Optimization of a maintenance scheduling problem through simulation

by

RIHAN SCHALKWYK
26332362

Submitted in partial fulfilment of the requirements for the degree of

BACHELORS OF INDUSTRIAL ENGINEERING

in the

FACULTY OF ENGINEERING, BUILT ENVIRONMENT AND INFORMATION TECHNOLOGY

UNIVERSITY OF PRETORIA

October 2009

Executive Summary

A maintenance scheduling problem was identified at a petrochemical company. A change in technology used in certain vessels (henceforth referred to as V1) prompted the investigation into changing the current schedule to coincide with the maintenance done on the vessel's peripheral equipment (henceforth referred to as vessels V2 and V3). The proposed plan is to do mini-GO's (General Overall) followed by major-GO's. This change will reduce equipment downtime significantly, but could also place an increased workload on maintenance teams as projects might start to overlap more.

To show to management the effects these proposed changes will have on the system as a whole, a simulation model was built using Arena. Using various tools available a good representation of the system was modelled without making it too complex. A model of the current system was built (As-Is model) and then after proven representative of the current system, another model was built, reflecting the proposed changes to the system (To-Be model). The maintenance schedule of the vessels (time between GO's) was then optimized through multiple simulation runs. Afterwards the transition from the As-Is to the To-Be model was studied as well.

The recommended changes to the system are as follows:

- One mini-GO followed by a major-GO
- Mini- and major-GO's to be three years apart
- The duration of the cycles (to complete maintenance on all 40 vessels) of mini- and major-GO's to be six years
- To gradually increase the time between scheduled maintenance activity start times according to their sequence in the current GO cycle to 54 days apart each
- The re-engineering of vessel V2 due to abnormal failure rates

When these changes are implemented, vessel reliability will increase significantly and the sum total amount of time spent on maintenance projects (when not including the re-engineering of V2, the sum total was reduced from 31, 793 days in the As-Is model, to 18, 478 days in the To-Be model with the recommended system inputs).

Table of Contents

List of Figures

List of Tables

Introduction & Background

At a certain petrochemical company, certain vessels are used to ensure that Production has all the raw material needed.. All equipment is serviced (GO – General Overall) according to a strict maintenance schedule and the different types of downtime are classified as follows:

- **V1 GO**
  The vessels are stripped, repaired, put back together and introduced back into the system during this GO-period with all haste possible as each hour of downtime results in major production losses.
- **Maintenance on peripheral equipment**
  Peripheral equipment includes V2 and V3. These maintenance actions in themselves are also a major source of production losses as all the vessels linked (V1, V2 and V3) are taken offline for the duration of operations.
- **Failure downtimes**
  Failures occur in any system and the equipment in question is no exception. With the introduction of new technology in V1, these failures have been reduced significantly and consequently also an increased lifetime.

With the introduction of the new type of technology in V1, it has been proposed to revise the maintenance schedule to combine operations for all vessels. This change will result in the elimination of unnecessary downtime and significant production increases.

Management, however, is resistant to change as maintenance teams are already stretched thin and any additional activities will probably result in more overlapping of projects. They have only been shown figures representing downtime reductions however. This means that the influence of this change needs to be shown on the system as a whole before management's decision concerning the matter may be swayed.

Project Aim

A simulation model of the equipment maintenance process will be built using Arena 11.The problem of unnecessary downtime needs to be addressed as well as resource utilisation. At the end of the project, management should be advised whether or not to incorporate proposed changes into the maintenance schedule. If changes should be made, the specific run times of equipment between maintenance operations should be recommended by means of showing system wide implications thereof (specifically the effect on maintenance's resource utilisation).

Project Scope

The following will be in scope of the project:

- Building of a simulation model for both As-Is and To-Be states
- To show the influences of the changes on the maintenance department's resources
- Normal operational activities should be taken into account together with maintenance activities
- Equipment failures' effect on the system

The following will be out of scope:

- Detailed (hour to hour) breakdown of maintenance activities
- Interaction of system with other company areas
- Effect of material shortages or any other abnormal external influences on the maintenance operations

Literature

When dealing with non-repairable items, one has to replace the item once failure occurs. This pattern is represented by the Bathtub-curve (Figure 1) and as can be seen, failure rates are at their lowest for a certain time around the middle of item life. This idea of an increased possibility for failure near the end of an items life forms the basis for reliability engineering (O'CONNOR, P. D. et al., 2002). At the petrochemical company this increased possibility of failure for equipment is of great concern.

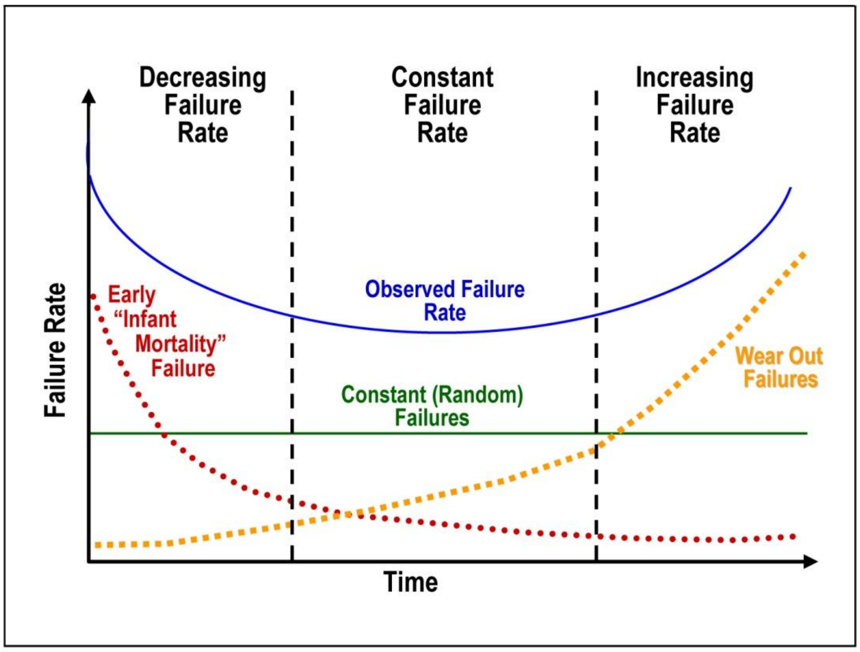

Figure 1 The Bathtub Curve (SZUBINSKI, H., 2009)

Maintenance at the petrochemical company can thus be defined as "defending machinery equipment against deterioration" according to M.G. Petrescu (PETRESCU, M. G. and Duţă, R., 2008).

In these companies different strategies are combined to achieve this goal. For example: Corrective Maintenance, Preventative Maintenance and Mean-Time-To-Failure (Refer again to Figure 1), only to name a few (PETRESCU, M. G. and Duţă, R., 2008). Maintenance schedules are drawn up for equipment in the plant (V1 and its peripheral equipment, v2 and V3, in this specific case) and maintenance teams work in a year round "shut down mode" to meet production demands.

When looking at the maintenance schedules themselves, a lot of work has been done by researchers around the world on the field including S.A. Oke and O.E. Charles-Owaba (OKE, S. A. and Charles-Owaba, O. E., 2007) (OKE, S. A. and Charles-Owaba, O. E., 2005) (OKE, S. A., 2004). Many of these studies focus on developing different techniques for maintenance scheduling and optimising total cost such as simulated annealing, genetic algorithms, tabu search, integer programming and the probabilistic approach. Although these methods yield their respective optimal results, the researcher finds that there are still many unanswered questions and urges researchers to find answers to these.

One of these fields is simultaneous scheduling of both operational and maintenance activities in a resource-constrained environment (OKE, S. A. and Charles-Owaba, O. E., 2005). This field touches on the issue at hand where a broader picture model is required as well as the fact that the amount of resources available for maintenance work is restricted. The work of O.E Charles-Owaba specifically goes into some depth on the Gantt charting side of maintenance scheduling which is also in use at the petrochemical company. These charts form the basis for each maintenance project's management.

As each maintenance project is approached and carried out in the same fashion, a lot of historical data for these projects are present. Work has been done in the project management field (MEYER, P. H. and Visser, J. K., 2006) where simulation was used to better predict project lifetimes using these historical data. An important matter also raised in this paper is that actual data needs to match estimated data.

Looking at maintenance scheduling from a simulation perspective, work have been done on the subject by a few researchers including J.K Visser and G.Howes (VISSER, J. K. and Howes, G., 2007). In this work they used simulation as a technique to optimise maintenance teams for a service company. Monte Carlo Simulation was used to study the service company in question which was largely due to the stochastic nature of the system.

When combining a few of the above mentioned ideas, one can begin to formulate a solution to the problem at hand. Simulation offers a way to generate data for past, present and future operations and the effects that these will have on the whole system. With a simulation model it would also be possible to include both operational and maintenance activities and analyse in depth the effect that any changes to the system would have on the resources at hand.

With this in mind special techniques could (and should according to P.S. Kruger (KRUGER, P. S., 2003)) be used in the simulation model to make it as representative as possible of the problem at hand. K.H. Concannon showed off Visual8's SIMUL8-planner at the 2003 Winter Simulation Conference (CONCANNON, H. et al., 2003). With this tool one is able to schedule resources and start times of activities in the simulation program in much the same way as Gantt charts are used in practice (Figure 2). Using schedules like this one can build a relatively simple model to accurately reflect the system and then use the scheduler to bridge the gap between user and model. This functionality is available with programs such as Arena as well (KELTON, W. D. et al., 2006).

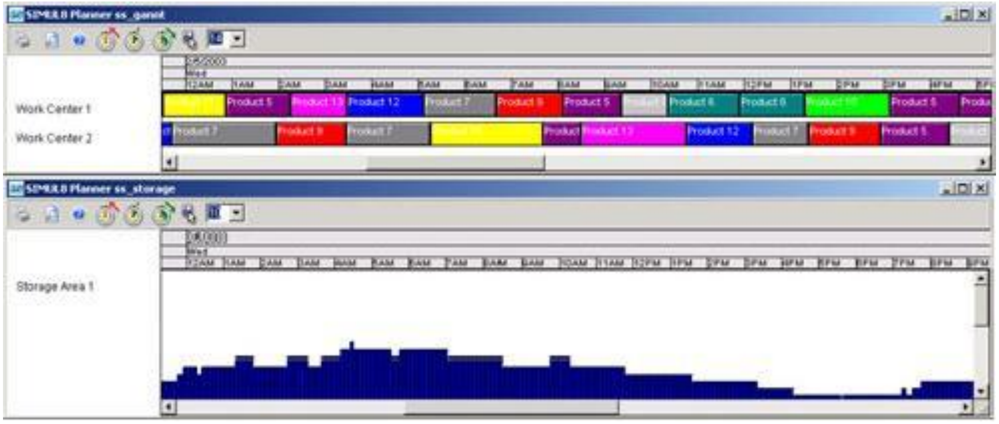

Figure 2 SIMUL8-planner Gantt chart (CONCANNON, H. et al., 2003)

All the while work is underway on solving the problem at hand, one should also be sure to keep in mind that management is currently still unconvinced that any changes should be made. This means that they are *resistant to change*. This is a change management problem and is specifically addressed by Dianne Waddel and Amrik S. Sohal (WADDELL, D. and Sohal, A. S., 1998). They point out the fact that resistance should not be viewed as detrimental to a project, but rather as a tool to be utilised. Resistance should be managed in the sense that it usually points out that the proposed change might not be well thought through or just plainly wrong.

Taking this into account, one should be careful not to get carried away with the modelling of the new system, but to keep it as true to the current system as possible. More than that, one should have the objective that “changes should minimize the disruption to the application system” (KRAMER, J. and Magee, J., 1990). This changeover period when the maintenance schedule changes from the current to the proposed, careful attention should be given towards how the models react in this specific period and not only the before and after simulations compared to each other (Figure 3). A positive answer resulting from this might well sway management’s vote on the matter.

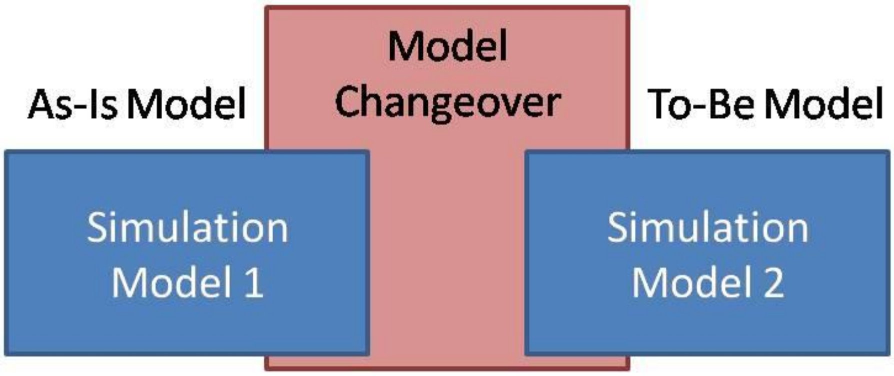

Figure 3 Illustration of model changeover

Data Analysis

Considerable time was spent getting to know the system during the first part of the project. This knowledge was then used to accurately model the system in Arena. Data however remains elusive in some ways as the maintenance teams are currently in the final stages of replacing old vessel parts with new reengineered parts. This presents a problem in the sense that no actual failure data exists for the new vessels. Expert opinion was thus gathered on many facets of the project to help fill any gaps and prevent inaccurate assumptions.

Minimum, maximum and mean times fit over a triangular distribution for all inputs was deemed adequate for building a high level model of the system. Here follows a list of all process as well as failure times:

Maintenance Activity	Activity Time			
	min	avg	max	Unit
Mechanical Strip V1	3	3	5	Days
Welding V1	28	31	34	Days
Mechanical Box Up V1	7	8	10	Days
Mechanical Strip V2/V3	1.5	2	2.5	Days
Mechanical Box Up V2/V3	4.5	5	5.5	Days
Mechanical Strip Failures V1	1	1.5	2	Days
Welding Failures V1	2.5	3.5	10.5	Days
Mechanical Box Up Failures V1	1.5	2	2.5	Days
Mechanical Strip Failures V2	4	5	6	Hours
Welding Failures V2	13	15	17	Hours
Mechanical Box Up Failures V2	3	4	5	Hours
Mechanical Strip Failures V3	8	10	12	Hours
Welding Failures V3	1.8	2	2.2	Days
Mechanical Box Up Failures V3	8	9	10	Hours
Proposed Mini-GO Mechanical Strip V1	1.5	2	2.5	Days
Proposed Mini-GO Welding V1	5.5	6	7	Days
Proposed Mini-GO Mechanical Box Up V1	1.5	2	2.5	Days

Table 1 Maintenance activity times

Vessel Failure	Time Till Failure			
	min	avg	max	Unit
V1 Initial Failure	1855	2200	2920	Days
V1 Secondary Failure	1095	1460	1825	Days
V2 Initial Failure	200	250	300	Days
V2 Secondary Failure	150	200	250	Days
V3 Initial Failure	1100	1200	1300	Days
V3 Secondary Failure	900	1000	1100	Days

Table 2 Vessel failure times

With regard to the maintenance schedules themselves (optimizing the new proposed maintenance schedule is the main objective of the project), maintenance teams have a target of 40 vessels to service within a four year period and this results in overlapping of projects. The scheduled arrival of entities into the simulation model (in the As-Is model) needs to reflect this overlapping (Figure 4). Afterwards, one can set out to increase resource utilization by optimizing the time between mini- and major-GO's in the To-Be model (Figure 5) to decrease the amount of vessel downtime and project overlapping.

