Hierarchical robustness model for business processes

Natalja Kleiner[a], Peter C. Lockemann[*,a]

[a] FZI Forschungszentrum Informatik, Karlsruhe, Germany

Abstract. *Most business processes today can be easily modelled and controlled by advanced business process management systems. When it comes to processes, that are driven by outside events and require fast reactions to contingencies in order to stabilize them, e. g., transport or production processes, business process management systems often seem to reach their limits. In this paper we introduce a robustness model which is based on a business process model of an undisturbed transport process and extends it by a generic contingency detection model. We employ a hierarchical organization for deriving corrective actions with least possible modifications to the original transport process.*

Keywords. Logistics • Failure Management • Failure Model • Complex Event Processing

1 Motivation

Business processes are usually modelled by workflow or business process techniques. Special software – workflow or business process management systems – is employed to drive the process across the actions in the model. The underlying assumption is that most of the actions take place in the information processing world and can easily be controlled by the management system.

Processes in logistics, e. g., in production or transport, follow a sequence of steps, sometimes with some alternatives provided. Hence one would expect that these processes could easily be modelled by workflow or business process techniques. Contrary to the assumptions above, however, the processes are driven by outside events in the real world and not a piece of software. Hence, a process management system by itself would not make sense. Further, the major actions are real-world activities such as loadings, transports, deliveries. Just a few actions reflect information processing activities, mostly in a supporting role, e. g., by registering the state of the process or by automating accompanying paperwork.

* Corresponding author.
E-mail. lockemann@fzi.de

We claim in this paper that business process models could still form an important base in logistics, though in a more special manner. Numerous contingencies may arise along a transport process. Suppose that a business process model reflects the regular, undisturbed transport process, then what we plan to achieve is to use the model as a framework for deriving corrective or re-planning actions in a systematic fashion.

The article is organized as follows. In section 2 we briefly touch on related work. In section 3 we give an example scenario. Section 4 introduces a generic model for the disturbances along the transport process. Section 5 builds on the model and develops a system that drives the necessary corrections to the process. Section 6 concludes the paper.

2 Related work

Systems, which withstand disturbances, are called *robust*. To be more precise, Wikipedia defines robustness ‘as the ability of a system to resist change without adapting its initial stable configuration’, a definition that is only helpful if one specifies what is meant by ‘change’ or ‘stable’. Also, the definition has a static flavour. A more process-oriented, dynamic view describes robustness as

the capability of a process to function reliably even under unfavourable conditions (Vogel et al. 2009). Again, whether a process is considered robust depends on the pertinent definitions of 'reliable' and 'unfavourable'.

In their general classification of system dependability (or reliability) aspects, Laprie et al. (1992) distinguish two ways of coping with failures in dependable systems: fault prevention and fault tolerance. Fault prevention is concerned with how to prevent fault occurrence, and is, to a large degree, a design issue and requires design rules which help to avoid introducing failures in a system. Fault tolerance deals with how to provide a service complying with the specification in spite of faults. Since transport logistics is driven by external forces, extraneous faults and failures seem unavoidable. Hence fault tolerance is the dependability issue. One of the first authors to consider robustness in a logistics context have been Wieland and Wallenburg (2012).

Planning for robustness relies on a list of expected failures and describes alternatives to be taken when a particular failure arises. Hagen and Alonso (2000) suggest that due to the control system complexity one should separate failure handling aspects from the normal flow of control. They demonstrate the principle with an approach from a transaction perspective and propose atomicity and exception handling as the two fundamental techniques to deal with fault tolerance. In case of a failure, an application or parts of it are rolled back to a previous consistent state (backward recovery). From this state, the computation continue by following alternative or compensation execution paths (forward recovery).

Such an approach makes sense if most of the actions are confined to information processing. In transport logistics, by the time a failure has been detected, the process has left too many irreversible traces in the real world to have a chance to return to a previous state. Hence, forward recovery is the only way to proceed. Standard approaches in workflows are either to dynamically modify the workflow at the point of failure, or to provide a set of mini-workflows to execute in place of the failed action. For an example, see, e. g., Lanz et al. (2010). As part of the Workflow Pattern Initiative, Russell et al. (2006) group unanticipated events into classes which are related by similarities in terms of conditions under which they may arise. Based on these, they develop patterns, i. e. generic recurring constructs, to incorporate in a workflow. Cognini et al. (2016) introduce richer sets of modelling constructs in the form of business process fragments and variants that include a wide repository of constraints.

