000	0.437	0.380	0.441	0.482	0.527	0.527	0.364
Mun_D	0.549	0.431	0.437	1.000	0.540	0.672	0.673	0.809	0.809	0.562
Mun_E	0.395	0.413	0.380	0.540	1.000	0.596	0.516	0.666	0.666	0.812
Mun_F	0.499	0.435	0.441	0.672	0.596	1.000	0.879	0.744	0.744	0.649
Mun_G	0.531	0.425	0.482	0.673	0.516	0.879	1.000	0.658	0.658	0.563
Mun_H	0.487	0.592	0.527	0.809	0.666	0.744	0.658	1.000	1.000	0.563
Mun_I	0.487	0.592	0.527	0.809	0.666	0.744	0.658	1.000	1.000	0.687
Mun_J	0.466	0.416	0.364	0.562	0.812	0.649	0.563	0.563	0.687	1.000

D Similary vs. complexity

Process	Mun1	Mun2	GED	SPS	Unified	CFC	Density	CC	Unified
GBA_1	Mun_A	Mun_G	0.667	0.250	0.258	51	0.087	0.020	39
GBA_1	Mun_C	Mun_G	0.665	0.200	0.232	33	0.095	0.023	30
GBA_1	Mun_C	Mun_I	0.798	0.735	0.638	15	0.194	0.049	13
GBA_1	Mun_D	Mun_G	0.719	0.466	0.422	36	0.095	0.019	33
GBA_1	Mun_D	Mun_I	0.950	0.929	0.894	7	0.238	0.091	7
GBA_1	Mun_E	Mun_B	0.841	0.774	0.702	15	0.232	0.044	13
GBA_1	Mun_F	Mun_H	0.803	0.649	0.601	13	0.167	0.043	13
GBA_1	Mun_G	Mun_F	0.711	0.364	0.362	34	0.108	0.026	28
GBA_1	Mun_H	Mun_B	0.837	0.573	0.598	17	0.178	0.048	14
GBA_1	Mun_I	Mun_A	0.942	0.788	0.815	11	0.214	0.061	10
GBA_1	Mun_I	Mun_B	0.896	0.781	0.763	13	0.262	0.047	11
GBA_1	Mun_I	Mun_F	0.879	0.917	0.814	7	0.238	0.084	7
GBA_1	Mun_J	Mun_D	0.801	0.491	0.519	9	0.214	0.072	9
GBA_1	Mun_J	Mun_I	0.793	0.459	0.495	9	0.214	0.073	9
GBA_2	Mun_A	Mun_B	0.898	0.756	0.754	20	0.114	0.023	24
GBA_2	Mun_A	Mun_C	0.867	0.526	0.606	27	0.084	0.018	32
GBA_2	Mun_C	Mun_G	0.867	0.526	0.606	25	0.088	0.021	29
GBA_2	Mun_E	Mun_F	0.938	0.837	0.835	16	0.144	0.029	18
GBA_2	Mun_E	Mun_J	0.944	0.858	0.852	18	0.135	0.028	20
GBA_2	Mun_F	Mun_C	0.863	0.574	0.625	26	0.091	0.018	31
GBA_2	Mun_F	Mun_G	0.941	0.912	0.876	12	0.167	0.034	15
GBA_2	Mun_F	Mun_H	0.932	0.784	0.803	19	0.121	0.025	22
GBA_2	Mun_G	Mun_J	0.894	0.736	0.739	18	0.121	0.025	22
GBA_2	Mun_H	Mun_D	0.897	0.628	0.688	33	0.083	0.021	32
GBA_2	Mun_I	Mun_A	0.891	0.710	0.723	19	0.121	0.025	22
GBA_2	Mun_I	Mun_D	0.827	0.488	0.545	39	0.063	0.016	43
GBA_2	Mun_I	Mun_E	0.924	0.830	0.817	19	0.110	0.023	24
GBA_2	Mun_J	Mun_E	0.944	0.858	0.852	18	0.135	0.028	20
GBA_3	Mun_A	Mun_B	0.758	0.495	0.477	38	0.078	0.031	30
GBA_3	Mun_A	Mun_F	0.796	0.617	0.577	30	0.071	0.027	30
GBA_3	Mun_B	Mun_H	0.741	0.475	0.449	37	0.069	0.018	39
GBA_3	Mun_C	Mun_G	0.770	0.504	0.493	30	0.069	0.019	35
GBA_3	Mun_D	Mun_E	0.823	0.503	0.548	28	0.081	0.027	28
GBA_3	Mun_D	Mun_F	0.841	0.529	0.581	33	0.068	0.022	34
GBA_3	Mun_E	Mun_J	0.882	0.698	0.708	25	0.103	0.035	22
GBA_3	Mun_F	Mun_C	0.793	0.295	0.413	33	0.063	0.021	36
GBA_3	Mun_G	Mun_B	0.776	0.639	0.567	29	0.081	0.028	28
GBA_3	Mun_G	Mun_D	0.911	0.789	0.784	17	0.124	0.052	15
GBA_3	Mun_G	Mun_I	1.000	1.000	0.982	8	0.194	0.113	7
GBA_3	Mun_I	Mun_F	0.875	0.660	0.681	30	0.078	0.026	29
GBA_3	Mun_J	Mun_B	0.801	0.634	0.590	26	0.081	0.033	25
GBA_3	Mun_J	Mun_E	0.882	0.500	0.609	25	0.103	0.035	22
MOR	Mun_A	Mun_C	0.773	0.509	0.499	42	0.067	0.012	50
MOR	Mun_C	Mun_A	0.773	0.509	0.499	42	0.067	0.012	50
MOR	Mun_C	Mun_E	0.860	0.709	0.690	31	0.073	0.015	39
MOR	Mun_C	Mun_J	0.739	0.524	0.471	40	0.062	0.012	49
MOR	Mun_D	Mun_J	0.754	0.467	0.458	29	0.080	0.016	36
MOR	Mun_E	Mun_F	0.767	0.475	0.476	45	0.057	0.011	56
MOR	Mun_E	Mun_H	0.770	0.504	0.493	32	0.066	0.017	38