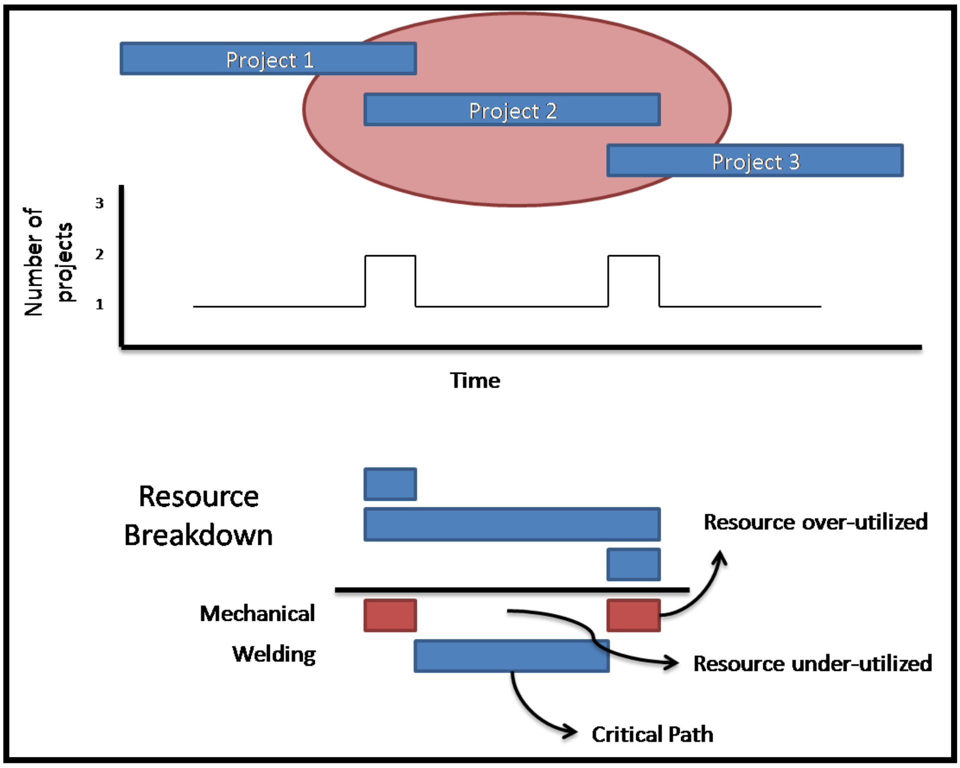

Figure 4 Illustration of current schedule and resource breakdown

Simulation Model

Conceptual Design

When considering the scheduling problem at hand, the proposal has been made to introduce "mini-GO's" on a much shorter interval than previous GO's to coincide with peripheral equipment's maintenance schedules. V1 will be repaired (no replacement of major-GO parts), maintenance done on V2 and V3 and then put back into operation for another cycle until the time when a "major-GO" can be done. Major parts will then be replaced as well as maintenance done on peripheral equipment. Scheduling of these projects will overlap even more (Figure 5), but when breaking activities down to resource level, one will get much better understanding of resource utilization.

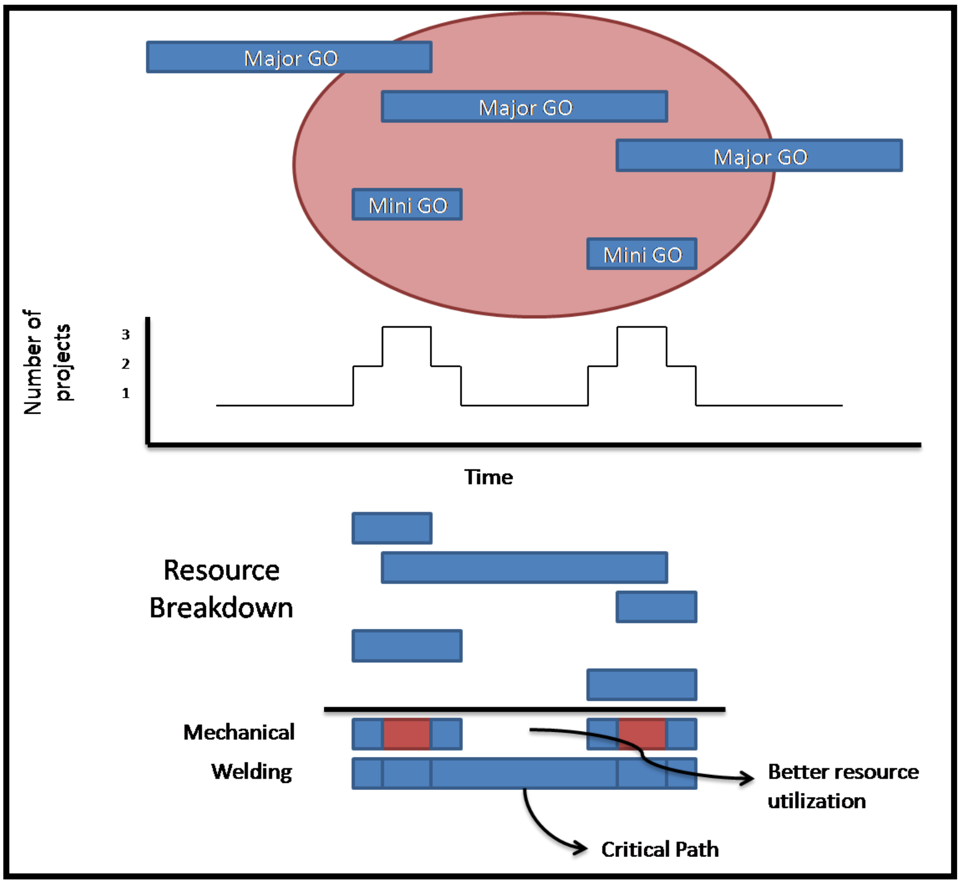

Figure 5 - Illustration of proposed changes to schedule

As stated earlier, the purpose of the project is to sell these proposed changes to management and the chosen method is a simulation of the system using Arena. Thus beginning with a high level model and working your way down, in terms of model detail, one can start to simulate the system towards achieving this means.

A basic simulation model (Figure 6) consists of inputs, the model itself and outputs. Here is a detailed list outlining the required inputs and outputs:

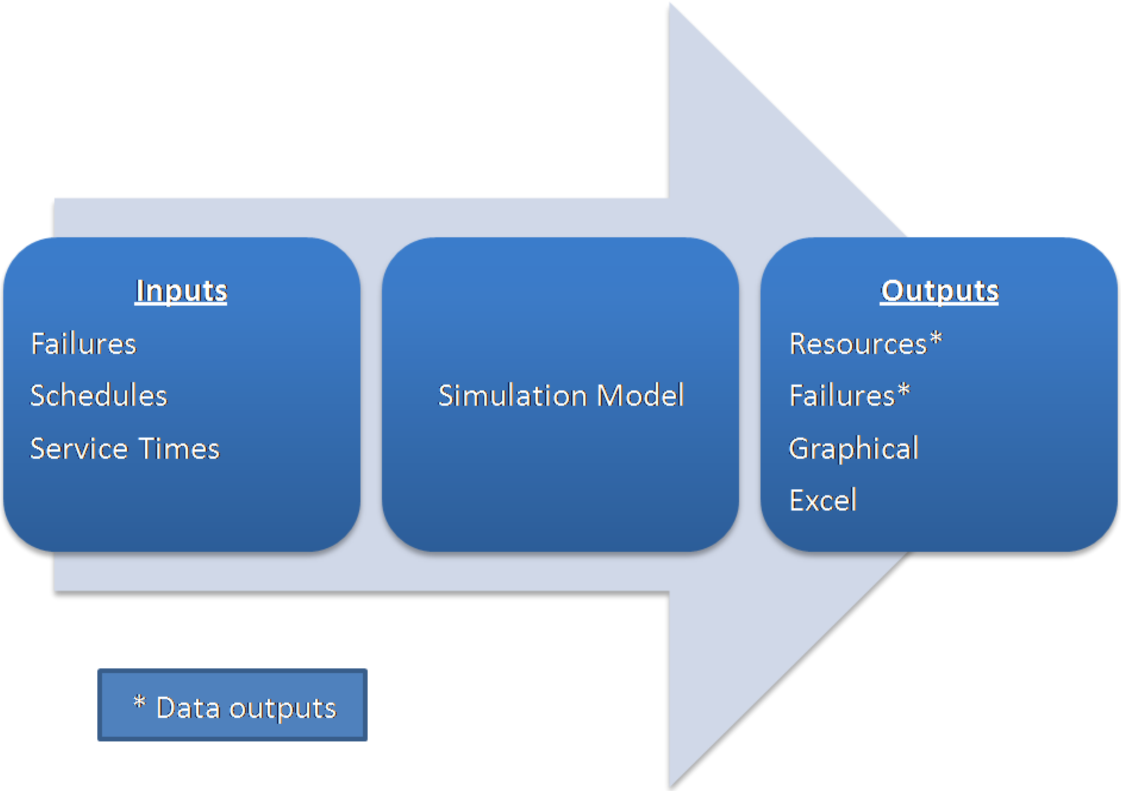

Figure 6 Basic flow of a simulation model

Inputs

- **Resource Failures**

  Resources here include all vessels. It is important to capture failures specifically in the As-Is model (the simulation model built of the current system to be used as basis for comparisons) to show management exactly how flawed the current system is. There exists, however, very little data to indicate when the new major parts for V1 vessels will start failing, but expert opinion was gathered and used as model inputs.

  When building these failures into the model, care should be taken to make sure that the resources that fail are given some priority in the queue for maintenance (Figure 7) as these resources have to be put back to use as soon as possible.

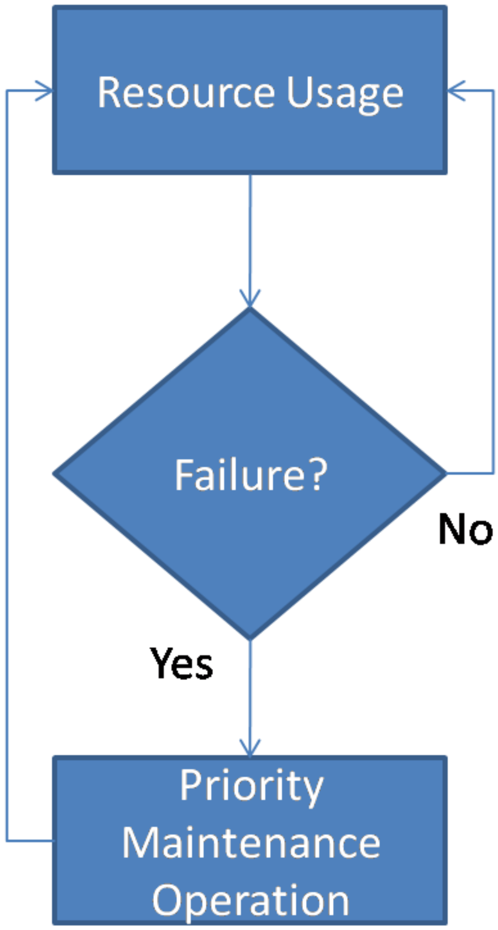

Figure 7 Failure logic

- **Scheduled Resource Run-Times**
  The run-times for V1 and peripheral equipment must be controlled via a schedule (or terminating usage of resource via maintenance schedule) in the simulation. This will provide the tool with which one can quickly make a change and see what the influence on the system as a whole will be. A method to be used in constructing a high level model is making use of resource sets to combine large quantities of the same type of equipment (this will simplify the model considerably). Individual attribute values will then be used to assign entities to specific resources within the set.
- **Maintenance Service Times**
  The basic high level processes for maintenance projects are mechanical strip of equipment, welding and then mechanical box-up. This clearly indicates the two main resources of the maintenance department (also refer back to Figure 4 and 5) and the scheduling of these activities is essential. In general the welding activities form the critical path of the project while mechanical activities are a bit more flexible. The simulation of this process is essential to show the effects of multiple maintenance projects on resources to management.

When considering the distributions to be used with the above data, it has been decided to use triangular distributions wherever expert opinion had to be used for maintenance times and failure rates.

Outputs

- **Data output**
  Using data generated during the run of a simulation, one can gain useful information such as:
  - utilization of resources
  - amount of resource downtime
  - number of concurrent maintenance projects
  - average number of active resources (especially all vessels in this case) in the system

  Information on failures will also be useful as it will show:

  - total number of failures
  - MTBF (mean time between failures)
  - MTTR (mean time to repair)
  - amount of resource downtime due to failures
  - time of failure in relation to maintenance schedule
- **Graphical Output**
  During the run of the simulation, it is useful to see how the system reacts before it has run to completion. Details such as follows can provide much needed day-to-day information on the system:
  - Number of active resources
  - Number of ongoing maintenance projects
  - Number of failures
  - Total equipment downtime and associated cost
- **Output to MS Excel**
  One of the many useful functions of Arena is the ability to export data generated during the simulation to Excel. From here it is possible to compare generated data to actual data and determine whether simulated data is accurate or not. Only when the As-Is model has been proved accurate, can changes be made and further comparisons made.

Additional Tools

When the normal tools available for simulating the process don't provide sufficient means, a tool that can be used to overcome these restrictions is Visual Basic for Applications (VBA). What makes VBA so useful is the fact that it can control and interact with various applications (not only Arena, but also MS Excel, MS Word etc) and provide a means to build complicated logic into a program without having to go to extreme lengths using the built in features of the program to achieve the same goals.

This can be used to control the usage of resources as well as proposed simultaneous usage periods of vessels. Building some of these complex logic structures into the Arena model can be tedious and often result in flawed structures. VBA can also be used to export data to Excel during the run of the model. This data can then in turn be used to draw graphs in Excel as well, so that this run-time info may be available for study after the simulation has completed its run (Figure 8).

