2II55 Business Process Management Systems

Final Report

Tim van Dalen (0744839) Bram Kohl (0746107) Jasper Selman (0741516)

March 29, 2015

Abstract

This document gives the results of the 2II55 Business Process Management Systems project. We describe our model, how it uses the patterns discussed in the lectures and give simulation results. We show the changes we made to our model and how these affect performance.

1 Model 1

1.1 Description

In this section we describe in detail how we model the description of the exercise, as given in Appendix A. To make this description easier to understand we will use figures of parts of the model to show how we modeled the process, but it should be possible to understand this without the figures. A complete overview of the process modeled in Bizagi can be found in Appendix B.

At the start of the process a student requests an application to study in Eindhoven. This is the start event in our process. From this start event we move on to the first task which is called *Register request*. In this task an administrative employee handles the request from the student (by entering it in the system of the TU/e) and the process moves on to two tasks which happen simultaneously.

The first task is done by a mail department employee, which is gathering all the forms which have to be sent, disregarding whether the request was on time or not, to the new student. This task is named *Gather standard forms*. The second simultaneous task was that a mail department employee checks if the request was made on time, if not he adds a request for a motivation letter to the package which will be send to the student. If the request was on time he does nothing but communicates that the request was on time. The name of this task is *Add motivation letter request*.

When both these tasks are finished the process goes to the next task, which is *Send enrollment package*. In this task a mail department employee sends the package to the student. The next task is *fill out and send back forms*. In this task the student has to fill out the forms he got and send them back to the university. If this is done within two weeks the process moves on to the next task. If this is not done in two weeks, the whole process ends and the request expires. This is modeled by a timer on the task *fill out and send back forms*, if this timer expires the process continues to an end event.

If the forms were send back on time the next task is *Check if forms are complete*. In this task an administrative employee checks if all the forms are filled in. If this is not the case the process

goes to the task *Add explanation missing items*. In this task the mail department employee adds the explanation for the missing items to the package which will be send again. The process now continues to the point right after that the request was registered in the system. This also means that a mail department employee checks if the request is still on time or that in this new case a motivation is required. This is because it is possible that a students first request was just in time for the deadline, but because it is two weeks later now and he has to do it over again the deadline has passed.

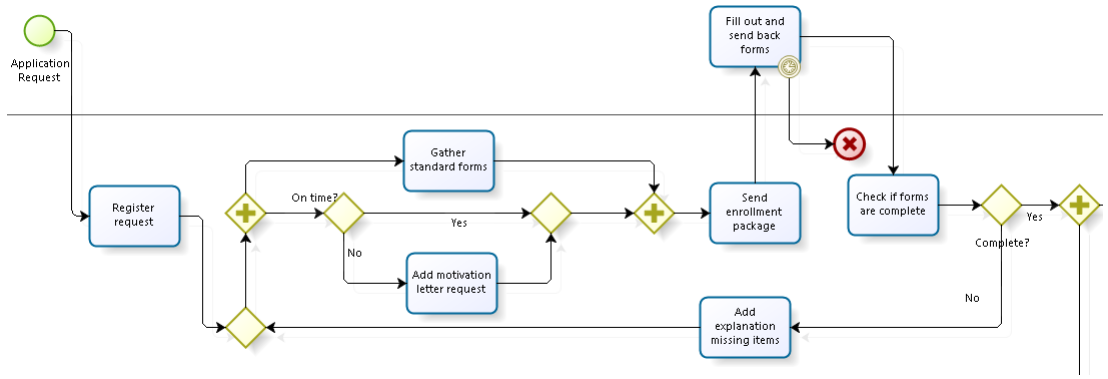

Figure 1: The first part of the model in Bizagi

If the forms were all complete the process continues and up to three tasks are done simultaneously (depending if a motivation letter was required). The first task is *Check prerequisites*, in this task a administrative employee checks if all the general prerequisites are met. If so, the process continues but has to wait until all three tasks are finished. If the prerequisites are not met the next task in the process is *Notify student*. In this task a mail department employee notifies the student that his request has failed because some of the prerequisites are not met. After this the process has terminated.

The second simultaneous task is *Check financial prerequisites*. In this task a financial department employee checks these prerequisites. If these are not met, the same process flow as with the normal prerequisites is followed and the process terminates. If met, the process has to wait for the (potential) last task.

The last task is optional. This task is *Check motivation letter*. This task needs only to be done if a motivation letter was required, so before a departmental executive board employee checks the motivation letter, an automatic check is done to see if this task needs to be executed. If not the process skips this task and the process can continue if the first two prerequisites checks are completed. If the motivation letter was not good enough the process terminates again via the above mentioned way. If the letter was good the process can continue if all three tasks are completed.

Though in the process description in Appendix A it is mentioned that these tasks take place after each other, we choose to do them simultaneously because they need different resources and do not depend on each other. This choice leads to a shorter process flow time, but a higher percentage of inefficient work. This is because one of the checks might result in a failure. In that case both the other checks should not have been done, but we value the flow time more than the efficiency.

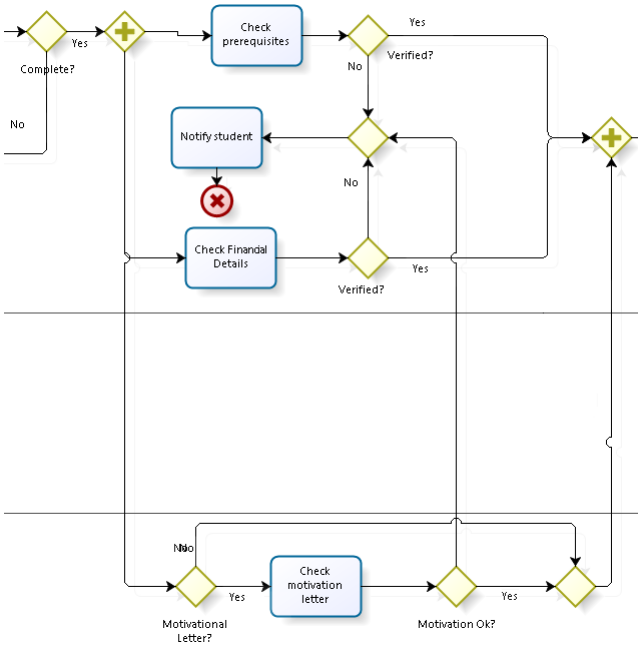

Figure 2: The second part of the model in Bizagi

Once all the prerequisites are checked, the registration of the student cannot fail anymore. At this point the process splits into two flows. The first flow is the final part of the registration of a student and the second flow is the order and payment of the laptop. We have explicitly chosen that a laptop is not ordered earlier than this, because otherwise it would be possible to order a laptop for a student and the registration of that student fails. This would cause that the NSC could order too much laptops which would be expensive.