As we shall demonstrate below, failures in transport logistic are of varying severity, and the scope of their effect across a network may differ considerably. Current solutions do not seem to account for those variations, at least not in a systematic fashion. We present a novel approach that organizes failures into a hierarchy, where those on higher levels have a wider effect than those on lower levels. We associate with each failue a compensating action with a concomitent reach of the effect. We escalate the actions up to the level where a suitable action can be found.

3 An example: Open logistics networks

The overall objective of transport logistics is to provide the desired goods in the correct volume at the right time and right place. Stakeholders are the suppliers, customers and transport carriers are the intermediaries. Since stakeholders of all three kinds interact in numerous ways, they form a network. An open network is one where stakeholders may freely enter or disconnect, and are free to enter temporary collaborations, e. g., to share transfer orders, improve the utilization of load capacities, or to help out in damaging situations. Particular challenges arise in a clocked network with carriers, which must follow a rigid timetable, e. g., railway companies. Figure 1 illustrates the principle.

Below on the left, goods are collected from three suppliers via separate trucks and the consolidated in a first transition hub (so-called hub-and-spoke principle), whereas on the right a single truck collects the wares from two suppliers and delivers

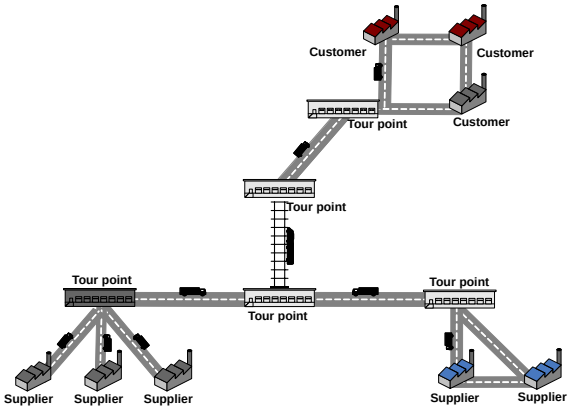

Figure 1: Example of a logistics network

them to a second hub (a so-called round trip). Both hubs will send trucks (perhaps after consolidating the previous loads with wares from further sources) to a so-called railport. Note that nothing has been said on whether the loads belong to one or more orders or whether they go to one or more customers. The rail hub by necessity consolidates a large number of orders into the load of a complete freight train which leaves according to a predetermined schedule. Likewise, at the other end a corresponding distribution over several legs will take place. From now on we will refer to points of loading or unloading and hubs collectively as tour points and to the transport between two tour points by a single vehicle as a tour.

The entire transport chain can be modelled by any suitable formalism, e. g., BPM or UML. In the remainder we abstract from any particular formalism.

4 Basic elements of failure management

4.1 Robustnesss

Whether a logistics network is robust or not is in the eye of the beholder. Since there are many stakeholders, one can expect that each of them holds an individual view on robustness. If we take the network as a whole, then, according to section 3, we should consider a logistics network robust if an order placed by a customer with a supplier is delivered at the specified place and time in correct composition and volume (see Magnus and Thonemann 2007), and that this should hold for every customer-supplier relationship within the transport system.

During transport, deviations from schedules or routes are the norm. Fortunately, most of them remain 'under the radar', i. e., are not noticed at all or considered inconsequential. For a deviation, that can no longer be safely ignored, but should at least be tracked because it could endanger the robustness of the transportation process, we use the term *exception*. Not every exception will have repercussions on the normal course of events and actions. Those that do will be termed *contingencies* because they require some sort of counteraction. *Failure* serves as a generic term to cover both, exceptions and contingencies.

4.2 Exceptions and Contingencies

Transport systems are more or less continuously monitored by collecting various measurement data from outside, using technical devices like RFID scans, GPS tracking, thermometers, pressure meters, and, sometimes, human observations. As long as these do not raise an alarm, a disruption and the resulting deviation remain unobserved. When they do, a deviation may qualify as an exception or a contingency.

On the physical level the exceptions and contingencies have to do with disruptions, which generally result in delays. Take traffic jams, detours, driver's indispositions, truck or train breakdowns, missing, incomplete, defect or incorrect shipments at a supplier, non-available ramps or storage areas at hubs, unpreparedness at the customer site, incorrect transport documents, vehicle replacements of the wrong type, to name a few. Delays may result in failures of much farther reach. Take a delay due to vehicle breakdown. Suppose a replacement vehicle can be found. This will affect the entire truck fleet of a transport company or even a second company. And even after one has been found, it may turn out that it has insufficient load capacity to pick up all the goods along the tour it was scheduled to do. In the worst case the transport may miss a scheduled train and wait for the next train, causing delivery of the order

at the customer site to be late by many hours or even a day. Disruptions may not only originate on the physical level. Consider emergency repairs in a vehicle fleet nullifying a tour plan, or the cancellation of parts of an order or an increase of volume of an order.

