Delft University of Technology

FACULTY MECHANICAL, MARITIME AND MATERIALS ENGINEERING
Department Marine and Transport Technology

Mekelweg 2
2628 CD Delft
the Netherlands
Phone +31 (0)15-2782889
Fax +31 (0)15-2781397
www.mtt.tudelft.nl

Specialization:	Transport Engineering and Logistics
Report number:	2012.PEL.7712
Title:	**Performance improvement of a caravan storage-system**
Author:	A.R. Dorrepaal

Title (in Dutch)	Prestatieverbetering van een caravanstallingssysteem

Assignment:	computer assignment
Confidential:	yes (until 31-12-2013)
Initiator (university):	-
Initiator (company):	-
Supervisor:	Dr. Ir. H. P. M. Veeke
Date:	July 15, 2012

Delft University of Technology

FACULTY OF MECHANICAL, MARITIME AND MATERIALS ENGINEERING
Department of Marine and Transport Technology

Mekelweg 2
2628 CD Delft
the Netherlands
Phone +31 (0)15-2782889
Fax +31 (0)15-2781397
www.mtt.tudelft.nl

Student:	A.R. Dorrepaal
Supervisor (TUD):	Dr.ir. H.P.M. Veeke
Supervisor (Company)	-

Assignment type:	Computer
Creditpoints (EC):	10
Specialization:	PEL
Report number:	2012.PEL.7712
Confidential:	Yes, until 31-12-2013

Subject: **Performance improvement of a caravan storage-system**

Introduction

In order to reduce space loss at caravan storage company's caravans are stored close to each other, often in rows. A lot of caravan storage company's do not make use of stacking optimization due to their small scale. For large scale caravan storage company's there might be opportunities to optimize the sequence of caravans in rows. In this case it is assumed that caravans are stored in rows with a maximum capacity of four caravans per row and with a capacity of the caravan-park of two thousand caravans.

Problem

The caravan storage system is a suboptimal system. But to improve or substitute the current system, investments are required. These investments bring costs, while it is not clear in what magnitude performance will be improved. In order to be able to quantify the improvements a simulation model is required. With the results a more accurate investment decision can be made.

Delft University of Technology

FACULTY OF MECHANICAL, MARITIME AND MATERIALS ENGINEERING
Department of Marine and Transport Technology

Mekelweg 2
2628 CD Delft
the Netherlands
Phone +31 (0)15-2782889
Fax +31 (0)15-2781397
www.mtt.tudelft.nl

Assignment

The assignment is to build a simulation model of a caravan storage system in order to be able to quantify possible performance improvements. In this model, the current scenario and possible to-be scenarios shall be applied and the performance of these scenarios will be investigated.

Execution

1. Investigate current caravan storage system
2. Create a simulation model
3. Develop an experimental plan
4. Verify the simulation model
5. Interpret results
6. Study relevant literature

The supervisor,

Dr. ir. H.P.M. Veeke

Table of contents

Table of contents II
Preface IV
Summary VI
Summary (Dutch) VIII

Introduction 1
General introduction 1
Goal 3

1. Conceptual model 5
1.1 Performance indicators 5
1.2 Parameters 5
1.3 conceptual model 6

2. PDL 11
2.1 Classes and attributes 11
2.2 Processes 12
2.3 Initiation 15

3. Experiments 17
3.1 Input 17
3.2 Output 21
3.3 Experimental plan 23

4. Verification and results 25
4.1 Calculation 25
4.2 Model 26
4.3 Verification 27
4.4 Results 29

Conclusion 31
Conclusion 31
Recommendations 31

Appendix A 33

Preface

This report was made based on a simulation assignment. The assignment is executed in the last year of the study Production Engineering and Logistics. The report gives clear insight in the goal of the simulation, the model and the results.

In this report it is investigated whether the performance of a real life caravan storage system can be improved or not. To realize this, the current situation is simulated and alternatives are assessed.

In chapter 1 the conceptual model is developed. In chapter 2 the PDL is shown. In chapter 3 it is explained what are the input parameters and which experiments will be executed. In chapter 4 the model is verified by tracing and manual calculations and the results of the executed experiments are shown. Finally the conclusions are presented which can be derived from the results.

Summary

In order to reduce space loss at caravan storage company's caravans are stored close to each other, often in rows. A lot of caravan storage company's do not make use of stacking optimization due to their small scale. For large scale caravan storage company's there might be opportunities to optimize the sequence of caravans in rows.

In this case it is assumed that caravans are stored in rows with a maximum capacity of four caravans per row and with a capacity of the caravan-park of two thousand caravans.

The caravans are ordered by the customers and the caravans are retrieved from storage by the CTV (Caravan Transport Vehicle) based on the orders. After being on holiday, the customer returns the caravan to the company. The arrived caravans wait to be stored by the CTV again. The process starts again after being stored until the next order is received for that caravan.

The order flow does have a strongly seasonal pattern, wherein busy periods arise during summer, and relatively quiet periods during winter. The amount of orders is lognormal distributed over the customers. Also, the length of the caravans is taken into account. The length does have a normal distributed pattern.

Retrieval and storage by the CTV can be done in different manners. Currently, the next leavetime of the caravan is taken into account by means of colored wires. Nevertheless, this is only the case at the second half of the year. At the first half of the year, the caravans are stored without taking the next leavetime into account. The only constraint is the length of the caravan.

A simulation model has been built to assess the performance of the current scenario. Also performances have been determined of two other scenarios. The performance is measured in terms of the amount of pick-up moments. The first scenario is based on the way of working of the first half year of the current scenario. No rearranging of rows takes place, and caravans are placed in a row, by only taking into account the length of the caravan. In the second (TO-BE) scenario it is assumed that the next leavetime of the caravan is known. Based on a length and time constraint, the caravans are selected from the arrived caravans, the relating row is rearranged when necessary and the caravans end up stored in the row.

The scenario of 'no rearranging rows' is manually calculated. Based on this calculation, tracing and visualization, the simulation model is verified.

From the results of the experimental plan it can be concluded that the TO-BE scenario does result in the highest performance increase, although the 'no rearranging rows' scenario approaches the TO-BE

scenario as second best. Both do show a clearly difference in performance compared to the AS-IS situation.

Summary (Dutch)

Om ruimteverliezen te besparen slaan caravanstallingen hun caravans vaak dichtgepakt op elkaar op, vaak in rijen. De meeste caravanstallingen maken geen gebruik van optimalizatie, vanwege de kleine schaal waarop ze opereren. Echter, voor de grotere caravanstallingen zijn er wellicht mogelijkheden tot het optimalizeren in de volgorde van stallen van de caravans in de rijen.

In dit geval is aangenomen dat het caravanpark bestaat uit rijen met een maximum capaciteit van vier caravans per rij, en dat het caravanpark in totaal een capaciteit heeft van tweeduizend caravans.

The caravans worden besteld door de klanten en vervolgens worden de caravans uit de caravanstalling gehaald door de CTV (Caravan Transport Vehicle), aan de hand van de klantorders. Nadat de klanten op vakantie zijn geweest retourneren ze hun caravan aan de stalling. Vervolgens wachten de gearriveerde caravans om opnieuw gestald te worden door de CTV. De caravans worden gestald totdat er weer een nieuwe order volgt. Hierna begint het proces weer van voren af aan.

De bestelstroom heeft een sterk seizoensgebonden patroon. Hierin komt duidelijk naar voren dat er in de zomer relatief drukke perioden voorkomen en in de winter rustige perioden. Het aantal orders is lognormaal verdeeld over de klanten. Ook de lengte van de caravans wordt in de simulatie meegenomen. De lengte heeft een sterk normaal verdeeld patroon.

Het klaarzetten en stallen van de caravans door de CTV kan op verschillende manier gerealiseerd worden. In de huidige situatie wordt de volgende vertrekdatum van de caravans meegenomen aan de hand van gekleurde draden. Dit wordt echter alleen toegepast in de tweede helft van het jaar. In de eerst helft van het jaar worden de caravans opgeslagen zonder rekening te houden met de volgende vertrekdatum van de caravans. In dit geval is de lengte van de caravan de enig beschouwde parameter.

