IFN650 Assignment 1 - 2023 Semester 1

Finance Domain Process Modelling on WoPeD

Kavya Kore n10840371

Pages Used Within Limit: 14/15

Due: 21 April 2023 (including the 48hr extended period)

Table of Contents

1.0 Process Model Description

The finance company process that I will be modelling and analysing is a loan application process at a fictional bank, which I have named KBANK. This process involves several steps, including application review, credit check, and approval or denial of the loan. The primary goal of this process is to ensure that the company provides loans to customers who are most likely to repay them, while minimising the risk of financial loss.

It is understood that the process depicted below may not fully reflect every step of a typical bank's loan process, because doing so would result in a very complex model with a lot more exceptions possibly. Another language such as YAWL might be a more appropriate way to do this rather than a workflow net. For the purpose of this assignment, the process has been drafted in a way which can demonstrate different workflow net constructs and patterns to provide an interesting capacity planning and simulation scenario.

1.1 Process Flow

The loan assessment process of KBANK starts with the submission of a loan application by the applicant. The application must be filled in online and then submitted through the online portal of KBANK. Once received, the application is registered by the Online Loan System. A Service Staff will then need to review the application on the system (checking completeness and validity of the entries). If the outcome of the application review is Not OK, there are two possibilities:

1. If the issues are major, the Service Staff will contact the applicant (by email or phone number based on provided contact details in their application) and request a resubmission of their application. The Service Staff will then cancel the application on the portal, and finally archive the application, and the process ends.
2. If the issues are minor and can be rectified by the Service Staff, then they will go ahead and rectify the issues, which then makes the review outcome as OK, and the process continues as below.

Therefore, if the application review is OK, then the Service Staff member will prepare to handover the application onwards by first initiating two checks. These are to be done in parallel: a customer credit history check (to be performed by a Financial Assessor) and a property appraisal (to be performed by a Property Officer). Once both tasks are completed, a Loans Officer will be advised to take over and conduct a loan risk assessment based on the results that were obtained from those previous checks, which may involve aspects such as a risk weighting calculation which will inform the next task. The Loans Officer then makes an informed decision about the applicant's eligibility based on all data that has been made available. If the applicant fails eligibility (it is Not OK), a Service Staff will reject the application and then archive the application, and the process ends. Alternatively, if the Loans Officer is on a borderline regarding the decision, especially if an unexpected market event takes place, they may request for a revision of the earlier two checks (credit & property). This request will be handled by a Service Staff who once again initiates the check, and the checks repeats as discussed before.

On the other hand, if the application passes the eligibility test, it is marked as approved by the Loans Officer into the system. The Loans Officer then checks for any inclusion noted within the application. Basically, applicants are allowed to nominate up to one (and only one) complementary discounted inclusion that they wish to add-on to their loan: either a credit card, or a home insurance policy. If a credit card is required, a Financial Assessor will prepare the credit card quote. Otherwise, if home insurance is required, a Property Officer will prepare the home insurance quote. After either of these finishes (or in the situation where an applicant wants none of them), an acceptance package is prepared by the Loans Officer. Afterwards, the acceptance package is safely sent digitally to the via KBANK's online portal, again done by the Loans Officer.

The acceptance package not only contains the proposed loan's repayment agreement, but also one of the quotes if the applicant had requested an inclusion. The applicant must digitally sign and return this to KBANK within 14 days. However, because KBANK believes in putting customers first and offers extra flexibility, it is possible that the applicant decides they need to make some sort of change to their loan. Therefore, there are 4 possibilities that could take place:

1. Firstly, the Online Loan System may receive **a loan amendment** from the applicant (such as if they need to take out an extra amount). This means that a revision is required, so the Loans Officer will need to request for that revision to a Service Staff. The Service Staff once again initiates a customer credit history check and property appraisal and the process repeats as discussed before.

2. Alternatively, the Online Loan System may receive **an inclusion change** from the applicant. This means that a Loans Officer will need to once again check the inclusions, and then pass on to the appropriate Financial Assessor or Property Officer to prepare a quote as mentioned earlier.

3. Otherwise, which should be the case most of the time, the applicant is happy and therefore the Online Loan System **receives the signed acceptance package** back. This means that the application is finalised, so a Service Staff member just needs to archive the process, and the process ends.

4. Lastly, if **no reply is received in 14 days**, the application must be cancelled by a Service Staff, and afterwards archived by a Service Staff too.

In short, regardless of whether the application was ultimately approved, rejected or cancelled, the final task is always done by a Service Staff which is archiving the application on the system.

The process is handled digitally where most tasks need to be done either by or on the online loan system.

2.0 Process WF-net

Using WoPeD, I developed a workflow net of the above process flow, as shown below in Figure 1.

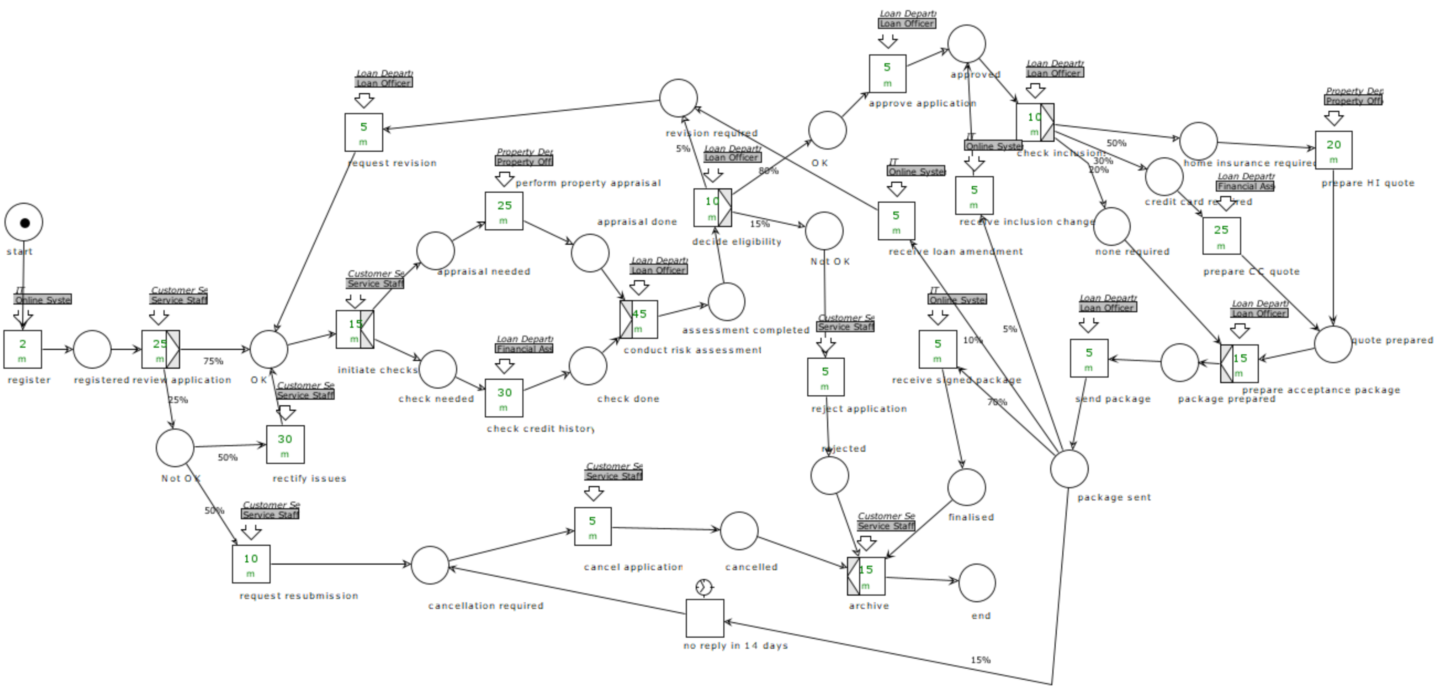


Figure 1: Workflow Net of KBANK's Loan Application Process

Due to the size of the model and being very horizontal, the full model is included in the Appendix, please refer to this. It contains over 23 tasks including a mixture of structured and unstructured loops, many decision points and two times a parallel sequence flow. This will be sufficient to demonstrate verification, capacity planning and run simulations later on.