We shall describe the registration part first. After the prerequisites check the process moves on to *Register student*. In this task the administrative employee registers the student at the university. After that the departmental administrative employee registers the student in the study program of the department corresponding to the chosen study program. This is all done in task *Register student at Department*. At last the student is notified by a mail department employee that the registration is complete via the task *Notify Successful Registration*. Important to note is that once this task is completed it is not possible anymore to pay for the laptop via a bank. When this part of the process is completed, the process has to wait for the other part to finish before the process can terminate.

Now we handle the second part. After the prerequisites were approved the process also continues to one of the tasks *Order separate notebook* or *Add notebook to Bulk order*. Which one of the tasks will be executed depends on whether the registration was on time or not. If it was on time than the task *Add notebook to Bulk order* is executed and otherwise *Order separate notebook* will be done. Both these tasks are done by an employee of the Notebook Service Center, further on mentioned to as NSC employee.

Now the process continues to the task *Notebook arrives*. This is the task where a NSC employee receives the notebook(s). When this task is finished, there are now three possible scenario's. In the first scenario the student wants to pay via a bank and the registration is not yet completed. In this case the process will continue to the task *Payment via bank Student*.

In the second scenario the student wants to pay via a bank, but the registration is already completed. In this case the process continues to the task *Pay Laptop in Cash*, since it is not possible anymore to pay via bank. In the third and last scenario the student wants to pay in cash and the process continues to *Pay Laptop in Cash*. The payment via the bank will be done automatically, but for the payment in cash we also need a NSC employee to receive the money.

After one of these two tasks are completed the process moves on to *Pick up Laptop*. In this task the NSC employee gives the laptop to the new student. After this task, this part of the process is finished and if the registration is also finished the process can terminate, otherwise the process has to wait until the registration is complete.

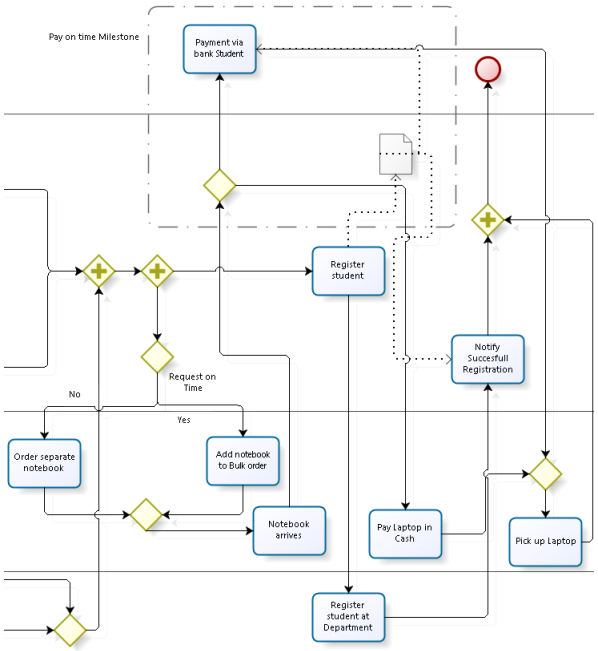

Figure 3: The last part of the model in Bizagi

1.2 Workflow patterns

1.2.1 Parallelism

Figure 4 shows the parallelism pattern used in our model.

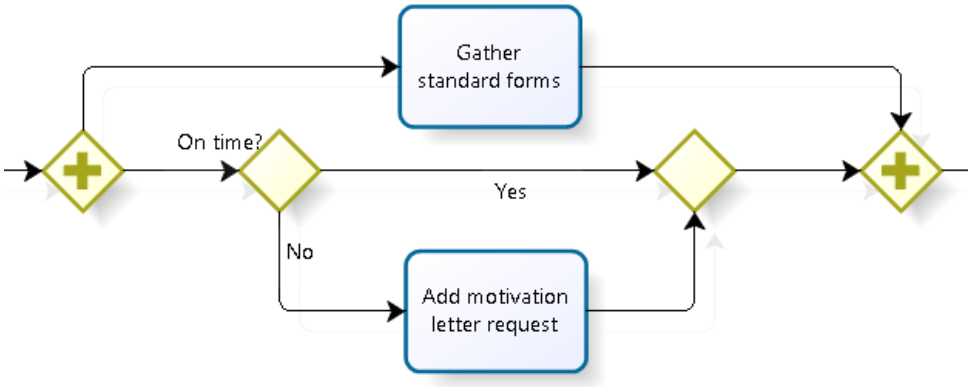

Figure 4: Parallelism workflow pattern

We use parallelism here to decrease the flow time and increase resource utilisation. Since gathering standard forms and adding a letter to the mail package can be done at the same time, the total time required to finish this part of the flow can be lower.

1.2.2 Implicit XOR-split

Figure 5 shows the implicit XOR-split pattern used in our model.

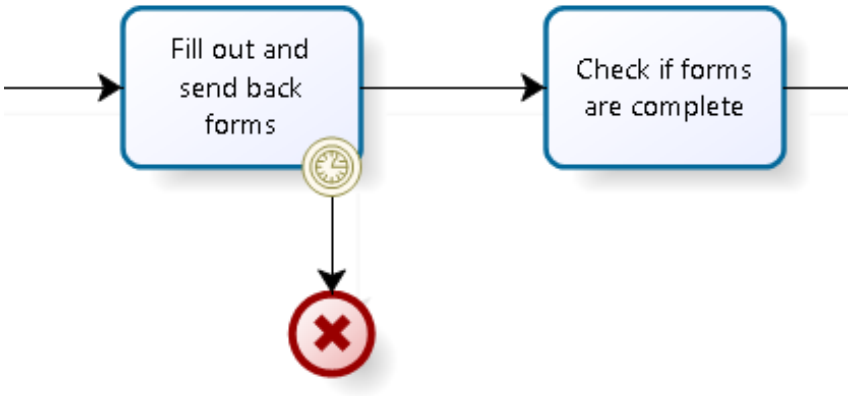

Figure 5: Implicit XOR-split workflow pattern

We use the deferred timer choice to end the model when the applicant does not reply on time.

1.2.3 Explicit XOR-split

Figure 6 shows the explicit XOR-split pattern used in our model.