Een simulatiemodel is gecreeerd om de prestaties van het huidige scenario te toetsen. Daarnaast worden ook de prestaties van twee andere scenarios bepaald. De prestatie wordt bepaald aan de hand van het aantal benodigde aanpikmomenten. Het eerste scenario is gebaseerd op de manier van werken zoals die van de AS-IS situatie gedurende het eerste half jaar. Er vindt geen herschikking van rijen plaats en het stallen van de caravans wordt alleen uitegevoerd aan de hand van de geschikte lengte van de caravan. In het tweede (TO-BE) scenario wordt aangenomen dat men ten alle tijde op de hoogte is van het volgende vertrekmoment van een caravan. Aan de hand van de lengte en het volgende vertrekmoment worden caravans geselecteerd uit de gearriveerde caravans. Vervolgens vindt er een herschikking van de rij plaats indien nodig en worden daarmee ook gelijk de geselecteerde caravans gestald.

Het scenario waarin geen herschikking van de rijen plaatsvindt is doorgerekend. Op basis van deze berekening, visualizatie van de simulatie en 'tracing' is het simulatiemodel geverifieerd.

Uit de resultaten van de experimenten kan geconcludeerd worden dat het TO-BE scenario tot de hoogste prestaties leidt. Het scenario waarin de rijen niet herschikt worden, benadert de prestaties van het TO-BE scenario. Beide scenario's tonen een groot verschil in positieve zin in vergelijking tot het huidige scenario.

Introduction

General introduction

The caravan park has a floor area of about 25000m2. In this area there are about 2000 caravans stored. The storage of these caravans is situated as follows.

The caravans are stored in rows. Every row has a capacity of 4 caravans. With 2000 caravans in total, and with the knowledge of the caravan storage to be full during winter time, 500 rows are needed to have enough capacity for the 2000 caravans.

The caravans are moved by a 'Caravan Transport Vehicle' a CTV. This moving of the caravans is done by two persons. One person is driving the CTV and the other person walks behind the caravans to instruct the driver and guide the caravan in the right way without colliding the storage building or other caravans.

The caravan storage company serves people who want to go on holiday with their caravan and therefore the company is a company with a seasonal pattern in the activity of the customers. During winter time the storage is completely filled, and often some extra caravans are left and put on some different places at the company. There is no activity. During the spring, the sheds become more and more empty. Customers order their caravans and this order pattern is at its maximum during the beginning of the summer holidays. At the end of the summer holidays, customers massively return their caravans and the storage becomes full again, heading to winter.

During the season, a lot of movements are necessary to place caravans outside the storage when customers order their caravans. Also the return of the caravans requires a lot of movements. Due to the fact that the storage needs to be completely filled in the winter, often caravans in a row are interchanged by the CTV to put them in the right order. The order of the caravans in a row is determined by a system of marks which is explained below.

Currently two situations can be distinguished regarding the storage of the caravans. During the spring until mid summer the caravans are just put back in front of a random row. After the summer, when a caravan arrives at the company, the caravan gets a mark. These marks are green, brown or black. The meaning of these colors is as follows:

Green – Keep the caravan in front, because the customer will order the caravan at least one time more this year.

Brown – Put the caravan in the middle of the row, because the customer is not sure whether to order the caravan again this year or not.

Black – Put the caravan at the back of the row, because the customer will order the caravan again in spring next year.

As one can understand, in the late summer and fall a continuous interchange of caravans in rows takes place, to put the caravans in the right order according their mark. It is questionable whether this current mark system is the optimal way to fill the storage again for the winter.

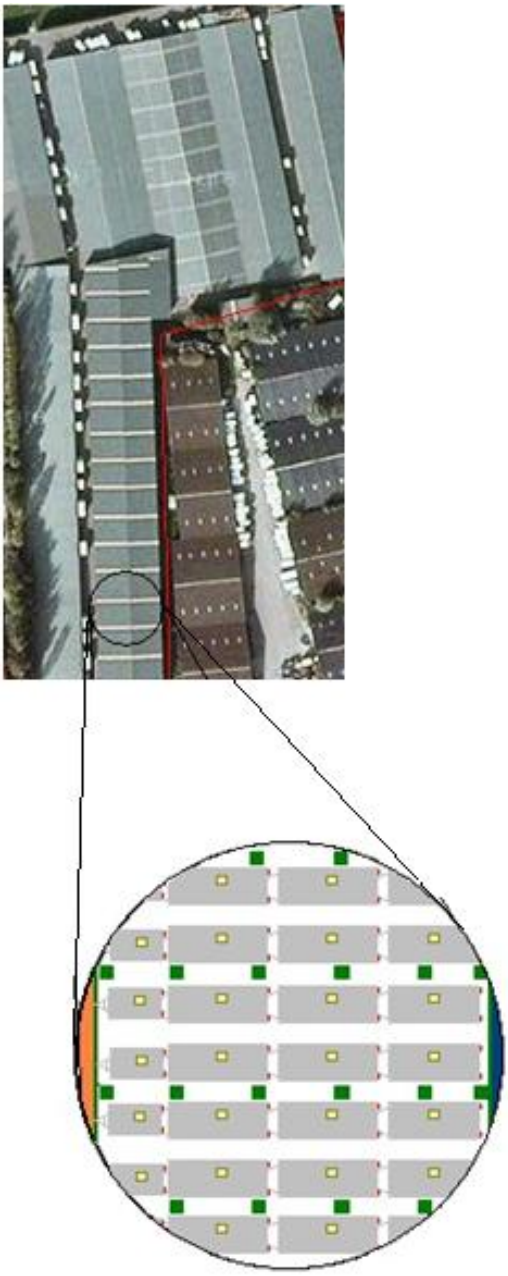

Figure 1 – schematic depiction of the rows

Goal

In the current system the rows continuously need to be adjusted in the right order. This system requires a lot of labor and do not seem to be the optimal way of working.

Currently all information regarding the location of the caravans and the orders of the customers is registered and handled manually (paperwork). By using a system to register the required information of the caravans, there can be made use of the benefit that for every decision moment a decision can be made based on all available data in the IT system. This system requires an investment. Therefore it is interesting to know the performance increase of the handling of the caravans when install a new registration system. This performance needs to be measured in terms of pickup moments. Regarding this, the goal of the simulation is as follows:

> *"Determine the performance of the current storage system and investigate possibilities to increase this performance."*

To realize this, it is necessary to simulate the current situation and compare this current situation to a new situation, in which it is assumed that on every moment all information regarding the caravans is available. By means of this available data a decision can be made in order to store the caravan on the most optimal location.

1. Conceptual model

1.1 Performance indicators

The performance of the system should be measured. Knowing performance numbers, decisions can be made whether the design parameters should be varied and which combination is best. The objective of this simulation project is to obtain a decrease in pick-up moments.
Therefore the relevant performance indicators are:

Amount of pick-up moments [#]
Amount of pick-up moments in the fourth quarter [#]

The pick-up moments are simply counted by every time the CTV is picking a caravan to move that caravan. In this indicator also taken into account are the movements which are necessary to obtain a caravan which is not directly reachable by the CTV, but for example is located in the back of a row. There are more performance indicators, but these are not directly relevant to the goal of the simulation project. These parameters are listed in paragraph 2.2, model output.

1.2 Parameters

A list of all parameters in the system is evaluated (Table 1). There is a difference in parameters which are a (fixed) constraint, parameters which are distributed and parameters which will be varied in the experiments.
The parameters are divided into two groups. The first group of parameters depends on the configuration we can choose; the second group of parameters depends on the given input of the system.

In this experiment the main parameter to change is the parameter 'scenario', in which can be chosen between the as-is situation and the to-be situation. The number of rows is assumed to be a constraint, as in real the caravan storage has about 500 rows. In every row 4 caravans are placed. Because the storage is exploited at full capacity, it is assumed that the amount of caravans is 4 times the amount of rows.

Besides of the experiments with the scenarios, also the influence of the remaining parameters in the configuration group will be used, to discover their influence on the results.

In the input part of the table, constraints are defined. These constraints are properties of the caravans and the rows.