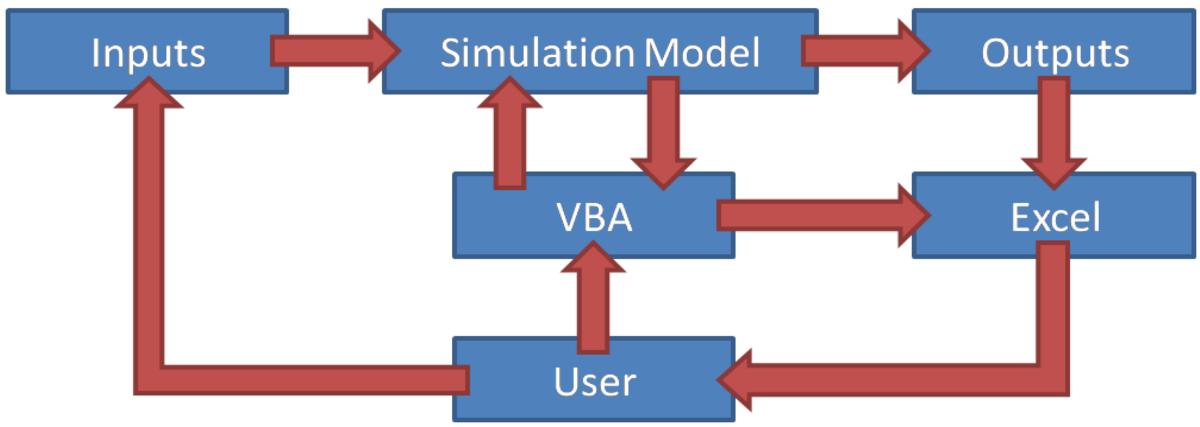

Figure 8 VBA and its interaction with user and applications

As-Is Model

It is important to know that when a maintenance activity takes place, the whole vessel is taken offline and this goes for V1 as well as V2 and V3. This concept is integral to the building of the simulation model. Whenever a maintenance activity takes place, the resource usage is interrupted for the duration of the activity. To achieve this in the model, conventional modelling techniques using high level "process blocks" to represent resource seizure, delay and release cannot be employed. More advanced process blocks are thus used to control the flow of maintenance activities throughout the system.

Furthermore, maintenance schedules of the vessels are independent of each other as well as maintenance done due to failures. Consequently each schedule (as well as failures) has its own control logic in the system. These control loops link with each of the other control mechanisms throughout the model by means of a specific attribute that identifies each entity (Figure 10).

Figure 9 Common entity creation and sequential attribute assignment

As each of the entities enters the system, two sequential attributes (a_GGseq and a_GGseq2) are assigned to them that are used to control the flow throughout the model. This is achieved by making use of "signal" and "hold" blocks. The "signal" blocks send a signal (the attribute value of the active entity) out to all the "hold" blocks and all entities corresponding to that attribute value are released to continue processing. See Appendix A for attribute and variable descriptions.

An in depth discussion follows for each part of the model (see Appendix C for more model views):

Vessel Control Processes

Each of these processes starts off with a "create" block (specific introduction delays between arrivals of entities) that, together with the "delay" block later on, represents the maintenance schedule of that specific vessel. After the entity attributes have been assigned, the entity triggers a signal (a_GGseq2) that releases the initial entity in the Resource Control to seize the corresponding resources. The entity is then delayed for the duration of the time between scheduled maintenance activities (Figure 11) and upon being released for a maintenance activity, is firstly sent through a scan block to make certain that there are no vessels of the same type that have failed, waiting to be repaired. If the entity gets to move on, it triggers yet another signal (a_GGseq) that releases the resources at Resource Control and sends the vessel for maintenance.

Figure 10 Maintenance schedule control

The entity then enters an “assign” block where the variable v_GOtype’s value associated with that specific entity is changed to indicate whether it is being sent for a V1 GO, V2 or V3 maintenance activity. The value of v_downGO is also adjusted to indicate that the resource’s failure loop should be terminated (see Vessel Failure Control below). After it receives the signal (a_GGseq2) that maintenance is complete, the entity is re-entered into the “delay” block to repeat the process.

Vessel Failure Control

All 40 entities are created simultaneously at the beginning of the run and then assigned individual sequence attributes as well as a value based upon the observed failure distribution. The entities are then held up and only released when receiving the initial signal from Vessel Control Processes. The entity released then triggers the assignment of initial values for v_downGO, v_V1fail_time1 (v_V2fail_time2 and v_V3fail_time3 in the cases of V2 and V3 failures) and an attribute value reflecting the current simulation time (Figure 12).

Figure 11 Failure time while-loop

The entity then enters a while-loop where the terminating conditions are that either the failure occurs (e.g. v_V1fail_time1 = 0) or the vessel is sent for maintenance. Inside the while-loop the value of v_V1fail_time1 is decreased by one (the “one” reflecting the base time unit of a day in the model) until either of the conditions are met. This makes the execution of the model considerably slower, but more accurate as failure is based on time between maintenance activities and not just the general lifetime of the vessel.

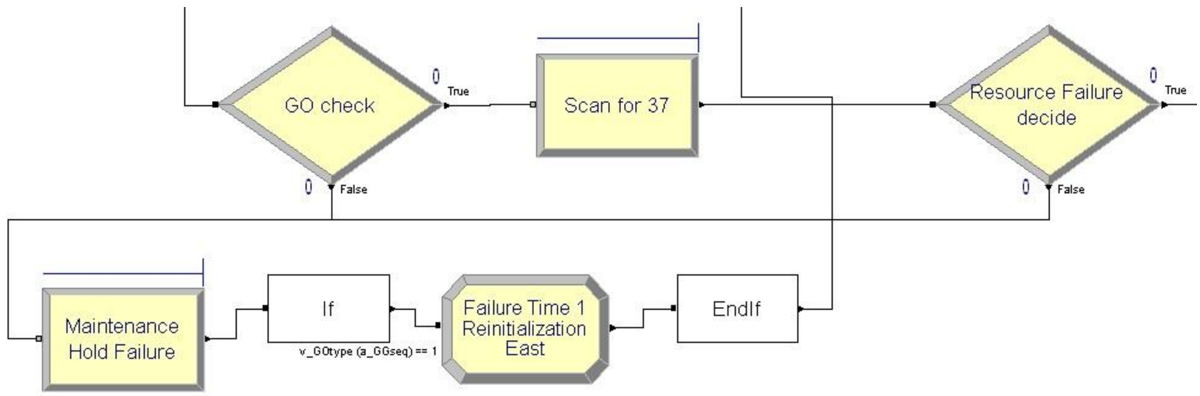

Figure 12 Failure decide tree

After an entity leaves the while-loop (Figure 13) it is either sent onwards for repair or placed on hold depending on its terminating condition. If it is sent for a repair, it enters a "scan" block where it is held up until there is a maintenance team available to work on it. This follows on the fact that many failures aren't critical and can be delayed for incorporation into the maintenance schedule to meet production's demand that 37 vessels per plant must be operational at all times (this however isn't absolute as when critical failures does occur, there doesn't exist any choice in whether to delay the repair activities or not). On the other hand if the entity is placed in a "hold" block, it is due to a maintenance activity scheduled for that specific time. The entity then awaits the signal that the routine maintenance is completed before it is released and the attribute, a_failT1 (a_failT2 and a_failT3 in the case of V2 and V3), is then reassigned a value based upon the original failure distribution if the if-statement is true. The If-statement tests to see whether the maintenance activity done was specifically for that vessel in question.

When an entity is sent for a maintenance activity (Figure 14), additional information regarding that is required. It enters the first "record" block where the counter for the total number of failures is incremented. The MTBF is recorded in the second block using the difference between TNOW and the value of a_tnow. The attribute, a_tnow, is then reset at an "assign" block to help pinpoint the MTTR. Together with this assignment the variable v_numFAIL is increased as well as the variables v_GOtype and v_downGO are adjusted to reflect the failure state of the vessel.

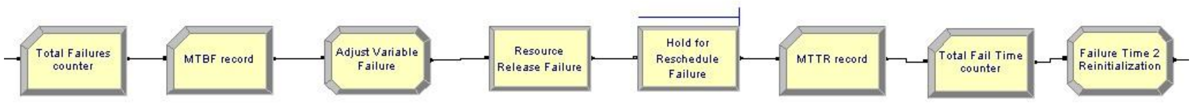

Figure 13 Failure repair

The entity then triggers a signal to be sent to release the associated resources and for the maintenance activities to begin. For the duration of the maintenance activities, the entity is placed in a "hold" block where it awaits the signal from the Maintenance that the maintenance is done. The MTTR is then recorded as well as the Total Fail Time increased with the repair time before the entity is passed on to be assigned a new value for its next failure time. This new value is based on a different distribution than the original failure distribution, as the vessels tend to fail more often once they have failed for a first time. The original distribution is assigned once the vessel has been sent for a scheduled GO.

After the entity has either triggered a maintenance activity or been placed in hold for the duration of a scheduled maintenance activity, the entity is sent back to the "assign" block right before the while-loop where the cycle is reinitialized.

Vessel Resource Control

As with the failure control, the resource control creates 40 entities at the beginning of the simulation run. After these entities has been assigned their sequential attributes they are immediately placed in a "hold" block and await the first signal to arrive from the schedule controls. The entity the proceeds to a "seize" block where V1, V2 and V3 resources associated with that specific entity are seized.

As there are 40 vessels per plant, 40 resources were created for the V1 vessels themselves, as well as for V2 and V3 vessels. This gives a total of 120 equipment resources associated with each plant[1]. To enable an entity to seize the correct resource (without making use of an unnecessary amount of individual "seize" blocks), resource sets were created for V1, V2 and V3 resources. The desired resource is then located within the set by making use of the first sequential attribute, a_GGseq.

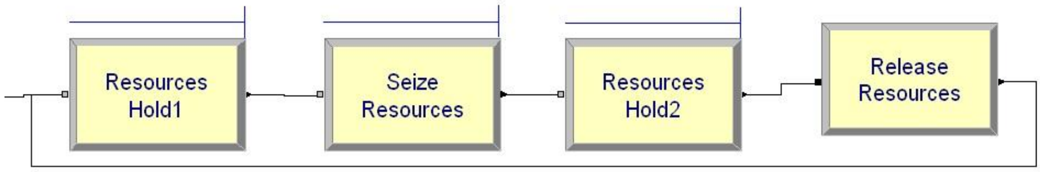

Figure 14 Resource control

The entity then moves on immediately (note that in this model the "seize delay release" actions of a basic process is split up and modelled separately as there are more than one event that can lead to one resource being seized and released) and is placed in hold again. Here the entity awaits a signal from either the failure or schedule controls after which it then proceeds to release the resources it previously seized.

After the resources have been released, the entity loops back to the first "hold" block it encountered, but the difference (from the second time onwards) is that the signal is sent from the maintenance control after the activity has been finished.

Vessel GO's and Maintenance

After initial attribute assignments, the entity is held until a signal (a_GGseq) is received to indicate that a maintenance activity is required. The entity is then sent to the desired maintenance processes according to the entity's value of v_GOtype. The first block the entity then encounters (Figure 15) increments the value of v_numGO. This serves the purpose of keeping track of the number of ongoing maintenance projects (as mentioned management is concerned that maintenance team's resources will be stretched too thinly). The attribute a_GOtime is assigned the value of the current simulation time.

[1] As 120 resources need to be created per plant (that gives 240 in total) the sheer amount of work to create these by hand would have taken very long. A simple VBA code was written to create all of these resources automatically. A sample of this code is given in Appendix B.

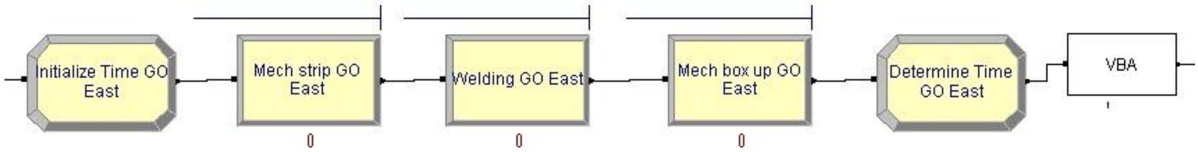

Figure 15 Maintenance activities and VBA block fire

The entity then proceeds through three processes (mechanical strip, welding and mechanical box up) that represent the basic breakdown of the three stages of maintenance operations (the exception is the scheduled maintenance activities of V2 and V3 which only includes the strip and box up). Each of these types of activities has its own resource. Upon completion of the maintenance the value of v_numGO is decremented again and a_GOtime is assigned the value of the difference between the attribute value and the current simulation time.

The entity then fires a VBA block whose function is to draw a block in Excel to indicate the duration of the specific vessel's maintenance project. After the simulation model has run to completion this Excel chart resembles a Gantt chart of the whole maintenance plan as drawn up by the simulation model. The VBA code to achieve this functionality of the model is given in Appendix B[2].

After the counter for the total amount of maintenance activities has been increased with the same value of a_GOtime, the entity then signals (a_GGseq2) to indicate that the maintenance work is done whereupon it is looped back to the initial "hold" block where it once again await the next maintenance request.

Excel Data Exporting

One entity is created (Figure 16) at time zero and then proceeds to a delay block where it is delayed one day (the base time unit of the model). At the end of the day the entity proceeds to an assign block where the current value of v_numGO (the total number of currently active maintenance projects) is assigned to the entity's attribute a_numGO. It then fires a VBA block that exports the value of a_numGO to Excel (the same workbook as the one used for the project Gantt chart, only a different worksheet) where a value is recorded for every day of the simulation run. After the simulation run has ended, a graph is drawn up for these values in Excel so that the day-to-day behaviour of the model can be studied.

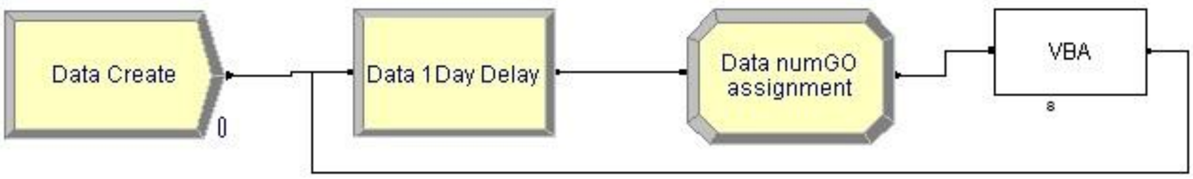

Figure 16 Data exporting

To-Be Model

The To-Be model uses the basic system logic of the As-Is model as basis for its own. The main differences between the two models are that the maintenance schedules for V1, V2 and V3 are combined and that extra logic was added to the Vessel GO's and Maintenance processes to control

---

[2] The code given only includes the first VBA block (that of the main V1 GO's) and will be expanded to include all of the maintenance types in the future.

the flow of entities for mini- and major-GO's. The rest of the processes remain exactly the same (resource and failure control as well as data exporting).