We conclude that although deviations may occur on a local level, their effects may reach much farther. A careful analysis of logistics networks shows that one can distinguish five spheres of influence that form a hierarchy (see figure 2). As mentioned above, some deviations may even originate on higher levels.

4.3 Buffers

A time-honored approach to robustness is to build some slack into the business process. Basically one adds some reserve capacity to the resources employed in the process (Bretzke 2010). For example, to overcome delays one allows more time than absolutely needed for the tour, or earlier arrivals than in the exact plan. Likewise, one may provide larger load capacity than minimally needed. Of course, this comes at a price because additional or larger trucks must be kept in reserve. We refer to the spare resources as *buffers*.

In designing failure systems one has first do define what one considers resources. Are they exclusively of a physical nature such as trucks or personnel, or also conceptual such as tour times or tour point sequences? Given these one can then assign buffers to them. Next, one can estimate how far their influence may reach, and arrange them within the hierarchy of spheres. For example, tour time buffers or load capacity buffers belong to the tour level because they can be used to alleviate disruptions of a single tour. Further examples for this level are alternative routes or alterations in the sequence of tour points. Typical examples on the transport level, where an entire transport from a supplier to a customer is considered, are excess loading personnel, several scheduled trains or unused warehouse capacities in the hubs. On the order level we observe buffers like delivery time or order splits. Typical for the fleet level are reserve trucks, and for the top, network level alternative suppliers, customers or transport carriers. Figure 2 illustrates the hierarchical arrangement of buffers.

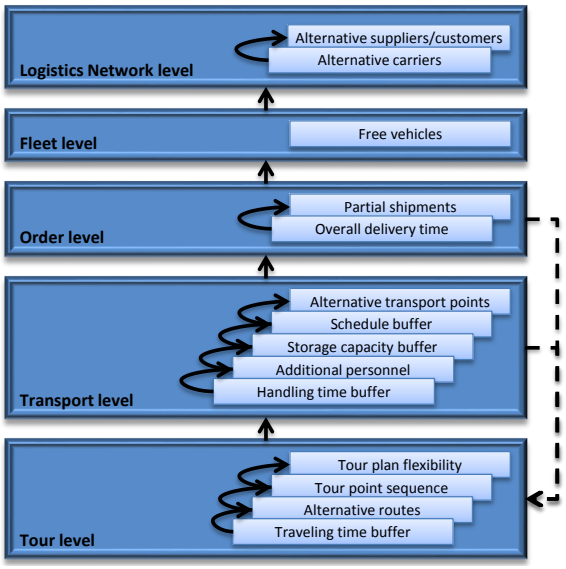

Figure 2: Buffer hierarchy

4.4 Tolerance intervals

Even though the business plan merely seems to describe the logistics process rather than drive it, it evidently plays an active role in case of failures. Certain buffers are statically specified as part of the plan, but are then consumed whenever spare resources are used for overcoming an exception. By drawing on the buffers and dynamically adjusting them, the business plan seems to occupy the driver's seat once an exception occurs.

For example, compensating for a delay may consume half of a delay buffer at a given tour point. Since a delay buffer can be associated with, an entire round trip the other half is still available for further delays along the run. Hence, the resource represented by a buffer seems to shrink dynamically. We refer to the dynamic counterpart of a buffer as a *tolerance interval*.

5 Failure handling

5.1 Deviation detection

As mentioned, the external system raises an alarm if a worthwhile disruption has been observed. This

alarm is usually referred to as an *event*. In fact, many events caused by observations may only become meaningful if they are considered in a larger context. Just consider temperature measurements, where a dangerous situation is recognized early enough after successive values show a growing tendency. Or suppose that GPS tracking indicates that a truck is behind its expected position, and the traffic situation indicates that late arrival seems unavoidable.

Consequently, that makes it less than straightforward to detect an exception. On arrival of an event, the event must be checked of whether it may become part of a complex event, conditions must be checked, the data accompanying the event must be compared with a specified target value, and finally a decision be taken whether the event qualifies as a deviation. Deviation detection, therefore, must be captured in the form of a set of rules, e. g., ECA (event-condition-action) rules.

5.2 From deviation to contingency

Once a deviation has been detected one must determine which resources are affected, and hence which sphere should be examined. Since we know the current position in the workflow we can infer the resources. Take late arrival at a tour point. Then trucks, personnel and loading ramps are candidates. Associated with each resource are certain quality characteristics, e. g., truck schedule, load capacity, ramp assignments. These define the buffers to be inspected. Note that more than one resource may be affected, e. g., given a delay both the schedule of a truck and the work schedule of its driver should be examined.