Group	Description	Abbreviation
Configuration	Scenario	Scenario
	Number of Rows	# of Rows
	Allowed difference-percentage of required caravan length	%length difference
	Start of autumn (start composing rows)	Store in rows from week
	Maximum length of ACQueue after end of day	Store until ACQueue <
	Percentage of customers that tells color correct/incorrect	-
Input	Leavetime of caravan	Leavetime
	Arrivaltime of caravan	Arrivaltime
	Upper works length of caravan	UWLength
	Orderamount of caravan	Orderamount
	Length of row (# of caravans)	Rowlength
	Length of row (meter)	Totallength

Table 1 - system parameters

1.3 conceptual model

The model exists of five component classes. The main component class is the Caravan Transport Vehicle, the CTV. The CTV moves the caravans in and outside the storage. This storage exists of rows, in which the caravans are placed. The list of ordered caravans is composed every day again by the component class orderlist. When the caravans are retrieved from storage, the customer takes the caravan on holiday until the arrivaltime of the caravan, generates a new leavetime and arrivaltime for the caravan and returns the caravan in the ACQueue. This main process is shown in the figure below.

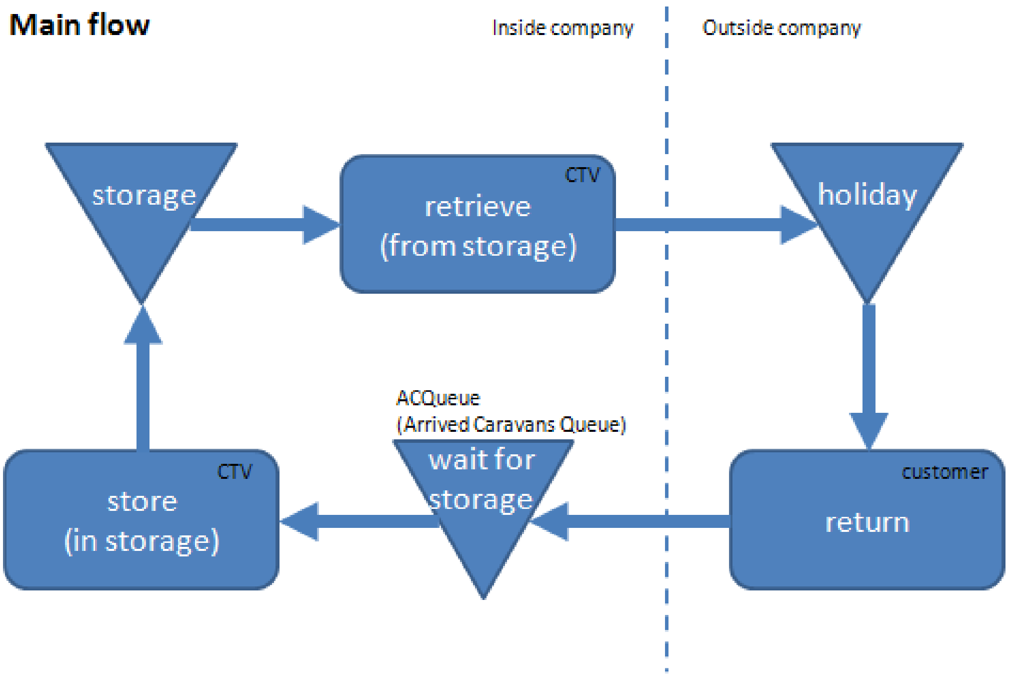

Figure 1 - main flow

Caravan

The caravan contains multiple attributes. It contains a UWLength, which gives the value for the Upper works length of the caravan, needed to decide whether the caravan fits into the row or not. Furthermore the attributes leavetime and arrivaltime gives the values of the days of the next leavetime from the storage of the company to the holiday and succeeding arrivaltime to the ACQueue of the company.

Row

The row has the attribute RowQueue, in which the caravans are stored. The attribute Caravanslength gives the sum of the length of the caravans stored into the RowQueue. The attribute Totallength gives the value of the totallength of the row. In this simulation the average of all Upper Works Lengths of all caravans is chosen, which is 18 meter. This average is chosen because theoretically all caravans must be fit in the whole storage of the company during winter.

The classes CTV, orderlist and customer contain processes, in which the process of the orderlist is not shown in the figure above. The orderlist generates a list of ordered caravans every day, by searching the rows if it contains caravans which have a leavetime within two days of Tnow. This process of the orderlist is shown in the figure below.

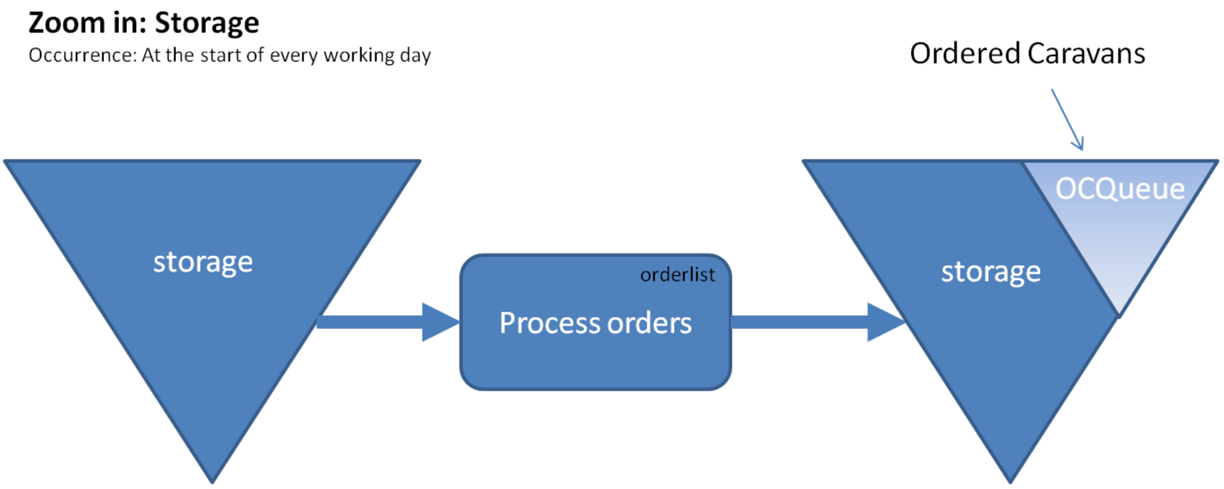

Figure 2 - creation of OCQueue

Customer

The customer searches in the HolidayQueue which caravans need to return to the company within a time period of 2 days from Tnow. After that, the next leavetime of this caravan is searched in the database. The caravan gets this leavetime and with the value of this leavetime and the leavetime after the next leavetime, the arrivaltime is calculated. This is done with the choice of a value from a uniform distribution between these two leavetimes. Then, for the as-is situation, the customer process also gives a color to the caravan, which gives an indication for the next ordermoment from the customer. In the form the percentages of correct and incorrect colors can be chosen. After these ‘administration’ steps, the caravan is returned to the company and has entered the ACQueue of arrived caravans.

CTV

The CTV stores the caravans in the storage and retrieves the caravans from the storage. During springtime more caravans are retrieved than stored and therefore the caravans which are returning from holiday in spring won’t receive a colored wire and won’t be stored by considering the next leavetime of the caravan. In spring a workday of the CTV exists of: first retrieving the ordered caravans and after that, storing the arrived caravans from ACQueue.

In the autumn the CTV works in a different manner. Three manners are possible:

- Continue as the CTV does in spring.
- AS-IS (use colored wires for selection)
- TO-BE (use next leavetimes for selection)

In the case of 2 and 3 the way of working is as follows. When an ordered caravan is found, that caravan is retrieved and the caravans in front of this retrieved caravan are placed outside. After that, the CTV searches in ACQueue for a correct caravan to put in that row. The amount of pick up moments is counted, including the pick ups for the caravans in front of the retrieved caravan (twice). When there are still a lot of caravans waiting in the ACQueue and all ordered caravans are retrieved, the CTV tries to find a row for every arrived caravan to store it in. This is done without the arranging of the rows itself. The arrived caravans must be fit in front of the row. When after this cycle there are still caravans waiting to be stored in ACQueue, the CTV searches for the emptiest row, pulls out as much caravans as needed, fills the row till it’s full with selected caravans from ACQueue, and rearrange the row in the best configuration, dependent of the leavetimes of the caravans.

Regarding the first manner, the selection of the arrived caravans is only based on the caravanlength. In the second manner (AS-IS), the selection of the arrived caravans is based on the caravanlength and the colored wires. When executing the third manner (TO-BE), the selection of the arrived caravans is based on the caravanlength and the leavetime of the caravan.

