

## **HNU Working Paper**

Nr. 41

Oliver Kunze, Emanuel Herrmann

### **City Crowd Logistics Processes**

12 / 2019

---

Dr. Oliver Kunze, Professor für Logistik,  
Emanuel Herrmann, Wissenschaftlicher Mitarbeiter innovative Logistik,  
Hochschule für angewandte Wissenschaften Neu-Ulm  
University of Applied Sciences  
Wileystraße 1, D-89231 Neu-Ulm

## **Abstrakt**

City Crowd Logistics (CCL) ist ein neues Konzept für innerstädtische Transporte auf der letzten Meile, die von der „Crowd“ und in Ausnahmesituationen von professionellen Kurieren durchgeführt werden. Diese Transporte können in getrennten Teiltransporten von verschiedenen Kurieren durchgeführt werden. Wir präsentieren ein Prozessmodell zur Durchführung von CCL-Diensten.

Freie Schlagwörter:

Urbane Logistik, Logistik der letzten Meile, Crowd Logistics, Prozesse

## **Abstract**

City Crowd Logistics (CCL) is a new concept for urban last mile transports which are conducted by the crowd and by professional couriers in exceptional situations. These transports can be carried out in separated legs by different couriers. We present a process model to conduct CCL services.

Keywords:

Urban Logistics, Last Mile Logistics, Crowd Logistics, Processes

JEL-Klassifikation: M00 , M15

# City Crowd Logistics Processes

Oliver Kunze\*, Emanuel Herrmann\*

*\*HNU Neu-Ulm University of Applied Sciences, Wileystraße 1, 89231, Neu-Ulm, Germany*

© The Authors

Keywords: Urban Logistics, Last Mile Logistics, Crowd Logistics, Processes

---

## Abstract

City Crowd Logistics (CCL) is a new concept for urban last mile transports which are conducted by the crowd and by professional couriers in exceptional situations. These transports can be carried out in separated legs by different couriers. We present a process model to conduct CCL services.

## Content

1. Introduction .....	4
1.1. Background.....	4
1.2. A Concept for City Crowd Logistics .....	4
1.3. Problem.....	5
1.4. Methodology.....	6
2. Process Model .....	6
2.1. Actors .....	6
2.2. Process Overview .....	8
2.3. Sub Processes .....	9
2.3.1. Order Entry .....	10
2.3.2. Find Execution Options .....	12
2.3.3. Split Order .....	14
2.3.4. Assign Shipments .....	16
2.3.5. Execute & Monitor .....	23
2.3.6. Manage Minor Deviations .....	26
2.3.7. Manage Major Deviations .....	27
2.3.8. Remuneration.....	31
3. Critical Discussion.....	32
4. Glossary.....	33
5. References .....	33

## 1. Introduction

### 1.1. Background

In recent years, the demand for B2C e-commerce has risen substantially and as of 2017 total revenue amounts to 48.7 bn € in Germany with an average annual growth rate of 23.5% since 2000. This increasing trend for convenience shopping entails several traffic related issues in urban areas. Most notably, more online-shopping requires a higher level of CEP-logistics, which is mostly conducted by vans nowadays. This causes a range of problems particularly in urban areas. Delivery vehicles often park in second lane, block pedestrian sidewalks and subsequently create unnecessary obstacles and congestion. Additionally, delivery vehicles cause noise pollution and contribute to greenhouse gas emissions. In this challenging environment, last mile delivery becomes increasingly more difficult, as customers want their goods delivered fast and residents demand that their living areas are uncongested and quiet while city officials require emission reductions. Therefore, logistics service providers are not only required to locally optimize their activities, but also incorporate environmental, social and political issues in their daily operations (Taniguchi, 2014). Optimizing the transportation flow given the limitations and restrictions of an urban environment is the goal of City Logistics, and Intelligent Transportation Systems (ITS) have become an integral component. ITS aids transportation in coordination (Pan, Trentesaux, Ballot and Huang, 2019), intermodal transport (Crainic, Perboli & Rosano, 2018) and other transportation related topics such as traffic management (e.g. congestion pricing, compare Aboudina & Abdulhai, 2017) or route optimization (Taniguchi & Shimamoto, 2004).

ITS and the prevalence of smartphones have also made it possible to outsource (transportation) tasks to the crowd. Therefore, crowdsourced logistics has become a viable option to mitigate urban freight transportation related issues. (Buldeo Rai, Verlinde, Merckx & Macharis, 2017). Crowd couriers are recruited from existing traffic flows in order to transport parcels within an urban area. Crowd couriers can be divided into different categories, however this is a complex issue, as the differentiation between consumer & provider, employee & self-employed and professional & non-professional service provision is not always clear. Possible categories for different types of crowd couriers are subcontractors, who perform delivery tasks for established service providers (e.g. DHL, UPS), professional drivers, who work for courier companies and have excess capacity as well as casual drivers such as students, freelancers, etc. (Buldeo Rai, Verlinde, Merckx & Macharis, 2017). Due to utilizing existing traffic flows which relies on already existing infrastructure, delivery orders can be conducted without adding additional traffic, while at the same time have the potential of being cheaper due to the disposition of non-professional, not permanently employed individuals (O'Byrne, 2016). However, the downside of relying on an independent workforce is the fact, that there is uncertainty in both the times when they are available and the tasks they are willing to carry out. Therefore, in order to guarantee the service agreement, the provider may still have to hire professional workers or services to some degree (Sampaio, Savelsbergh, Veelen turf & van Woensel, 2019).

### 1.2. A Concept for City Crowd Logistics

City Crowd Logistics (CCL) is a new concept for urban last mile transports which are conducted by the crowd and in exceptional situations by professional couriers. These transports can be carried out in separated legs by different couriers. CCL is a specific instance of the physical internet (PI) - details on PI see (Crainic & Montreuil 2016).

The key idea of CCL is that a central dispatching software (dispatcher) splits transport orders into consecutive transport legs and offers these legs to several couriers which bid for these legs. The dispatcher then assigns a chain of consecutive transport legs to the couriers ensuring that the order arrives on time at the destination. The transport items can be stored temporarily at router locations, where couriers drop the items after their assigned leg transport and their successors pick them up for the next leg transport. This process is illustrated in Figure 1

CCL Order Execution Options

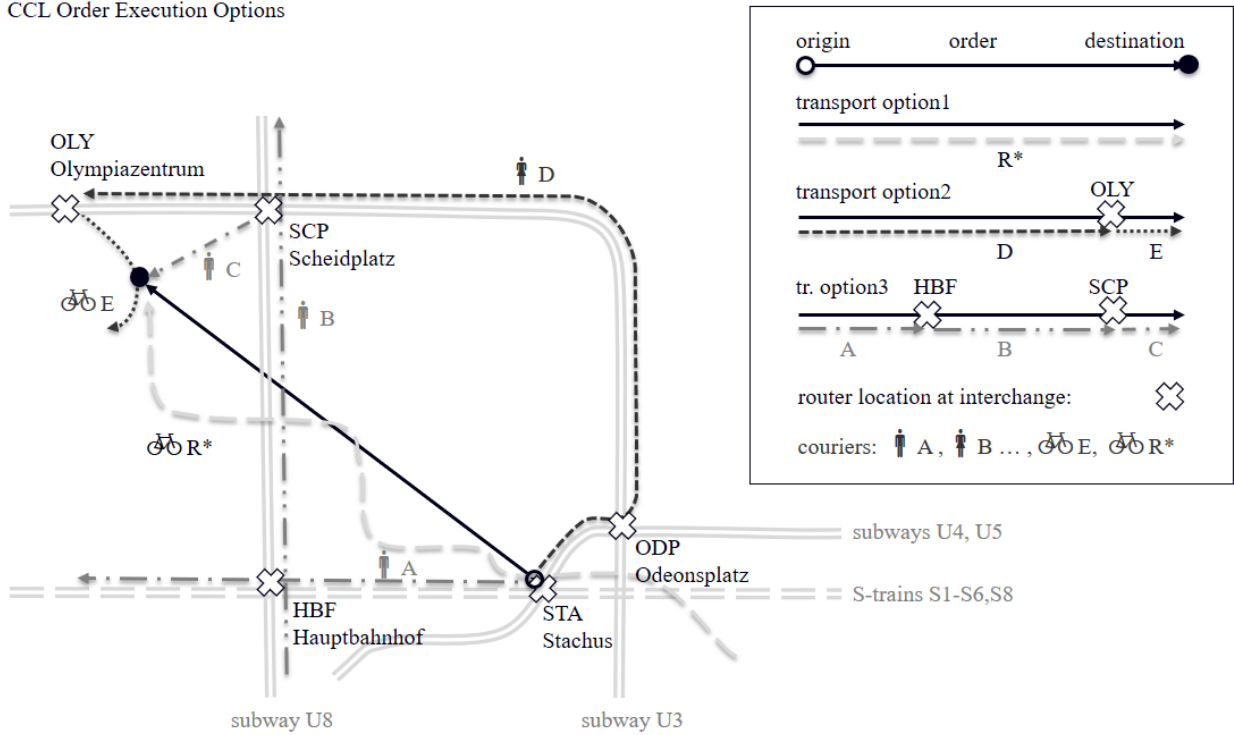


Figure 1 - CCL Order Execution Options

An order shall be transported from the origin at STA to the destination near OLY. One option (1) to transport this order is to employ a professional bike courier R\* who can transport the order directly from origin (O) to destination (●) by bike at a fixed tariff. Alternatively, the order could be transported by the crowd – i.e. by individuals who travel anyhow and agree to carry the items along their way for an adequate payment. Thus another option (2) would be a subway commuter D on its way from STA to OLY, who picks up the item at STA and drops it at OLY, where cyclist E picks up the item and carries it to the destination. A third option (3) would be a commuter A heading westwards who picks up the item at STA and drops it at HBF. There, commuter B who is heading north, picks it up and drops it at SCP. Finally, a local pedestrian C agrees to pick up the item at SCP and bring it to the destination. One assumption of CCL is that the crowd will usually carry the items at a cheaper price than the professional courier. Another assumption of CCL is that in a delay situation a professional courier can carry the item, ensuring the latest arrival time of the order which was agreed between the customer and the dispatcher.

### 1.3. Problem

One challenge to set up such services is a process model, which covers the relevant aspects of such a service. The research question which we try to answer is: “How could such a process look like?”.

Note that other research activities currently deal with further aspects of CCL like number and optimal locations of the routers (boxes), optimized multi-courier routing through the network, and price negotiating algorithms. These aspects are excluded from this paper.

## 1.4. Methodology

We took a straight forward methodology to define these processes. We used the meta process structure presented by (Kunze, Baumgärtel et al 2012) which was tailored for multi leg full truck load transports for multiple carriers, and adapted the processes to meet the CCL needs. The draft processes were presented to the CCL panel\*, and modified based on the panels feedback. After this review, the processes were accepted by the panel.

The processes were modeled by means of UML (unified modelling language) using use case diagrams (UML UC) and UML activity diagrams (UML AD).

We want to point out, that the presented process template may need further refinement depending on the specific implementations of possible CCL services. Still, it can be used as a first beta version for further enhancement.

## 2. Process Model

We defined the process model by means of one overall process and several sub processes.

### 2.1. Actors

The following actors are involved in the CCL process:

**Courier:** Person who transports shipments.

Couriers can be distinguished by their commercial role:

- **crowd courier** – including:
  - pro bono courier (courier who will provide courier services without remuneration)
  - private courier (non-professional courier who will provide courier services for adequate remuneration)
- **professional courier** (employed to do courier services)

Couriers can also be distinguished by the means of transport they mainly use, e.g.:

- bike courier
- car-courier
- pedestrian courier (walking, only)
- public transport courier (use of public transport and walking)
- other courier (e.g. skateboarder, e-scooter-driver, ...)

Note1: As crowd couriers might use different means of transport to execute one transport leg (e.g. foldable bike and subway), this differentiation is not unique.

Note2: In the future, ground drones (also termed robots or automated ground vehicles) might be used to execute CCL transport legs. In our model these non-human couriers could be represented by the role professional courier, as it is most likely that a ground drone operator would charge a predefined tariff price for the use of a ground drone.