Here follows a detail description of all model changes to the To-Be model (see Appendix D for more model views):

Vessel Control Processes

As with the As-Is model, the initial resource seizure signals are sent from the Vessel Control Processes. The entities then proceed to the delay module (Figure 17) that represents the schedule delay between maintenance activities. The value for this delay is assigned via an attribute value, a_GOdelay, which is initialized in the "Entity count Shed" assignment block. After being released for a GO activity, the entity then enters a scan block to check whether there aren't more than two vessels in the queue for failure repairs. If not, then the entity continues onwards, otherwise it is held until the number of vessels in line for maintenance has decreased.

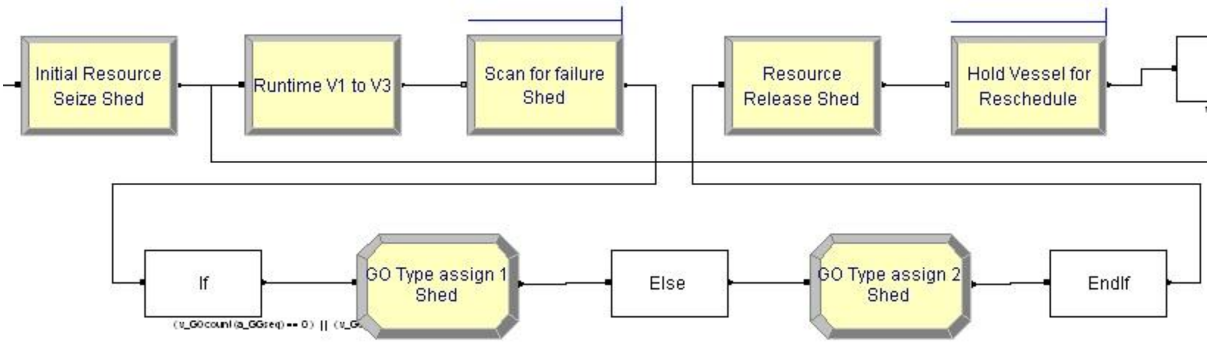

Figure 17 Combined maintenance schedule

The entity must then be sent for either a mini- or a major-GO depending on the value of the value of v_GOcount (initially set to 0). If the value is zero or one, the v_GOtype variable is assigned to one, indicating a mini-GO. If not, the v_GOtype value is changed to two, indicating a major-GO. The entity then triggers a signal, a_GGseq, to be sent to release the specific resource and start the vessel on its GO process. The entity is then held in a hold block for the duration of the maintenance activities and is only released another signal, a_GGseq2, is received.

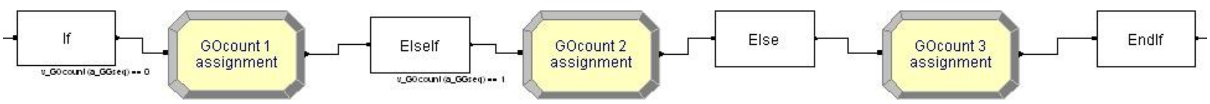

Figure 18 v_GOcount and delay time assignments

Before the entity is sent back to the initial delay block for its next scheduled runtime period, it is sent through a series of if-statements (Figure 18). Depending on the value of v_GOcount, it is either incremented (values of zero and one) or reset to zero. The value of a_GOdelay can also be changed in this part of the model to allow the control of whether the times between mini- and major-GO's are equal or not.

Vessel Failure Control

The failure processes of the vessels remain exactly the same as in the As-Is model, with the only exception of the exclusion of the if-statement (Figure 19) following the hold block where entities go when being sent for a scheduled maintenance activity (mini- or major-GO). This control logic is excluded because there is only one maintenance schedule for V1, V2 and V3 now. So after every GO (whether mini or major) a new value is assigned to a_failT1, a_failT2 and a_failT3 according to the relevant distribution before being sent back to the Variable Initialization Failure assign block.

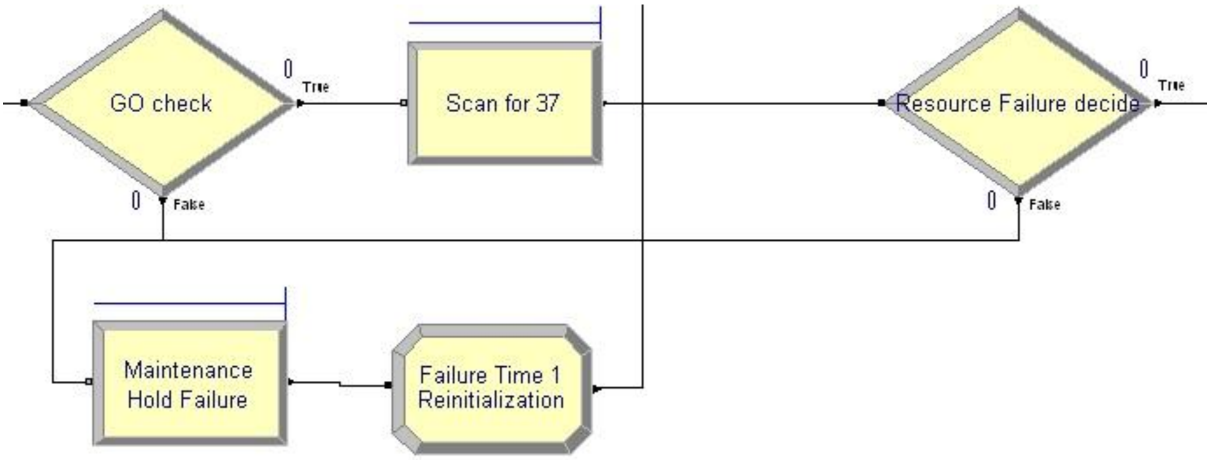

Figure 19 Decision logic without if-statement

Vessel Resource Control

The processes for resource control remain unchanged from those of the As-Is model.

Vessel GO's and Maintenance

A new V1 maintenance type is included in the To-Be model to represent the mini-GO's. The rest of the maintenance activities remain unchanged as no extra work is added or taken away and thus the times remain unchanged. The only noteworthy difference in the processes, however, is the resources they seize. It has been proposed that more staff will be recruited for the sole purpose of looking after the mini-GO's. Thus the mini-GO's have their own V1, V2 and V3 resources that operate independently from the resources used for failure maintenance and major-GO activities.

Extra controls were added to determine where entities should go in case of a mini- or major-GO. After the initial decide block, entities that have a corresponding v_GOtype value of one or two are sent to a common assign block (Figure 20) where the value of v_numGO is incremented (this is because even though V1, V2 and V3 maintenance projects are underway, it is seen as only one overall GO project). An entity then proceeds through a separate block where the entity is duplicated twice. The original entity proceeds to a decide block where it is sent to the initialization assign block of either the mini- or major-GO.

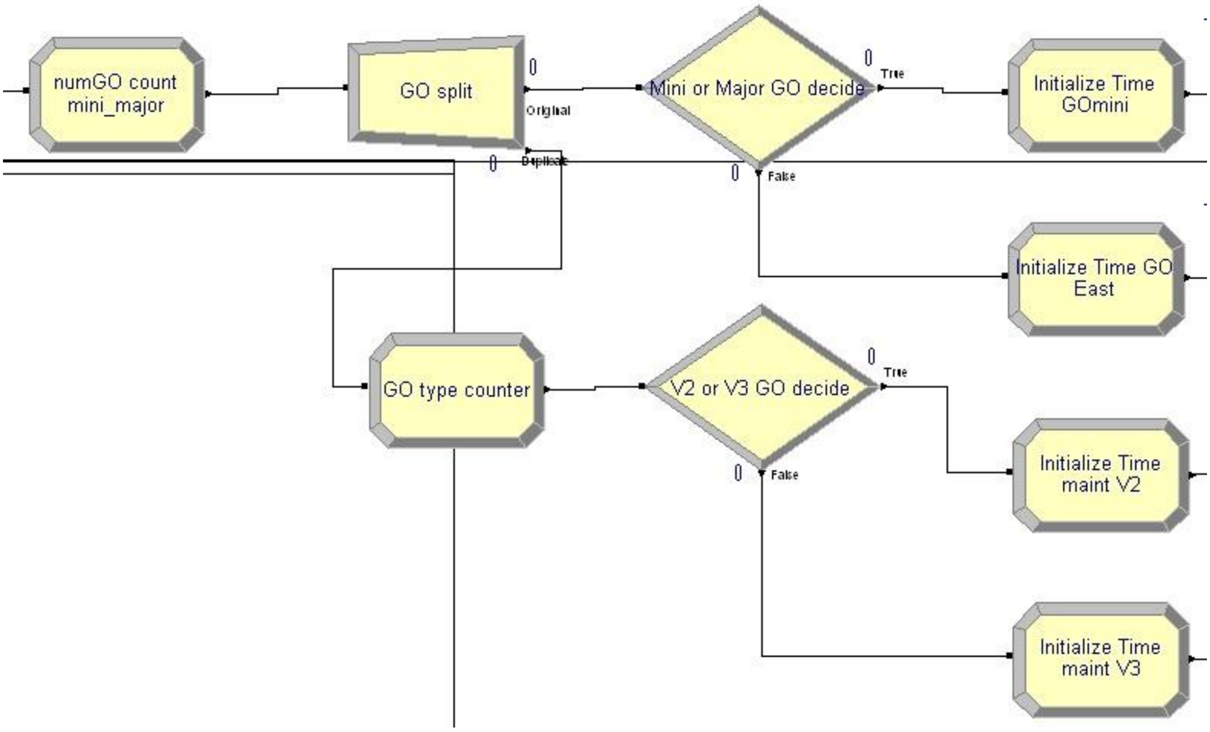

Figure 20 Mini- and Major-GO decision logic

The duplicated entities are sent to a assign block where a variable, v_equalcount, is incremented. An entity then proceeds to yet another decide block where it is sent to the initialization blocks of either V2 or V3 maintenance processes according to whether the current value of v_equalcount is an odd or even number.

Upon completion of the maintenance activities and after the VBA blocks have been fired, entities proceed to the batch block (Figure 21) where the duplicates are permanently joined together with the original entity according to the value of their a_GGseq attribute. Afterwards the single entity continues to the assign block where v_numGO is decremented again where after it is passed on to the Total Repair Time recording block.

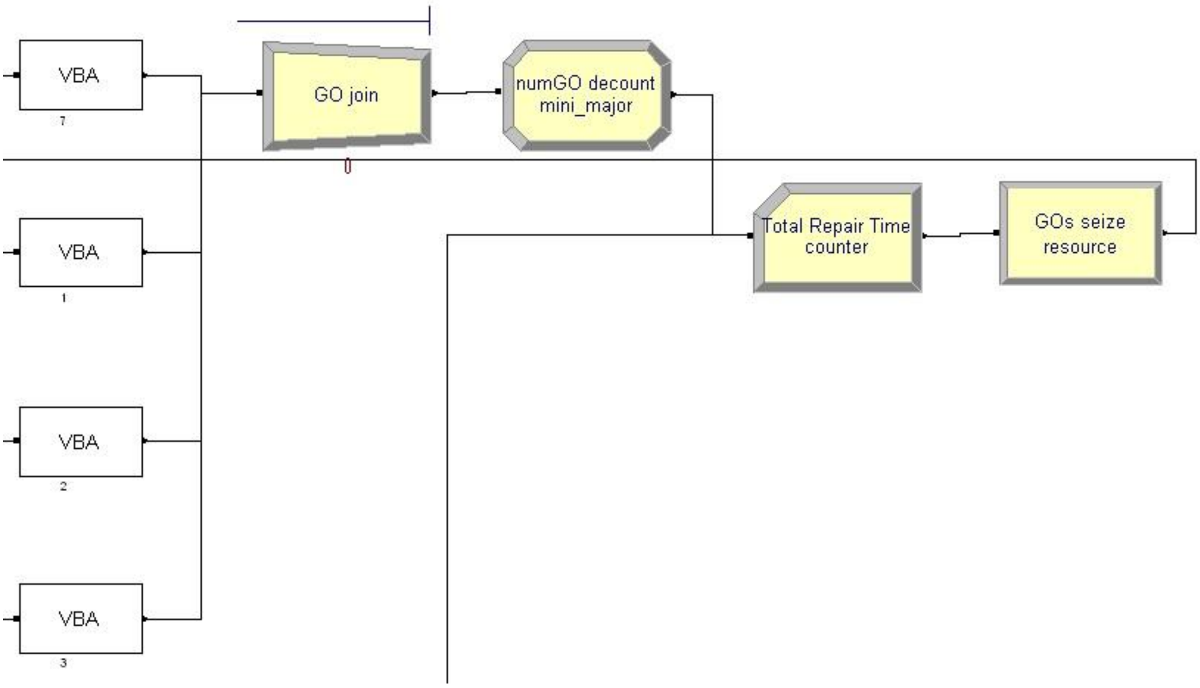

Figure 21 Duplicate termination

Excel Data Exporting

The same process is followed as in the As-Is model.

Optimization Methodology

Upon completion of the To-Be model, it was repeatedly executed to determine the optimal maintenance schedule times between mini- and major-GO's. The three main inputs that can be changed are the following:

- Introduction delay of entities created
- The choice between one or two mini-GO's
- The time between scheduled GO activities for a specific vessel

Various inputs were entered and the model executed both with animation and in batch run mode each time. This is because one is interested in how the system reacts to these inputs both from a day-to-day perspective as well as total run time report figures. The model's warm up period is set to the time when the first major-GO's will start (failures, mini- and major-GO's will be present in the system then) and the run length set to 20142 days (which is round about 55 years).

In the day-to-day observations, the model's graph output is of crucial importance as it depicts the number of ongoing maintenance projects. One cannot simply look at minimum, maximum and average figures at the end of the simulation run to determine whether or not the maintenance resources will be over or under utilized. Thus it can clearly be seen when and for how long abnormal activity takes place that might have been missed otherwise.

The reports that Arena compiles at the end of each run, however, contains some data (such as user specified data e.g. MTBF and MTTR tallies, total repair time counters and time persistent variables such as that of v_numGO) that are still extremely useful. These data are then recorded and compared with other simulation runs and together with observations from the animated simulation runs, useful conclusions can be drawn and eventually an optimal maintenance strategy determined.