Next, we must examine the candidate buffers and their tolerance intervals on the given hierarchy level. This should be done in a certain order. The arrows in figure 2 give an example. For each interval we subtract the deviation associated with the failure from the current value. If the interval remains above zero then the buffer could accommodate the deviation and therefore the deviation can be classified as an exception. If necessary, we go on and inspect the next buffer. For example, if a truck arrives late but within tolerance, and the driver can take a rest within tolerance all intervals remain above zero and the effect remains strictly local. If not, we may try to alter the sequence of tour points next. If this keeps us within tolerance – now for the entire tour – we still keep the effect within the tour level. However, because the correction now involves re-planning the tour, the deviation has morphed into a contingency. Re-planning is usually complicated. In finding a sequence, the tolerance intervals of the other tour points must be observed. In case of finding an alternative route one must employ routing algorithms. Re-planning algorithms are again associated with the buffers.

If the failure remains (locally) an exception, then it will not by itself affect the buffers of the successive transport points, because each buffer has been designed independent of all the other buffers. However, on arrival at the next point the intervals must be properly adjusted.

Whenever the intervals within a given level have been exhausted, i. e., at least one tolerance interval has fallen below zero we have a contingency that cannot be dealt with on the current level. Suppose that a truck has been delayed for too long to load wares at a transport point within the tolerance interval. One remedy could be to start a second truck on time to pick up the wares. However, this affects resources beyond the tour level. Figure 3 illustrates the procedure.