The selection based on the length of the caravan is as follows. A maximum required length difference (in percents) is filled in the form. With that length difference a calculation is made if the caravan is sufficient to select or not.

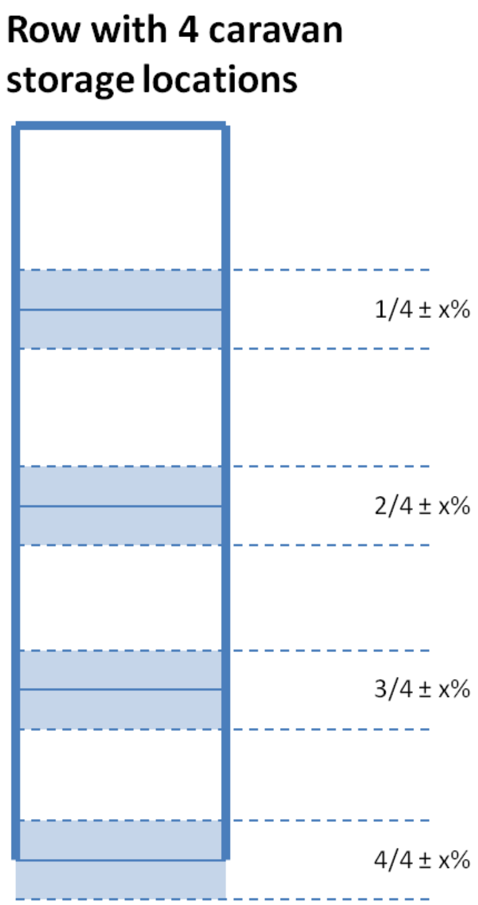

Figure 3 - schematic view of row

Schematically it is shown in the figure above. When, for example, already two caravans have been stored in the row, the sum of the lengths of the two caravans in the row and the length of the new caravan gives a length which must be into the range as marked blue in the figure above. This means that it is possible for a row to receive a long caravan, but only when there are short caravans in the row already.

2. PDL

2.1 Classes and attributes

CTV	= Class	
Orderlist	= Class	
Customer	= Class	
Caravan	= Class	
Row	= Class	
CTV (active)		
- Handlingtime	: Integer	// Handling time of CTV (value = 1)
- Process		
Orderlist (active)		
- Process		// Process creates OCQueue
Customer (active)		
- Process		// Process returns caravans to company
Caravan (passive)		
- CaravanNumber	: Integer	// Number of the caravan
- UWLength	: Double	// Upper Works length of the caravan
- Leavetime	: Integer	// Leavetime of the caravan
- Arrivaltime	: Integer	// Arrivaltime of the caravan
- Color	: Integer	// color of the caravan (green, none, brown or black)
Row (passive)		
- RowNumber	: Integer	// Number of the row
- RowQueue	: TomasQueue	// Queue for storing caravans
- Caravanslength	: Double	// Sum of the Upper Works length of the caravans
- Totallength	: Double	// Total length of the row (= 18m)
- firstleavetime	: Integer	// The first leavetime of the caravans in Row
- firstcolor	: Integer	// The first color of the caravans in Row
Global objects		
Storage	: TomasQueue	// Queue of all caravans in storage
HolidayQueue	: TomasQueue	// Queue of all caravans on holiday
OCQueue	: TomasQueue	// Queue of all ordered caravans
ACQueue	: TomasQueue	// Queue of all arrived caravans
AllrowsQueue	: TomasQueue	// Queue of all rows
AllCaravansQueue	: TomasQueue	// Queue of all caravans
TempQueue	: TomasQueue	// Temporary Queue used to sort selected rows
justoutsideQueue	: TomasQueue	// Queue for storing caravans while rearranging Row

2.2 Processes

CTV Process

{Retrieves and stores caravans from and to storage}

```
Repeat
        Wait 1
                If spring then
                        {Retrieve}
                        While OCQueue.length > 0 do
                        Find caravan in allRowsQueue
                        Add caravan to holidayQueue
                        Determine firstleavetime of Row
                        Calculate # of pick ups

                        {Store}
                        For i= 1 to ACQueue.length
                                For j = 1 to allrowQueue.length
                                        If caravan.length = OK
                                                Add row to TempQueue

                                Select emptiest Row from TempQueue
                                Add caravan to Row
                                Determine new Row.firstleavetime
                                Calculate # of pick ups

                If autumn then
                        If AS-IS
                                {Retrieve & Store}
                                While OCQueue.length > 0 do
                                   Find caravan in allRowsQueue
                                   Put caravans in front just outside
                                   Determine firstcolor of Row
                                   For i= 1 to ACQueue.length
                                        If caravan.length = OK and Row.firstcolor > caravan.color
                                                Add caravan to TempQueue

                                        Select caravan from TempQueue with lightest color
                                        Add selected caravan and caravans 'just outside' in Row at right
                                        order
                                        Determine new Row.firstcolor
                                        Calculate # of pick ups

                                {Store without removing caravans from Row}
                                For i= 1 to ACQueue.length
                                        For j = 1 to allrowQueue.length
                                                If caravan.length = OK and Row.firstcolor > caravan.color
                                                        Add row to TempQueue

                                                Select emptiest Row from TempQueue
```

```
                    Add caravan to Row
                    Determine new Row.firstcolor

               Calculate # of pick ups


          {Store with removing caravans from Row}
          Repeat until ACQueue < x            //x is filled in form
            Select emptiest Row
            For i= 1 to ACQueue.length
               If caravan.length = OK
                    Add caravan to TempQueue

               Select caravan from TempQueue with lightest color
               Add caravan to Row sorted
               Count amount of caravans in front
               Calculate # of pick ups
               Determine new Row.firstcolor

     If TO-BE
          {Retrieve & Store}
          While OCQueue.length > 0 do
            Find caravan in allRowsQueue
            Put caravans in front just outside
            Determine firstcolor of Row
            For i= 1 to ACQueue.length
               If caravan.length = OK and Row.firstleavetime > caravan.leavetime
                    Add caravan to TempQueue

               Select caravan from TempQueue with latest leavetime
               Add selected caravan and caravans 'just outside' in Row at right
               order
               Determine new Row.firstleavetime
               Calculate # of pick ups

          {Store without removing caravans from Row}
          For i= 1 to ACQueue.length
               For j = 1 to allrowQueue.length
                    If caravan.length = OK and Row.firstleavetime >
                    caravan.leavetime
                         Add row to TempQueue

                    Select emptiest Row from TempQueue
                    Add caravan to Row
                    Determine new Row.firstleavetime
                    Calculate # of pick ups


          {Store with removing caravans from Row}
          Repeat until ACQueue < x            //x is filled in form
          Select emptiest Row
          For i = 1 to ACQueue.length
               If caravan.length = OK
```

```
                    Add caravan to TempQueue

                Select caravan      from TempQueue with latest leavetime
                Add caravan to Row sorted
                Count amount of caravans in front
                Calculate # of pick ups
                Determine new Row.firstleavetime
```

Orderlist Process

{Generates orderlist ‘OCQueue’ of caravans which are ordered to go on holiday}

```
Repeat
    Wait 1
    For i= 0 to storage.length-1
        If (myCaravan.leavetime – Tnow) < 2 then
            OCQueue.add(myCaravan)
    For j = 0 to ACQueue.length-1
        If myCaravan.leavetime – Tnow) < 2 then
            ACQueue.remove(myCaravan)
            OCQueue.add(myCaravan)
```

Customer Process

{Returns caravans from Holiday to ACQueue by selecting the caravans in Holiday which approaches the arrivaltime. Also generates new leavetime and arrivaltime of returning caravans.}

```
Repeat
    Wait 1
    For i= 0 to holiday.length-1
        If (myCaravan.arrivaltime – Tnow) < 2 then
            ACQueue.add(myCaravan)
                Search myCaravan in database
                Search current myCaravan.leavetime in database
                Search next leavetime in database

        Adjust myCaravan.leavetime
        Calculate next arrivaltime
        Adjust myCaravan.arrivaltime
        Determine color
        Adjust myCaravan.color
        HolidayQueue.remove(myCaravan)
```