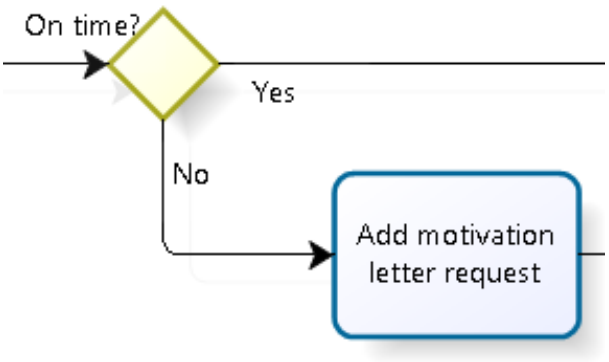

Figure 6: Explicit XOR-split workflow pattern

The explicit XOR-split is used a number of times throughout the model to make direct choices based on user input or state. In this case, we use it do decide whether or not it's necessary to add a request for a motivation letter.

1.2.4 OR-join

Figure 7 shows the OR-join pattern used in our model.

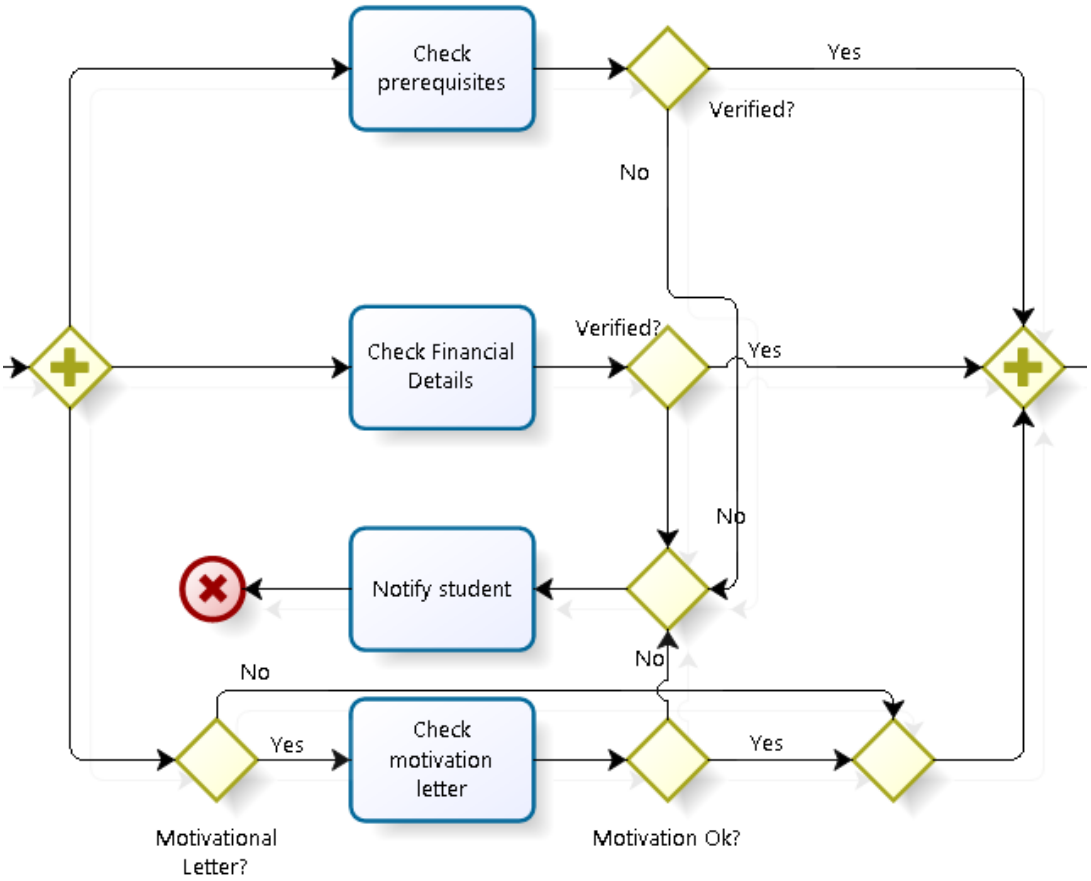

Figure 7: OR-join workflow pattern

We model the OR-join pattern as a parallel split with choices. In Figure 7, we make use of this pattern for the different types of checks on the applicant.

1.2.5 Iteration

Figure 8 shows the iteration pattern used in our model.

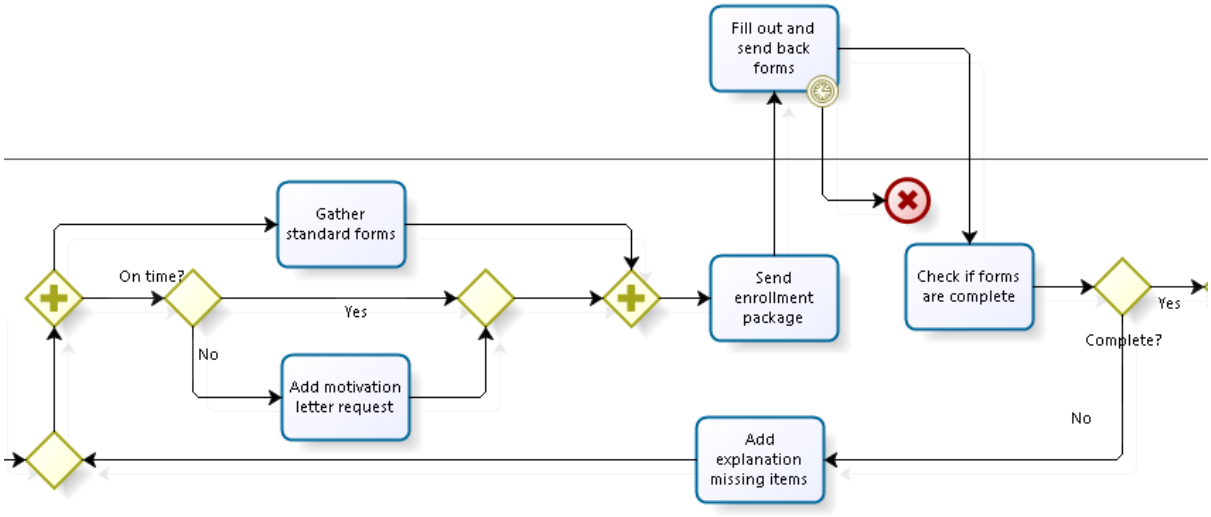

Figure 8: Iteration workflow pattern

Our implementation of the iteration pattern is a simple loop that will repeat the information gathering part of the model. If an applicant does not fill out the forms completely, we step back to sending the enrollment package.

1.2.6 Milestone

Figure 9 shows the milestone pattern used in our model.