In KBANK, there are 12 hypothetical staff and 1 system utilised in the process, across 5 resources roles within 4 groups (departments), also visualised below:

- **2 Financial Assessors** (Loan Department)
- **4 Loan Officers** (Loan Department)
- **1 Online Loan System** (IT Department)
- **2 Property Officers** (Property Department)
- **4 Service Staff** (Customer Service Department)

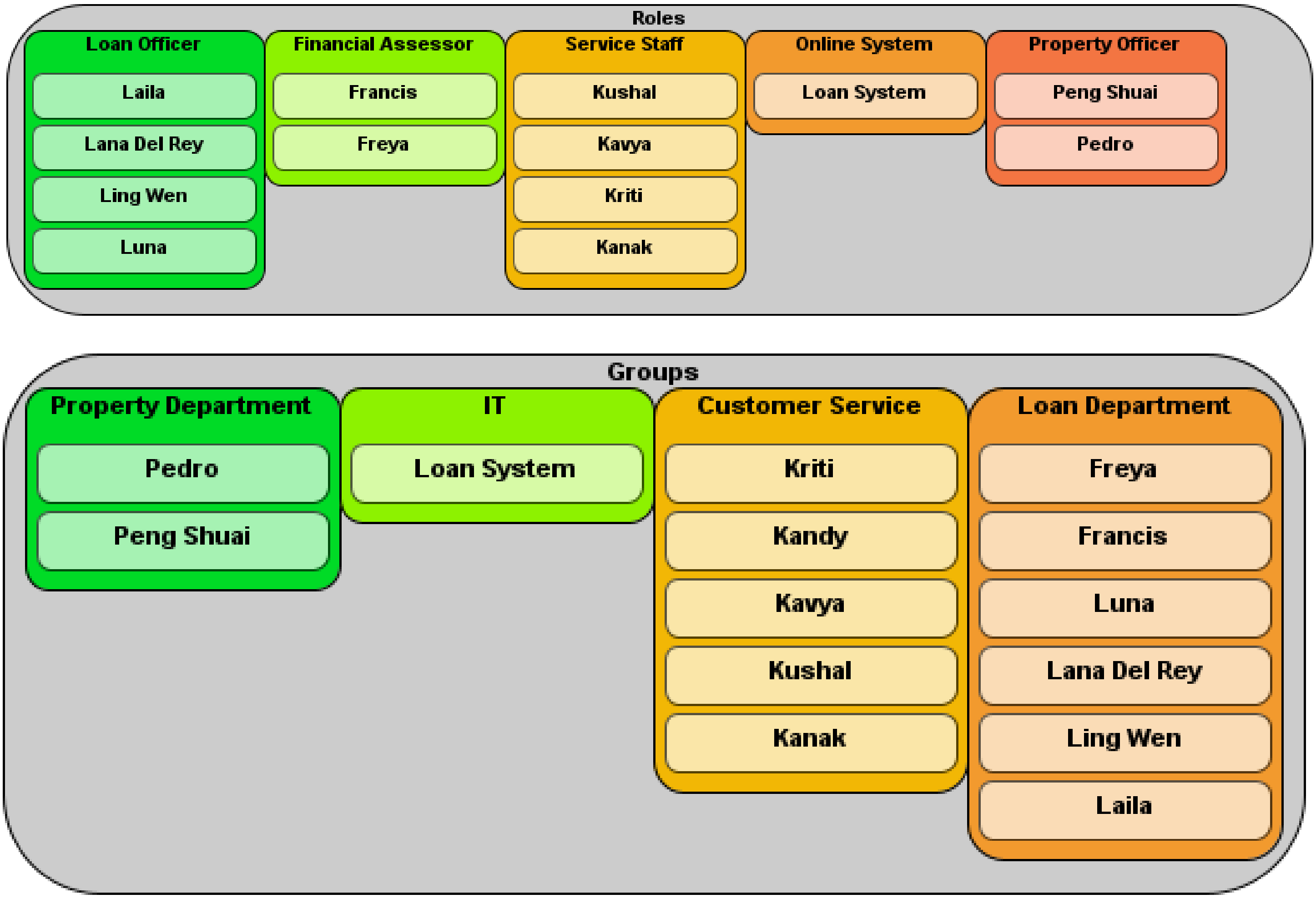


Figure 2: Roles and Groups within KBANK

Initially, purposely more staff have been assigned to the roles which have the most frequent and/or intensive tasks. These may be altered later on to tweak the planning and simulation. For example, if there is an unexpected absence from a staff member, sickness or leave period. Section 4.0 and 5.0 will later consider if the resources provided are sufficient or not, and also consider the resource utilisation through sample run throughs of the process.

3.0 Process Verification (including unsound models)

To ensure the soundness of the process model, I have conducted process verification against the provided three soundness criteria.

Firstly, I created a reachability (coverability) graph to make sure the model is operating as expected, as per the Figure 3 on the right. Visibly, there is one starting place and one ending place, which means that there is never a situation where unlimited tokens are created or a token ending up in an inappropriate marking. The graph seems messy at times which is because there are situations where a token needs to go back to an earlier part of the process to do some rework, but otherwise continues as per normal. Never will a token be ‘stuck’ in the model, or is there a situation where a task is ‘dead’, because there is always a way to progress along markings until the ending (p14) is reached.

Secondly, I also played the token game to test out several execution paths and make sure they make sense.

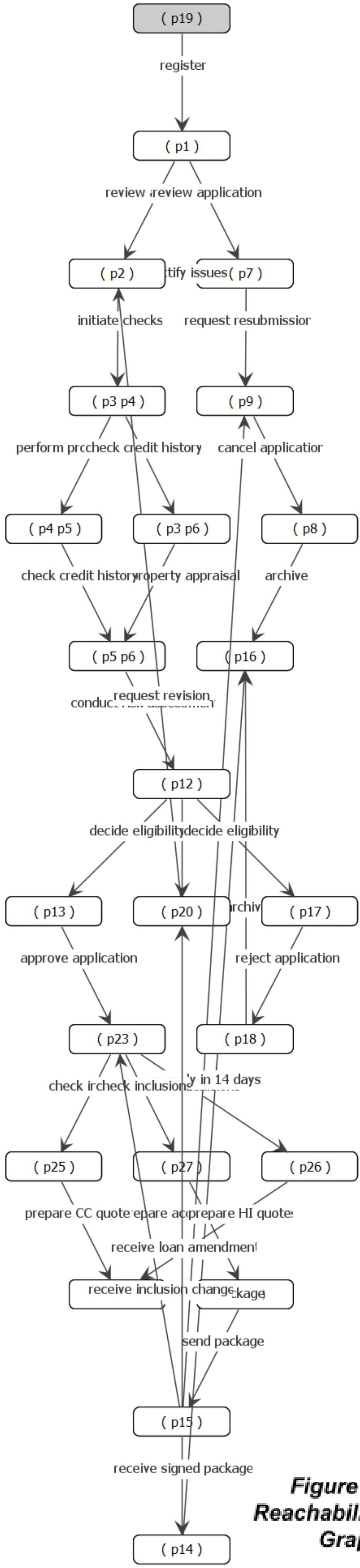


Figure 3: Reachability Graph

3.1 Example 1: Violation of ‘Option to complete’

This property refers to the ability of every token in the model to eventually complete the process. Every state M which is reachable from the initial state, needs to have a firing sequence from M that leads to the final end state.

A common cause of this violation is when there is a deadlock in the process somewhere, often the result of a mismatch between the split and join mechanisms used. Below is a modified example from my process where the bottom XOR join has been replaced with AND join:

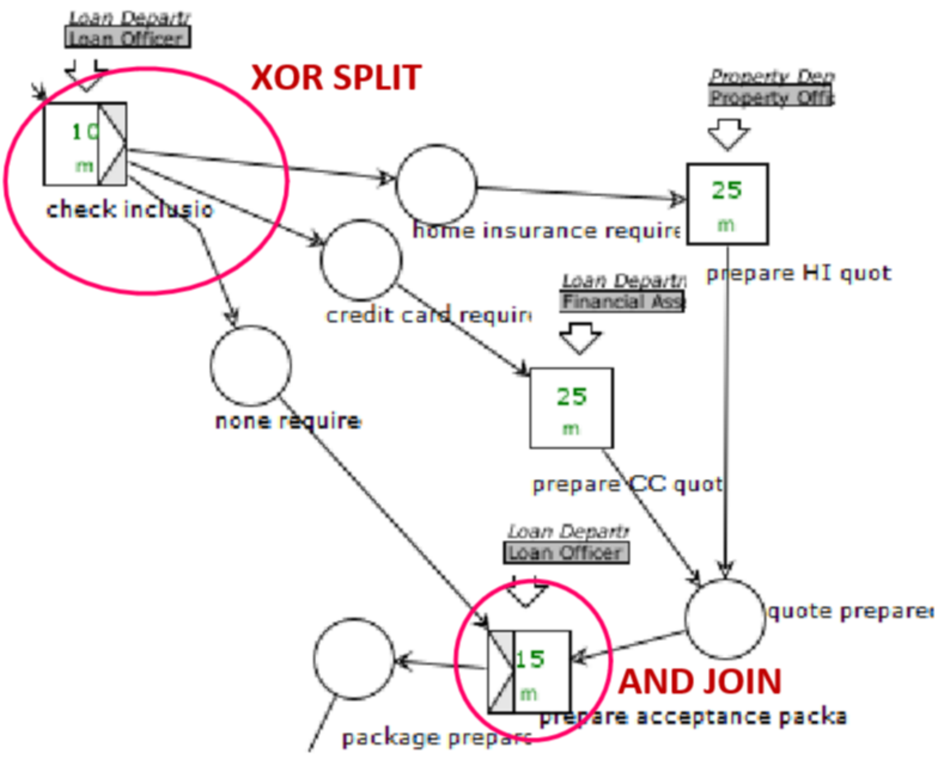


Figure 4: No option to complete

This creates a situation where only **one** token exits the XOR split, but the AND join will keep on waiting for **two** tokens from its incoming places. A token game simulation can illustrate this first:

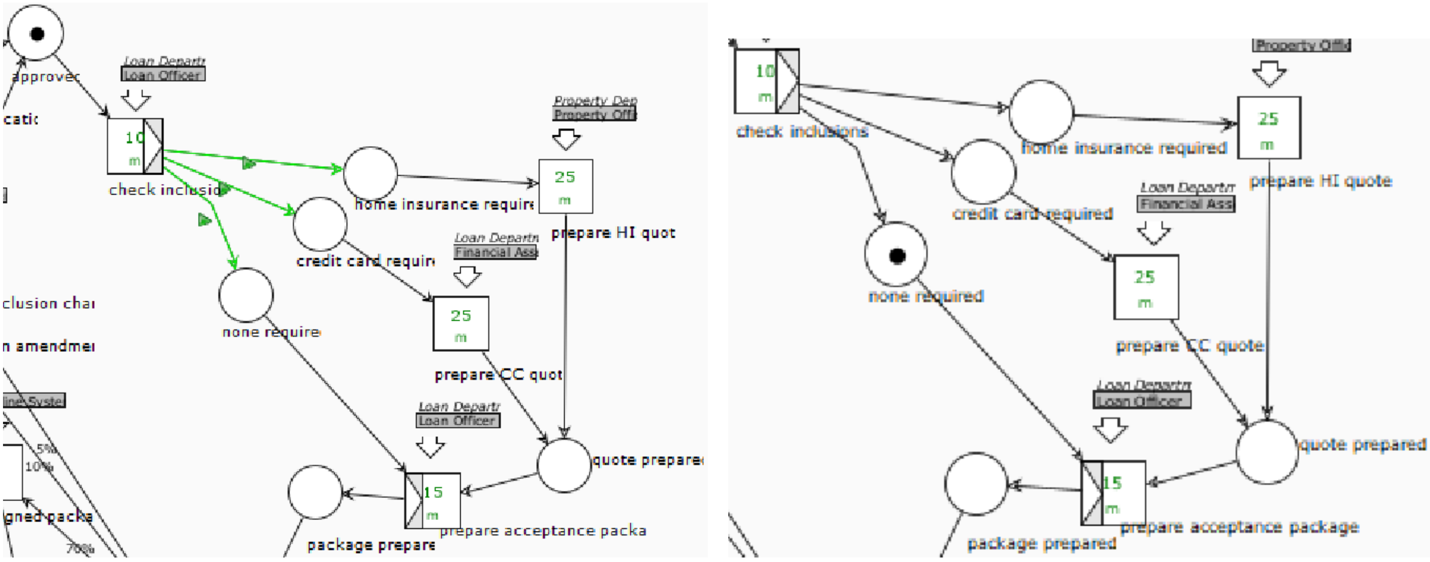


Figure 5: Token game example

It is clear that the "prepare acceptance package" task cannot be completed. The reachability graph confirms this (see right).

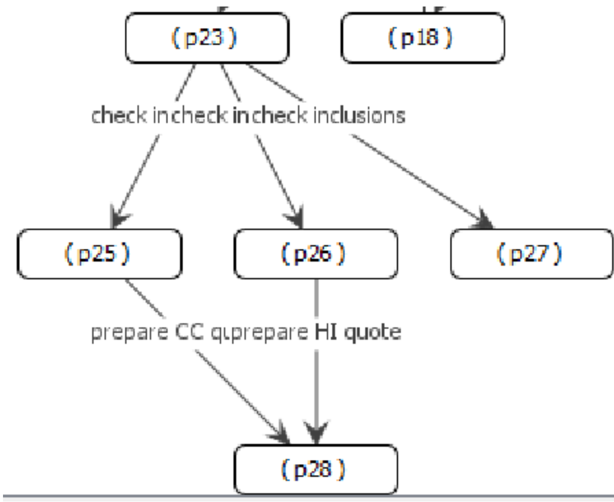


Figure 6: Reachability snapshot

There are two "ending" markings in this broken model. One is p27 (1 single token in the none required place) which was shown above. The other possibility is 1 single token in p28 (the quote prepared place), which is also just before the AND join.

This is clearly violating the soundness of the WF-net because a WF-net should not have two ending markings, just one. And these markings are definitely not the correct end marking.

That is why it is important to make sure that a split and join remains in pattern, that's why I have used an XOR join in the original model.

3.2 Example 2: Violation of 'Proper completion'

Violating proper completion means that the process model does not maintain a synchronisation of a particular case. Like here, it may mean accidentally creating an extra token for the applicant and not making sure it is consolidated before the end of the process (so more than one token for the applicant ends up in the final place). A model can demonstrate how this might happen, please see Figure 7 on the right.

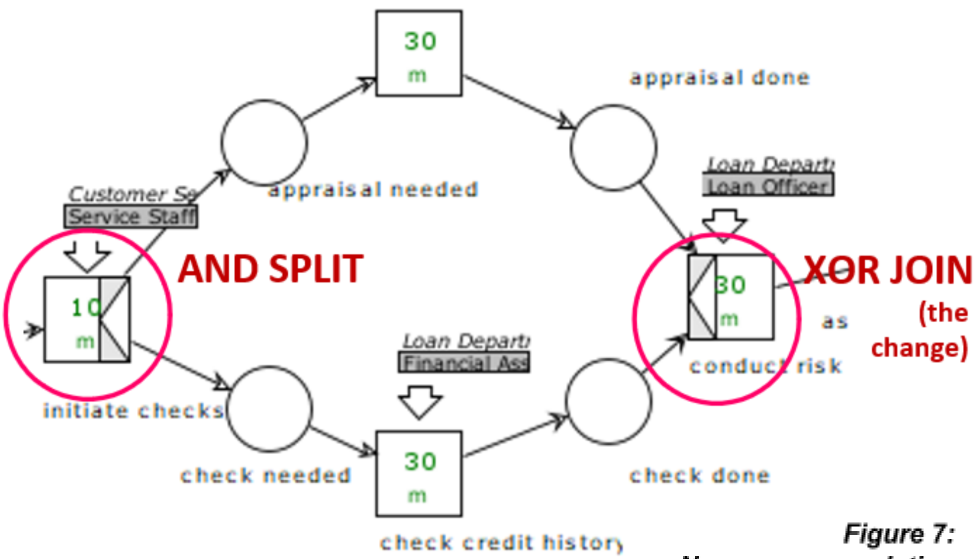


Figure 7: No proper completion

This has a problem because the AND split will first spit out **two** tokens into the next places, but then after those tasks are completed, the XOR join will simply pass each token forward in the model which means that later there will still be **two** tokens for that applicant. Figure 8 below shows this.

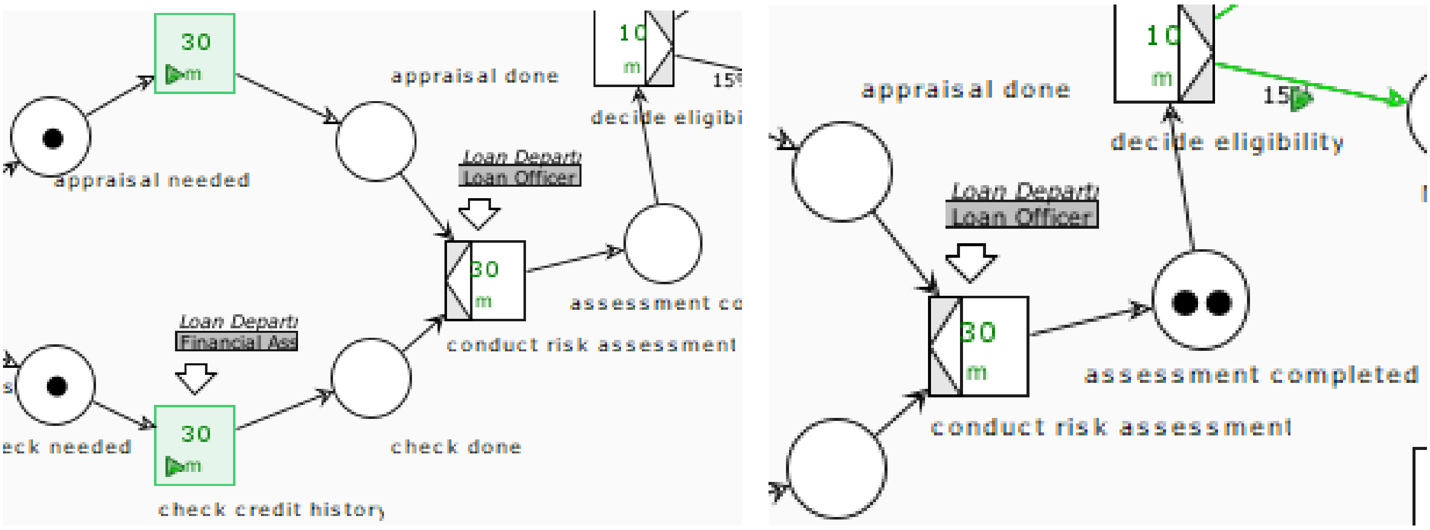
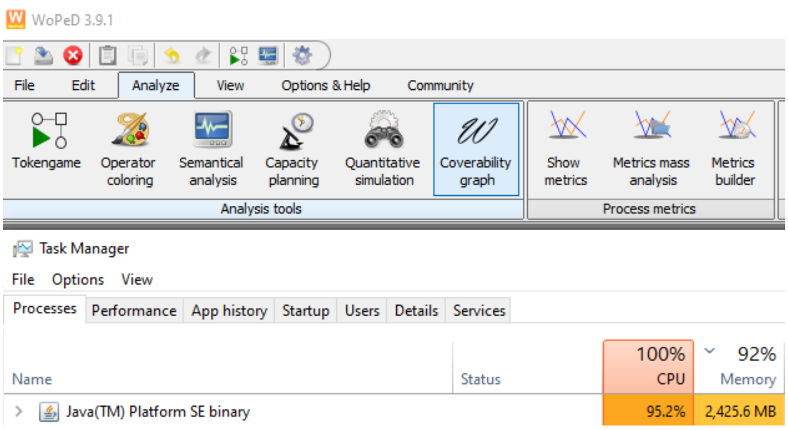