**Customer:** Person (or company) who wants a good to be shipped from origin to destination – the customer places the order and pays for it.

Note: For simplicity reasons we assume in our model, that the customer and the recipient of the items are one and the same. In cases, where the customer and the recipient of the items are not identical, the order placement, the payment process and the transport monitoring event communication are performed with the customer (who in turn informs the recipient), whilst the destination of the item is the location of the recipient

**Dispatcher:** A system, which assigns transport order legs to individual couriers, bills the customer for the performed transport and credits the couriers for their services.

---

\* CCL panel = board of company representatives monitoring and controlling the progress of the CCL-research project

The use case diagram of CCL with its actors is depicted in Figure 3.

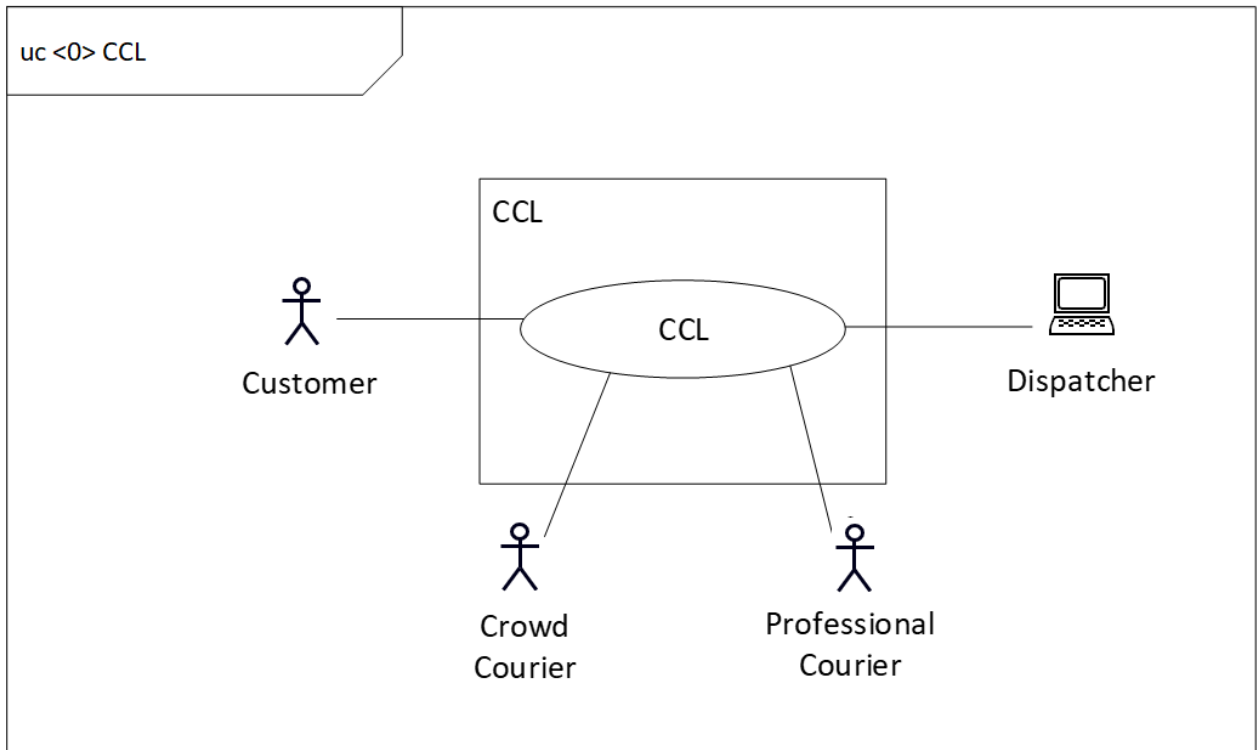


Figure 2 - CCL Use Case Diagram

In the following UML-AD-diagrams, the roles defined in Figure 2 are represented by swim-lanes. If no role distinction is made, the single swim lane is named CCL-System.

2.2. Process Overview

The overall CCL-process can be modelled as a process model for the transportation of one item. This process model is depicted in Figure 4.

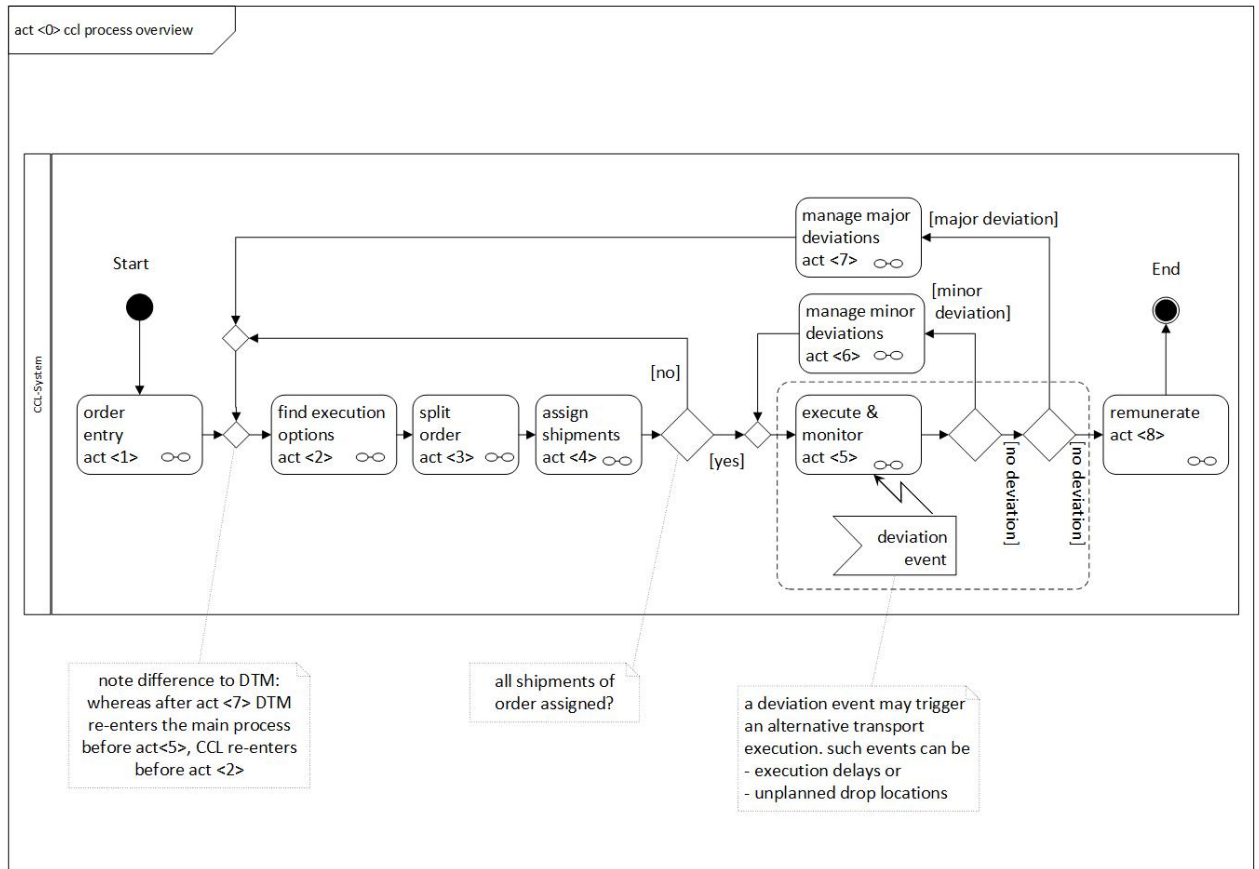


Figure 3 – CCL Process Overview

This overall process model is built up by several sub processes:

1. order entry act<1>: this sub process defines the order entry process (including billing)
2. find execution options act <2>: this sub process checks, which theoretical routing option there are to transport the order (for examples see Figure 1)
3. split orders act <3>: this sub process choses a theoretical routing option<sup>†</sup> and splits the order based on decision criteria into transport legs (i.e into individual consecutive shipments).
4. assign shipments act <4>: each shipment is assigned to a courier
5. execute & monitor act <5>: after the shipments of the order have been assigned to couriers, the planned transport is being monitored. Based on these monitoring results either no changes are necessary and the transports are executed as planned (continue with act <8>), or minor deviations occur (which means the original assignments are kept, and only arrival and departure times are modified – see act <6>), or the

<sup>†</sup> Or several theoretical routing options – for details see the relevant sub-process)



deviations are of a magnitude, that a re-assignment of shipments to other couriers or even a different shipment-split is required (see act <7>)

6. manage minor deviations act <6>: This sub-process manages the minor deviations
7. manage major deviations act <7>: This sub-process manages the major deviations
8. remunerate act <8>: This sub-process includes the compensation of the couriers for their services as well as the compensation of the customers in case of service level violations (i.e. significant delays).

Note: As we used the DTM process [details see (Kunze, Baumgärtel et al 2012)] as blueprint for the CCL-process, we want to point out a significant difference between the CCL-Process and the DTM-Process. Whereas DTM returns from “manage major deviations” act<7> to “execute and monitor” act<5>, CCL returns from “manage major deviations” act<7> to “find execution options” act<2>.

### 2.3. Sub Processes

We now define the different sub-processes.

2.3.1. Order Entry

The order entry process shall enable the customer to order a transport. It shall also check correctness of data (e.g. address & weight checks) and decide on acceptance of the transport order (=closure of transport contract).

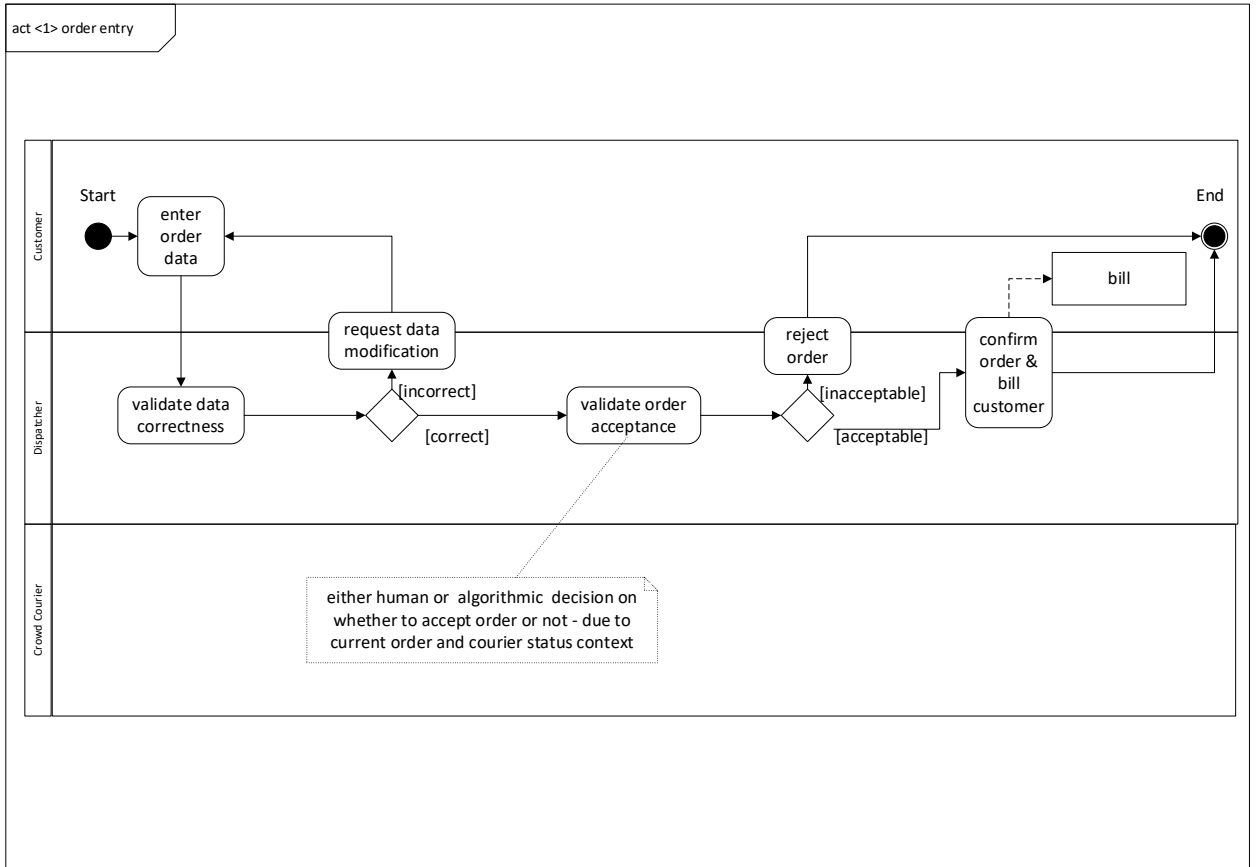


Figure 4 – Order Entry

#### 2.3.1.1. *Input*

The input data for this process are

- Location of origin (incl. full address)
- Location of destination (incl. full address)
- Weight & volume of item
- Availability of item for pickup: dd.mm.yyyy & hh:mm
- Latest due date for delivery: dd.mm.yyyy & hh:mm

#### 2.3.1.2. *Output*

The output of this process is either an order confirmation (including the billing for the order) or a rejection of the order e.g. due to lacking order data or because the dispatcher cannot confirm the order (e.g. due to overload).