Results and Recommendations

After having executed the To-Be model several times with different input values, the data presented in Table 1 was recorded (also see Appendix E for the Excel output Gantt charts and v_numGO graphs of the As-Is and To be model with recommended inputs). Throughout this exercise, a much better understanding was gained of how exactly the system components influence each other and how the schedule might further be improved.

Intro delay (days)	0 to mini 1 (days)	mini 1 to mini 2 (days)	mini 2 to major (days)	numGO avg	numGO max	numFAIL avg	numFAIL max	Total Failures	Cross-over no activity (years)
27	1095		1095	1.3681	7	0.1742	4	3001	2
32	1095		1095	1.1825	5	0.1784	4	3061	2
35	1095		1095	1.1609	5	0.1776	3	3059	2
37	1095		1095	1.1593	5	0.1718	4	3066	2
27	730		1095	1.5897	8	0.1695	4	2969	1.5
32	730		1095	1.4087	5	0.1717	4	3008	1.6
35	730		1095	1.4016	5	0.1715	3	3030	1.2
37	730		1095	1.3720	5	0.1704	4	3006	1.1
27	730		1460	1.3781	8	0.1752	4	2990	2.6
32	730		1460	1.1766	5	0.1755	4	3017	2.5
35	730		1460	1.1659	5	0.1766	4	3046	2.2
37	730		1460	1.1640	5	0.1752	4	3028	2.1
27	730		730	1.8262	8	0.1604	4	2888	0.8
32	730		730	1.6803	5	0.1637	4	2935	0.6
35	730		730	1.6441	5	0.1630	4	2936	0.2
37	730		730	1.6509	5	0.1634	4	2944	0.05
54	1095		1095	1.1242	5	0.1784	4	3067	0.2
45	912.5		912.5	1.3192	5	0.1745	4	3068	0.2
54	730	730	730	1.3052	5	0.1655	4	2859	0.2
37	1095	1095	1095	0.9773	5	0.1790	4	2900	5
37	912.5	912.5	912.5	1.1342	5	0.1734	4	2913	3.4
68	912.5	912.5	912.5	1.0840	5	0.1747	4	2921	0.2

Table 3 To-Be model results

The major factor, it seems, that is critical for the system to become stable throughout the whole run cycle is the introduction time of entities into the system. This is the current maintenance schedule, or on the other hand the hypothetical maintenance schedule of the future. To explain this further, one can consider the current 4-year maintenance plan for the V1 vessels. At the moment the vessels are round about 37 days apart from each other (to fit into the four years available to GO each of the 40 vessels). If one was to let the GO's of the different vessels coincide, one would see that problems arise if the major-GO's aren't sufficiently spaced apart from each other.

It was found that the original proposed plan for two mini-GO's followed by a major-GO, each three years apart, resulted in a five year period of little or no activity for maintenance teams (row highlighted in red in Table 1) if the vessels were 37 days apart from each other. In other words there would be a period of four years in which the maintenance teams will have too much work (mini-GO's together with overlapping major-GO projects) followed by two cycles of low activity where only failures occur and mini-GO's are done. The other problem here will be that the V1 vessels will be pushed to their maximum theoretical life spans of 9 years. This isn't wise as the previous V1 vessels showed that failures might occur much more rapidly from the second half of the life span onwards.

It is thus a better solution to the problem at hand to lengthen the time in which the 40 major-GO's are to be performed (increasing the introduction delay) from four years to six years (row highlighted in green in Table 1). One mini-GO will be done after three years and after a following three years the major-GO will be done. This does well mean that V1 will not be utilized till the end of its life span, but reliability will definitely increase. This specific schedule also has less overlapping projects than other. With an introduction delay into the system of 54 days, it usually has 10 days available between major-GO's into which a mini-GO project could fit.

When looking at specifically the sum amount of days spent on maintenance projects, one can see the significant improvement in the system when comparing the As-Is model to firstly the nine year (two mini-GO's) schedule and then secondly to the six year (one mini-GO) schedule (Table 2).

Model	Total Repair Time (days)
As-Is	31, 793
To-Be (9 years)	21, 598
To-Be (6 years)	18, 478

Table 4 Sum total of maintenance projects

Another point to note is that the crossover from the current schedule (As-Is) to that of the new schedule (To-Be) needs to be considered as well. When keeping the current introduction delay of vessels into the schedule (37 days apart) one comes across the problem that an extended period of "low activity" exists between the current V1 GO cycle and the next major-GO cycle (Figure 22). To keep this from happening, the introduction delay must be extended (e.g. from 37 to 54 days) and thus extending the period over which the maintenance teams can work on GO projects. This will only result in a once off period of low activity as vessels' run times must be gradually changed to fit into the future schedule.

Lastly one can see that the total amount of failures do not really decrease by much (irrespective of the model input) and if examined closely, *all* failures are V2 failures. As the failures occur so regularly (one every five days) it would be strongly recommended that the V2 vessels be redesigned the schedule cannot possibly be adjusted to pre-empt these failures.

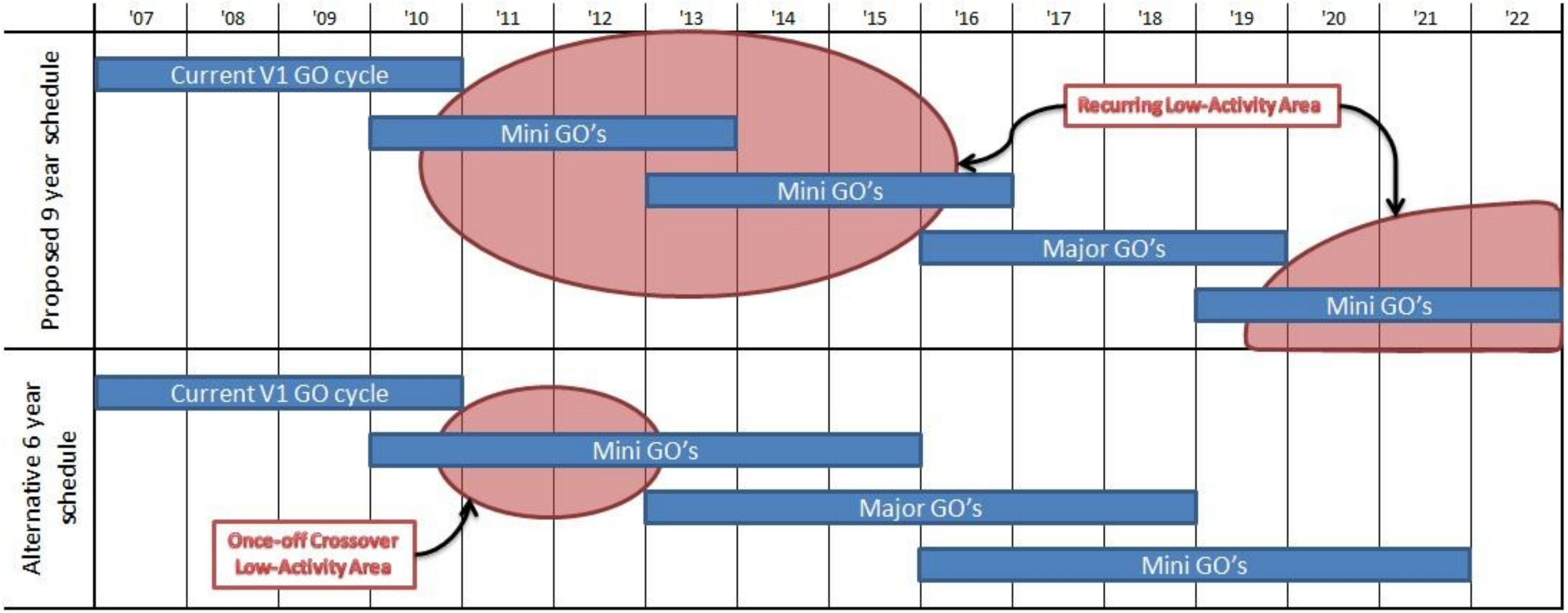

Figure 22 Comparison of schedule alternatives

Conclusion

After careful system and data analysis two simulation models was built to represent the system in its current state as well as in the proposed future state where all vessel maintenance schedules are combined. The As-Is model was used to compare with results obtained from repetitive execution of the To-Be model with various model inputs.

The inputs to the To-Be model that yielded the best overall results was a schedule that only includes one mini-GO and one major-GO, both three years apart, but each stretching over a period of 6 years. To achieve this it is necessary to delay some of the vessels that underwent maintenance later on in the current GO cycle to achieve a delay between start times of major-GO projects of 54 days (ideally). This result in constant maintenance-resource utilization as well as a significant reduction in the sum total of time spent on maintenance projects (18, 478 days down from 31, 793 days).

When switching over to the new schedule, a period of two years will occur when there will be no major-GO projects. This crossover period will only occur once and can be used to maintain vessels V2 and V3 as well as repair failures.

It is also strongly recommended that the re-engineering of vessel V2 should be considered. The only failures that occur throughout a simulation run (after the warm up period) are those of vessel V2. It is not possible to adjust the scheduled maintenance activities to pre-empt failures that occur every 5 days. That is basically every 200 days apart per vessel, compared to the current maintenance schedule in which a vessel is only replaced every three years.

Implementing these proposed changes in the maintenance system of the company in question will increase equipment reliability significantly as well as result in much better utilization of available maintenance resourc

References

ALBERTYN, M. and P. S. KRUGER. 2003. Generic building blocks for simulation modelling of stochastic continuous systems. *South African Journal of Industrial Engineering*. **14**(2), pp.47-61.

CONCANNON, H., I. HUNTER, and M. TREMBLE. 2003. SIMUL8-Planner Simulation-Based Planning and Scheduling. *In*: *2003 Winter Simulation Conference, 2003*. New Orleans:, pp.1488-1493.

KELTON, W. D., R. P. SADOWSKI, and D. T. STURROCK. 2006. *Simulation with Arena*. McGraw Hill Book Company.

KRAMER, J. and J. MAGEE. 1990. The evolving philosophers problem: dynamic change management. *IEEE Transactions on Software Engineering*. **16**(11), pp.1293-1306.

KRUGER, P. S. 2003. The art of simulation modelling. *South African Journal of Industrial Engineering*. **14**(1), pp.39-50.

MEYER, P. H. and J. K. VISSER. 2006. Improving project schedule estimates using historical data and simulation. *South African Journal of Industrial Engineering*. **17**(1), pp.27-37.

O'CONNOR, P. D., D. NEWTON, and R. BROMLEY. 2002. *Practical reliability engineering*. John Wiley and Sons.

OKE, S. A. 2004. Maintenance scheduling : description, status and future directions. *South African Journal of Industrial Engineering*. **15**(1), pp.101-117.

OKE, S. A. and O. E. CHARLES-OWABA. 2007. A Fuzzy Linguistic Approach of Preventative Maintenance Scheduling Cost Optimisation. *Kathmandu University Journal of Science, Engineering and Technology*. **1**(3).

OKE, S. A. and O. E. CHARLES-OWABA. 2005. An inflation-based maintenance scheduling model. *South African Journal of Industrial Engineering*. **16**(2), pp.123-142.

PETRESCU, M. G. and R. DUţă. 2008. PREDICTIVE MAINTENANCE TOOLS. *In*: *3rd International Conference on Manufacturing Engineering, 2008*. Chalkidiki, Greece:, pp.837-846.

SZUBINSKI, H. 2009. *Sci Fi Reality*. [online]. [Accessed 19 May 2009]. Available form World Wide Web: < HYPERLINK "http://benabb.wordpress.com/2009/05/10/bath-tub-curve-sci-fi-reality/" http://benabb.wordpress.com/2009/05/10/bath-tub-curve-sci-fi-reality/ >

VISSER, J. K. and G. HOWES. 2007. A simulation technique for optimising maintenance teams for a service company. *South African Journal of Industrial Engineering*. **18**(2), pp.169-185.

WADDELL, D. and A. S. SOHAL. 1998. Resistance: a constructive tool for change management. *Management Decision*. **36**(8), pp.543-548.

Appendix A

Attribute Name	Description
a_GGseq	An attribute that uniquely identifies an entity and is used for releasing entities from holding blocks, identifying an entity's allocated resource in a set and changing the value of a variable matrix.
a_GGseq2	An attribute that uniquely identifies an entity and is used for releasing entities from holding blocks.
a_tnow	An attribute that is used to store the current simulation time in Vessel Failure Control. This is then used to record the MTBF and MTTR values.
a_GOtime	An attribute that is used to store the entity's maintenance duration. It is used for the Total Repair Time counter as well as being sent to Excel where it is used to populate a Gantt chart of maintenance projects.
a_numGO	An attribute that is used to store the value of v_numGO and then sent to Excel where it is used for populating a graph for the whole simulation run.
a_GOdelay	An attribute that is used to store the entity's delay time until its next GO (be it mini- or major-GO).
a_failT1	An attribute that is used to store the original failure time for V1 as assigned by a distribution.
a_failT2	An attribute that is used to store the original failure time for V2 as assigned by a distribution.
a_failT3	An attribute that is used to store the original failure time for V3 as assigned by a distribution.