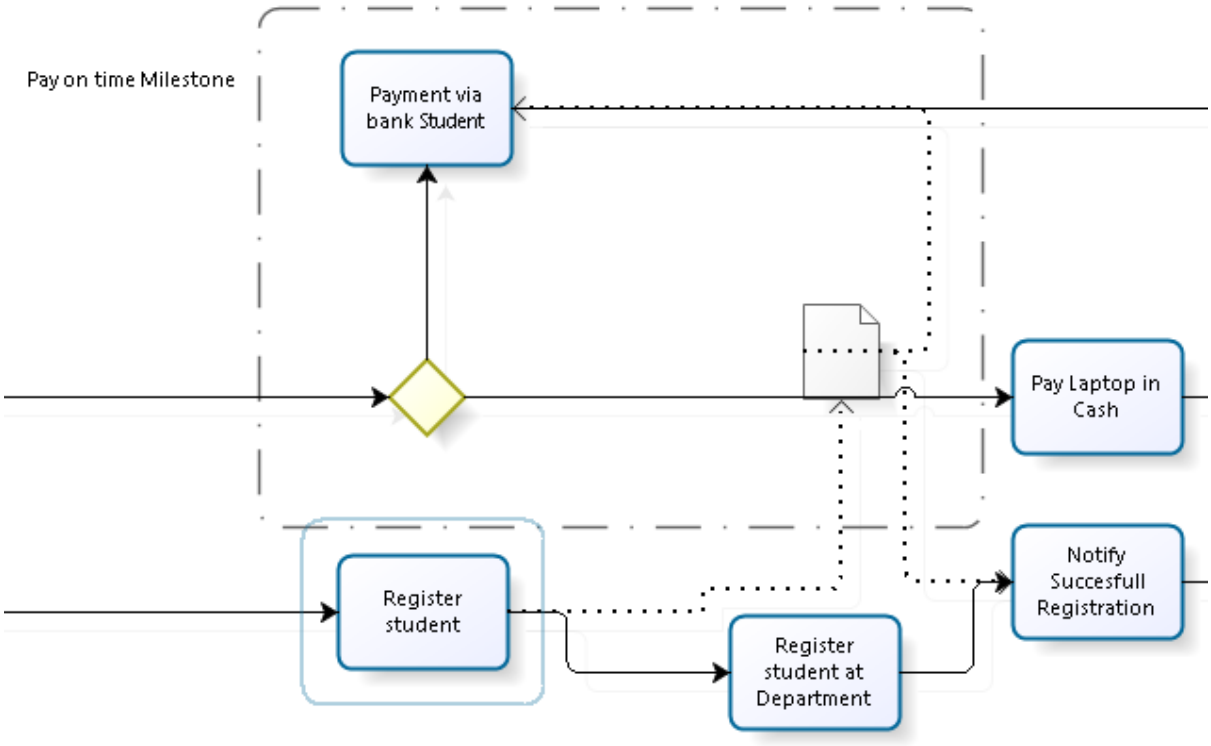

Figure 9: Milestone workflow pattern

We use the milestone pattern to make sure that applicants can only pay via bank transfer while the registration is still running. After the registration is complete, applicants can only pay in cash.

1.3 Soundness

Below, we use two methods to show that our model is sound. First, we draw the block diagram of our model, then we show the results from ProM.

1.3.1 Block diagram

Figure 10 shows the block diagram of our model. Since the diagram is hierarchical, the model is sound.

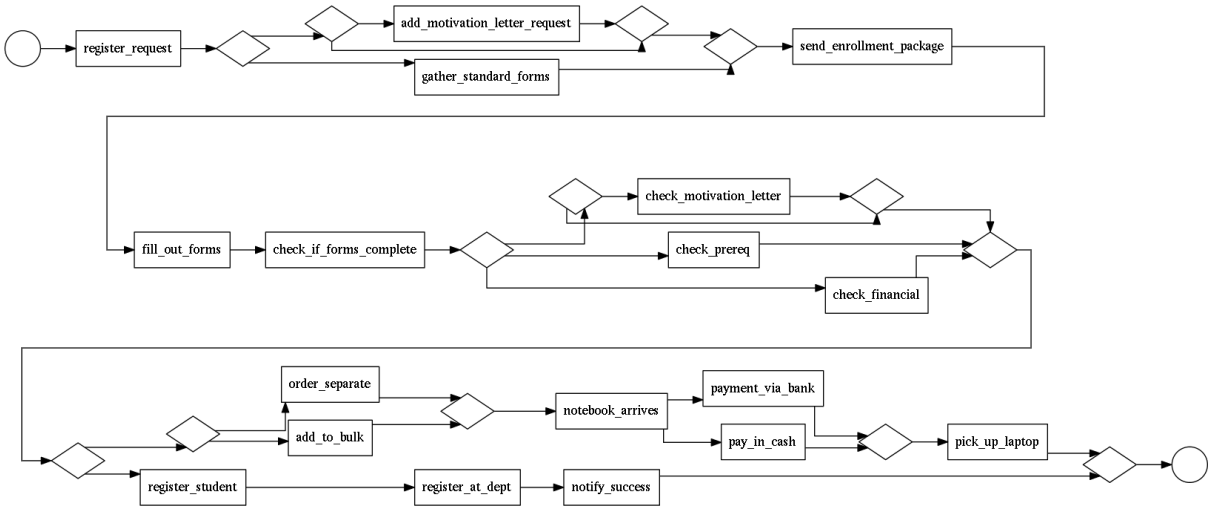

Figure 10: Block diagram of the model

1.3.2 ProM

We used ProM to check the soundness of our model. The resulting net can be seen in Appendix E.

The Woflan analyzer gave the results we expected: aside from the fact that we have three ending places, the net is valid.

2 Analysis of model 1

2.1 Performance choices

Our focus for performance lies on the flow time and the resource utilization. Therefore we chose to perform the prerequisite check, financial details check and (when required) motivational letter check in parallel. Though this is less efficient, as when one the prerequisite check fails, we could would not have had to execute the other two checks, and similarly when the financial detail check fails, we would not have had to check the motivational letter. Despite this, we implemented it this way as when the student does pass these checks, it is faster when done in parallel and thus we decrease the flow time.

2.2 Simulation setup

In this subsection, we explain the time distributions of the tasks, the probabilities for the gateways (when applicable), the resources required per task and the resources which are available for the tasks. We use the truncated normal distribution to describe the time each task takes. This way, we prevent it from ever giving unrealistic values, like negative durations. We write the truncated normal distribution as norm(*mean*, *standard deviation*, *minimum value*, *maximum value*), where each value is in minutes.