#### 2.3.1.3. *Comments*

The rules for an order rejection have to be specified in the relevant business context. An order rejection may be triggered by simple rules, top level KPIs, complex algorithmic decisions or by a manual decision of the dispatcher.

2.3.2. Find Execution Options

This process step is used to find possible alternative routing options through the CCL-Network within the actual context. These options are assessed with respect to the input data (see below). Based on this assessment, a sorted list of "best" options is computed, which is used in a later process step as basis for possible assignments to couriers.

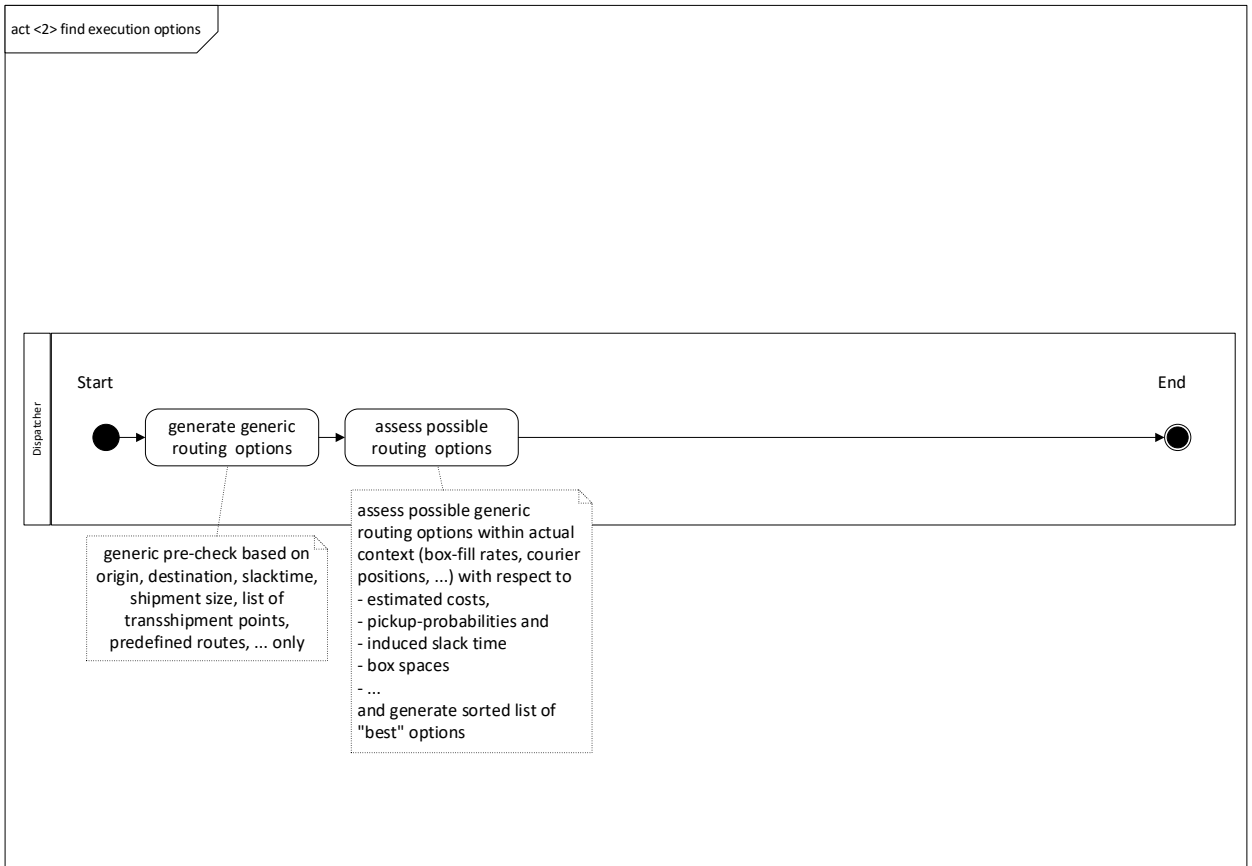


Figure 5 – Find Execution Options

The generation of generic routing options may result in a very long list of options. Therefore, the subsequent routing option assessment helps to narrow down the multitude of routing options to those, which are realistic.

### 2.3.2.1. *Input*

Input for this sub-process are data on the current and anticipated network status as well as order data:

- current and anticipated box-fill-rates,
- current and anticipated courier positions,
- current and anticipated order load
- pickup - probabilities along the routing option and
- historic transport costs
- ...
- and available slack time for order fulfillment

### 2.3.2.2. *Output*

Based on the input data, the estimated costs per option are computed, and the options are sorted by ascending estimated costs.

### 2.3.2.3. *Comments*

Note that the details of this assessment are not specified in any detail here. During a ramp-up-phase, most of the needed data (especially estimates based on historic data) may not be fully available, yet. Therefore, simple initial rules may be used to sort the transport options (e.g. sort by network distance via unoccupied boxes) until better data for the assessment of routing options are available. However, once these data are available, one can improve the route-option assessment function accordingly.

### 2.3.3. Split Order

This sub-process splits the original transport order into a chain of subsequent shipments. Each shipment needs to be assigned to a courier in a later step.



Figure 6 – Split Order - Process

#### 2.3.3.1. *Input*

- Transport order data
- Sorted list of possible routing options

#### 2.3.3.2. *Output*

- Consecutive chain of shipments (which are not assigned to couriers, yet)

#### 2.3.3.3. *Comments*

Note1 – it is an option in this process step to convert a full transport order into one shipment (from origin to destination), only. I.e. the word “split” is to be used in a more generic sense

Note2 – in order to be able to assign a shipment to a courier, the shipment which covers of the original transport order has to be generated prior to the assignment.

Note3 - if several assignment options are to be tried out in the subsequent sub-process, this sub process “split order” needs to generate several spit-instances (e.g. A to B via C and A to B via D) for the same order (multi option mode). Status handling has to make sure that unused split instances are deleted later.

2.3.4. Assign Shipments

The sub-process “assign shipments” can be run in three variants:

- Assign shipments all-in-one: in this variant all shipments of the order are assigned to different couriers in one sub-process without any unassigned shipments.  
This key idea of this variant is to sell all shipments to the crowd in one bundle – this reduces the risk of not finding couriers for the downstream shipments.
- Assign shipments sequentially: in this variant the “next” unassigned shipment is assigned to a courier – the remaining succeeding shipments remain unassigned.  
This key idea of this variant is to sell shipments sequentially to the crowd and leaving possible optimization room for later assignments.
- Assign shipments partially: in this variant “any” unassigned shipment of an order is assigned to a courier – the remaining shipments (either preceding or succeeding the assigned shipment) remain unassigned.  
This key idea of this speculative variant is “cherry picking”, i.e. sell shipments if attractive offers are made and hope that the rest can be sold at good prices later.

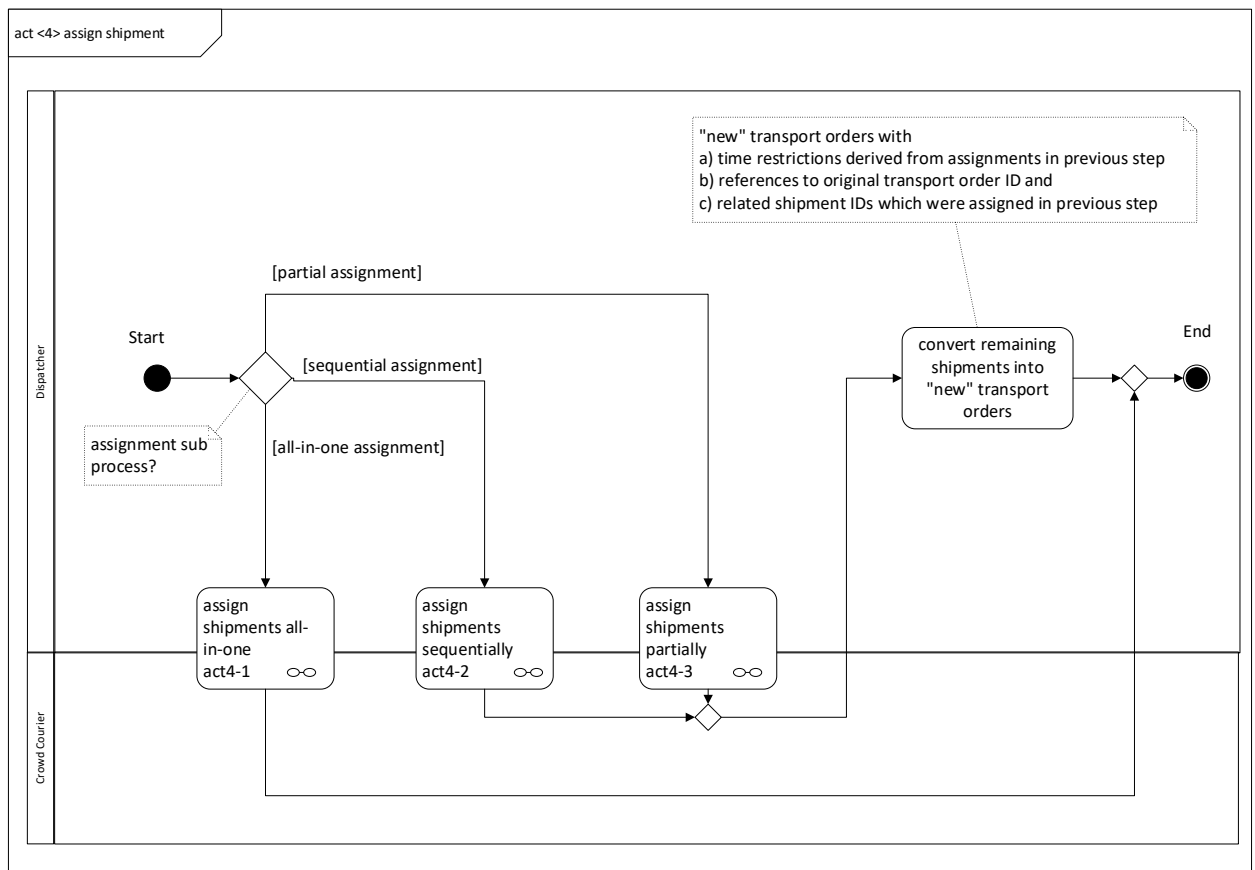


Figure 7 – Assign Shipment – Process



2.3.4.1. Assign Shipments “All in One”

The conceptual idea behind this sub-process is to assign all shipments of an order in one step without leftovers (i.e. without leaving any shipment of the order unassigned).