Figure 8: Token game example

Those two tokens can then continue on in the model, but even worse is that there is a possibility that if a rework loop is activated "revision required" (probability 5%), up to infinite tokens could be generated by this wrong process. See Figure 9 on the right for the continued token game. This is a significant behavioural problem in the model. In fact, I tried to make a coverability graph using WoPeD but it caused the program to hang without warning (below).



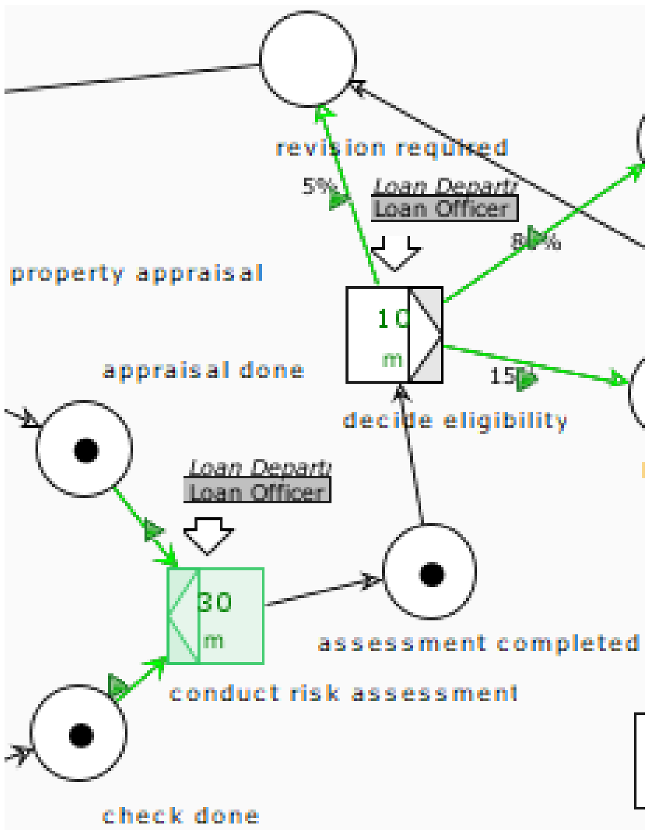


Figure 9: Crashing coverability graph and continued token game

The exact same thing happened when I tried to do a Semantic analysis on the net - WoPeD freezes. The freezing is most possibly the result of the infinite token problem and further proves that the model is not behaving correctly. Therefore, it is important once again to make sure that gateways are matching where appropriate to prevent behavioural issues like that. That is why I made sure that I had a matching AND join with my AND split and it was controlled so that nothing can interfere and come in between the middle part of the gateway procedure.

3.3 Example 3: Violation of 'No dead transitions'

This property refers to how every transition (task) in the process should be 'fireable'. There should never be a situation where a task cannot be completed in the process. To demonstrate this, I have added an extra "bad" place going into a transition which means that it technically is an "and-join" task and needs a token from each of its previous places, see Figure 10a below.

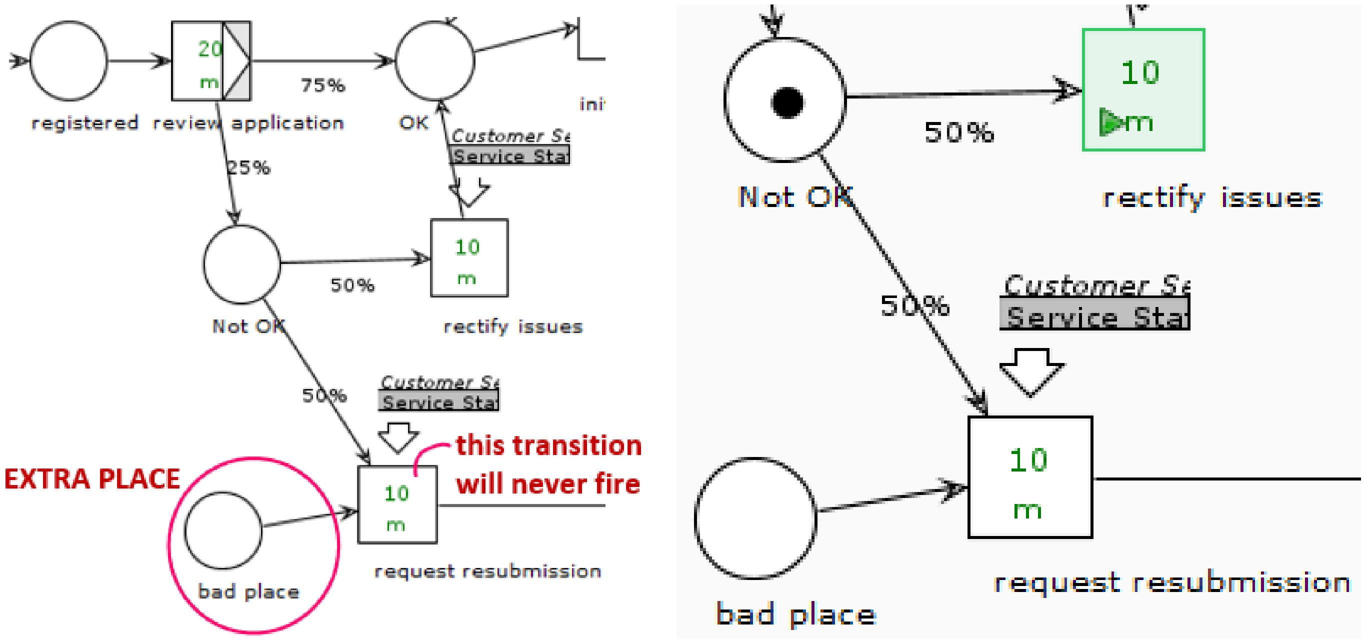


Figure 10a: Dead transition due to a bad place, and token game proving this

Running the token game clearly shows that the bottom "request resubmission" task is dead and will never be possibly completed in the process execution. The semantic analysis on WoPeD proves this, Figure 10b:

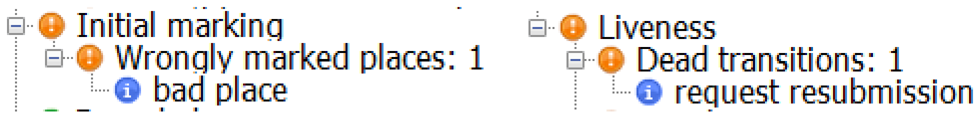


Figure 10b: Semantic analysis

The other issue that is picked up is that bad place is an extra initial marking which is not allowed in WF-net standards. The focus is the right-hand issue, the dead transition of request resubmission. By not having any places that are not appropriate for transitions, I have prevented such situations in my proper WF-net.

3.4 Verification Summary

To conclude, when my process was finally complete and I had made manual checks, I also used the WoPeD semantic analysis tool to double check that my model really is structurally correct and sound, which was confirmed by the tool as you can see below in Figure 11.