For the calculation of the confidence intervals, we run 30 replications of the Time Analysis and Resource Analysis using the *What-If Analysis* of Bizagi. Furthermore, we run the Time Analysis simulations for 60 days to make sure everything finishes on time, and the Resource Analysis simulations for 28 days, as this is the top of the 95% confidence interval for the total time and

we do not want the additional time to lower the resource utilization.
In figure 11 we show the time distributions and performers of each task. In figure 12 we show the probabilities used for the gateways. Finally, in figure 13 we show the amount of resources used for the different roles.

Task name	**Time distribution (min)**	**Performers**
Register request	norm(4, 1, 0, 12)	Administrative Employee
Gather standard forms	norm(2, 1, 0, 8)	Mail Department Employee
Add motivation letter request	norm(2, 1, 0, 8)	Mail Department Employee
Send enrollment package	norm(3, 1, 0, 10)	Mail Department Employee
Fill out and send back forms	norm(7200, 7200, 0, 40320)	/
Check if forms are complete	norm(4, 2, 1, 10)	Administrative Employee
Add explanation missing items	norm(3, 1, 0, 8)	Mail Department Employee
Check prerequisites	norm(10, 2, 2, 20)	Administrative Employee
Check financial details	norm(8, 2, 0, 20)	Financial Department Employee
Check motivation letter	norm(15, 10, 1, 45)	Departmental Executive Board Employee
Notify student	norm(4, 1, 0, 10)	Mail Department Employee
Order separate notebook	norm(10, 2, 2, 20)	NSC Employee
Add notebook to Bulk order	norm(5, 1, 0, 10)	NSC Employee
Notebook arrives	norm(4320, 1440, 0, 20160)	/
Register student	norm(10, 3, 2, 20)	Administrative Employee
Register student at Department	norm(5, 1, 0, 10)	Departmental Student Administration Employee
Notify Succesfull Registration	norm(4, 1, 0, 10)	Mail Department Employee
Payment via bank Student	norm(1440, 4320, 0, 20160)	/
Pay Laptop in Cash	norm(3, 1, 1, 8)	NSC Employee
Pick up Laptop	norm(1, 1, 0, 10)	NSC Employee

Figure 11: The time distributions and required resources per task

Gateway name	**Probabilities**
On time?	Yes: 85%, No: 15%
Complete?	Yes: 75%, No: 25%
Verified? (Prerequisites)	Yes: 85%, No: 15%
Verified? (Financial details)	Yes: 90%, No: 10%
Motivational letter?	Yes: 15%, No: 85%
Motivation ok?	Yes: 40%, No: 60%
Request on Time	Yes: 85%, No: 15%
Payment method	Via bank: 80%, Cash: 20%

Figure 12: The probabilities for the gateways (where applicable)

Resource name	Amount
Administrative Employee	1
Mail Department Employee	1
NSC Employee	1
Financial Department Employee	1
Departmental Executive Board Employee	1
Departmental Student Administration Employee	1

Figure 13: The available resources

2.3 Simulation results

In this subsection we describe the simulation results. All simulation results can be found in the provided Excel files, which have a second sheet for the Resource Analysis. Here we describe the most important parts of the simulation of model 1, which are: the number of students registered and failed, the flow time, the number of students that have a late request (either right away or due to incomplete form submission), the number of instances failed on the prerequisites check, the number of instances failed on the financial details check and the resource utilization.

In model 1, there are on average 658 $\pm$ 4.7 students who were able to register completely. As there is a total of 1000 students on each run, there are 342 students who failed to register. Of these 56.4 $\pm$ 2.8 fail registration due to not sending the registration forms back in two weeks. 92.5 $\pm$ 3.2 fail because their prerequisites are not sufficient. 136.3 $\pm$ 3.3 fail for having incorrect financial details and the remaining 56.9 fail as they did not submit their registration request on time and their submitted motivation letter was insufficient.

The flow time of model 1 is 25876 $\pm$ 250.6 minutes, which is about 18 days. When we take into account that sending the forms back, receiving the notebook and processing the payment can take some time, this seems like a reasonable amount of total time.

There are 195.6 $\pm$ 4.6 students which send a registration request too late. This includes both the students that send it late initially as well as the ones that have to resubmit the forms because they did submitted them incomplete and are late on the second attempt.

The confidence intervals for the utilization of each role can be found in figure 14. So we use the NSC employees a bit too much. This can be explained by the fact that in model 1, we did not split the arrival of the notebook and waiting for the arrival of the notebook into two tasks. Due to this, the NSC employee is busy waiting for as long as the laptop is on the way. We fixed this in model 2.

3 Model 2

3.1 Description

In this section we describe how we altered the process from Section 1 so that is comprehends to the new additional requirements, which are stated in Appendix C. We also made some

Resource name	Utilization
Administrative Employee	59.2 ± 0.22
Mail Department Employee	28.9 ± 0.17
NSC Employee	90.9 ± 0.82
Financial Department Employee	18.4 ± 0.08
Departmental Executive Board Employee	5.5 ± 0.20
Departmental Student Administration Employee	8.1 ± 0.05

Figure 14: The available resources

improvements and these are also described in this section. A complete overview of the second bizagi model van be found in Appendix D.

The first change we made was in the first part right after the task *fill out and send back forms*. Now before we go to *Check if forms are complete* we do an automatic check to see if the forms contain a valid IBAN. If not we consider the forms as incomplete and the flow continues to *Add explanation missing items*. If it was correct the process continues to the task *Check if forms are complete*. Since this check is done automatically, we do not need any employees for this.

The second change regards the payment via the bank. Before we had a task *Payment via bank Student* and we assumed that this payment would always succeed. Now we do not do that anymore. After this first payment task there is another task, *Offer Payment to External Service*. This service determines whether the payment succeeds, is delayed, fails or is invalid. This decision is made by a random distribution which will be explained later. The way the flow continues depends on the result of this task.

If the result is successful than the flow continues to the task *Pick up Laptop*. If the payment fails than the student has top pay for the laptop in cash when he comes to pick it up. This means that the flow continues to the task *Pay Laptop in Cash*. If the payment is delayed then the flow returns to the task *Offer Payment to External Service*, where it is tried again to do a payment. The random distribution is chosen in such a way that it is not possible that one IBAN results in an infinite loop of delayed payments and therefore a deadlock in the process. The last option was that the payment was invalid, this would be the case when the amount to be payed was negative but we do not allow these amounts in our model and therefore this option will never occur.