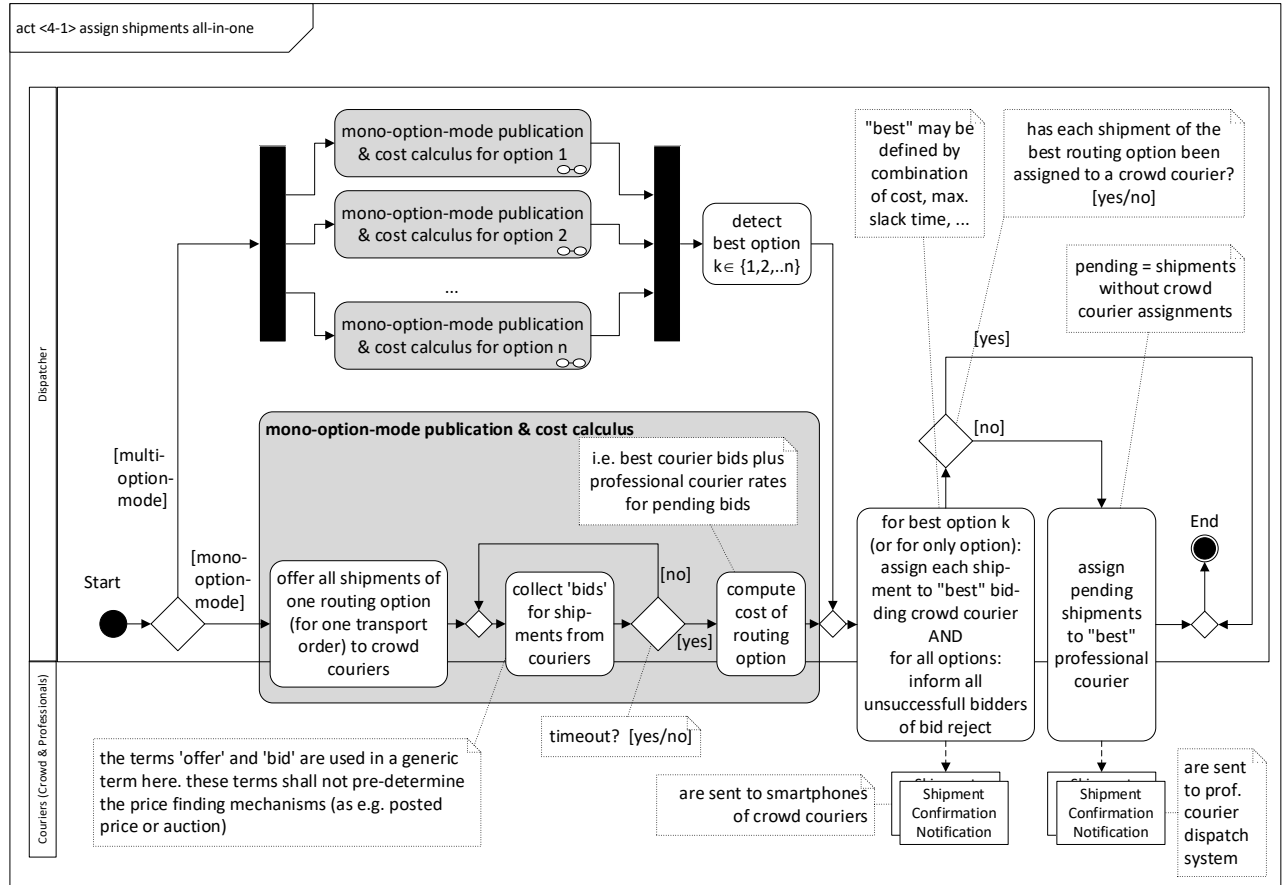


Figure 8 - Assign Shipments “All in One” - Process

This sub process can be run in a “mono-option mode” – i.e. there is only one split instance and only one set of subsequent shipments which represent the original transport, or it can be run in a “multi-option mode” – i.e. several split instances (and thus several sets of subsequent shipments) are offered (not assigned!) to the couriers in parallel.

In the *mono option mode*, the chain of subsequent shipments is offered to couriers, and bids from the couriers are collected, until a timeout occurs. If all shipments got a bid from a crowd courier, the couriers with the lowest bids are assigned to the relevant shipments. If not all shipments got a bid from a crowd courier, the remaining shipments are assigned to a professional courier. Those couriers which had placed a non-minimal bid for a shipment get a rejection notice.

Note – especially in the case where not all shipments got a bid from the crowd, and a professional courier needs to be hired anyway to conduct at least one shipment, it is possible, that it is a better option (i.e. it is cheaper and faster) to assign the whole un-split transport to a professional courier. In this case, the “best option” is one single shipment which is identical to the original transport, and this one shipment is assigned to a professional courier, only (i.e. without the involvement of any crowd courier).

In the *multi-option mode* the mono-option mode is executed for each of the different split-instances in parallel (i.e. different chains of singular shipments are offered to the crowd in parallel).

Note – in the multi-option-mode one can anticipate some level of frustration on the courier’s side, if the timeout is not short enough, because many couriers will be informed after a while, that their bids were not considered. This “frustration” is one aspect of “entry deterrence” in auction theory – further details see e.g. (Klemperer 2004).

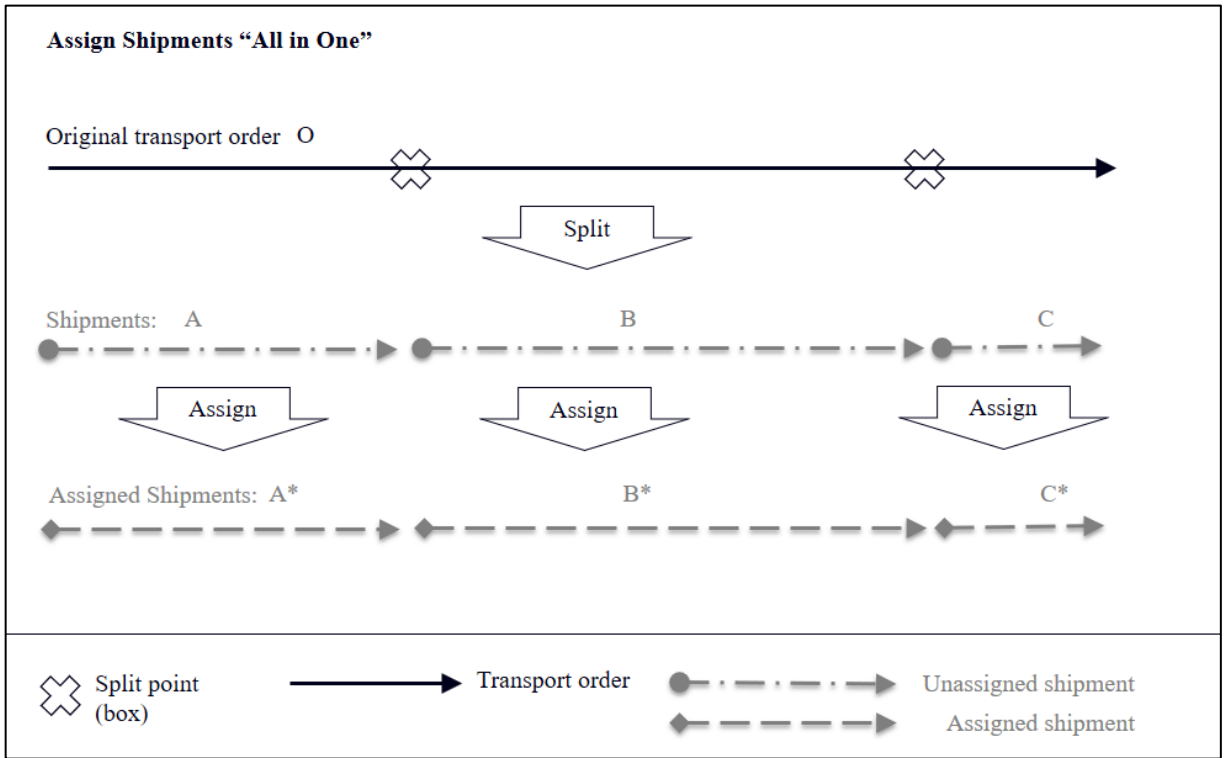


Figure 9 - Assign Shipments "All in One" - Scheme

At the end of this process step the original transport order O is split into different shipments (here A, B and C), and these shipments are assigned 1:1 to a courier each. The physical execution of the subsequent shipments A, B and C therefore fulfills the original transport order O.

2.3.4.2. Assign Shipments Sequentially

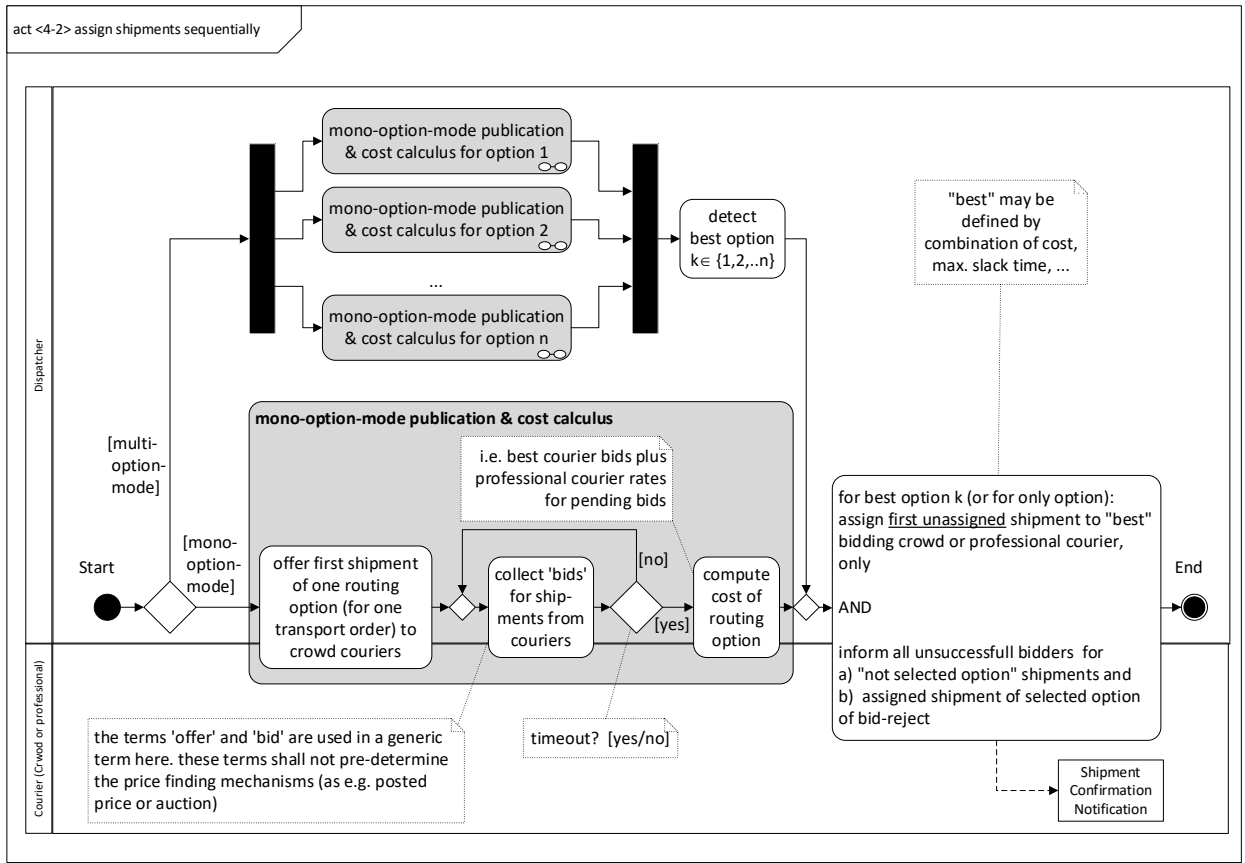


Figure 10 - Assign Shipments Sequentially - Process

This sub process also can be run in a “mono-option mode” – i.e. there is only one split instance and only one set of subsequent shipments which represent the original transport, or it can be run in a “multi-option mode” – i.e. several split instances (and thus several sets of subsequent shipments) are offered (not assigned!) to the couriers in parallel.

Note that in this sequential assignment variant only the “next” shipment is assigned whereas the subsequent (possible) shipments remain unassigned and are converted into a “new” transport order which represents the remainder of the original transport order (see Figure 7).