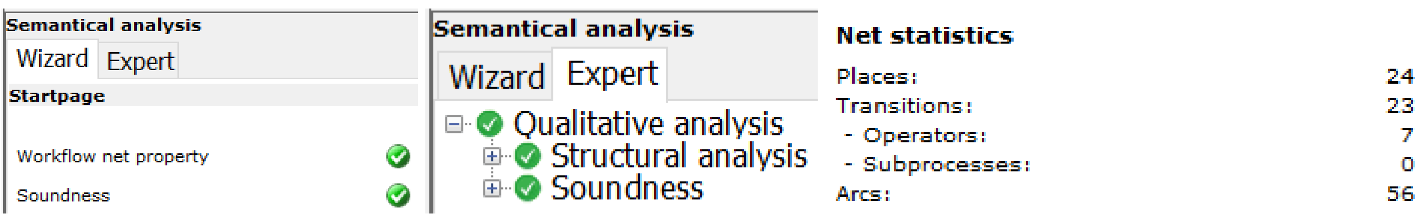


Figure 11: Semantic analysis on my final WF-net

4.0 Process Capacity Planning

By analysing the average time it takes to perform each task and overall each case, and also understanding the capacity requirements for resources within KBANK, process capacity planning can produce some very valuable insights. As discussed earlier, and as visible in Figure 12 on the right, there are 12 human staff members in the organisation along with 1 system for the sake of this process. It is assumed that initially all of these staff members are <u>only</u> responsible for doing work on this specific business process, and nothing else apart from standard admin activities and work breaks. This allows an initial high resource allocation. An 8 hour standard work day is assumed and will be consistently applied to both scenarios.

What will be most interesting to look at is when two parameters are changed:

1. The number of arrivals per day
2. The resource allocation percentage

The main idea is to try to engage in some stress testing to see how KBANK's resources may behave throughout and whether it is sufficient or not.

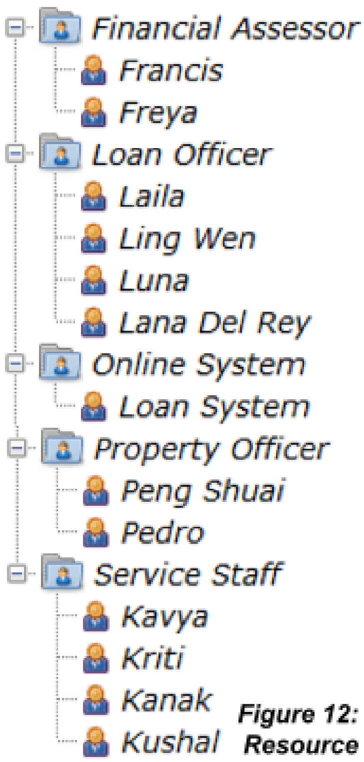
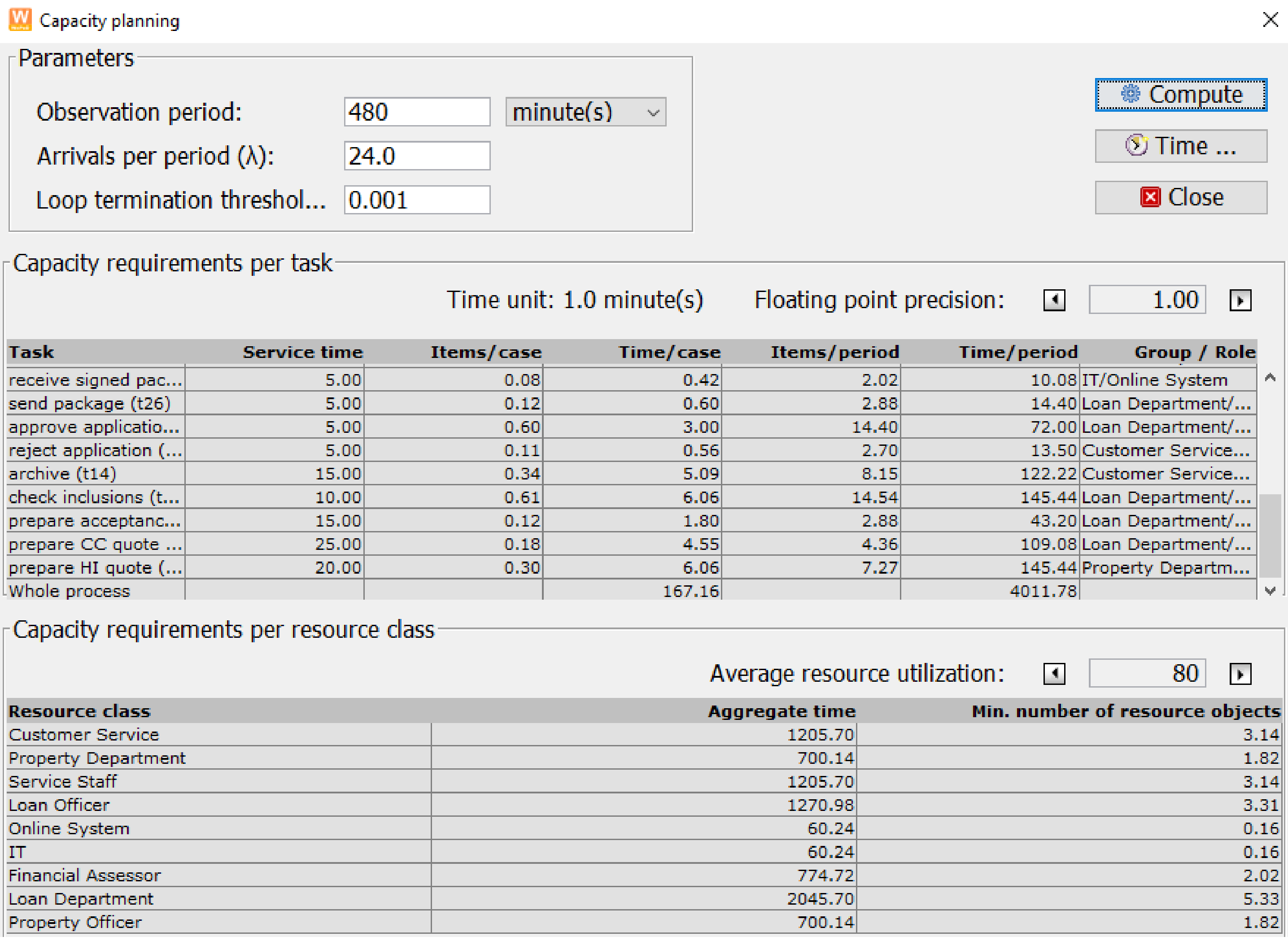


Figure 12: Resource distribution

4.1 Scenario 1: **24** arrivals per 8 hour day, **80%** resource utilisation



Task	Service time	Items/case	Time/case	Items/period	Time/period	Group / Role
receive signed pac...	5.00	0.08	0.42	2.02	10.08	IT/Online System
send package (t26)	5.00	0.12	0.60	2.88	14.40	Loan Department/...
approve applicatio...	5.00	0.60	3.00	14.40	72.00	Loan Department/...
reject application (...	5.00	0.11	0.56	2.70	13.50	Customer Service...
archive (t14)	15.00	0.34	5.09	8.15	122.22	Customer Service...
check inclusions (t...	10.00	0.61	6.06	14.54	145.44	Loan Department/...
prepare acceptanc...	15.00	0.12	1.80	2.88	43.20	Loan Department/...
prepare CC quote ...	25.00	0.18	4.55	4.36	109.08	Loan Department/...
prepare HI quote (...	20.00	0.30	6.06	7.27	145.44	Property Departm...
Whole process			167.16		4011.78	

Resource class	Aggregate time	Min. number of resource objects
Customer Service	1205.70	3.14
Property Department	700.14	1.82
Service Staff	1205.70	3.14
Loan Officer	1270.98	3.31
Online System	60.24	0.16
IT	60.24	0.16
Financial Assessor	774.72	2.02
Loan Department	2045.70	5.33
Property Officer	700.14	1.82