2.3 Initiation

Simulation initiation

```
{All the element classes need to be initiated}
HolidayQueue            = TomasQueue.Create('HolidayQueue')
Storage                 = TomasQueue.Create('StorageQueue')
OCQueue                 = TomasQueue.create('OrderedCaravansQueue')
ACQueue                 = TomasQueue.create('ArrivedCaravansQueue')
AllrowsQueue            = TomasQueue.Create('allrowsQueue')
AlleCaravansQueue       = TomasQueue.Create('AlleCaravansQueue')
tempQueue               = TomasQueue.Create('tempQueue')
justoutsidequeue        = Tomasqueue.Create('justoutside')

// Rows {Create the rows}
For i = 1 to total number of Rows
        NewRow:=Row.create('Row')
        NewRow.RowNumber = i
        NewRow.Caravanslength = 0
        NewRow.Totallength = 18   //only UpperWorks length (totallength of caravan not counted in)
        NewRow.RowQueue = TomasQueue.create('RowQueue')
        NewRow.EnterQueue(allrowsQueue)
        NewRow.firstleavetime = 10000
        NewRow.firstcolor = 4

// CTV {Create the CTV and start CTV}
newCTV = CTV.Create('CTV')
newCTV.Start(TNow)

// Orderlist {Create the orderlist and start orderlist}
newOrderlist = Orderlist.Create('Orderlist')
newOrderlist.Start(TNow)

// Customer {Create the customer and start customer}
newCustomer = Customer.Create('Customer')
newCustomer.Start(TNow)

// Caravan {Create the caravans and initiate attributes of caravan}
For j = 1 to 4*(total number of Rows)
    NewCaravan = Caravan.create('Caravan')
        NewCaravan.CaravanNumber:= j
        Search caravan in database
        Determine NewCaravan.UWLength from database

        Search next leavetime in database
        Adjust myCaravan.leavetime

        Calculate next arrivaltime
        Adjust myCaravan.arrivaltime
        NewCaravan.Color = 2                    //color 2 = no color
        NewCaravan.EnterQueue(AlleCaravansQueue)
        NewCaravan.EnterQueue(Storage)
        Select Row (non-full Row)
        NewCaravan.EnterQueue(Row)
```

3. Experiments

3.1 Input

As is made clear in part 1, the model uses several input parameters. The values of these parameters are listed below. Each of the values is explained. It is also explained whether the parameter is varied in the experiments or not.

Scenario	AS-IS/TO-BE
# of Rows	500
%length difference	20%
Store in rows from week	26
Store until ACQueue <	0
Percentage of customers that tells color correct/incorrect	Correct: 80% Unknown: 15% Incorrect: 5%
Leavetime	Caravan dependent
Arrivaltime	Caravan dependent
UWLength	Caravan dependent
Orderamount	Caravan dependent
Rowlength	4 caravans
Totallength	18 meters

Table 2 - input values

Scenario

The main input parameter of the simulation model is the choice of the scenario. In the AS-IS scenario, caravans are stored based on the color it has, and in the TO-BE scenario the caravans are stored on their next leavetime.

# of Rows

The number of Rows is chosen to be 500. This equals an amount of caravans of 4* 500 = 2000. This is also the case of the real caravan storage. Due to realistic output regarding the real situation, the experiments are run with 500 rows.

% length difference

This parameter must be set as small as possible, with still all caravans be stored during winter. When this parameter is too small, the caravans in ACQueue do not fit any more in any row, and thereby the storage keeps partly empty while the ACQueue still consists of caravans in winter.

Store in rows from week

In the real (AS-IS) situation the rows are rearranged (in order to store arrived caravans) from week 26. It might be interesting to consider storing rows already from the beginning of the year in the TO-BE situation.

Store until ACQueue smaller than

With the use of this parameter it makes it possible to make use of more arrived caravans without the need to rearrange rows already.

Percentage of customers that tells color correct/incorrect

This parameter can be used to investigate the effectiveness of the current system of colored wires in the AS-IS scenario. The color of a returned caravan can be green (number 1 in the source code), none (number 2 in the source code), brown (number 3 in the source code) or black (number 4 in the source code). It is assumed that in the autumn every caravan gets a colored wire (green, brown or black). Two cases are possible:

- The caravan will be picked up in current year.
- The caravan will be picked up in next year.

For both of the cases, the input form makes it possible to decide which percentage of the wires is correct, which percentage of the customers doesn't know it yet and which percentage of the customers gives the incorrect answer.

For example, if the case is case 1, and the form states that 60% is incorrect, there is a 60% chance that the caravan gets the color black.

Orderamount

Every customer orders his caravan a certain amount of times during the year. From data it is known that the distribution of the amount of caravans over the amount of orders per year is shaped as given in the figure below. The most caravans are only ordered once a year. The distribution is likely to be a lognormal distribution.

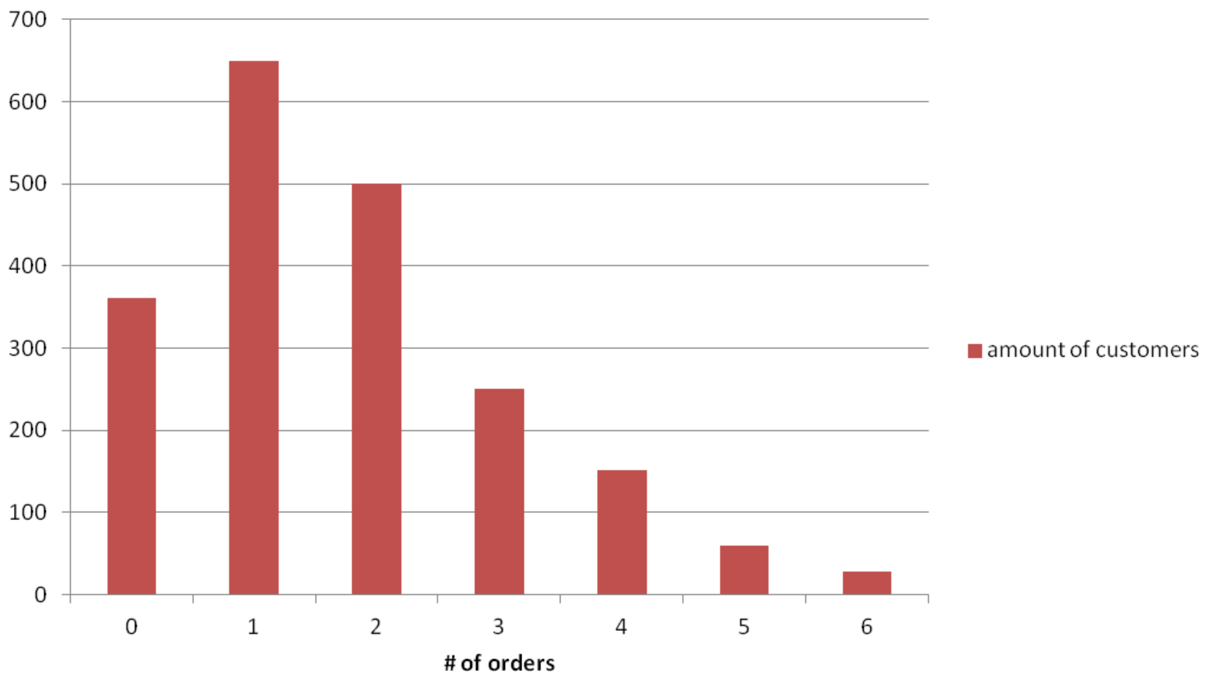

Figure 4 - distribution of customers - the amount of orders per customer

Leavetime

Also from data it is derived that there is a strongly season-influenced order pattern over the year. This is shown in the figure below, where the amount of orders per day is given per day, starting from day one of the year until day 260. It is assumed there are 52*5 = 260 working days per year. Although the company is open on Saturday, the orders for Saturday and Monday after the Saturday are summed. This creates the pattern below. Although there is a seasonal effect, it is not possible to catch this pattern in a, f.e. normal or sinusoidal, distribution. Therefore these data is used every year again.