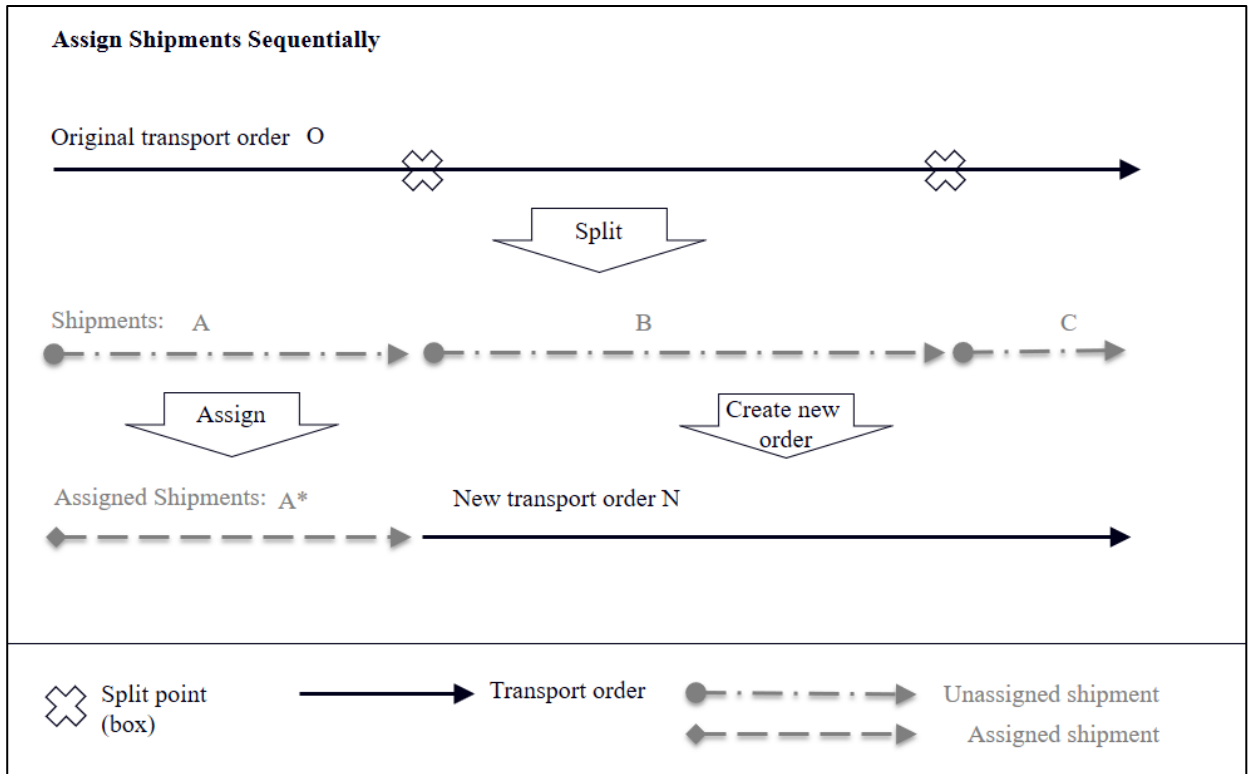


Figure 11 - Assign Shipments Sequentially - Scheme

At the end of this process step the original transport order O is split into different shipments (here A, B and C), but only shipment A is assigned 1:1 to a courier. The remaining unassigned chain of shipments is re-grouped into the new transport order N in the next master process step (see Figure 7 right side). From then on, N is treated as any other transport order.

The benefit of this sequential assignment is to keep as much decision space as possible open for later assignment decisions – especially unnecessary time window constraints are avoided this way.

The drawback of this assignment method is that good offers for later shipments cannot be harvested before the next-in-line shipment has been assigned.

2.3.4.3. Assign Shipments Partially

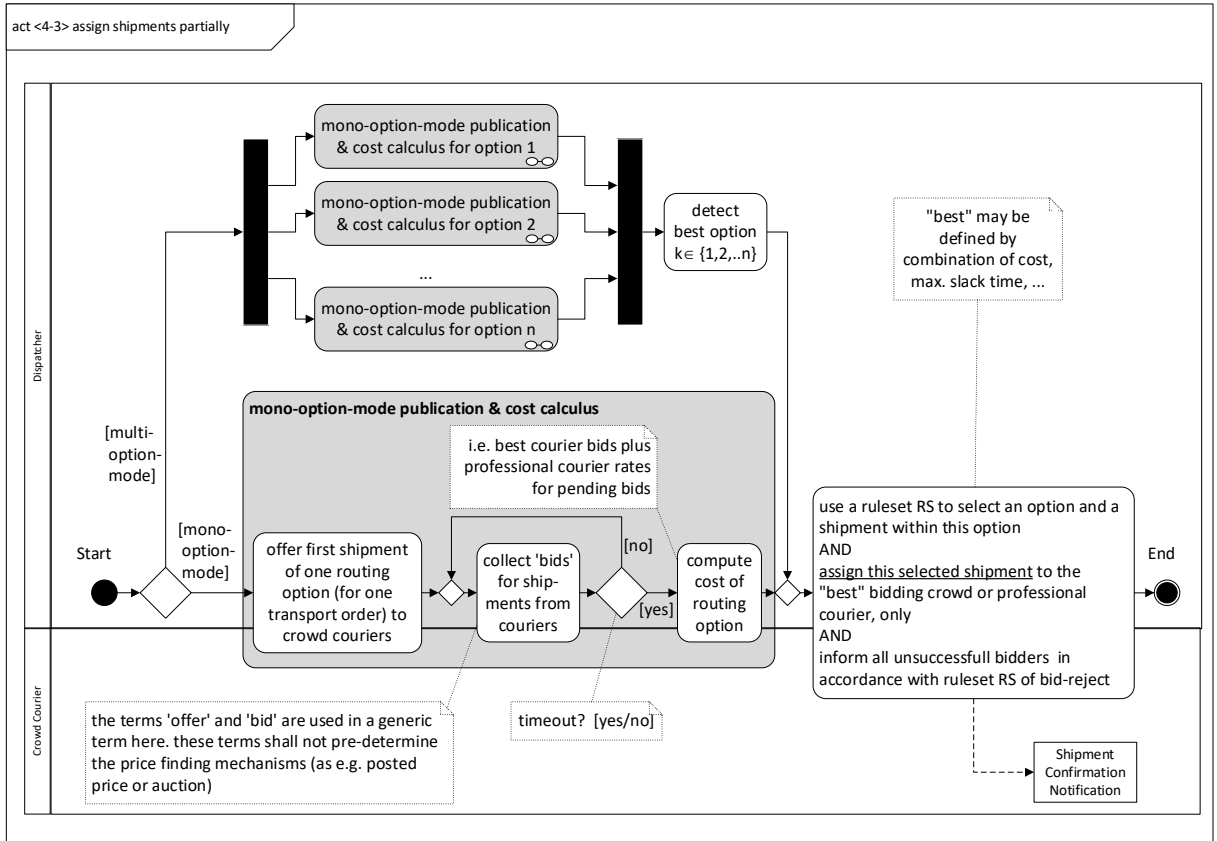


Figure 12 - Assign Shipments Partially - Process

This sub process also can be run in a “mono-option mode” – i.e. there is only one split instance and only one set of subsequent shipments which represent the original transport, or it can be run in a “multi-option mode” – i.e. several split instances (and thus several sets of subsequent shipments) are offered (not assigned!) to the couriers in parallel.

Note that in this partial assignment variant only one of several shipment is assigned whereas the preceding and subsequent shipments remain unassigned and are converted into a “new” transport orders which represents the remainder of the original transport order (see Figure 7).

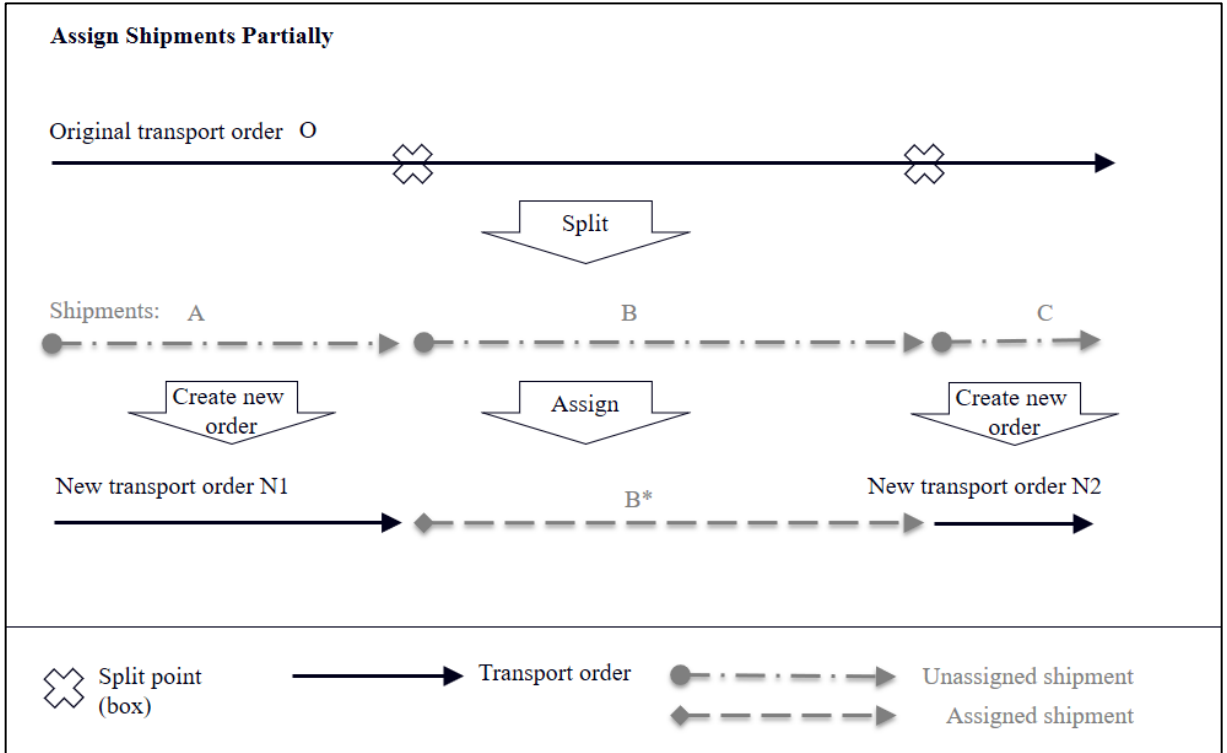


Figure 13 - Assign Shipments Partially - Scheme

At the end of this process step the original transport order O is split into different shipments (here A, B and C), but only shipment B is assigned 1:1 to a courier. In the next master process step (see Figure 7 right side), the remaining unassigned shipments A and C are converted into the new transport orders N1 and N2. From then on, N1 and N2 are treated as any other transport orders.

The drawback of this partial assignment is a creation of possibly unnecessary time window constraints for the different transport legs, as at the end all assigned shipments must form a synchronized transport chain.

The benefit is the potential to harvest low cost bids from the couriers (whilst hoping that the resulting new transports with probably tighter time-window can be assigned to a suitable courier later).

**2.3.4.4. Input**

- Transport order data
- Sorted list of possible routing options

**2.3.4.5. Output**

- Assigned shipments
- Optionally new transport order data (which represent the unassigned leftovers of the previous order data)

**2.3.4.6. Comments**

Which of the three assignment schemes is best cannot be decided from a conceptual point of view. Simulations and trial implementations should be conducted to test the performance of the different schemes in real life settings.