Variable Name	Description
v_EntityCount_GOs	A variable that is incremented in order to assign individual sequential attribute (a_GGseq) values of entities in Vessel GO's and Maintenance. The initial value is zero.
v_EntityCount_res	A variable that is incremented in order to assign individual sequential attribute (a_GGseq) values of entities in Vessel

	Resource Control. The initial value is zero.
v_EntityCount_shed1	A variable that is incremented in order to assign individual sequential attribute (a_GGseq) values of V1 entities in Vessel Control Processes (used for the single schedule in the TO-Be model as well). The initial value is zero.
v_EntityCount_shed2	A variable that is incremented in order to assign individual sequential attribute (a_GGseq) values of V2 entities in Vessel Control Processes. The initial value is zero.
v_EntityCount_shed3	A variable that is incremented in order to assign individual sequential attribute (a_GGseq) values of V3 entities in Vessel Control Processes. The initial value is zero
v_EntityCount_V1fail	A variable that is incremented in order to assign individual sequential attribute (a_GGseq) values of V1 entities in Vessel Failure Control. The initial value is zero
v_EntityCount_V2fail	A variable that is incremented in order to assign individual sequential attribute (a_GGseq) values of V2 entities in Vessel Failure Control. The initial value is zero
v_EntityCount_V2fail	A variable that is incremented in order to assign individual sequential attribute (a_GGseq) values of V3 entities in Vessel Failure Control. The initial value is zero
v_GOtype	A 1D variable matrix consisting of 40 rows. Rows are accessed through the entity's attribute a_GGseq and the individual value is used to indicate what type of maintenance activity is to be performed on the vessel in question. Values of 1 through to 6 are assigned (1 through to 5 in the case of the To-Be model).
v_EntityCount2_GOs	A variable that is incremented in order to assign individual sequential attribute (a_GGseq2) values of entities in Vessel GO's and Maintenance. The initial value is 100.
v_EntityCount2_res	A variable that is incremented in order to assign individual sequential attribute (a_GGseq2) values of entities in Vessel Resource Control. . The initial value is 100.
v_EntityCount2_shed1	A variable that is incremented in order to assign individual sequential attribute (a_GGseq2) values of V1 entities in Vessel Control Processes. . The initial value is 100.
v_EntityCount2_shed2	A variable that is incremented in order to assign individual sequential attribute (a_GGseq2) values of V2 entities in Vessel Control Processes. . The initial value is 100.

v_EntityCount2_shed3	A variable that is incremented in order to assign individual sequential attribute (a_GGseq2) values of V3 entities in Vessel Control Processes. . The initial value is 100.
v_EntityCount2_V1fail	A variable that is incremented in order to assign individual sequential attribute (a_GGseq2) values of V1 entities in Vessel Failure Control. The initial value is 100.
v_EntityCount2_V2fail	A variable that is incremented in order to assign individual sequential attribute (a_GGseq2) values of V2 entities in Vessel Failure Control. The initial value is 100.
v_EntityCount2_V3fail	A variable that is incremented in order to assign individual sequential attribute (a_GGseq2) values of V3 entities in Vessel Failure Control. The initial value is 100.
v_V1fail_time1	A 1D variable matrix consisting of 40 rows. Rows are accessed through the entity's attribute a_GGseq and the individual value is used to determine the time till failure of that specific V1 vessel. The initial value is gained from a failure distribution obtained.
v_V2fail_time2	A 1D variable matrix consisting of 40 rows. Rows are accessed through the entity's attribute a_GGseq and the individual value is used to determine the time till failure of that specific V2 vessel. The initial value is gained from a failure distribution obtained.
v_V3fail_time3	A 1D variable matrix consisting of 40 rows. Rows are accessed through the entity's attribute a_GGseq and the individual value is used to determine the time till failure of that specific V3 vessel. The initial value is gained from a failure distribution obtained.
v_downGO	A 1D variable matrix consisting of 40 rows. Rows are accessed through the entity's attribute a_GGseq and the individual value is used to indicate whether a vessel is currently online (value of 1) or whether it is being sent for a maintenance activity (value of 0).
v_numFAIL	A variable that is incremented when a vessel has failed and is currently being repaired. It is decremented when the repair is done.
v_numGO	A variable that is incremented when any maintenance project is started. It is decremented when the project is finished.
v_GOcount	A 1D variable matrix consisting of 40 rows. Rows are accessed through the entity's attribute a_GGseq and the individual value is used to determine at what stage the vessel is in the GO cycle. After a mini-GO is done, the value is incremented and after the

	major-GO, it is reset to zero.
v_equalcount	A variable that is incremented every time a duplicate entity in Vessel GO's and Maintenance passes through it. It is in turn used to determine whether an entity should go for V2 or V3 maintenance.

Appendix B

Resource creation code examples[3]

```
Sub CreateResource3()


Dim k As Integer
   Dim tel As Integer
   Dim line1 As String
   Dim line2 As String

   tel = 37
   For k = 1069 To 1078
      Model.Modules.Create "BasicProcess", "Resource", 0, 0
      line1 = "object." + Format(k)
      line2 = "210GG" + Format(tel) + "_v3"
      Model.Modules(Model.Modules.Find(smFindTag, line1)).Data("Name") = line2
      tel = tel + 1
   Next k


End Sub
```

Resource set population[4]

```
Sub CreateSet2()

   Dim k As Integer
   Dim tel As Integer
   Dim line1 As String
   Dim line2 As String

   tel = 37
   For k = 31 To 40
      line1 = "Resource Name(" + Format(k) + ")"
      line2 = "210GG" + Format(tel) + "_v2"
      Model.Modules(Model.Modules.Find(smFindTag, "object.1095")).Data(line1) = line2
      tel = tel + 1
   Next k

End Sub
```

[3] This code was replicated for all resources created. The variables were changed to reflect different names.

[4] This code was replicated for all resource sets created in the same way as that of resource creation itself.

Gantt project chart population during simulation run

```
Option Explicit
'Global Variables
Dim oSiman As Arena.SIMAN
Dim nGOtimeInd As Long
Dim nEntityNrInd As Long
Dim nRowBase As Long
Dim nTimeC As Long
Dim nGOnumInd As Long
Dim exec1 As Boolean
Dim daytel As Long

'Global Excel Variables
Dim oExcelApp As Excel.Application
Dim oWorkBook As Excel.Workbook
Dim oWorkSheet2 As Excel.Worksheet
Dim oWorkSheet As Excel.Worksheet

Sub ModelLogic_RunBeginSimulation()

   Dim k As Integer

   Set oSiman = ThisDocument.Model.SIMAN
   nGOtimeInd = oSiman.SymbolNumber("a_GOtime")
   nEntityNrInd = oSiman.SymbolNumber("a_GGseq")
   nGOnumInd = oSiman.SymbolNumber("a_numGO")
   Set oExcelApp = CreateObject("excel.application")
   oExcelApp.Visible = True
   oExcelApp.SheetsInNewWorkbook = 2
   Set oWorkBook = oExcelApp.Workbooks.Add
   Set oWorkSheet = oWorkBook.Sheets(1)
   Set oWorkSheet2 = oWorkBook.Sheets(2)
   oExcelApp.Application.ScreenUpdating = False
   With oWorkSheet
      .Activate
      .Name = "gantt"
      .Columns("B:B").Select
      .Range(Selection, Selection.End(xlToRight)).Select
      oExcelApp.Selection.ColumnWidth = 1
      .Range("A1").Select
      oExcelApp.Selection.FormulaR1C1 = "Vessel"
      For k = 1 To 40
         oExcelApp.Selection.Offset(1).Select
         oExcelApp.Selection.FormulaR1C1 = k
      Next k
```

```
    End With
    With oWorkSheet2
        .Activate
        .Name = "numGO"
    End With
    nRowBase = 1
    nTimeC = 0
    exec1 = True
    daytel = 1

End Sub

Private Sub ModelLogic_RunEndSimulation()

    oWorkBook.Sheets("numGO").Select
    oWorkSheet2.Range("A1").Select
    oWorkSheet2.Range(Selection, Selection.End(xlDown)).Select
    oExcelApp.Charts.Add
    With oExcelApp.ActiveChart
        .ChartType = xlColumnClustered
        .SetSourceData Source:=oWorkSheet2.Range("A1:A20142"), PlotBy:=xlColumns
        .SeriesCollection(1).Name = "=""Number of current projects"""
        .Location Where:=xlLocationAsObject, Name:="numGO"
    End With

    oExcelApp.Application.ScreenUpdating = True

End Sub

Function gantt250()

    Dim FindRange As Range
    Dim FoundEnd As Boolean
    Dim k As Integer

    oWorkSheet.Activate
    oWorkSheet.Range("A1").Select
    FoundEnd = False
    Do While Not FoundEnd
        Set FindRange = oExcelApp.ActiveCell.CurrentRegion
        Set FindRange = FindRange.Offset(FindRange.Rows.Count + 1).Resize(1, 1)
        FindRange.Select
        If oExcelApp.ActiveCell = Empty Then
            FoundEnd = True
        Else
```

```
        FoundEnd = False
      End If
    Loop
    oExcelApp.Selection.FormulaR1C1 = "Vessel"
    For k = 1 To 40
      oExcelApp.Selection.Offset(1).Select
      oExcelApp.Selection.FormulaR1C1 = k
    Next k
    nTimeC = nTimeC + 250
    nRowBase = nRowBase + 42
```

End Function

Private Sub VBA_Block_1_Fire()[5]

```
Dim dDelay As Double
Dim dNr As Double
Dim dTNOW As Double
Dim k As Integer
Dim dif1 As Integer
Dim dif2 As Integer
Dim dif3 As Integer

dDelay = oSiman.EntityAttribute(oSiman.ActiveEntity, nGOtimeInd)
dNr = oSiman.EntityAttribute(oSiman.ActiveEntity, nEntityNrInd)
dTNOW = oSiman.RunCurrentTime
If exec1 = True Then
  nTimeC = dTNOW - dDelay
  exec1 = False
End If
If dTNOW > nTimeC + 250 Then
  If dTNOW - dDelay < nTimeC + 250 Then
    dif1 = dTNOW - dDelay
    dif2 = nTimeC + 250 - dif1
    dif3 = dDelay - dif2
    With oWorkSheet
      .Activate
      .cells(nRowBase + dNr, dTNOW - dDelay - nTimeC + 2).Select
      .Range(oExcelApp.Selection, oExcelApp.Selection.Offset(0, dif2)).Select
      With oExcelApp.Selection.Interior
        .ColorIndex = 3
```

[5] For each of the VBA blocks in Vessel GO's and Maintenance, this same piece of code was used each time except for different fill colours and patterns.

```
            .Pattern = xlSolid
          End With
        End With
      Else
        dif3 = dDelay
      End If
      gantt250
      With oWorkSheet
        .Activate
        .cells(nRowBase + dNr, 2).Select
        .Range(oExcelApp.Selection, oExcelApp.Selection.Offset(0, dif3)).Select
        With oExcelApp.Selection.Interior
          .ColorIndex = 3
          .Pattern = xlSolid
        End With
      End With
    ElseIf dTNOW - dDelay < nTimeC Then
      dif1 = dTNOW - dDelay
      dif2 = nTimeC - dif1
      dif3 = dDelay - dif2
      With oWorkSheet
        .Activate
        .cells(nRowBase - 42 + dNr, dTNOW - dDelay - (nTimeC - 250) + 2).Select
        .Range(oExcelApp.Selection, oExcelApp.Selection.Offset(0, dif2)).Select
        With oExcelApp.Selection.Interior
          .ColorIndex = 3
          .Pattern = xlSolid
        End With
        .cells(nRowBase + dNr, 2).Select
        .Range(oExcelApp.Selection, oExcelApp.Selection.Offset(0, dif3)).Select
        With oExcelApp.Selection.Interior
          .ColorIndex = 3
          .Pattern = xlSolid
        End With
      End With
    Else
      With oWorkSheet
        .Activate
        .cells(nRowBase + dNr, dTNOW - dDelay - nTimeC + 2).Select
        .Range(oExcelApp.Selection, oExcelApp.Selection.Offset(0, dDelay)).Select
        With oExcelApp.Selection.Interior
          .ColorIndex = 3
          .Pattern = xlSolid
        End With
      End With
```

```
End If

End Sub


Private Sub VBA_Block_8_Fire()

Dim varvalue As Double

varvalue = oSiman.EntityAttribute(oSiman.ActiveEntity, nGOnumInd)
With oWorkSheet2
   .Activate
   .cells(daytel, 1).Select
   oExcelApp.Selection.FormulaR1C1 = varvalue
End With
daytel = daytel + 1

End Sub
```