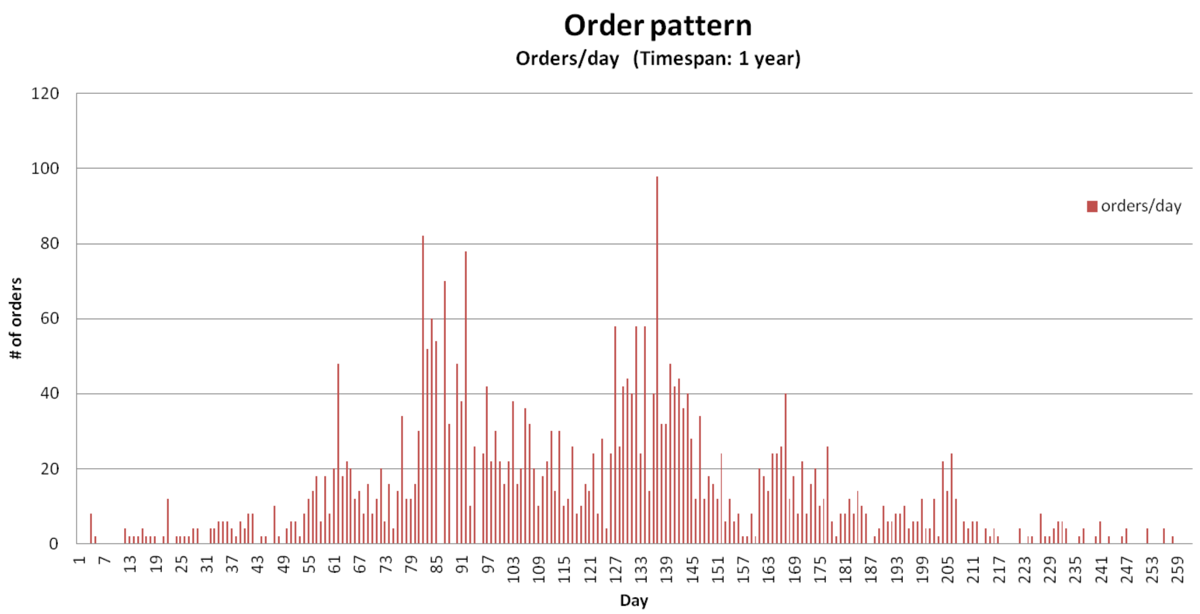

Figure 5 - order pattern

The leavetimes per caravan has been in a singular event calculated with the use of the order pattern over the year and the amount of orders per caravan per year. These data has been stored in the database and used in the second year again. This is done because, as explained above, it is not possible to use a distribution for this pattern. The two peaks are respectively the week of queen's day and the beginning of the summer holiday.

Arrivaltime

The arrivaltimes of the caravans are not registered and therefore there is no information available about the arrivaltimes. Because of this, the arrivaltime is calculated by the customer process for every returning caravan. This is done via the next algorithm:

```
If nextleavetime in current year then
        myCaravan.arrivaltime =  myCaravan.Leavetime + (randomvalue*(nextleavetime -
        myCaravan.Leavetime))
Elseif nextleavetime in next year then
        myCaravan.arrivaltime = myCaravan.Leavetime + (randomvalue*(random*(260-
        myCcaravan.leavetime)))
```

This source code above manages all caravans to be arrived when the next year starts. The extra random variable is used to avoid a perfectly linear arrival pattern of the caravans in the last months of the year.

UWLength

The UWLength dataset has been derived from the current real Upper Works Lengths of the caravans in the company. In the figure below the distribution of this input is shown. It shows the amount of caravans as function of the Upper Works length in meters. The figure has the marks of a normal distribution. The average of the length is 4.5 meter.

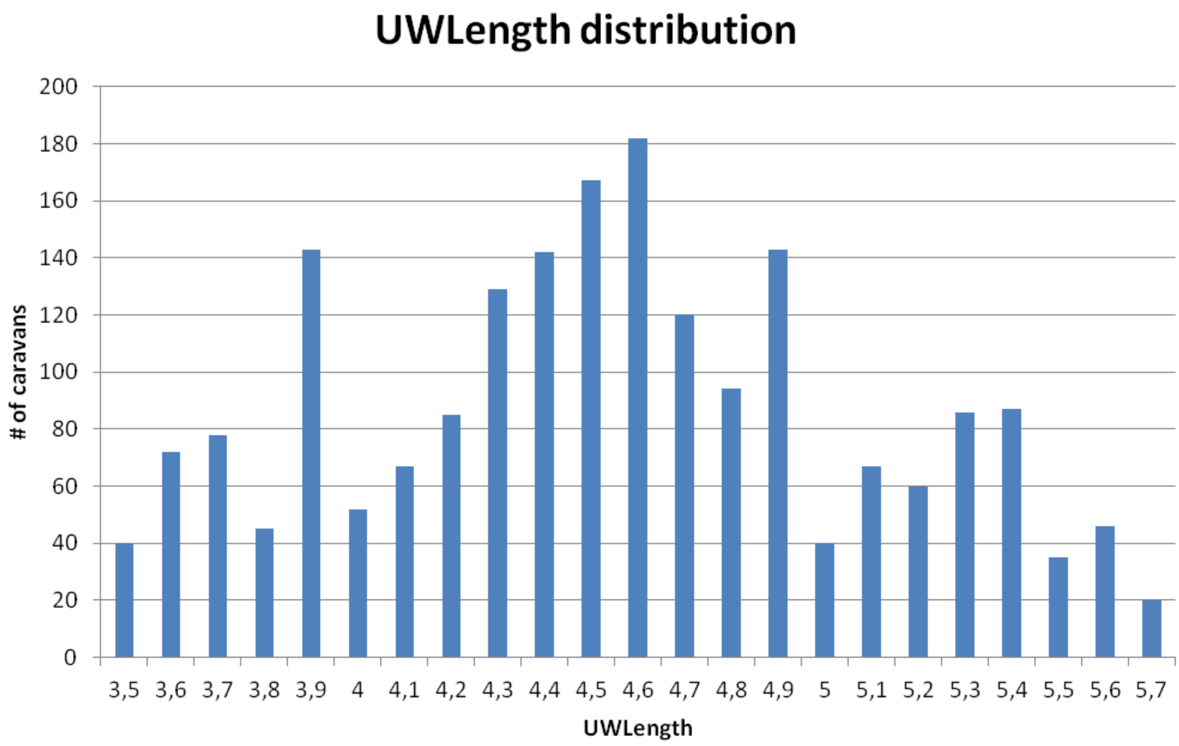

Table 3 - Upperworks length distribution of the caravans

Rowlength

The rows do in any case have a fixed capacity of 4 caravans per row. This has been like the real situation.

Totallength

The total length of a row expressed in meters must be equal to the multiply of the average UWLength of the caravans and the amount of caravans per row. This is equal to 4.5*4 = 18 meters. This is necessary to ensure all caravans are able to become stored while not using a high length difference percentage.

3.2 Output

The output of the model exists of the amount of pick-up moments in year 2 and the amount of pick-up moments in quarter 4 of year 2.

At the initialization of the model, the caravans are randomly stored in the storage. Therefore, no valid measurements can be done regarding the performance of the AS-IS and the TO-BE situation. During autumn of year 1, the caravans are stored in the way the selected scenario executes. During spring of the second year, the benefits can be taken of this way of execution.

Also, due to psychological effects it is desirable to minimize the work in the autumn. Therefore, it is interesting to assess the amount of pick-up moments in the last three months of the year as percentage of the total amount of pick-up moments. The graph of the cumulative pick-up moments is used for this performance indicator.

Furthermore it is important to assess the graphs of the ACQueue, OCQueue, Storage and HolidayQueue to ensure all caravans are stored at the end of every year. Besides, it is undesirable for the ACQueue to contain more than 30 caravans for multiple days without being able to store them.

3.3 Experimental plan

The main goal of the experiment is to investigate differences between the as-is scenario and the to-be scenario. Next to that, also the situation without time constraints (length constraint only) will be experimented. This makes three experiments, which will be run with length differences of 20, 15, 10 and 5 percent. This will be repeated for a ACQueue<20 and an ACQueue<0. The table used is shown below. The output values exists of the amount of pick-up moments in year 2 and the amount of pick-up moments of Q4 of year 2.