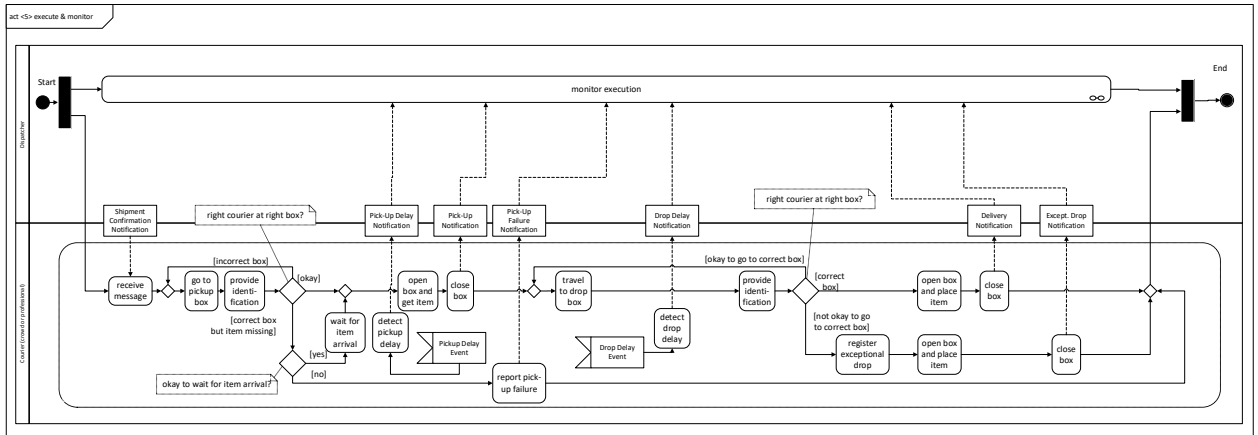


Figure 14 Execute & Monitor – Process

The “Execute & Monitor”-sub-process can be divided into two parts - i.e. pick-up and drop.

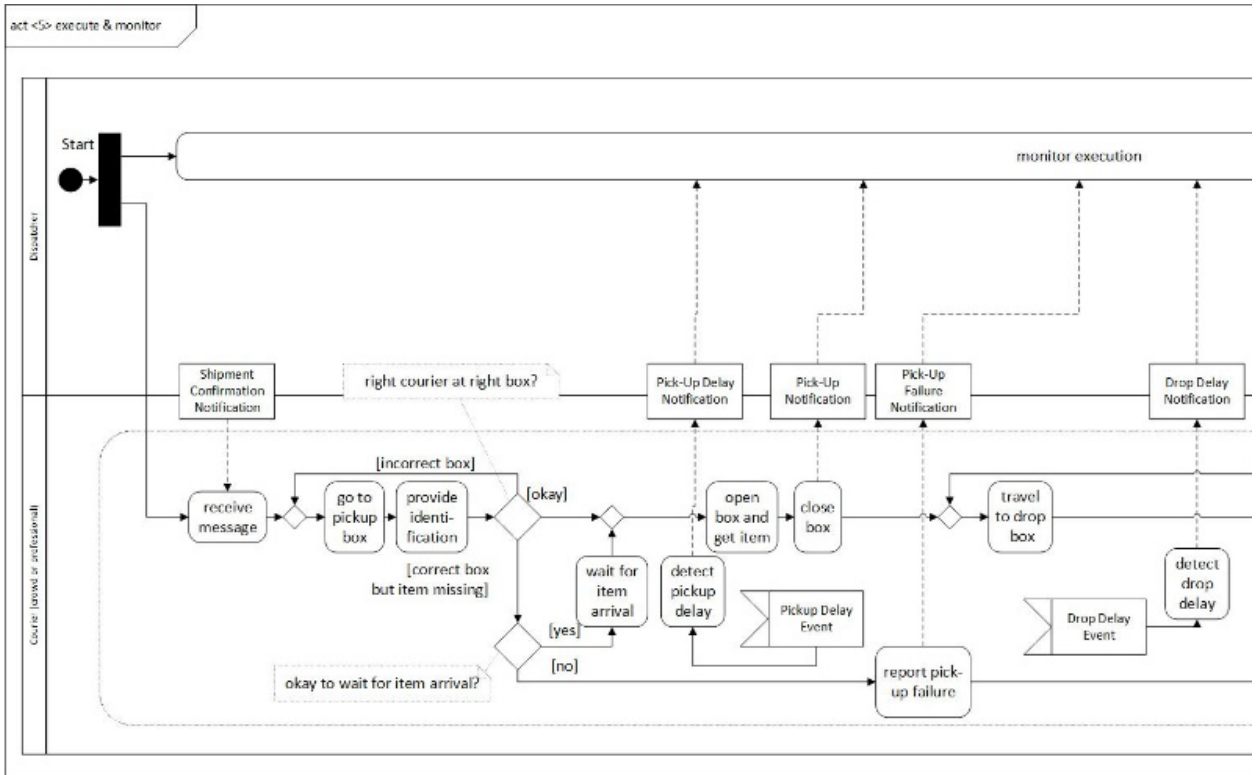


Figure 15 – Execute & Monitor Part 1 – Pick-Up

The sub-process starts with the shipment confirmation notification, which is sent to the courier. Once the courier shows up at the relevant box, it is checked, if the courier is at the correct box. If yes, he/she can open the box with his/her app and take out the relevant transport item. Then he/she closes the box and starts his/her journey to the drop box.

The following exceptions to the successful pick-up are modeled:

If the courier made a mistake and shows up at the wrong location, he/she should get an error message directing him/her to the correct location.

If the box location is correct, but the item is not there, yet, the courier should get an error message asking him/her to wait. If the courier agrees to wait, he/she waits till the item has arrived. If the courier doesn't agree to wait any longer a pickup-failure message is generated.

If the courier doesn't show up within the defined time window, a pickup-delay-notification is generated.

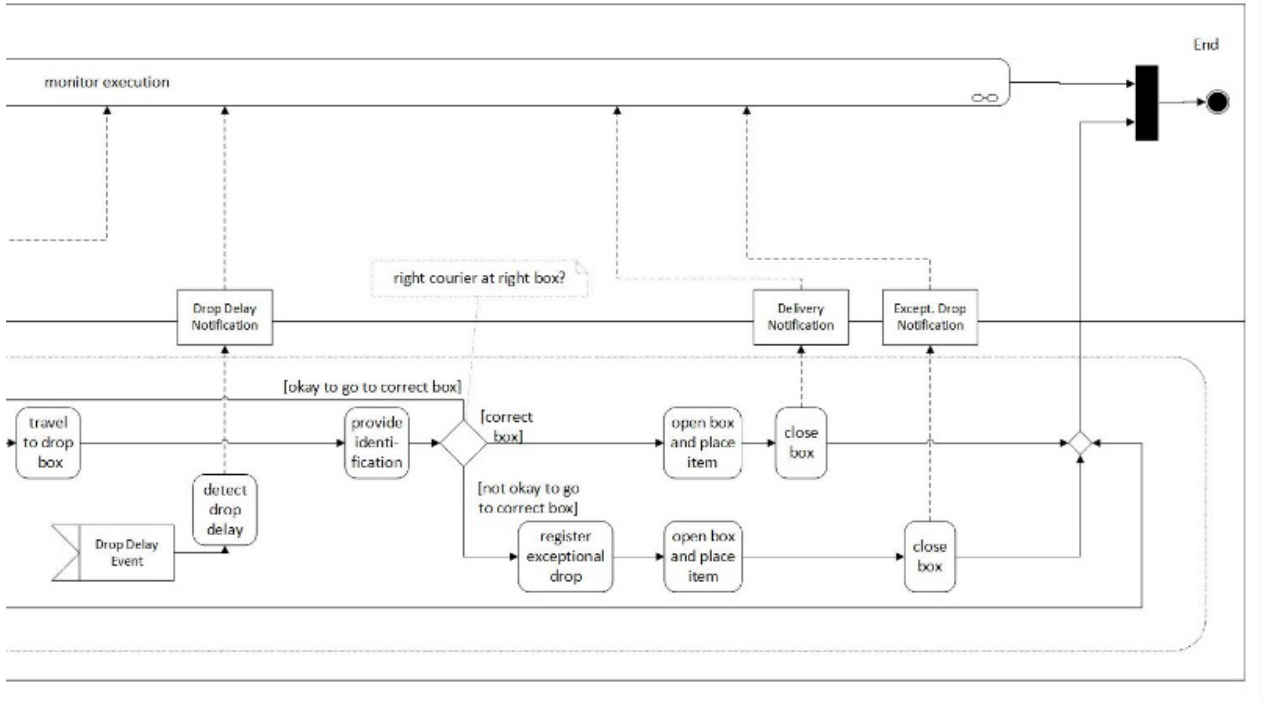


Figure 16 Execute & Monitor Part 2 – Drop

Once the item has successfully been collected by the courier, he/she travels to the relevant drop box. There the courier is identified, opens the box, places the item inside and closes the box, and a delivery notification is generated. Note that the identification of the item needs to be technically checked e.g. via bar-code-scan, RFID-read, etc.).

The following exceptions are foreseen in the model:

If the courier shows up at a wrong box, and agrees to go to the correct box he/she is re-directed.

If the courier shows up at a wrong box, and does not agree to go to the correct box he/she can still drop the item in the wrong box (exceptional drop notification). Note that this notification will require several subsequent re-organization tasks (see chapters on deviation management), and should reduce the courier's remuneration significantly.

2.3.5.1. *Input*

- Assigned shipments
- Item IDs
- Courier IDs
- Optional: Courier GPS-data (collected via app)

2.3.5.2. *Output*



- Delivery status information

#### 2.3.5.3. *Comments*

Note 1: A sub-process which allows the courier to reject the assigned shipment is not envisioned here. In case it turns out, that such a sub-process is required, it could be added to this process model in a later version.

Note 2: The option to drop the item in the wrong box (exceptional drop notification) requires a number of subsequent changes, which are not specified in detail here. Still this option will generate a defined intermediate delivery status. Such a defined status seems better, than the undefined status of the courier keeping the item and carrying it around. This status also could be used in a local box breakdown.

Note 3: The processes which deal with the different exception-status are defined below (see chapters on deviation management).

2.3.6. Manage Minor Deviations

A minor deviation is defined as a deviation which requires no re-scheduling actions, but simply triggers a message to couriers and customers about delays.

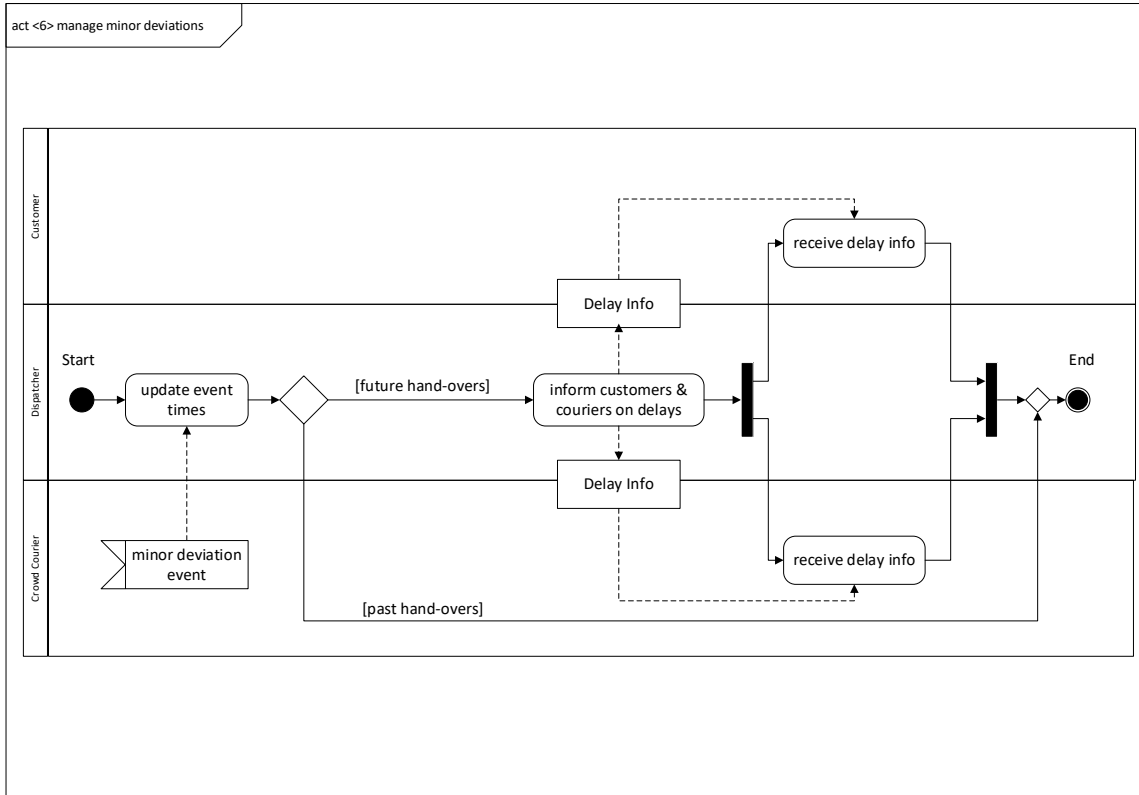


Figure 17 – Manage Minor Deviations

As delays may have an impact on future handovers as well as on the final delivery time, both couriers and customers are informed on relevant delays.