Appendix C

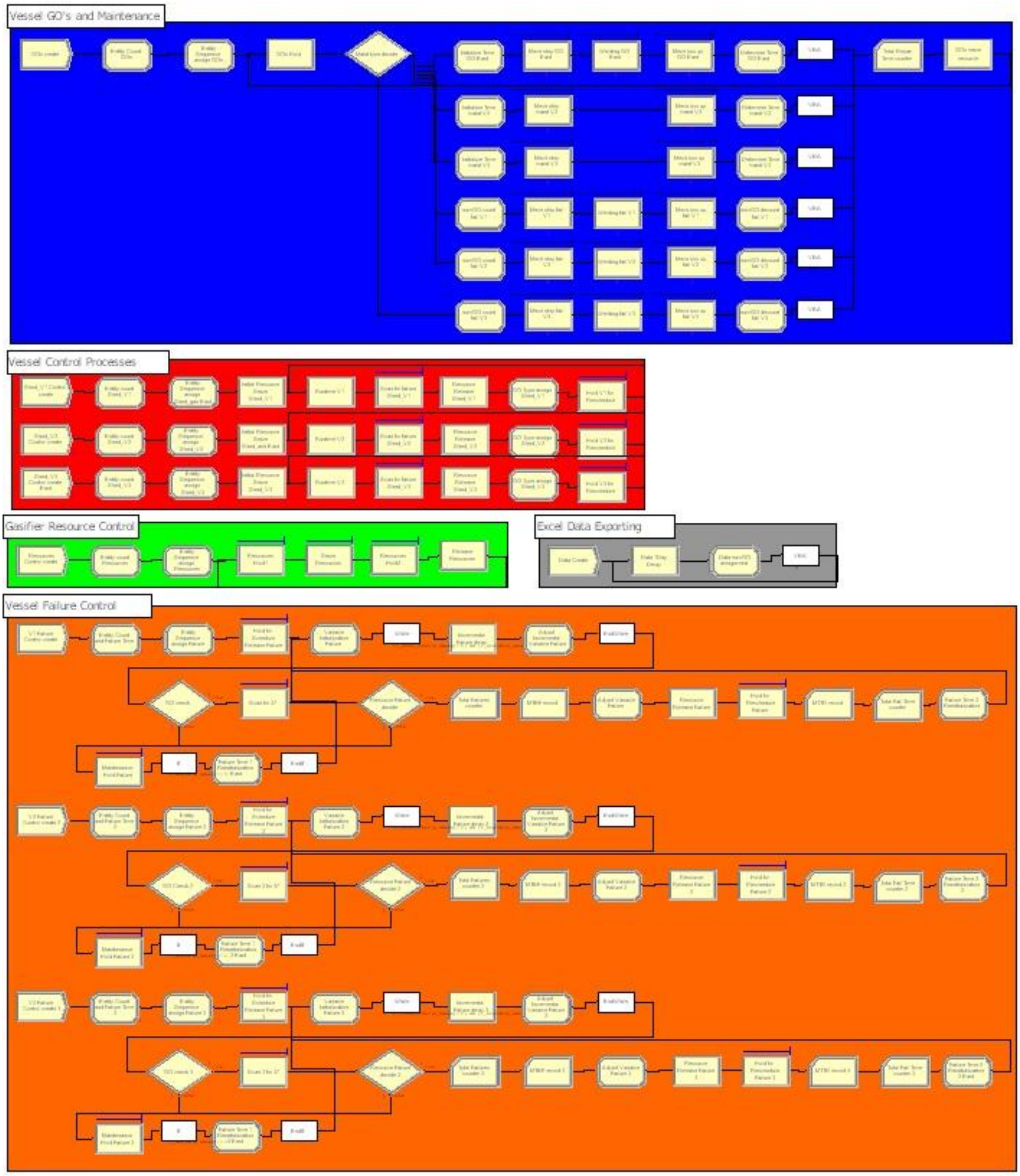

Figure 23 As-Is model whole system view

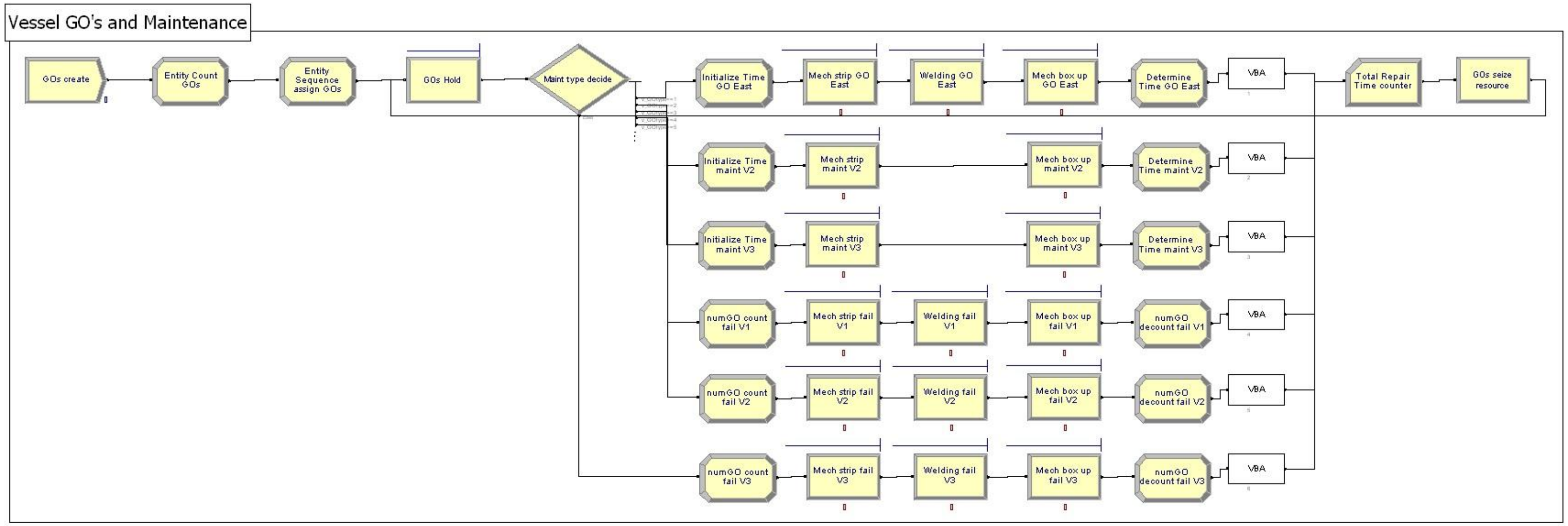

Figure 24 As-Is model Vessel GO's and Maintenance

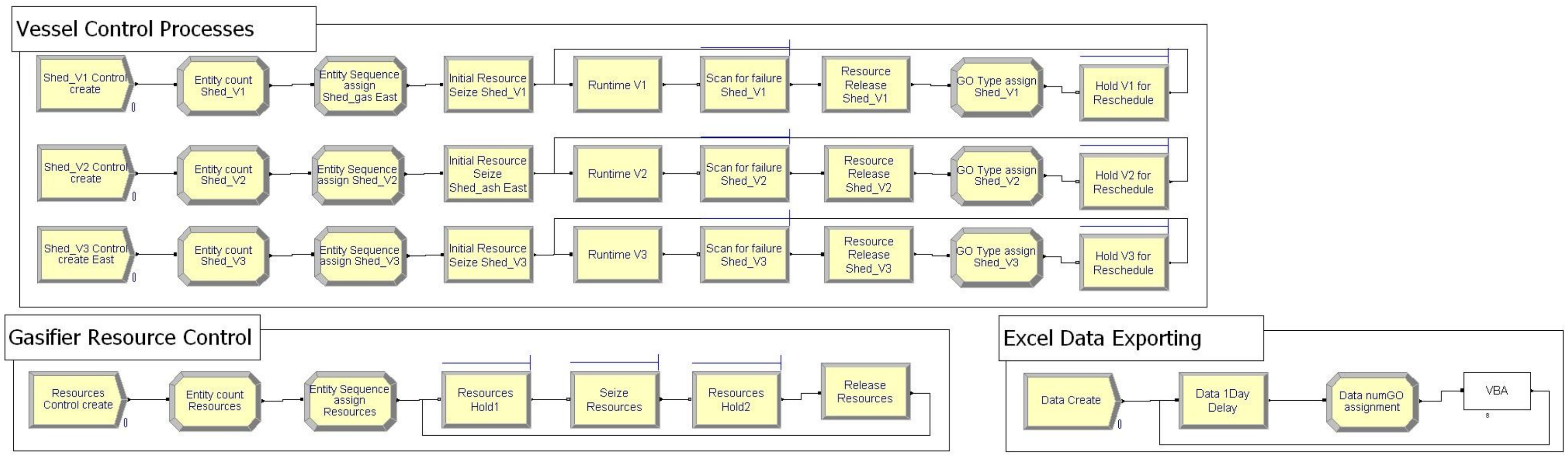

Figure 25 As-Is model control processes and data exporting

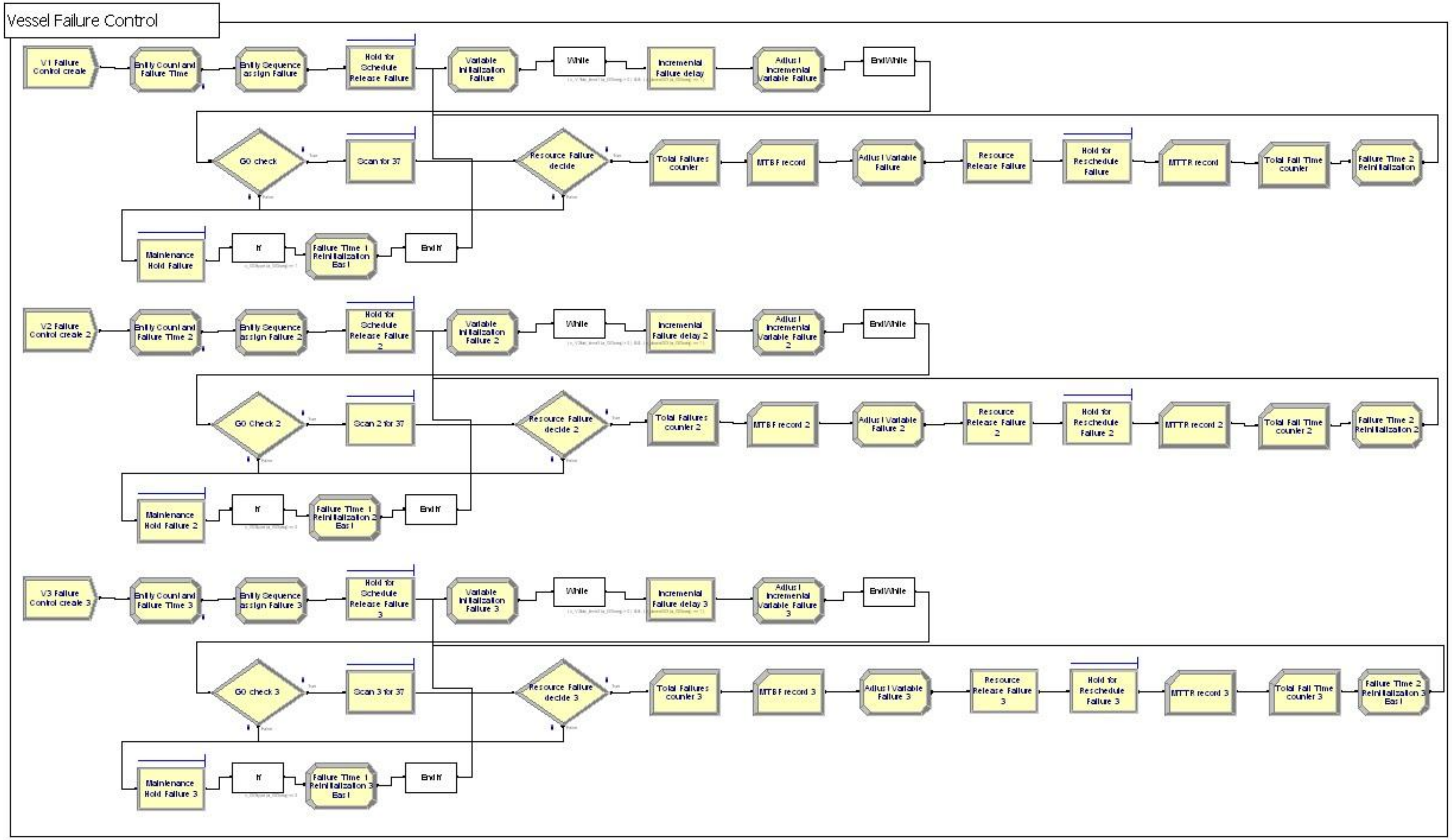

Figure 26 As-Is model Vessel Failure Control

Appendix D

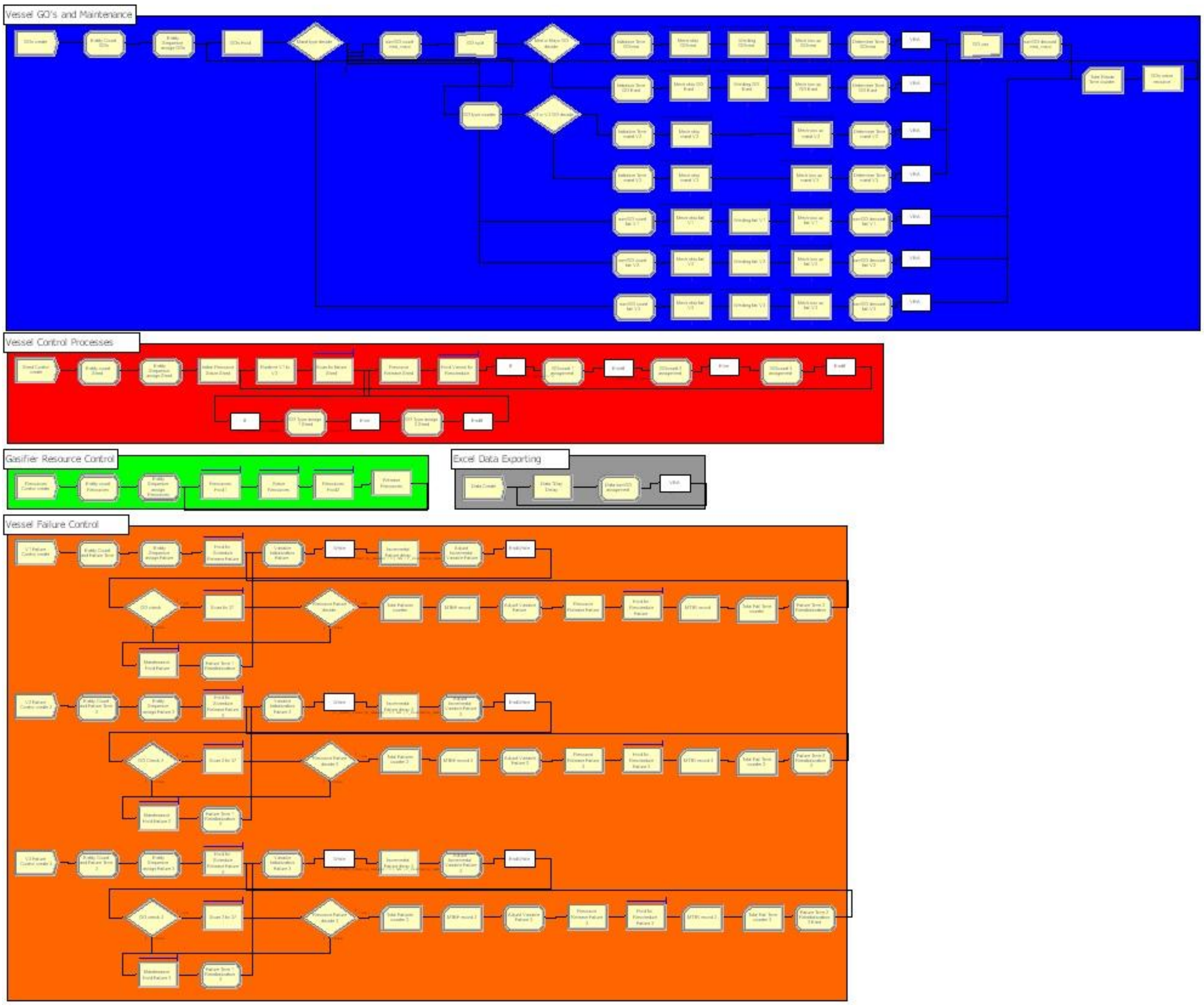

Figure 27 To-Be model whole system view

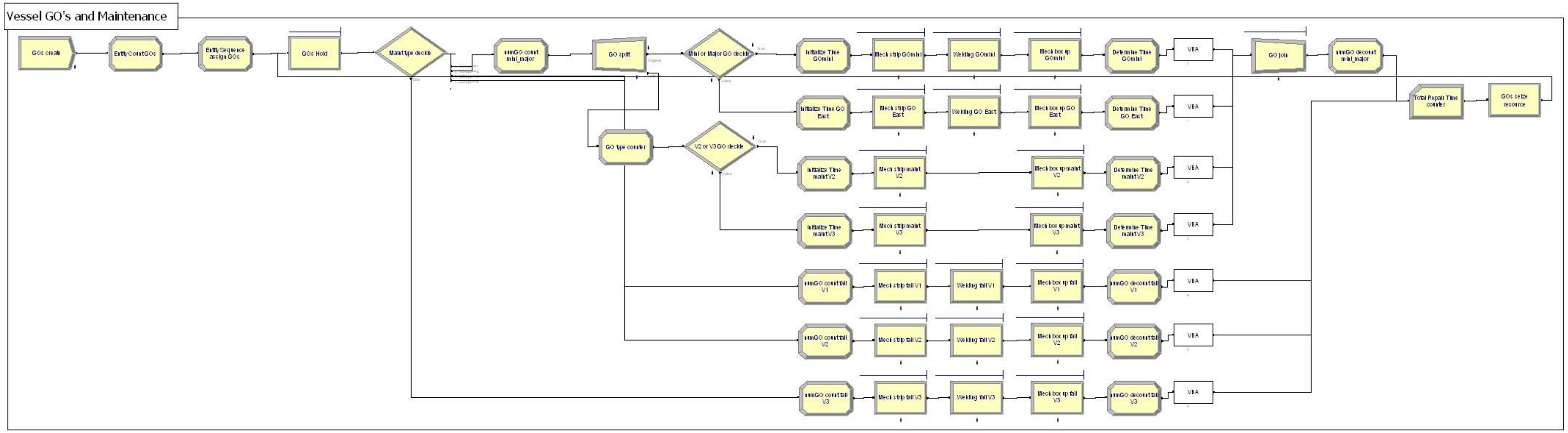

Figure 28 To-Be model Vessel GO's and Maintenance

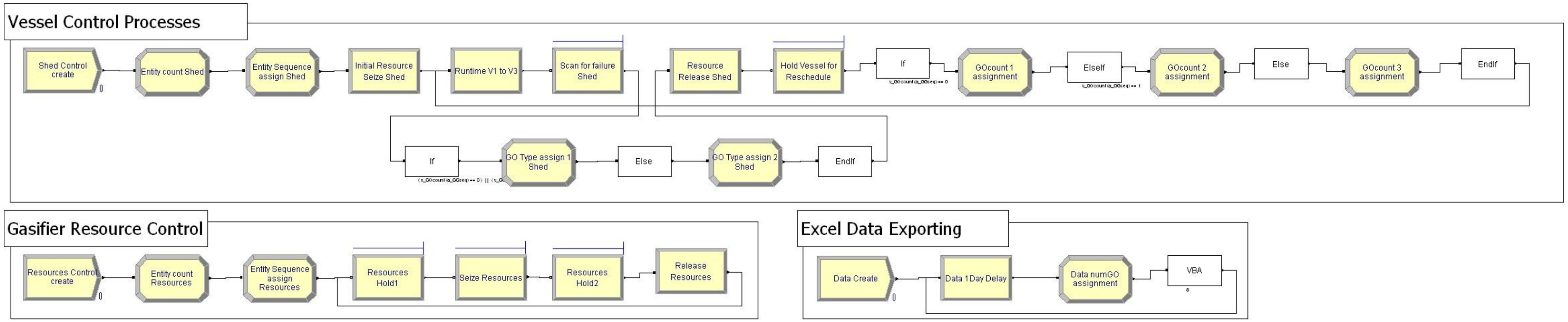

Figure 29 To-Be model control processes and data exporting