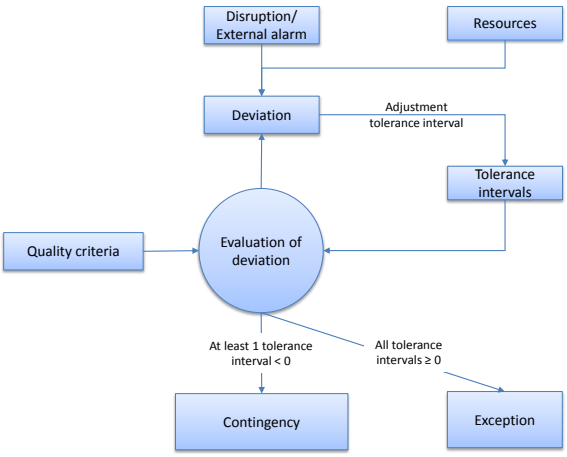

Figure 3: Failure handling on single hierarchy level

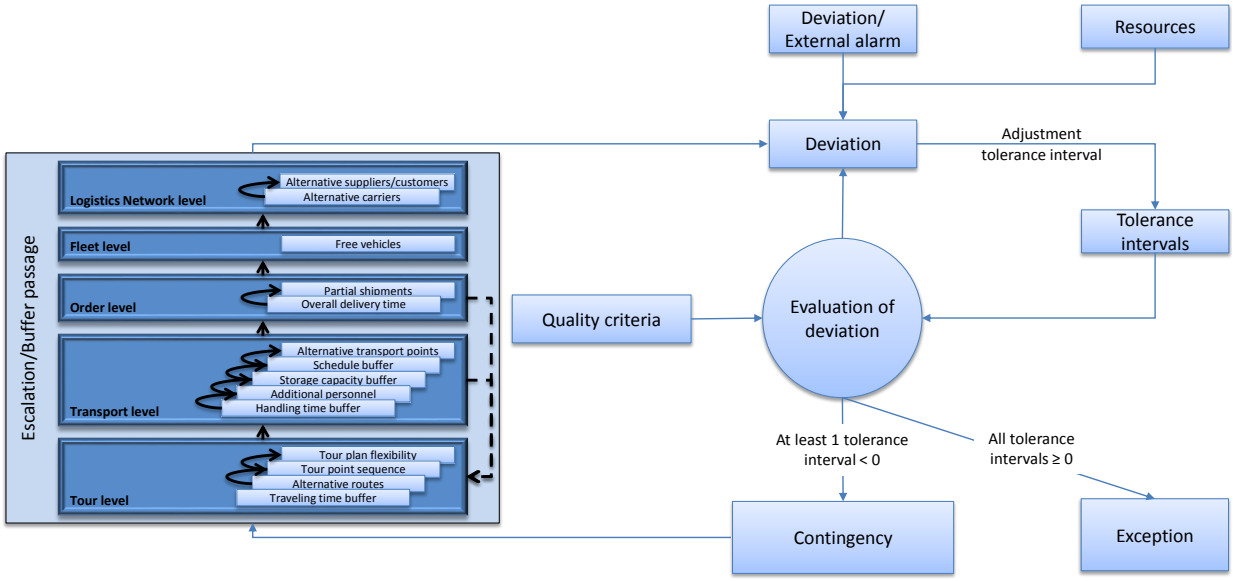

Figure 4: Escalated failure handling

5.3 Escalation

In the previous example the contingency had a reach beyond the current level: The suggested remedy cannot be dealt with locally but affects the carrier's fleet. A buffer reflecting spare trucks would have to be provided on a higher level, in this case the fleet level. This gives us a handle on how to treat contingencies: Escalate the contingency to a higher level and hope that the contingency is contained on that level. To give another example, in a tour level contingency one may examine the entire transport chain for additional slack due to delay. The transport level is aware of all the tour points along the transport chain and thus may inspect each of them whether they still could tolerate the original delay. It may discover that the railport offers, with its temporal and volume buffers, the possibility to take the next train without a violation of tolerance intervals further along the transport chain.

The basic idea of our approach is to try to resolve the contingency on higher levels of the hierarchy. Basically, the contingency is propagated to the next higher level (failure escalation). In principle, since each level in the hierarchy has its own set of resources with their commensurate buffers, the procedure of section 5.2 applies separately to each level in the hierarchy. Figure 4 illustrates the propagation within each level and the escalation across levels. Note that exceptions may originate on any level. For example, a customer may change an order while the underlying tour is already on its way, or a supplier may have to split an order into several parts.

Escalation is more complicated than propagation within a single level. Many or all tours within the transport chain must be individually replanned with the target utilizations and their buffers adjusted. Thus, escalation means change propagations up and down the hierarchy, and this perhaps several times (figure 4).

6 Conclusions

In general, business plans serve two purposes. First, they are a planning tool and as such document the business intentions. Second, by embedding them within a business process management system they become a vehicle to drive the business process. If large portions of the business process take place in the physical world, the second reason does not apply. Even then, as we have shown, a business plan may assume an active role, albeit as

the means to guarantee the robustness of a business process. To do so, one will have to embed the business plan into a failure management system, and augment the plan by resource buffers and associated re-planning algorithms.

Using the real-world example of a logistics network we were also able to show that if robustness is a matter of many resources and stakeholders resulting in spheres of influence of different width, a hierarchical approach eases the design of robustness and adds transparency.

We believe the approach is sufficiently generic to be applicable to other application domains, e. g., to production scenarios. However, further research is still needed to confirm the assumption.

The work reported is part of a dissertation by Natalja Kleiner. The paper is a bit of a historical reminiscence – Natalja is the second author's final student while H.C. Mayr was one of his early fellow researchers.

References

Bretzke W. (2010) Logistic Networks (in German). Springer Berlin Heidelberg

Cognini R., Corradini F., Polini A., Re B. (2016) Business Process Feature Model: An Approach to Deal with Variability of Business Processes In: Domain-Specific Conceptual Modeling: Concepts, Methods and Tools Karagiannis D., Mayr H. C., Mylopoulos J. (eds.) Springer International Publishing, pp. 171–194

Hagen C., Alonso G. (2000) Exception handling in workflow management systems. In: IEEE Transactions on Software Engineering 26(10), pp. 943–958

Lanz A., Reichert M., Dadam P. (2010) Robust and Flexible Error Handling in the AristaFlow BPM Suite In: Information Systems Evolution: CAiSE Forum 2010, Hammamet, Tunisia, June 7-9, 2010, Selected Extended Papers Soffer P., Proper E. (eds.) Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 174–189

Laprie J. C., Avizienis A., Kopetz H. (eds.) Dependability: Basic Concepts and Terminology. Springer-Verlag New York, Inc., Secaucus, NJ, USA

Magnus K., Thonemann P. (2007) Successful Supply Chain Cooperation Between Retailers and Consumer Goods Manufacturers: An Empirical Study of the Retailers Perspective (in German). Gabler Edition Wissenschaft. Deutscher Universitätsverlag

Russell N., van der Aalst W., ter Hofstede A. (2006) Exception Handling Patterns in Process-Aware Information Systems. BPM-06-04. BPM Center. http://bpmcenter.org/wp-content/uploads/reports/2006/BPM-06-04.pdf

Vogel O., Arnold I., Chughtai A., Ihler E., Kehrer T., Mehlig U., Zdun U. (2009) Software Architecture - Basics, Concepts, Applications (in German). Spektrum Akademischer Verlag

Wieland A., Wallenburg C. M. (2012) Dealing with supply chain risks: Linking risk management practices and strategies to performance. In: International Journal of Physical Distribution & Logistics Management 42(10), pp. 887–905