Figure 13: Capacity Planning Scenario 1

Table 1: Scenario 1, key summary across the five resource roles

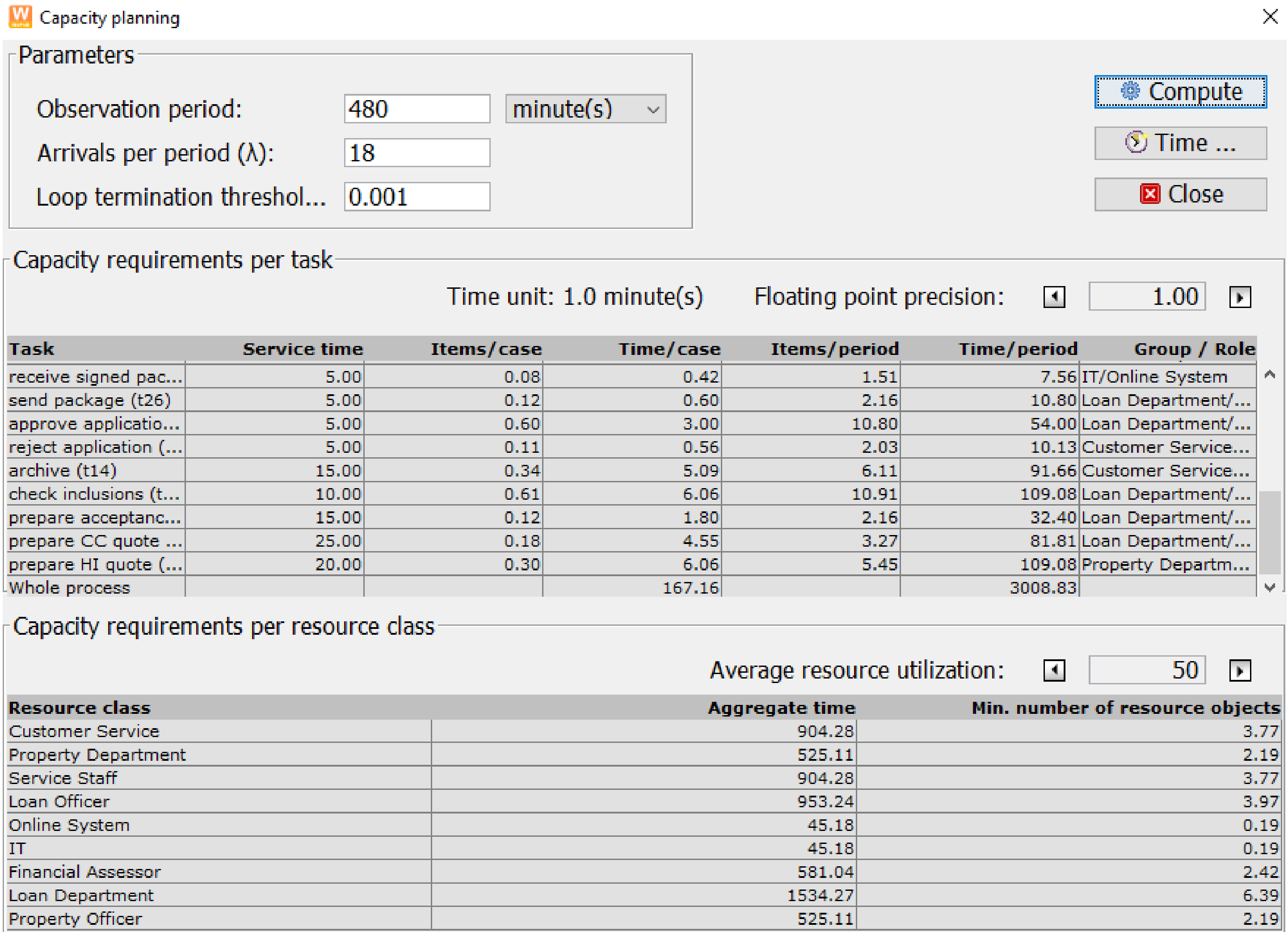
	Financial Assessor	Loan Officer	Online System	Property Officer	Service Staff
Min resource objects	2.02	3.31	0.16	1.82	3.14
# of staff	2	4	1	2	4

The first scenario considers a situation where KBANK receives a new loan application every 20 minutes as they are a local branch (which equals to 24 applications per 8 hour day). Meanwhile, the resource utilisation is set to 80%. The average time it takes per case is 167 minutes (2 hours, 47 minutes). The most resource-intensive tasks are:

- Conduct risk assessment (810 minutes per period)
- Check credit history (666 minutes per period)
- Review application (600 minutes per period, since this is always averaged at 20 minutes per case)

Fortunately, each of the above are performed by separate roles, respectively, Loan Officer, Financial Assessor and Service Staff. The main concern is the minimum resource objects required for the process. While most are within the range with a sensible margin, a concern is how the Financial Assessor is alarmingly just above the current staffing.

4.2 Scenario 2: **18** arrivals per 8 hour day, **50%** resource utilisation



Capacity planning

Parameters

Observation period: 480 minute(s)

Arrivals per period (λ): 18

Loop termination threshol... 0.001

Compute

Time ...

Close

Capacity requirements per task

Time unit: 1.0 minute(s) Floating point precision: 1.00

Task	Service time	Items/case	Time/case	Items/period	Time/period	Group / Role
receive signed pac...	5.00	0.08	0.42	1.51	7.56	IT/Online System
send package (t26)	5.00	0.12	0.60	2.16	10.80	Loan Department/...
approve applicatio...	5.00	0.60	3.00	10.80	54.00	Loan Department/...
reject application (...	5.00	0.11	0.56	2.03	10.13	Customer Service...
archive (t14)	15.00	0.34	5.09	6.11	91.66	Customer Service...
check inclusions (t...	10.00	0.61	6.06	10.91	109.08	Loan Department/...
prepare acceptanc...	15.00	0.12	1.80	2.16	32.40	Loan Department/...
prepare CC quote ...	25.00	0.18	4.55	3.27	81.81	Loan Department/...
prepare HI quote (...	20.00	0.30	6.06	5.45	109.08	Property Departm...
Whole process			167.16		3008.83	

Capacity requirements per resource class

Average resource utilization: 50

Resource class	Aggregate time	Min. number of resource objects
Customer Service	904.28	3.77
Property Department	525.11	2.19
Service Staff	904.28	3.77
Loan Officer	953.24	3.97
Online System	45.18	0.19
IT	45.18	0.19
Financial Assessor	581.04	2.42
Loan Department	1534.27	6.39
Property Officer	525.11	2.19

Figure 14: Capacity Planning Scenario 2

Table 2: Scenario 2, key summary across the five resource roles

	Financial Assessor	**Loan Officer**	**Online System**	**Property Officer**	**Service Staff**
Min resource objects	2.42	3.97	0.19	2.19	3.77
# of staff	2	4	1	2	4

This second scenario now considers a situation where the bank is facing tougher times. This might be due to increased interest rates and fewer people being able to afford a home loan anymore. So I have devised a situation where there is a 25% drop in new applications, to 18 applications per 8 hour day (just 2.25 per hour). Because of these economic conditions, the bank decides to reduce their staff's hours (or stand down some staff) and this can be simulated by reducing the average resource utilisation from 80% to 50%. Those are the only two changes, but the impacts are quite drastic.

Considering table 2 above, it is immediately evident that the Financial Assessor role is overstrained by over 20%. Even the Property Officer is overloaded by 10%. Both of these situations, when compared to Scenario 1, means that there is going to be a greater delay during the process and a knock-on effect. This is because both the Financial Assessor and the Property Officer both need to do tasks earlier on (the credit history and property appraisal checks), but also later on if a credit card or home insurance quote is desired. Delays in the process will only continue to escalate over time unless there is an intervention to artificially reduce the number of cases taken per day, or ideally, increasing resourcing to handle the shortcoming.

4.3 Comparing the scenarios: findings and recommendations

In short, the findings are:

- Staff deficit for an optimal process flow is evident for Financial Assessors, which is further heightened in the 2nd scenario which also affects the Property Officer.
- The most underutilised role, apart from the Online System, is the Service Staff in both scenarios.

Now, there are of course some limitations In using WoPed for this analysis. Firstly, it needs to be consistent time units across all tasks to make sure it calculates correctly. More critically, the resource utilisation rates can only be set for all of the resources at once (that it is not possible to adjust them for individuals). Having the ability to do that would have been very useful to consider using some resources more than others (as discussed above, the financial assessor and property officer are first to get strained and may need higher resource allocation). That would be the chief recommendation here, that the financial assessors and property officers staff resource utilisation remains higher, because reducing the staff hours here would be a dangerous move. Another "close call" here is the loan officer. It has reached, in Scenario 2, very close to crossing the staff availability, just a touch below 4, at 3.97. The recommendation here would be to make sure that the loan officer staffing is also continuously reviewed to make sure there are backups here, especially since on the basis of aggregate time, the Loan Officer needs to perform the greatest amount of work (1271 minutes in Scenario 1, and 953 minutes in Scenario 2, per day).

Another recommendation could be to upskill the service staff to assist with some of the other work to ensure they are being used efficiently, but the viability of this would need to be investigated including cost of training.

5.0 Process Simulation

Now that it is time to run a quantitative simulation for two realistic business scenarios which KBANK might experience.

5.1 Simulation 1: Complete as many cases within 8 hours - 10 simulation runs

The first simulation I had initially crafted is to simply complete an average of 24 cases per 8 hour day, similar to the situation described in section 4.1 earlier. I have chosen 10 simulation runs and ticked both condition boxes inside the termination rule:

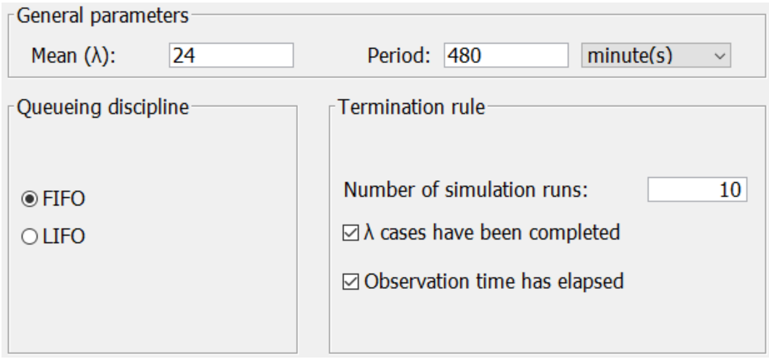


Figure 15: General parameters for Simulation 1

The interarrival time distribution and service time distribution have been kept consistent as their default values.