ACQueue< 20		No rearranging of rows	AS-IS Rearranging from wk 26	TO-BE Rearranging from wk 26
Length difference	20			
	15			
	10			
	5			

ACQueue< 0		No rearranging of rows	AS-IS Rearranging from wk 26	TO-BE Rearranging from wk 26
Length difference	20			
	15			
	10			
	5			

4. Verification and results

4.1 Calculation

Only the verification of the 'no rearranging'-scenario will be done, because due to the influences of the time constraints on the AS-IS and TO-BE scenario's, these cannot be calculated.

The average amount of orders per customer per year: 1.728 orders/ customer/ year. There are 2000 customers. This means a total amount of 3456 orders per year.

Consider to pick one caravan from a row. The next situations can occur:

- 25% chance of 4 caravans in the row.
- 25% chance of 3 caravans in the row.
- 25% chance of 2 caravans in the row.
- 25% chance of 1 caravan in the row.

When there are 4 caravans in the row, there will be on average 1.5 caravans in front of the ordered caravan. Even so, when there are 3 caravans in the row, there will be on average 1 caravan in front of the ordered caravan. When there are 2 caravans in the row, there will be on average 0.5 caravans in front of the ordered caravan and when there is 1 caravan in the row there will be no caravan in front of the ordered caravan.

Every caravan in front needs to be picked twice, one time to put it outside and one time to put it back when the ordered caravan has been removed from the row. Also, one pick up is required to remove the ordered caravan.

- Amount of pick-ups when 4 caravans in row = 1.5 * 2 + 1 = 4 pick-ups.
- Amount of pick-ups when 3 caravans in row = 1 * 2 + 1 = 3 pick-ups.
- Amount of pick-ups when 2 caravans in row = 0.5 * 2 + 1 = 2 pick-ups.
- Amount of pick-ups when 1 caravans in row = 0 * 2 + 1 = 1 pick-up.

Now calculating the average amount of pick-ups per ordered caravan:

Average # of pick-ups = 4 * 0.25 + 3 * 0.25 + 2 * 0.25 + 1 * 0.25 = 2.5 pick-ups per ordered caravan.

As described above, there are 3456 orders per year. This means that for the ordered caravans on average 2.5 * 3456 = 8640 pick-ups per year are required. There are as much arrived caravans as there are ordered caravans, so adding 3456 to 8640 gives a total of 12096 pick-ups per year.

4.2 Model

A view of the model is shown in figure… As one can see, on the left the input can be given and at the top of the form the control buttons make it able to start, resume or stop the simulation or to hide/unhide the charts and the grid. At the bottom the grid is shown which contains the database. In the database the customer number, the UWLength and the orders of the caravans are shown. At the top right the time in days is shown. In the middle the output charts are shown and at the right the output in terms of number of pick-up amounts is shown.

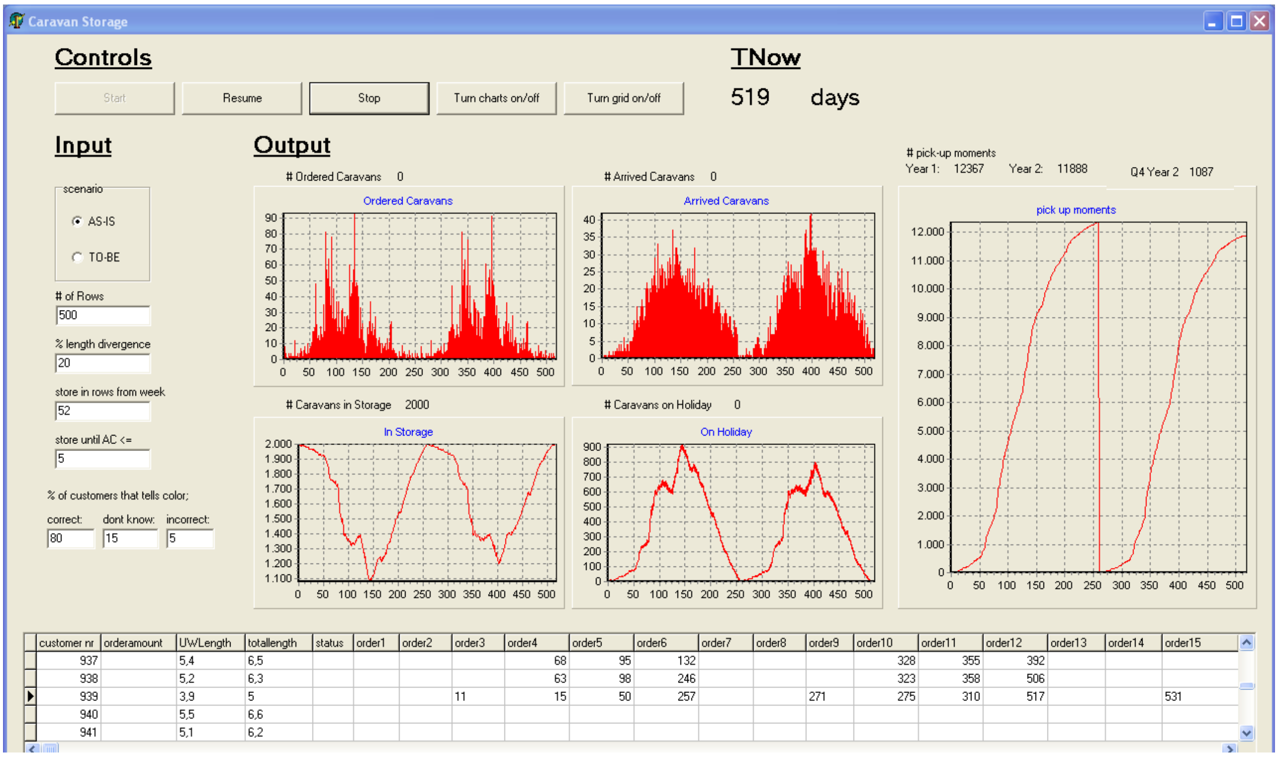

Figure 6 - model interface

4.3 Verification

The start of the simulation is shown in the trace form. The model starts as expected. The Classes, Queues and the Caravans are created.

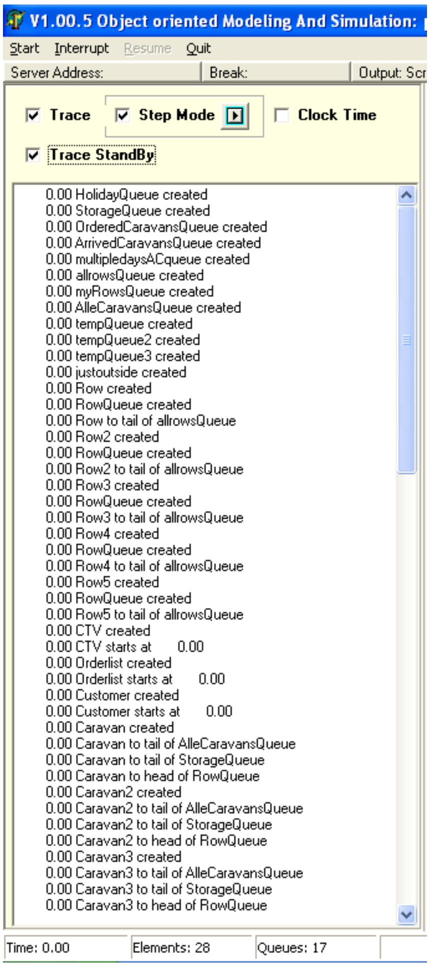

Figure 7 - tracing

Also the order pattern is the same as the order pattern shown in the input section.