After the first simulation we noticed that almost everytime before the laptop arrives, the registration was already completed. This was due to the fact that waiting fora laptop took much more time than completing the registration. This also resulted in a high resource utilization for the NSC employee since the employee was "busy waiting" in the *Wait for notebook to arrive* task. Therefore we splitted this task into two tasks. One is the original *Wait for notebook to arrive*, but now without using resources, and the *Notebook* arrive task. This task is done by the NSC employee. This resolved the "busy waiting" problem.

We resolved the problem with the registration by adding the restraint that if a student wants to pay via the bank, a registration can only be completed when the notebook has arrived. This means the task *Register Student* can be executed only when the notebook has arrived. If a student wants to pay in cash (most of the times this is the situation when a request is to late) the task *Register Student* does not have to wait for the notebook to arrive. We choose for this

because in this case it does not matter that the student cannot pay via the bank anymore after the registration is completed, since he wants to pay in cash. Another reason we chose for this is that we can imagine that, when a request is to late (so the academic year is already started) you want to finish the registration as soon as possible, such that the student can participate in the courses. The laptop is not immediately necessary to do this, but a complete registration is.

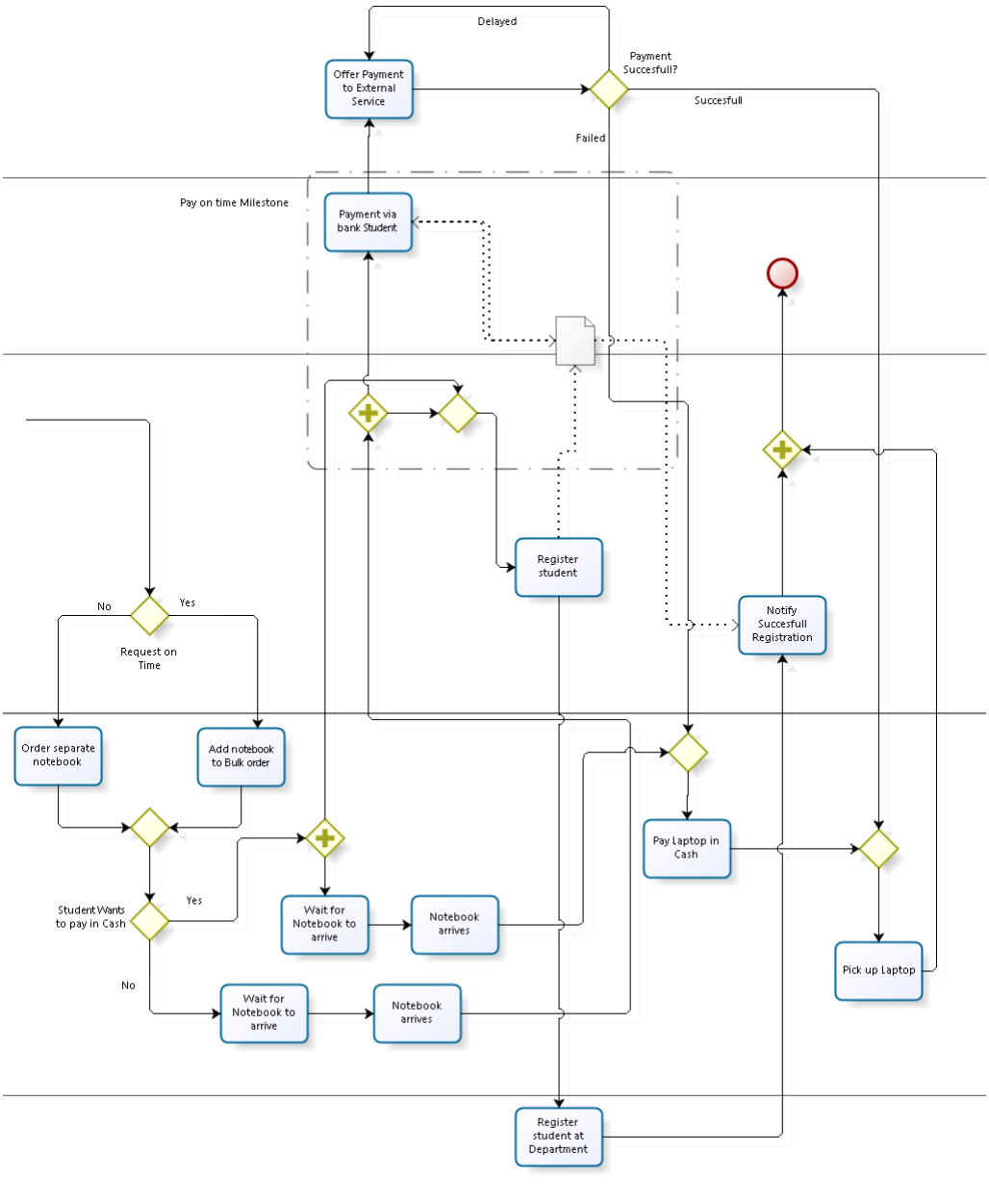

Figure 15: The changed part of the second model in Bizagi

3.2 Workflow patterns

The workflow patterns in version 2 of the model are the same as those described in Section 1.2.

3.3 Soundness

For the second model, we just used ProM to show soundness.

3.3.1 ProM

We used ProM to check the soundness of our model. The resulting net can be seen in Appendix E.

The Woflan analyzer gave the results we expected: aside from the fact that we have three ending places, the net is valid.

4 Analysis of model 2

4.1 Simulation setup

In this section we describe the simulation of model 2. First we describe the setup, as tasks have been added and some task time distributions and probabilities have changed. After this, we describe the results of the simulation of this model. In figure 16 we show the time distributions and performers of each task. In figure 17 we show the probabilities used for the gateways. Finally, in figure 18 we show the amount of resources used for the different roles.

4.2 Simulation results

In this subsection we describe the simulation results of the most important things that we expect to have changed since model 1, which are the flow time and the resource utilizations.

The flow time now is 39979.8 $\pm$ 229.3 minutes, which is about 27.8 days. This is a much higher flow time. This difference in flow time can be explained by the additional constraints.

The resource utilizations now are as shown in figure 19. As we can see, most roles have similar utilization as in model 1. Clearly, the NSC employee has much less utilization than in model 1 as it now no longer actively waits for notebooks to arrive.

5 Conclusion

Since we had already considered our design goals for model 1 very carefully, it was difficult to think of improvements for model 2. Because of this, we mainly focussed on implementing the additional requirements in model 2.