Figure 16 shows an overall summary of the process in this simulation, where the time ran out (480 minutes) before 24 cases were completed. According to the results, 11 cases were finished. The throughput rate was 11.38. The process completion time was approximately 196 hours, where 172.52 of hours were actual servicing time and 24.37 hours were spent waiting. These are reasonable figures which indicate that the process performs acceptably well, but it is clear that not that many cases actually get completed within a day.

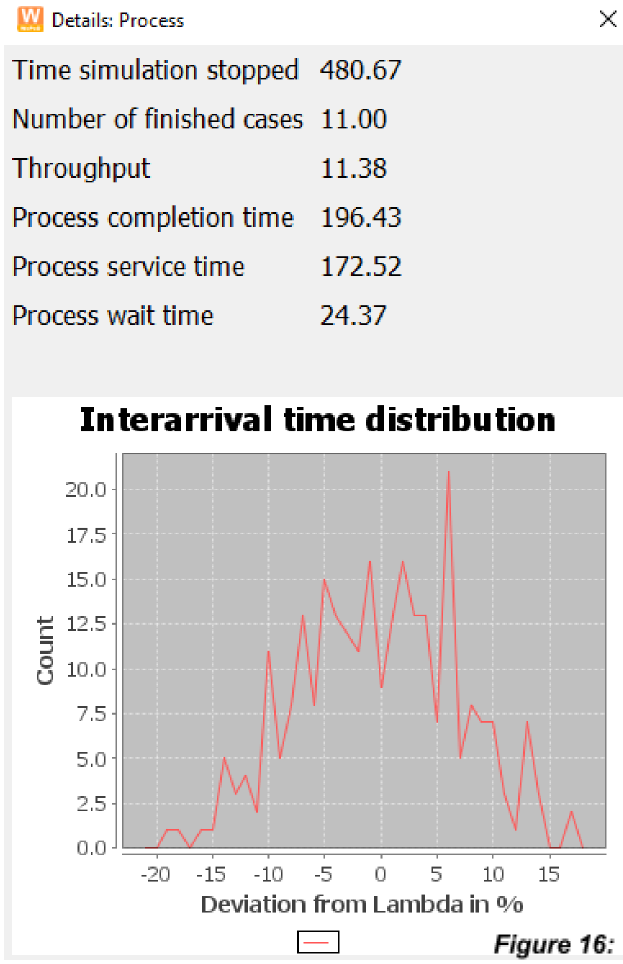


Figure 16: Simulation 1 summary

Resource utilization

Resource object	Utilization (%)
Kriti	62.90
Peng Shuai	57.98
Kandy	0.00
Kavya	60.80
Kushal	59.77
Loan System	17.99
Kanak	57.61
Freya	65.15
Ling Wen	66.42
Lana Del Rey	61.89
Francis	64.61
Laila	65.15
Luna	63.30
Pedro	62.00

Figure 17: Simulation 1 resource utilisation

According to Figure 17, the resource utilisation of the staff members falls within the range of 57% to 67%, with the outliers being Kandy (who is a spare employee that does not do any work, maybe the CEO) and the Online Loan System that only is utilised 18% of the time in this process. The remaining figures are sensible but it seems that underutilisation is quite rife, as the resources could be used better. The most utilised resource, which is Ling Wen, is a Loan Officer. Meanwhile, the most underutilised non-outlier is Kanak (Service Staff). But the next most overutilised and underutilised come from different roles, so there is no distinct pattern here to derive an insight from.

Name	#Cases	λ	ExecTime	ServiceTime	WaitTime	Details
prepare HI quote (t30)	0.96	7.6	25.04	19.82	5.23	...
decide eligibility (t12)	1.19	18.0	15.15	9.89	5.26	...
approve application (t13)	0.92	14.4	9.55	5.04	4.52	...
conduct risk assessme...	2.46	18.0	48.51	45.44	3.06	...
reject application (t15)	0.29	2.7	6.10	4.98	1.13	...
register (t17)	0.73	24.0	2.23	2.00	0.23	...
request revision (t21)	0.38	2.4	5.92	3.97	1.96	...
prepare acceptance pa...	1.16	15.1	19.77	14.94	4.83	...
rectify issues (t24)	0.17	3.0	30.38	30.38	0.00	...
check inclusions (t27)	1.05	15.2	15.10	9.98	5.12	...
send package (t26)	0.98	15.1	10.75	5.02	5.73	...
prepare CC quote (t29)	0.90	4.5	32.10	24.50	7.60	...
review application (t1)	1.93	24.0	25.07	24.92	0.14	...
receive signed packag...	0.45	10.6	5.12	4.98	0.14	...
perform property appr...	1.89	23.4	28.87	24.93	3.93	...

Figure 18: Simulation 1 process statistics including tasks

According to Figure 18, there are certain tasks which seem to be the most susceptible to increased waiting times. The biggest one here is the prepare credit card quote task, which must wait for a loan officer to not only approve the application but also check the inclusions first. The "prepare home insurance" quote is also affected in this same region. **A strong recommendation here would be to combine both tasks,** in other words the loan officer should check the inclusions while approving the application. This should save some time during the process and reduce friction. **Another recommendation** to consider is the "prepare acceptance package" task which is also commonly held up due to waiting, to actually **conduct it in parallel** to whichever "prepare quote" task takes place. For example, if the applicant wants a credit card, the Financial Assessor will prepare that credit card quote *while* the Loan Officer will prepare the acceptance package. Then simply, the quote can be attached to the package and this reduces having to wait at all for the task.

5.2 Simulation 2: Time it takes to complete 30 cases - 3 simulation runs

The second designed simulation considers how long it takes KBANK to complete 30 cases, with no early cut off due to time. I have chosen 3 simulation runs to circumvent needing to change the loops or unstructured patterns in the net, otherwise it will not be an authentic representation of the process modelled. I have also adopted a different queueing discipline, moving from FIFO in the previous simulation to LIFO for this one. The simulation will only end when 30 cases are completed.

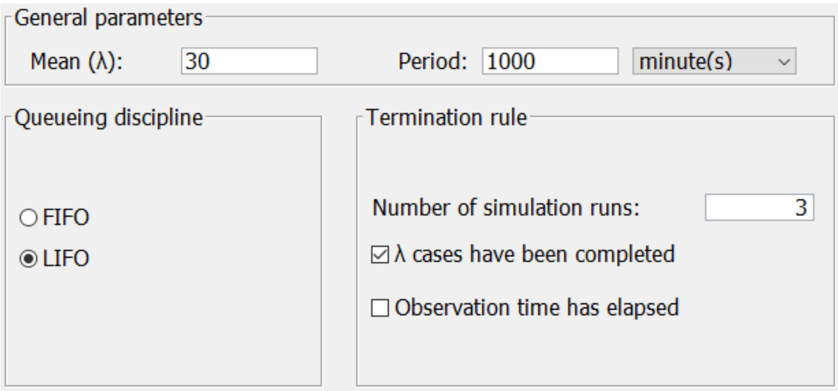


Figure 19: Simulation 2 general parameters

It will be interesting to see whether this leads to an increase in the efficiency of the resources inside KBANK.

This is basically an **increased demand simulation**, because the lambda parameter has increased from 24 to 30 (this represents a 25% increase). Figure 20 illustrates the overview of the results of this simulation. The most noteable figure to first consider is the time the simulation stopped, which is 1181 minutes (just under twenty hours: 19.68 hours). It is definitely not possible for KBANK to cope with this kind of demand in an 8 hour day, they need about 2.5 working days to process 30 loan applications (an average of just 12 loans finalised per day). It is very clear that an immediate conclusion that might be jumped to here, would be for KBANK to engage in a hiring round if they expect a persistent demand increase. However, it is more complex than that, because it might be more beneficial for business process optimisation efforts first to increase the efficiency of tasks (as discussed in 5.1, to consolidate tasks or parallelise tasks could reduce wait times). Further investigation is needed here.

Looking at the interarrival time distribution in comparison to Simulation 1, there seem to be more frequent negative deviations from the average which is expected since it has gone up. However, the impact has not been catastrophic.

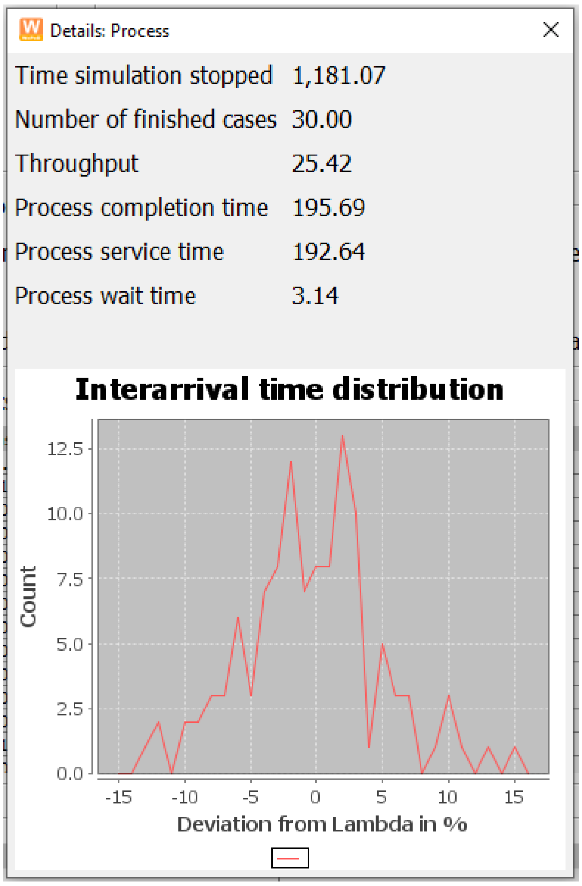


Figure 20: Simulation 2 summary

Next the utilisation statistics of the resources can be explored, see Figure 21.