Process	Mun1	Mun2	GED	SPS	Unified	CFC	Density	CC	Unified
MOR	Mun_F	Mun_E	0.767	0.475	0.476	45	0.057	0.011	55
MOR	Mun_G	Mun_E	0.895	0.864	0.804	27	0.096	0.021	29
MOR	Mun_H	Mun_C	0.739	0.437	0.427	37	0.055	0.013	48
MOR	Mun_H	Mun_G	0.768	0.556	0.517	32	0.066	0.017	39
MOR	Mun_H	Mun_J	0.757	0.470	0.463	34	0.060	0.015	44
MOR	Mun_I	Mun_C	0.740	0.299	0.359	36	0.075	0.014	43
MOR	Mun_J	Mun_H	0.757	0.470	0.463	34	0.060	0.015	44
$WABO_1$	Mun_A	Mun_I	0.769	0.681	0.581	12	0.100	0.044	16
$WABO_1$	Mun_B	Mun_C	0.748	0.416	0.427	18	0.076	0.032	23
$WABO_1$	Mun_B	Mun_G	0.794	0.488	0.511	14	0.096	0.037	19
$WABO_1$	Mun_D	Mun_B	0.794	0.488	0.511	14	0.096	0.039	18
$WABO_1$	Mun_D	Mun_I	0.952	0.948	0.906	10	0.129	0.065	12
$WABO_1$	Mun_E	Mun_H	1.000	1.000	0.982	5	0.267	0.094	6
$WABO_1$	Mun_E	Mun_J	1.000	1.000	0.982	5	0.267	0.094	6
$WABO_1$	Mun_G	Mun_D	0.952	0.948	0.906	10	0.129	0.065	12
$WABO_1$	Mun_G	Mun_E	1.000	1.000	0.982	5	0.267	0.094	6
$WABO_1$	Mun_I	Mun_E	1.000	1.000	0.982	5	0.267	0.094	6
$WABO_1$	Mun_J	Mun_A	0.769	0.681	0.581	12	0.100	0.044	16
$WABO_1$	Mun_J	Mun_E	1.000	1.000	0.982	5	0.267	0.094	6
$WABO_1$	Mun_J	Mun_F	0.878	0.778	0.743	15	0.105	0.037	18
$WABO_2$	Mun_A	Mun_G	0.872	0.776	0.736	130	0.036	0.012	92
$WABO_2$	Mun_A	Mun_H	0.872	0.776	0.736	130	0.036	0.012	92
$WABO_2$	Mun_D	Mun_C	0.925	0.887	0.846	70	0.032	0.015	71
$WABO_2$	Mun_E	Mun_A	0.836	0.596	0.608	144	0.032	0.012	103
$WABO_2$	Mun_H	Mun_F	0.943	0.778	0.811	63	0.043	0.020	55
$WABO_2$	Mun_I	Mun_F	0.915	0.669	0.728	63	0.044	0.020	54
$WABO_2$	Mun_J	Mun_C	0.880	0.548	0.631	65	0.031	0.014	72
$WABO_2$	Mun_J	Mun_G	0.901	0.617	0.687	55	0.032	0.017	64
$WABO_2$	Mun_J	Mun_I	0.906	0.732	0.749	44	0.042	0.021	48
WMO	Mun_A	Mun_D	0.773	0.432	0.461	100	0.026	0.009	107
WMO	Mun_A	Mun_E	0.776	0.497	0.496	80	0.033	0.009	88
WMO	Mun_C	Mun_B	0.791	0.367	0.446	71	0.038	0.012	73
WMO	Mun_C	Mun_H	0.825	0.452	0.525	60	0.049	0.012	62
WMO	Mun_D	Mun_A	0.773	0.432	0.461	99	0.026	0.009	105
WMO	Mun_E	Mun_A	0.776	0.497	0.496	83	0.032	0.008	93
WMO	Mun_E	Mun_B	0.758	0.400	0.429	89	0.029	0.009	96
WMO	Mun_E	Mun_C	0.831	0.601	0.606	82	0.032	0.009	92
WMO	Mun_E	Mun_J	0.784	0.470	0.491	68	0.035	0.011	75
WMO	Mun_F	Mun_A	0.802	0.431	0.490	67	0.043	0.010	72
WMO	Mun_F	Mun_E	0.785	0.476	0.496	62	0.038	0.011	71
WMO	Mun_G	Mun_J	0.805	0.457	0.507	61	0.046	0.012	62
WMO	Mun_J	Mun_D	0.757	0.373	0.414	83	0.028	0.013	84
WMO	Mun_J	Mun_F	0.811	0.565	0.566	59	0.040	0.011	70
WOZ	Mun_A	Mun_C	0.771	0.511	0.498	48	0.045	0.018	50
WOZ	Mun_B	Mun_I	0.745	0.592	0.511	51	0.056	0.015	51
WOZ	Mun_C	Mun_D	0.736	0.437	0.425	50	0.046	0.018	50
WOZ	Mun_D	Mun_G	0.831	0.673	0.641	27	0.058	0.019	37
WOZ	Mun_E	Mun_F	0.799	0.596	0.570	35	0.045	0.016	48
WOZ	Mun_E	Mun_I	0.895	0.666	0.704	30	0.058	0.020	37
WOZ	Mun_F	Mun_E	0.799	0.596	0.570	35	0.045	0.016	48
WOZ	Mun_J	Mun_E	0.875	0.812	0.757	41	0.046	0.016	49