The additional requirements in version 2 of the model obviously make the model more realistic. Because of these changes to the model, it is hard to compare the two versions. However, even though the second model has a higher flow time, we recommend implementing the changes, since there are more automated tasks integrated in the process.

Task name	Time distribution (min)	Performers
Register request	norm(4, 1, 0, 12)	Administrative Employee
Gather standard forms	norm(2, 1, 0, 8)	Mail Department Employee
Add motivation letter request	norm(2, 1, 0, 8)	Mail Department Employee
Send enrollment package	norm(3, 1, 0, 10)	Mail Department Employee
Fill out and send back forms	norm(7200, 7200, 0, 40320)	/
Check if forms are complete	norm(4, 2, 1, 10)	Administrative Employee
Add explanation missing items	norm(3, 1, 0, 8)	Mail Department Employee
Check prerequisites	norm(10, 2, 2, 20)	Administrative Employee
Check financial details	norm(8, 2, 0, 20)	Financial Department Employee
Check motivation letter	norm(15, 10, 1, 45)	Departmental Executive Board Employee
Notify student	norm(4, 1, 0, 10)	Mail Department Employee
Order separate notebook	norm(10, 2, 2, 20)	NSC Employee
Add notebook to Bulk order	norm(5, 1, 0, 10)	NSC Employee
Wait for Notebook to arrive (2x)	norm(4320, 1440, 0, 20160)	/
Notebook arrives (2x)	norm(2, 1, 0, 8)	NSC Employee
Register student	norm(10, 3, 2, 20)	Administrative Employee
Register student at Department	norm(5, 1, 0, 10)	Departmental Student Administration Employee
Notify Succesfull Registration	norm(4, 1, 0, 10)	Mail Department Employee
Payment via bank Student	norm(1440, 4320, 0, 20160)	/
Offer Payment to External Service	norm(10, 5, 1, 20)	/
Pay Laptop in Cash	norm(3, 1, 1, 8)	NSC Employee
Pick up Laptop	norm(1, 1, 0, 10)	NSC Employee

Figure 16: The time distributions and required resources per task

A Appendix Process Description

A.1 STU (imaginable)

At the Eindhoven University of Technology, the Student Service Centrum (STU) is the department that registers new students. This process starts when a new student makes an application request. An administrative employee (AE) registers the request, and a mail department employee (MDE) sends an enrollment package to the student. If the request is on time for the next academic year, this package contains two forms. In case the request is late for the next academic year, the package contains the same two forms and a request for a motivation letter. The first form asks for personal information, the study programs the student wishes to take, and financial details; the second form asks whether the student wants to participate in the notebook project, and if so, more details about this participation.

Gateway name	Probabilities
On time?	Yes: 85%, No: 15%
IBAN valid?	Yes: 95%, No: 5%
Complete?	Yes: 75%, No: 25%
Verified? (Prerequisites)	Yes: 80%, No: 20%
Verified? (Financial details)	Yes: 95%, No: 5%
Motivational letter?	Yes: 15%, No: 85%
Motivation ok?	Yes: 40%, No: 60%
Request on Time	Yes: 85%, No: 15%
Student Wants to pay in Cash	Yes: 20%, No: 80%
Payment successful	Successful: 60%, Delayed: 25%, Failed: 15%

Figure 17: The probabilities for the gateways (where applicable)

Resource name	Amount
Administrative Employee	2
Mail Department Employee	2
NSC Employee	5
Financial Department Employee	1
Departmental Executive Board Employee	1
Departmental Student Administration Employee	1

Figure 18: The available resources

Resource name	Utilization
Administrative Employee	56.8 ± 0.25
Mail Department Employee	28.4 ± 0.17
NSC Employee	14.3 ± 0.13
Financial Department Employee	18.3 ± 0.07
Departmental Executive Board Employee	5.4 ± 0.13
Departmental Student Administration Employee	5.9 ± 0.07

Figure 19: The available resources

If no answer is received within two weeks, the process ends without further message to the student. Once the forms are received, an AE checks whether they are complete and puts them in the system. If the forms are not complete, a new package is sent again by an MDE, with an explanation that something was missing. If a motivation letter was required, it is sent back with the package.

Once all forms are complete, the Notebook Service Centrum (NSC) orders a notebook, if the student opted to participate. Meanwhile, STU processes the request further. First, the prerequisites are checked. Second, the financial details are checked by a financial department employee (FDE). If these are not met, the student is notified and the process ends. Third, if applicable, the motivation letter is checked by a departmental executive board employee (DEBE). (If the DEBE rejects the motivation letter, the student is notified of this and the process stops.)

Fourth, the student is registered in the university by an AE of STU. Fifth, the student is registered in the study programs by the departmental student administration employee of the departments corresponding to the chosen study programs. Finally, the student is notified of the successful registration.

Only if the request is not on time for the next academic year, NCS orders a separate notebook on receipt of the request. Otherwise, the notebook is added to the bulk order for notebooks for the next academic year. Once student registration cannot fail anymore and the notebook has arrived, payment is requested from the bank of the student. In case the payment does not arrive before registration completes, the student has to pay at pick up. In both cases, NSC notifies the student, who picks up the laptop and pays if necessary.