Resource utilization

Resource object	Utilization (%)
Kriti	44.62
Peng Shuai	47.69
Kandy	0.00
Kavya	45.98
Kushal	40.37
Loan System	14.27
Kanak	40.26
Freya	54.77
Ling Wen	54.95
Lana Del Rey	56.16
Francis	46.45
Laila	55.96
Luna	54.29
Pedro	41.25

Figure 21: Simulation 2 resource utilisation

Most shocking is the drop in resource utilisation efficiency despite the fact that the case average was increased! But it is clear that before in Simulation 1, the minutes per case completed were 481/11 = 43.73 minutes. In Simulation 2, on average 1181/30 = 39.37 minutes were spent on each case completed. Although this represents an increase in case efficiency, especially when looked at with the higher Simulation 2 throughput rate of 25.42, it still means that most staff are only being used on average between 40% to 56% of the time. This is probably due to the sample size of the simulations being low (only 3). Despite this, the recommendation would still remain the same from Simulation 1: there is clearly room for KBANK to restructure the process better and this would make more sense before hiring new staff, as existing resources can be used more efficiently if they were given the opportunity to do so.

6.0 Conclusion

6.1 Summary of key insights/findings/recommendations

This project enabled me to gain a deeper understanding of a sample loan application process in the financial industry. It also provided me with the knowledge that even a perfect business model that works efficiently in a well-developed market may not necessarily work the same way in a less developed and challenging market. Adjustments may need to be made based on the market dynamics to ensure the process works effectively. Another important insight that I gained from this project is that a clear understanding of the process is paramount when modelling any process. Without a clear understanding, the outcome may be incorrect, as I experienced during the first attempt at modelling the application process where logic issues were encountered.

There are a variety of useful methodologies and tools available, such as WoPed, to create diagrams, flow charts, or visual interfaces. WoPed is a beneficial tool to model business processes and evaluate them in different dimensions to improve efficiency, productivity, and performance. Constructing a workflow net in WoPed and activating the token game helped me but also the coverability graph to gain a better understanding of control flows and the real processes at work, despite having little experience or background in this domain. On the other hand, those with process knowledge and background can appreciate the value of using such tools to enhance processes effectively. The semantic analysis also quickly helped me ensure that my model was sound.

The capacity planning simulation in WoPed provides valuable insights into the efficiency of the processes and is helpful in forecasting the utilisation of human and non-human resources involved in the process to appropriately manage them and deliver expected outcomes without blockages. Meanwhile, the quantitative simulation facility provides an understanding of the time taken to perform each task from start to finish. Proper analysis on this quantitative simulation can help improve the efficiency of individual tasks performed.

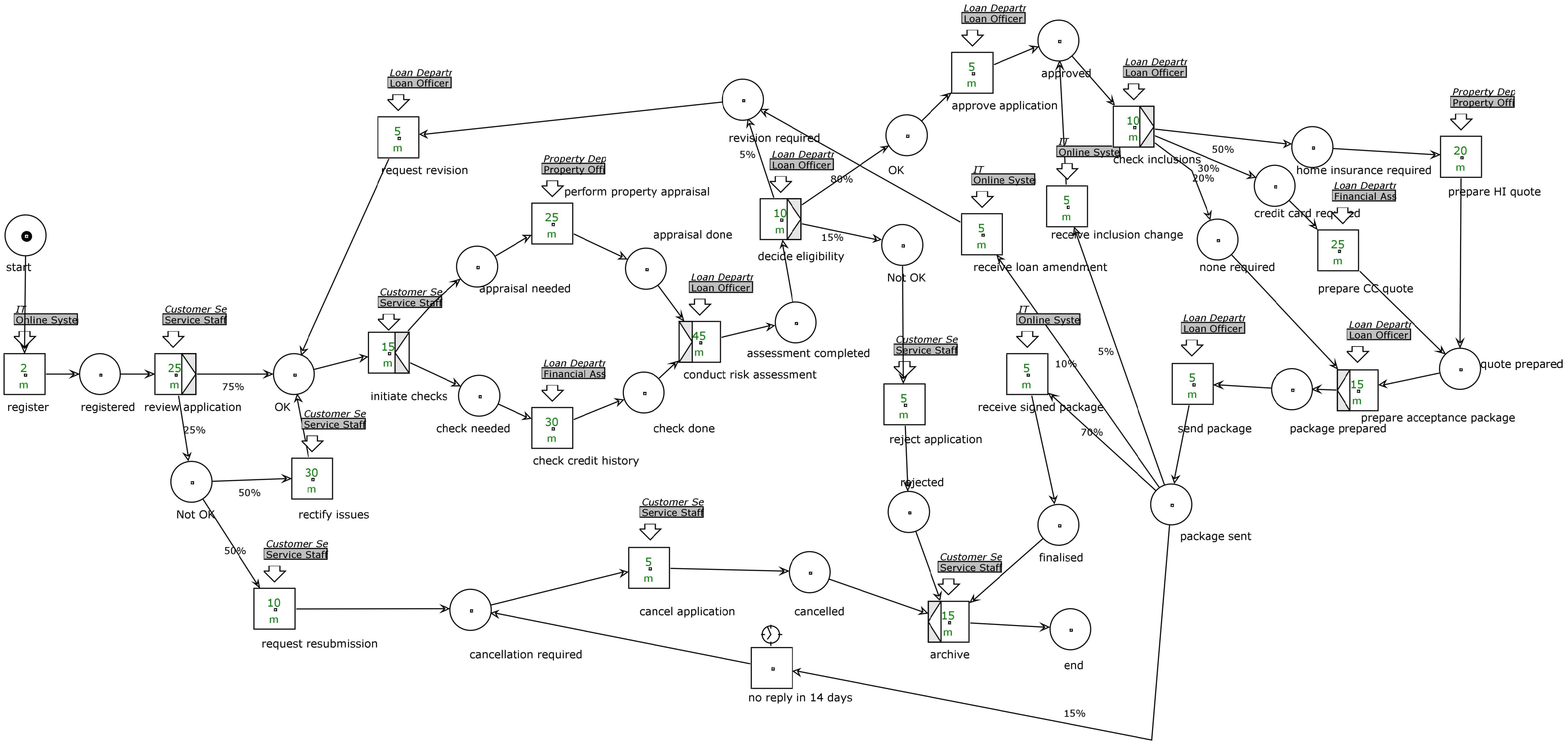
It is very evident that through the effective use of the WoPed tool, an individual or company can appropriately analyse, then evaluate, and eventually remodel a process to achieve improved outcomes, reducing their costs and waiting times, and generally eliminating wastes, thereby adding an increased value to their process. For KBANK, the key recommendation is to engage in Business Process Improvement initiatives to better restructure their process first to streamline it and reduce bottlenecks, rather than jumping to hire additional staff or contracted resources.

6.2 Challenges encountered

The WoPed tool sometimes lacks user-friendliness and does feel like it is some dated early 2000s software. Dragging elements onto the process diagram plane and especially connecting arrows can be counterintuitive. WoPeD also tends to freeze frequently while creating models and conducting the simulation exercises, which causes a great deal of frustration and disruption. Sometimes when trying to exit the application, it keeps saying there are unsaved changes even though I have already saved. Yet the tool must be restarted every time it freezes, which causes unnecessary interruptions and the risk of losing any unsaved work. However, despite this, WoPeD is a fantastic tool given that it is open-source and has a powerful array of features and it is actually quite straightforward to pick up.

7.0 Appendix

Large scale model attached below.



start
register
registered
review application
OK
Not OK
rectify issues
request resubmission
request revision
initiate checks
appraisal needed
check needed
perform property appraisal
appraisal done
check credit history
check done
conduct risk assessment
assessment completed
decide eligibility
revision required
OK
Not OK
approve application
approved
check inclusions
home insurance required
credit card required
none required
prepare HI quote
prepare CC quote
quote prepared
prepare acceptance package
package prepared
send package
package sent
receive loan amendment
receive inclusion change
receive signed package
finalised
reject application
rejected
cancel application
cancelled
cancellation required
no reply in 14 days
archive
end
75%
25%
50%
50%
5%
80%
15%
50%
30%
20%
5%
10%
70%
15%