2.3.6.1. Input

Automatically computed delay events based on:

- Planned arrival times
- Optional: GPS-based arrival time estimates
- Arrival times

2.3.6.2. Output

Delay notifications

2.3.6.3. Comments

One might add a check in this sub-process, which computes the chances to deliver on time, even if minor delays have already been detected. Based on this computation one might decide on whether the customer shall be informed or not.

2.3.7. *Manage Major Deviations*

A major deviation is defined as a deviation which requires re-scheduling actions.

One can differentiate between two cases here:

- A sole shipment which is currently carried by one courier to its destination is affected by the delay
- A chain of downstream shipments is affected by the delay (i.e. at least one downstream shipment is affected by a delay occurring upstream in the chain)

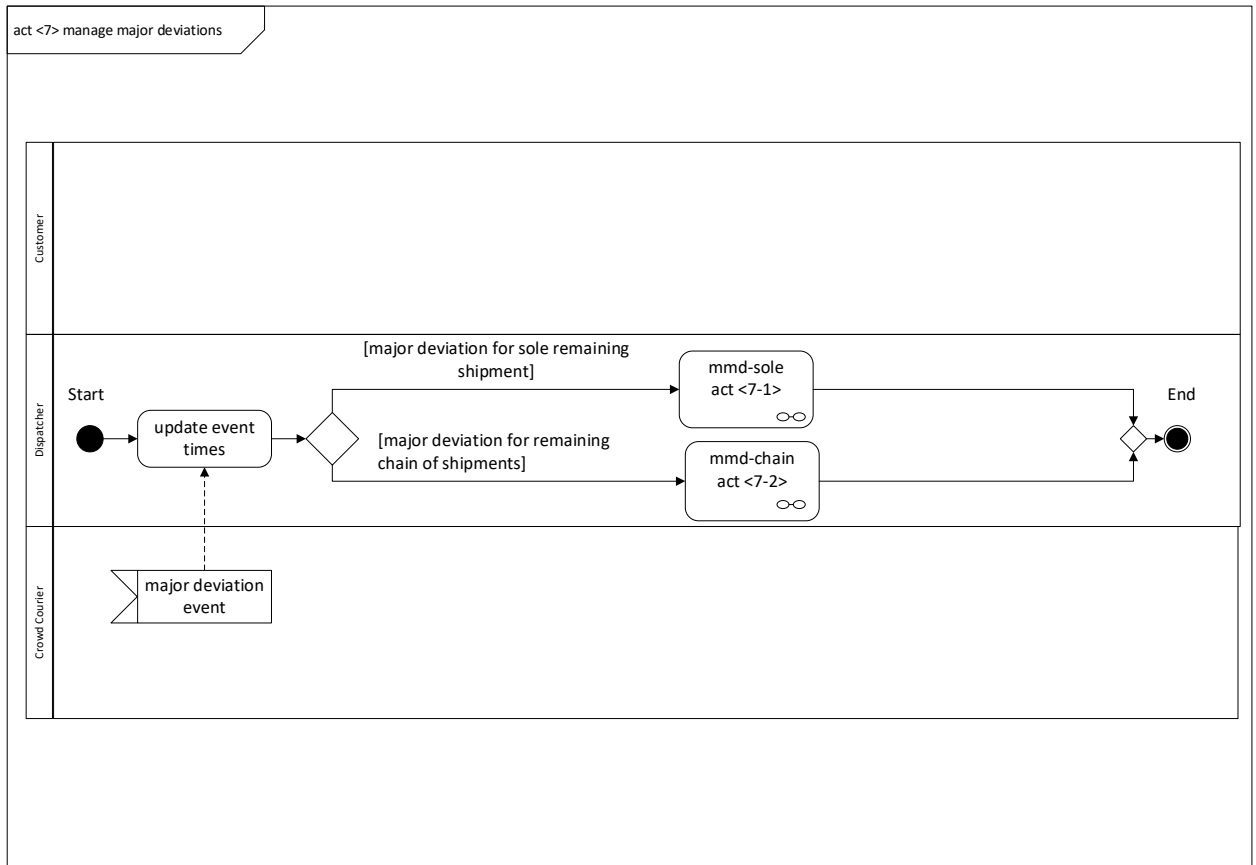


Figure 18 – Manage Major Deviations

2.3.7.1. Manage Major Deviations – Sole Shipment

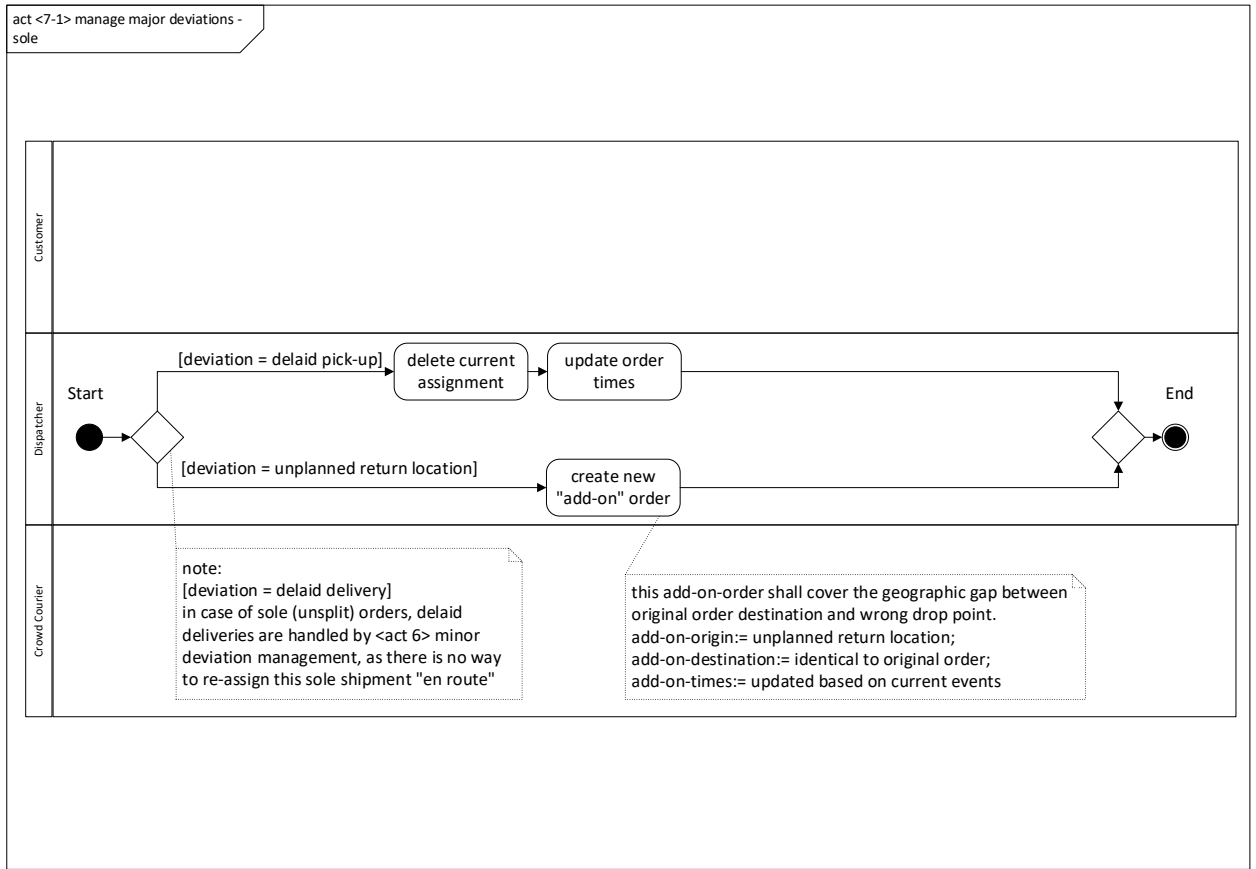


Figure 19 – Manage Major Deviations – Sole Affected Shipment

The following two problems can lead to a schedule change, if they occur for the last downstream shipment:

- The last pick-up is significantly delayed – in this case a replacement courier is to be scheduled. Thus the current assignment is deleted, and the now unassigned shipment is converted into a new transport order with updated time windows.
- The last drop is misplaced – i.e. the item is not dropped at the required location – in this case the remaining transport leg has to be fulfilled by means of a new “add-on” order (from misplacement box to customer) which is created automatically.
- Note: If the final drop at the customer location is delayed, this requires no scheduling changes. Therefore, such a delay is a minor deviation.

2.3.7.2. *Manage Major Deviations – Chain of Shipments*

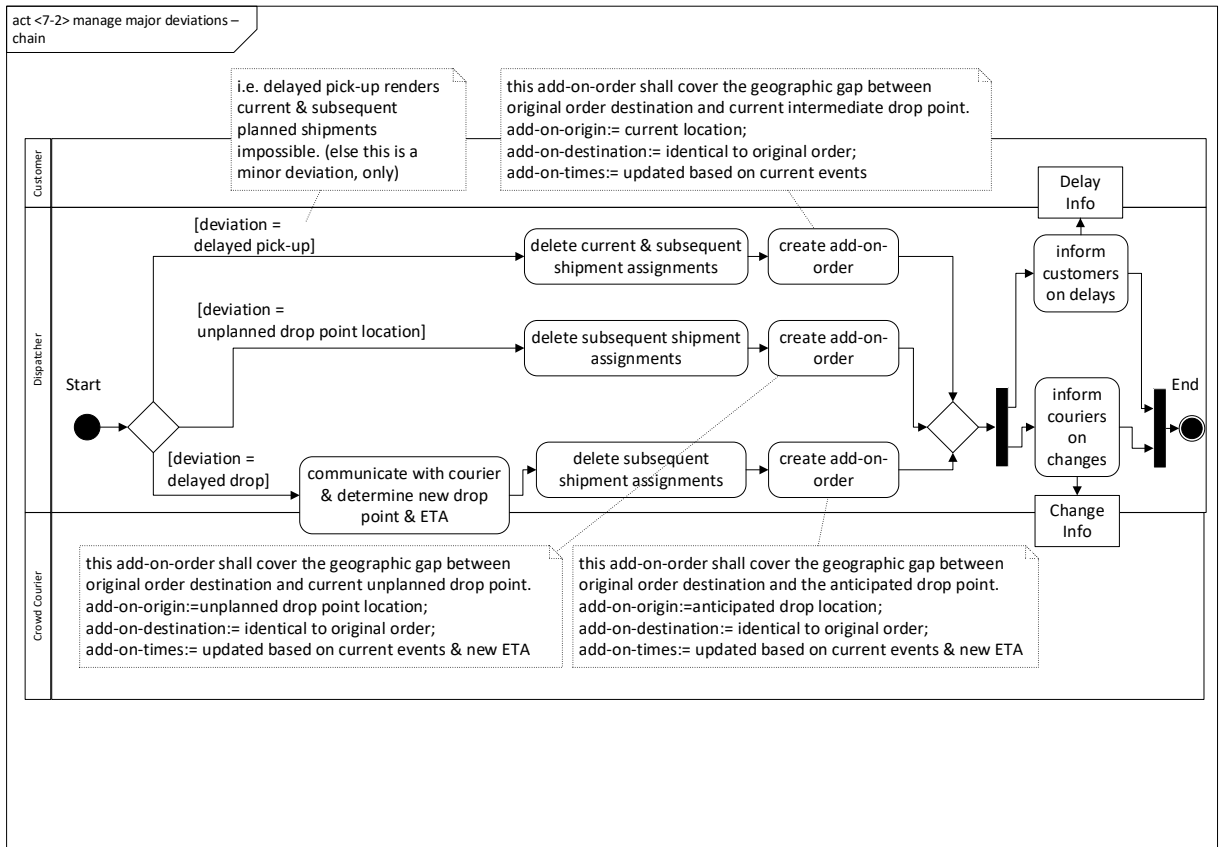


Figure 20 - Manage Major Deviations – Chain of Affected Shipments

Three events may have an effect on subsequent shipments:

- A delayed pick-up – in this case a replacement for the no-show-courier is needed. As it takes time to schedule the replacement it is most likely, that the subsequently assigned shipments cannot be carried out as planned. Therefore, the downstream-assignments are all cancelled, and a new transport order representing the remaining transport to the customer is created. Note that one might expand this sub-process at this point to introduce a check, whether subsequent shipments are affected or not – if yes, nothing changes, but if no, then it is sufficient to re-assign the non-picked-up shipment.
- A delayed delivery – in this case the delayed shipment is still ongoing and needs to be returned to a box before it can be processed further. Therefore, we suggest a communication with the delayed courier in order to determine when and where he/she can drop the item. Based on this communication result, the downstream shipments are deleted and a new transport order representing the remaining transport to the customer is created. Note that one might expand this sub-process at this point to introduce a check, whether subsequent shipments are affected or not – if yes, nothing changes, but if no, then it is sufficient to wait for the completion of the delayed shipment.
- A drop at an unplanned location – in this case the old downstream chain of shipments is deleted, and a new transport order representing the remaining transport to the customer is created.