B Appendix Bizagi Model 1

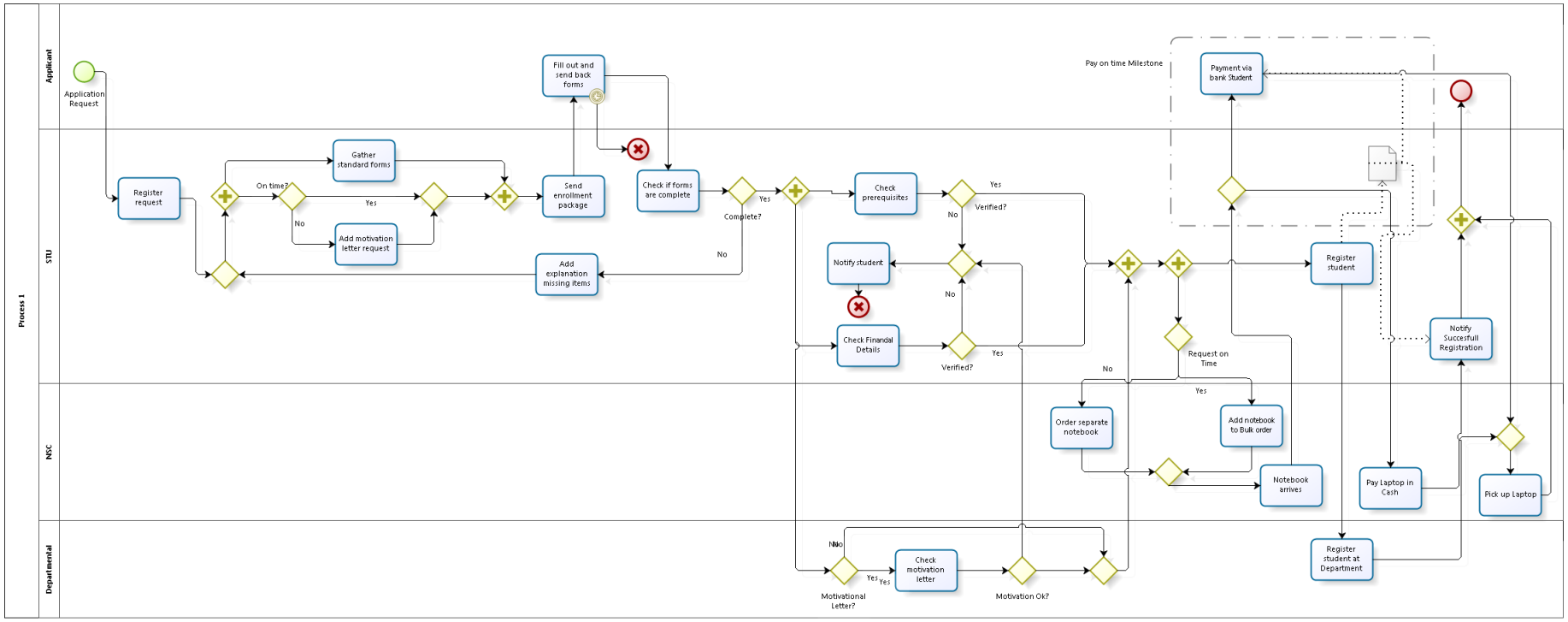

Figure 20: The first model in Bizagi

C Appendix Additional Requirements

As an additional requirement, the payment should be ordered to a "real" bank using a YAWL external service call (a codelet):

The payment is ordered to an external service. The payment can fail, succeed or be delayed. In case of delay, the payment should be ordered again a few times. If the payment fails, the student pays at pick-up. Please note that it is not guaranteed that repeatedly calling the external service will eventually result in success or failure.

D Appendix Bizagi Model 2

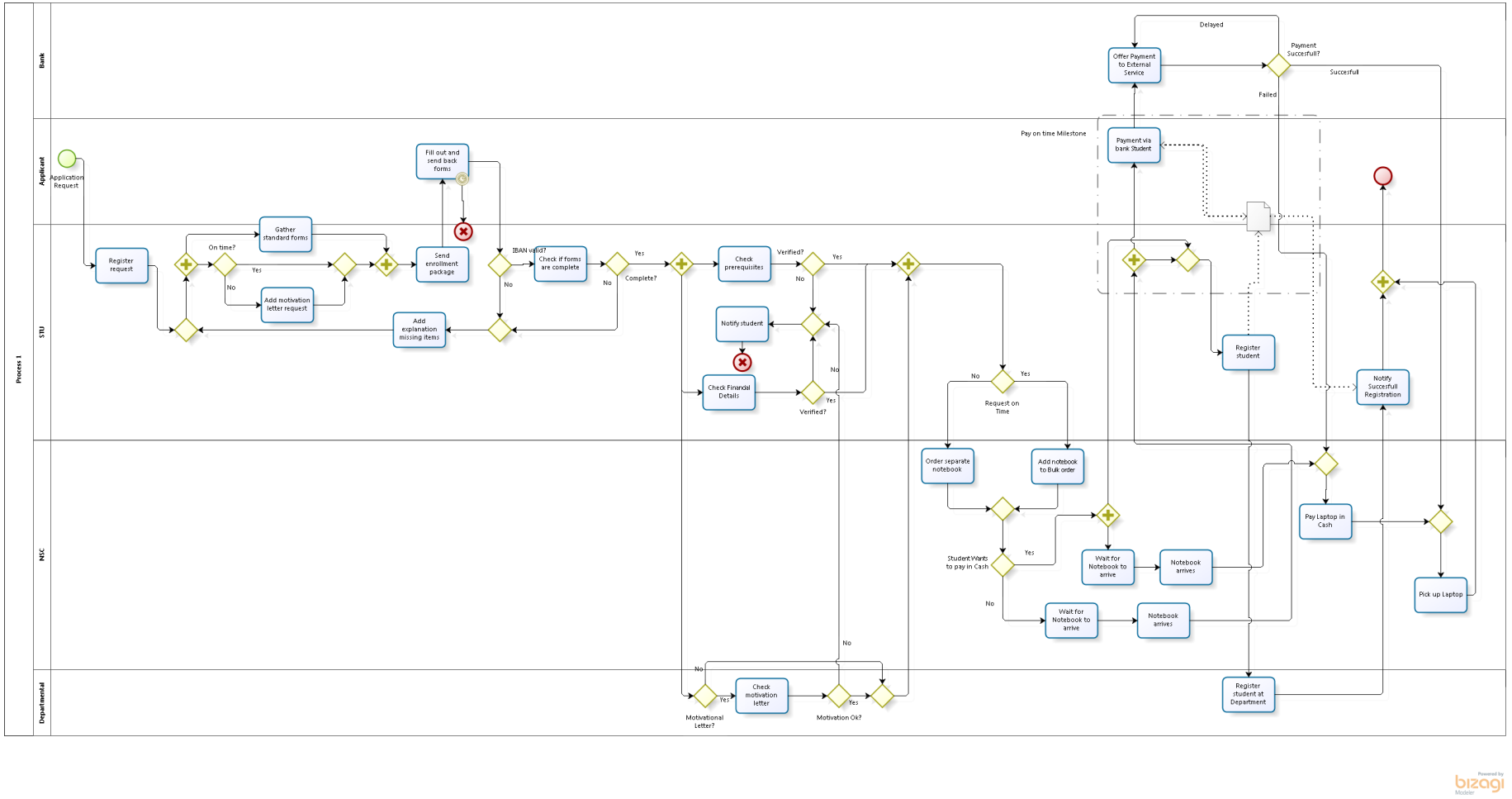

Figure 21: The second model in Bizagi

E Appendix ProM

Figure 22: ProM net for model 1

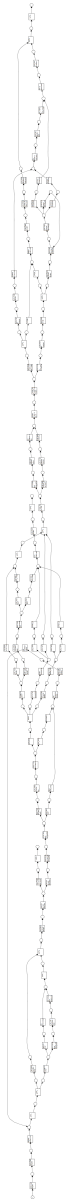

Figure 23: ProM net for model 2