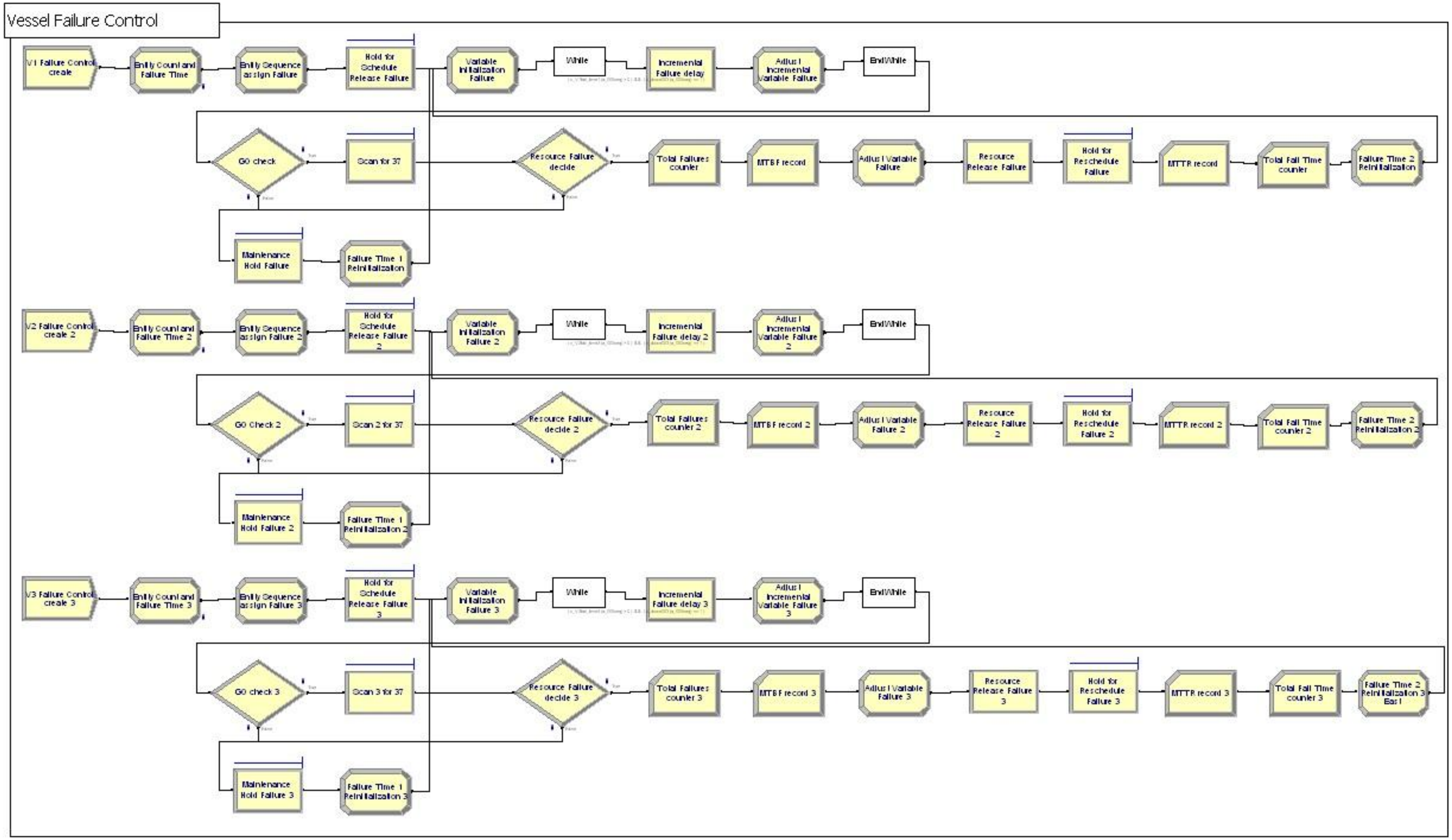

Figure 30 To-Be model Vessel Failure Control

Appendix E

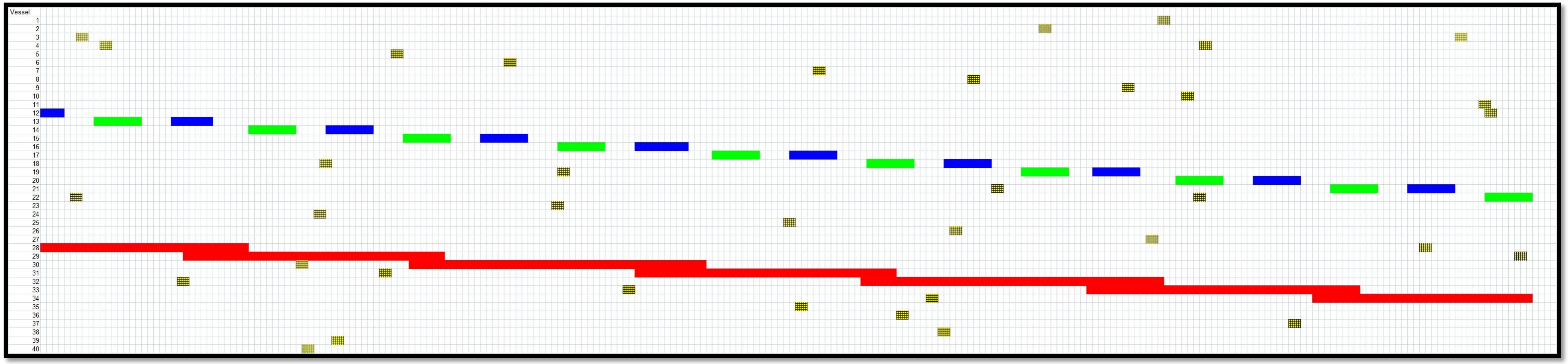

Figure 31 Abstract from the As-Is model's generated Excel Gantt chart

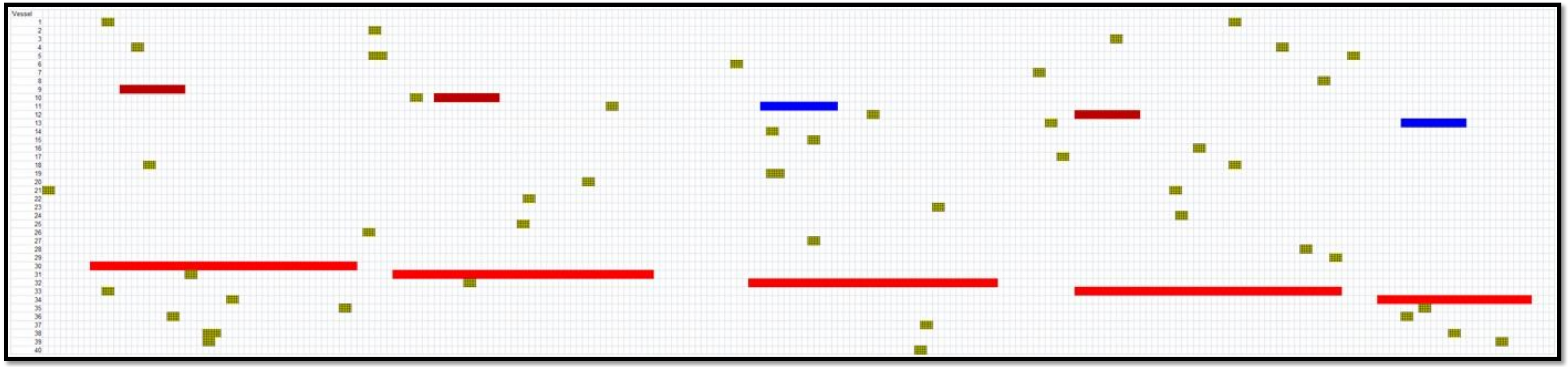

Figure 32 Abstract from the To-Be model's generated Excel Gantt chart

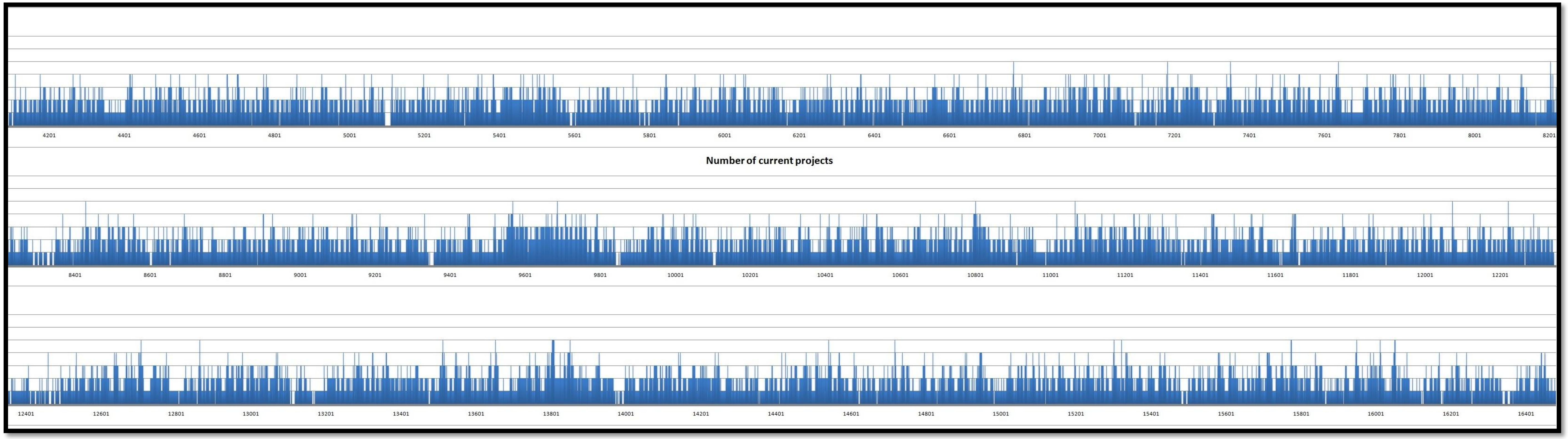

Figure 33 Abstract from the As-Is model's Excel graph for the number of current maintenance projects over time (days)

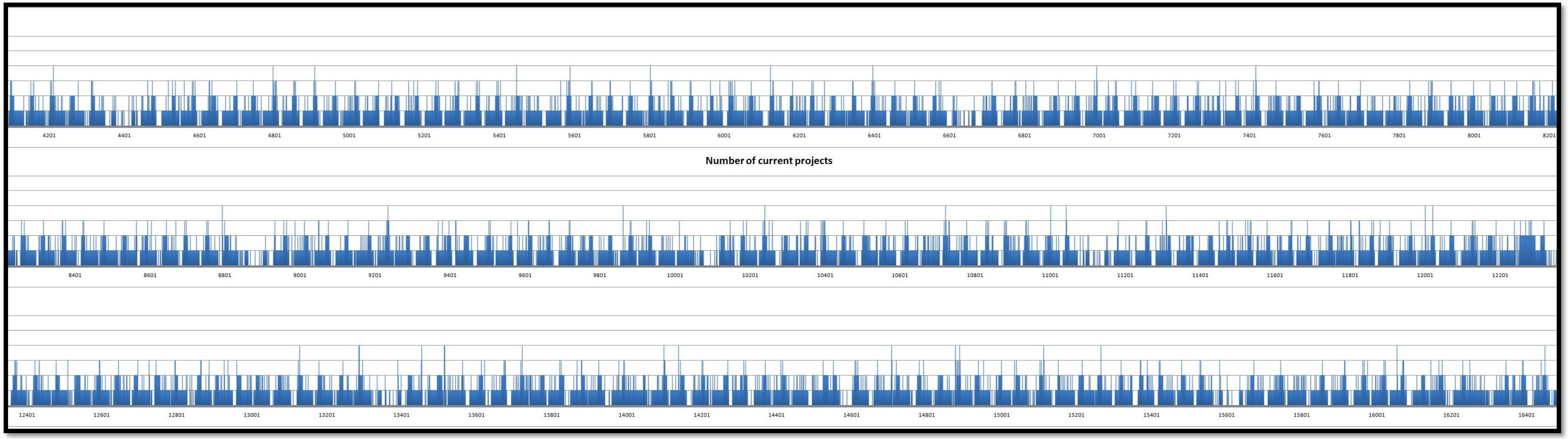

Figure 34 Abstract from the To-Be model's Excel graph for the number of current maintenance projects over time (days)