2.3.7.3. *Input*

- Delay events

2.3.7.4. *Output*

- Deleted assignments
- Deleted shipments
- New transport orders

#### 2.3.7.5. *Comments*

The management of major deviations for a chain of affected shipments is one of the worst case scenarios for a human dispatcher or for an autonomous dispatcher software. Therefore, it cannot be ruled out, that real life implementations of this sub-process will show needs for further process-refinements.

2.3.8. Remuneration

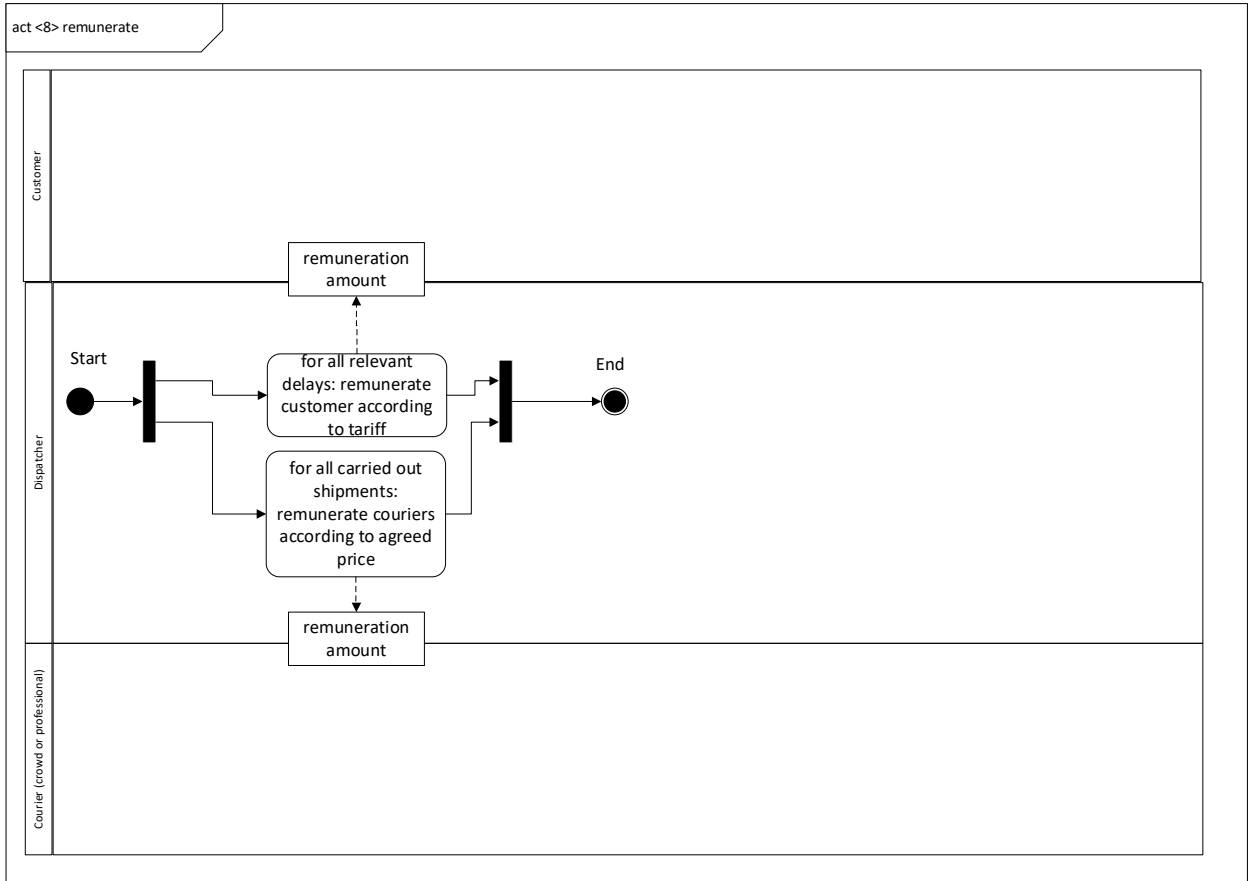


Figure 21 - Remuneration

As the customer has already been billed upon order placement (see Figure 5) it may be necessary to re-compensate the customer for delays.

This sub-process-model assumes, that the courier is only paid after the execution of his/her services.

2.3.8.1. Input

- Relevant remuneration details (as e.g. distance, negotiated prices)
- Deviations (no-shows, delays and/or alternative drop locations)
- Remuneration rule sets

2.3.8.2. Output

- Computed delay compensations for customers
- Computed remuneration amounts for couriers

### 2.3.8.3. *Comments*

This sub-process is built on the assumption, that each shipment is remunerated individually. Therefore, a theoretically possible tariff based bundle remuneration for professional couriers is not included in the above process scheme.

## 3. **Critical Discussion**

The presented process model is a conceptual model, only, which was developed by scientists. It is based on the basic conceptual ideas of the DTM-process model which has been presented by (Kunze, Baumgärtel et al 2012).

The only verification instance of this model has been the CCL-practitioners panel, where the model was presented, and no alterations were suggested at this point.

Because the process model has not been tested in daily operation, it should be considered a preliminary version (v0.5) of a process model in contrast to a version (v1.0) which has successfully been tested, a later version (v2.0) which would incorporate process improvements (that will have evolved from operational use) and maybe a standardized version (v3.0) which is defined by a standard setting process certification body.

Still, this process model can be considered as a twofold contribution to the scientific as well as to the practitioner's world. In the world of practitioners, it can be used as a starting point for any service provider who endeavors to start a city crowd logistics service. And in the scientific world, it is a step towards a generalized process concept for crowd shipping and/or co-operative shipping as envisioned in the concept of the Physical Internet (PI).



## Glossary

**Box:** A cross dock device to enable non-face-to-face hand-over of items from one courier to another. It can technically be implemented as a shelf of locker boxes which can individually be opened and closed by means of a mobile phone app and secure access codes. A box serves as a router in this concept.

**CCL:** City Crowd Logistics

**Courier:** Person who transports shipments.

Couriers are distinguished by their *commercial role* for this process model:

- *crowd courier* – including:

- pro bono courier (courier who will provide courier services without remuneration)
- private courier (non-professional courier who will provide courier services against remuneration)

- *professional courier* (employed to do courier services)

Couriers can also be distinguished by the *means of transport* they mainly use, e.g.:

- bike courier
- car-courier
- pedestrian courier (sheer walking)
- public transport courier (incl. walking)
- other courier (e.g. skateboarder, e-scooter-driver, ...)

But as crowd couriers might use different means of transport to execute one transport leg (e.g. foldable bike and subway), this differentiation is not unique

**Customer:** Person who wants a good to be shipped from origin to destination – the customer triggers the order and pays for it

**DTM:** Dynamic Truck Meeting

**Order:** Original transport order (origin to destination) as defined by the customer

**Router:** Transshipment point, where shipments can be dropped or picked up by couriers. Typically a router is a box or a shop or kiosk, which has space to store items temporarily and has agreed to operate as a router.

**Routing Option:** A routing option is a realistic possible option (with rough consideration of current routing context, but without any courier assignments, yet)

**Routing Variant:** A routing variant is a theoretic possibility for a routing (without consideration of current context)

**PI:** Physical Internet

**Shipment:** A geographical and time wise part of order (i.e. a sequential execution of all related shipments will execute the order)

**UML:** Unified Modeling Language

## References

- Aboudina & Abdulhai 2017: Aboudina, A. and Abdulhai, B. (2017). “A bi-level distributed approach for optimizing time-dependent congestion pricing in large networks: A simulation-based case study in the Greater Toronto Area”. In: Transportation Research Part C: Emerging Technologies 85, pp. 684–710.
- Buldeo Rai, Verlinde, Merckx & Macharis 2017: Buldeo Rai, H., Verlinde, S., Merckx, J., and Macharis, C. (2017). “Crowd logistics: an opportunity for more sustainable urban freight transport?” In: European Transport Research Review 9.39.
- Crainic & Montreuil 2016: Crainic T.G. & Montreuil B. (2016): Physical Internet Enabled Hyperconnected City Logistics, Transportation Research Procedia, Volume 12, 383-398, Elsevier

Crainic, Perboli & Rossano 2018: Crainic, T. G., Perboli, G., and Rosano, M. (2018). "Simulation of intermodal freight transportation systems: a taxonomy". In: *European Journal of Operational Research* 270.2, pp. 401–418.

Klemperer 2004: Klemperer, Paul (2014): "Auctions: Theory and Practice", Princeton University Press, Princeton & Qxford

Kunze, Baumgärtel et al 2012: Kunze, Oliver; Baumgärtel, Hartwig, Neitmann, Andreas and Rosemeier, Sebastian (2012): *Dynamic Truck Meeting, Vol. 1 Process Standard*, KIT Scientific Publishing, Karlsruhe

O’Byrne 2016: O’Byrne, R. (2016). "The Uberisation of Logistics". In: *MHD Supply Chain Solutions* 45.5, pp. 36–38.

Pan, Trentesaux, Ballot & Huang 2019: Pan, S., Trentesaux, D., Ballot, E., and Huang, G. Q. (2019). "Horizontal collaborative transport: survey of solutions and practical implementation issues". In: *International Journal of Production Research*. Published online. DOI: 10.1080/00207543.2019.1574040.

Sampaio, Savelsbergh, Veelenturf & van Woensel 2019: Sampaio, A., Savelsbergh, M., Veelenturf, L., and van Woensel, T. (2019). "Crowd-Based City Logistics". In: *Sustainable Transportation and Smart Logistics: Decision-Making Models and Solutions*. Ed. by J. Faulin, S. Grasman, A. Juan, and P. Hirsch. Elsevier, pp. 381–400.

Taniguchi 2014: Taniguchi, E. (2014). "Concepts of City Logistics for Sustainable and Liveable Cities". In: *Procedia – Social and Behavioral Sciences* 151, pp. 310–317.

Taniguchi & Shimamoto 2004: Taniguchi, E.; Shimamoto, H. (2004). "Intelligent transportation system based dynamic vehicle routing and scheduling with variable travel times". In: *Transportation Research Part C: Emerging Technologies* 12.3-4, pp. 235–250.

### Project Funding Note

This report is a result of the government-sponsored research project "CCL" which was funded by the German Federal Ministry of Economics and Energy based on a decision of the German federal parliament.

Gefördert durch: Bundesministerium für Wirtschaft und Energie aufgrund eines Beschlusses des Deutschen Bundestags

Gefördert durch:



aufgrund eines Beschlusses  
des Deutschen Bundestages