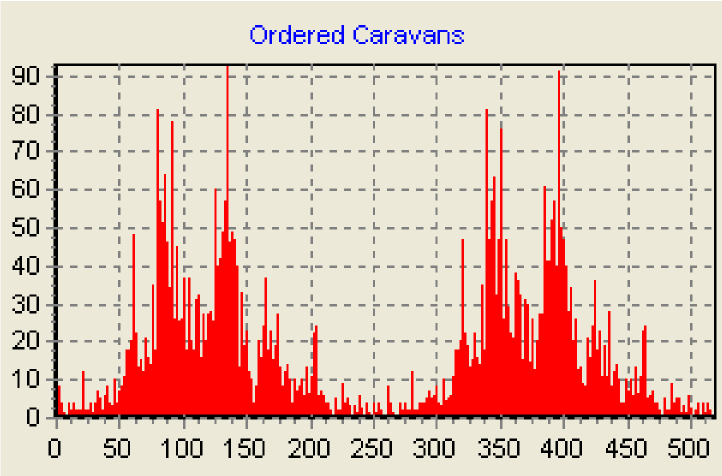

Figure 8 - verified order pattern

In the last chapter a calculation is done to verify the output of the model. This is done for the situation in which no rearranging of the rows take place. The calculated output has been 12096 pick-ups per year. In Appendix A the results are given. The average output of the model in the 'no rearranging' scenario needs to be calculated:

$$\text{Average \# of pickups} = \frac{12200 + 12164 + 11958}{3} = 12107$$

There is a minimal difference of 11 pick-ups between the calculated and the simulated output. In percentage calculated as follows:

$$\text{Deviation} = \frac{11}{12107} * 100 = 0.09\%$$

The combination of the correct tracing steps and the calculated deviation above, the simulation model is verified and thereby considered to be modeled correctly.

4.4 Results

The detailed results of the experimental plan are shown in Appendix A. The results are shown in the figures below. In these figures the scenarios are numbered as follows:

1. No rearranging of rows
2. AS-IS
3. TO-BE

As becomes clear, the total amount of pick-ups will be less by executing the TO-BE scenario. In Q4 the 'no arranging' scenario performs better.

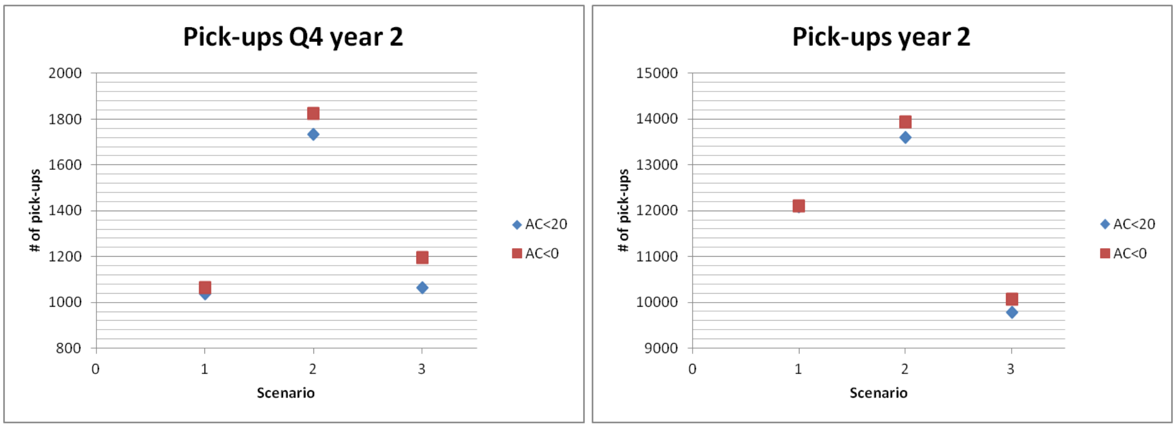

Figure 9 - Pick-ups year and pick-ups Q4

Storing the caravans until the ACQueue is less than or equal to 20 performs slightly better than a situation in which the ACQueue must be reduced to 0 every day. This effect does have more influence on the TO-BE scenario than in the other scenarios.

By considering the Q4 pick-ups as a percentage of the total pick-ups during that year, it becomes clear that the 'no rearranging' scenario performs best. This is shown in the following figure.

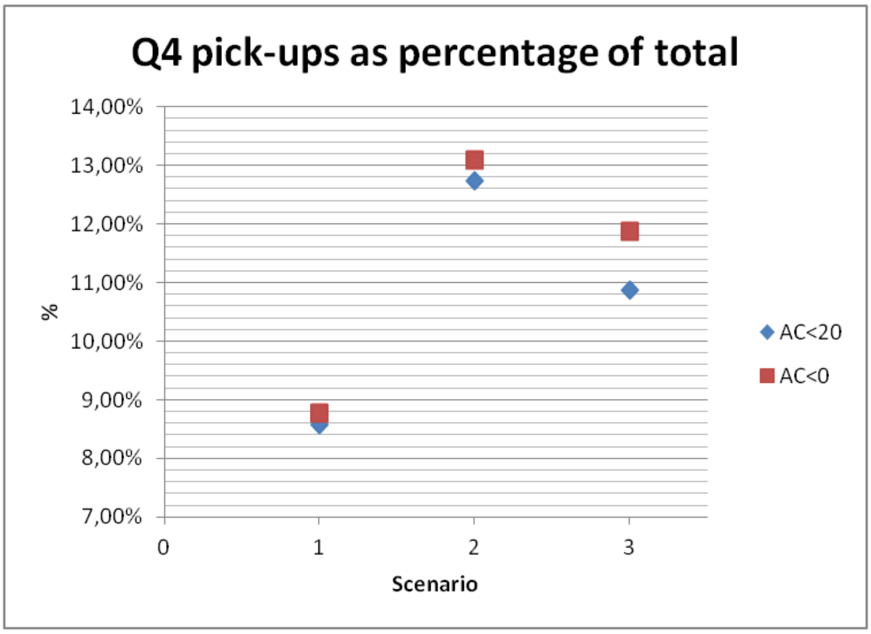

Figure 10 - Q4 pick-ups as percentage of total

It is interesting to calculate a possible cost reduction by implementing a system based on the TO-BE scenario. As becomes clear in figure…, a yearly reduction of 13600-9800 = 3800 pick-up moments is possible. By guess, an average time between two pick-ups is assumed of 3-6 minutes. Two employees are in charge to work the CTV. The cost price of an employee is assumed to be 35 euro/hour. The total cost reduction is calculated as follows:

Worst case: $\text{cost reduction} = 3800 * \frac{3}{60} * 2 * 35 = 13300 \text{ euro}$

Best case: $\text{cost reduction} = 3800 * \frac{6}{60} * 2 * 35 = 26600 \text{ euro}$

Assumed is a maximum required payback time of 1 year. Investing in a new registration system, which makes it able to execute the TO-BE scenario, may have a maximum cost price of 13300-26600 euro.

Conclusion

Conclusion

The goal of the simulation assignment is stated in the introduction.

"Determine the performance of the current storage system and investigate possibilities to increase this performance."

The performance of the current storage system is determined in terms of the amount of pick-up moments. Also alternatives were investigated. The AS-IS scenario does perform worst in comparison to the two alternatives. The TO-BE scenario does result in the best overall performance, but the alternative of 'no rearranging of rows' does result in the least amount of pick-ups in Q4.

It can be concluded that implementing the TO-BE system creates the most gains, although this system does also have the highest investment costs. On the other hand, using the 'no rearranging of rows'-alternative does not result in a costly investment, but does have a high improvement in the performance of the storage system either.

Recommendations

In the TO-BE situation it is assumed that it is possible to exactly determine the next leave time of the customers. Further investigation is required to stipulate the possibilities of determining the exact next leavetime of the customers. By implementing this into an improved simulation model, more accurate output can be obtained, and thereby a more accurate investment decision can be made.

Appendix A

AC<= 0

pick up moments

			AS-IS	TO-BE
		No rearranging of rows	Rearranging from wk 26	Rearranging from wk 26
Length difference	20	12200	13851	10041
	15	12164	13737	10054
	10	11958	14215	10074
	5	12184	N/A	N/A (Year 1: 11899)

pick up moments Q4

			AS-IS	TO-BE
		No rearranging of rows	Rearranging from wk 26	Rearranging from wk 26
Length Difference	20	1035	1753	1189
	15	1063	1840	1180
	10	1091	1882	1212
	5	1068	N/A	N/A

AC <= 20

pick up moments

			AS-IS	TO-BE
		No rearranging of rows	Rearranging from wk 26	Rearranging from wk 26
Length difference	20	11964	13737	9859
	15	12058	13456	9799
	10	12234	13605	9680
	5	12202	N/A	N/A

pick up moments Q4

			AS-IS	TO-BE
		No rearranging of rows	Rearranging from wk 26	Rearranging from wk 26
Length difference	20	1055	1803	1124
	15	1033	1736	1042
	10	1024	1657	1025
	5	1118	N/A	N/A