E Clusters

Cluster	GBA_1	GBA_2	GBA_3	MOR	$WABO_1$	$WABO_2$	WMO	WOZ
k	BDEFI	D	DFH	DHI	F	CDGHI	D	DEHIJ
	GJ	AEFGI	GIJ	BEFGJ	DEGHIJ	AB	CEFGJ	AFG
	C	BCHJ	ABCE	AC	ABC	EFJ	ABHI	BC
1	AF	BDEJ	CEG	BDE	DEH	DH	AEGI	F
	G	I	ABDI	ACGH	AFIJ	FIJ	DFJ	AGH
	BCDEHIJ	ACFGH	FHJ	FIJ	BCG	ABCEG	BCH	BCDEIJ
2	AJ	ACDE	EIJ	H	BDGHIJ	G	CE	BCG
	BDGH	BI	BD	ACGI	CEF	CFH	BFGHIJ	AEFIJ
	CEFI	FGHJ	ACFGH	BDEFJ	A	ABDEIJ	AD	DH
3	EIJ	BCEG	CEG	F	CG	ADHIJ	ACDE	DJ
	ACFH	ADF	AHJ	CDG	ABDJ	EFG	BFGHJ	EFI
	BDG	HIJ	BDFI	ABEHIJ	EFHI	BC	I	ABCGH
4	CEFI	CDH	BEIJ	BCD	G	AG	ABCIJ	BCJ
	BJ	ABG	AFH	AFGI	ACDEFHJ	BCDEFIJ	H	FGH
	ADGH	EFIJ	CDG	EHJ	BI	H	DEFG	ADEI
5	E	ABC	BH	CDGIJ	B	BCDEGIJ	E	CEIJ
	CFHJ	FGIJ	ACG	ABEH	ACDEF	AH	BI	ABDF
	ABDGI	DEH	DEFIJ	F	GHIJ	F	ACDFGHJ	GH
6	ABCF	ACDH	EHJ	BCHI	BDH	BH	AEH	ACHI
	DEIJ	BEIJ	BD	DEF	CEIJ	AEGJ	J	BDEGJ
	GH	FG	ACFGI	AGJ	AFG	CDFI	BCDFGI	F
7	F	BGHI	CEGJ	E	CEI	DEHI	BCFI	ABDEFGJ
	BCDH	CF	ABFHI	ABDIJ	DFGJ	FG	AEJ	H
	AEGIJ	ADEJ	D	CGH	ABH	ABCJ	DGH	CI
8	CEFIJ	ACJ	BEH	ACFJ	ACG	AGHI	CDFG	AEI
	BG	EH	ADFG	BDI	BDEHIJ	BDF	EI	BCDGH
	ADH	BDFGI	CIJ	EGH	F	CEJ	ABHJ	FJ
9	AEGJ	ADGJ	ADG	AIJ	J	BCDE	ABCEF	ABFIJ
	CH	EF	BCFH	BCDEF	ACDFGI	AF	IJ	EGH
	BDFI	BCHI	EIJ	GH	BEH	GHIJ	DGH	CD
10	BCDGI	FI	FHIJ	FGJ	DHI	CEFGIJ	AEGJ	E
	FH	AJ	ACE	BEH	AE	AB	BDH	ABCFGHI
	AEJ	BCDEGH	BDG	ACDI	BCFGJ	DH	CFI	DJ