

A Software Architecture for Transportation Planning and Monitoring in a Collaborative Network

Anne Baumgraß, Remco Dijkman, Paul Grefen, Shaya Pourmirza, Hagen Völzer, Mathias Weske

► **To cite this version:**

Anne Baumgraß, Remco Dijkman, Paul Grefen, Shaya Pourmirza, Hagen Völzer, et al.. A Software Architecture for Transportation Planning and Monitoring in a Collaborative Network. Luis M. Camarinha-Matos; Frédéric Bénaben; Willy Picard. 16th Working Conference on Virtual Enterprises (PROVE), Oct 2015, Albi, France. IFIP Advances in Information and Communication Technology, AICT-463, pp.277-284, 2015, Risks and Resilience of Collaborative Networks. <10.1007/978-3-319-24141-8_25>. <hal-01437896>

HAL Id: hal-01437896

<https://hal.inria.fr/hal-01437896>

Submitted on 17 Jan 2017

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



A Software Architecture for Transportation Planning and Monitoring in a Collaborative Network¹

Anne Baumgraß¹, Remco Dijkman², Paul Grefen², Shaya Pourmirza²,
Hagen Völzer³, and Mathias Weske¹

¹ Hasso Plattner Institut, University of Potsdam, Germany
{anne.baumgrass,mathias.weske}@hpi.de

² Eindhoven University of Technology, The Netherlands
{r.m.dijkman,p.w.p.j.grefen,s.pourmirza}@tue.nl

³ IBM Zurich Research Lab, Switzerland
hvo@zurich.ibm.com

Abstract. Transportation planners require software support to easily monitor and dispatch transportation resources, especially when transportation is multi-modal and when resources from different companies in their network are being used. We call such an application a transportation control tower. This paper presents a software architecture for transportation control towers. It focuses in particular on the novel aspects of the software architecture. These are: the ability to easily configure the monitoring of resources and tasks; the ability to create the statements for monitoring resources and tasks based on the transportation plan; and the ability to dynamically adjust the monitoring statements. A prototype of the software architecture is implemented and evaluated on three usage scenarios.

Keywords: collaborative networks, transportation, transportation management system, control tower, transportation planning

1 Introduction

On a daily basis, planners at transportation companies are working on the assignment of transportation resources to transportation orders. Each transportation order involves a large variety of activities and may involve many different parties. Of course, activities include the actual transportation of the goods, either by the company of the planner or by partner companies. However, they also include logistic activities around the transportation, such as collecting an empty container to transport the goods in, temporary storage of the goods in a partner of a warehouse, and transshipment by a terminal. They also include administrative tasks, such as reserving resources of transportation partners, filling out forms, inspection of the goods by customs, and

¹ A working version of this paper is available as A. Baumgrass, R.M. Dijkman, P.W.P.J. Grefen, S. Pourmirza, H. Voelzer, M. Weske. A Software Architecture for a Transportation Control Tower. BETA Working Paper WP-461. Eindhoven University of Technology, Eindhoven, The Netherlands, 2014.

billing. Transportation planners are responsible for planning and monitoring these activities. This task is made more complex by the fact that the transportation plan for a transportation order often changes, for example, due to changes in the availability or cost of transportation resources. This may have a rippling effect on other transportation orders as well.

To support these tasks, a transportation planner would benefit from software that helps him monitor the tasks that have been performed for the transportation order, the tasks that still have to be performed, and the transportation resources that are assigned to, or are available for, the transportation orders. We also call such software a ‘transportation control tower’ analogous to the control towers that monitor transportation resources in aviation.

There exist a large number of transportation management systems (TMS), which provide functionality for monitoring transportation resources. In previous work [4], we did a thorough analysis of a collection of 49 of these tools. The difference between such more traditional TMS and a control tower is that a control tower provides a more open ecosystem with advanced facilities for incorporating monitoring capabilities of partner resources and other data.

This paper presents an architecture for a transportation control tower. The architecture is based on an analysis of transportation scenarios from practice.

2 Usage Scenarios

This section presents three scenarios that were developed in collaboration with industry partners [9]. The scenarios represent situations that occur on a daily basis and that require advanced (control tower) functionality from transportation management systems to properly support transportation planners. The section ends with an overview of this functionality.

2.1 Multi-modal Planning

A transportation planner can plan a multi-modal route for a full container load. To do that effectively, the planner needs to have insight into real-time information about the transportation infrastructure, including both the current traffic conditions and transportation-related events, such as the waiting time at customs or the gate at the harbor.

Often, transportation companies make use of partners to do the transportation. To that end, the transportation planner needs insight into the availability of transportation partners. Using this information, the transportation planner can select the trucks that are closest to the pick-up location of the goods. These trucks can either be own assets or assets owned by transportation partners or single drivers.

After the transportation plan is created, the planner should be able to automatically distribute the transportation plan to the parties that are involved, including transportation partners. Subsequently, the execution of the transportation must be monitored, the geographical position of the resources involved must be tracked, as

well as the status of the transportation steps that have been performed and the steps that still must be performed.

2.2 Freight Shift

It happens every day that transportation capacity or demand shifts from one location to another. For example, an airplane that carries 10 containers may have to land at a different airport unexpectedly due to weather conditions. As another example, an airplane may suddenly have more transportation capacity available, because of the cancellation of a transportation order.

Such freight-shifts should be detected as quickly as possible to properly act on them. Nowadays, transportation companies often hear about it at a late stage, for example, when the airplane has already landed at another airport.

Once the freight-shift has been detected, transportation planners must change the plans around the affected locations. This involves re-planning the resources that were originally planned to pick-up or drop-off the cargo, possibly even if they are already en-route. It also involves re-planning additional capacity to pick-up or drop-off the cargo at the new location.

2.3 Inland Waterways

The use of inland waterway transportation usually is associated with transportation cost advantages and positive environmental performance. However, its reliability is influenced by varying water levels which might lead to restrictions or complete close down of the waterway for days. Since such situations can, to an extent, be foreseen, the creation of robust offline plans, which consider the risk of low or high water levels, and careful monitoring and prediction of water levels, can mitigate the need for ad-hoc online re-planning.

Based on historical data and information about the current water levels and their development, situations in which the planned transportation by inland waterway might lead to problems due to insufficient water depth can be detected. Since this information can be transmitted to the planner in ample time before the start of the transportation, the planned route can still be changed using other transportation alternatives.

The choice of other transportation alternatives is facilitated by the consideration of existing alternatives based on schedules as well as on information about the available capacities and routes shared by transportation partners or by assigning free vehicles to new routes.

2.4 Required Functionality

The scenarios above, point to a number of requirements that transportation control tower software should support. A detailed analysis of requirements can be found in [9], but on a high level of abstraction, a transportation control tower should support:

- the provisioning of information on the availability of transportation resources, including own resources, and partner resources;
- the provisioning of information on infrastructure availability;
- automated detection and prediction of disruptions of transportation infrastructure and resources;
- offline planning (before the transportation is executed);
- online planning (while the transportation is executed and taking real-time information about resource availability and disruption into account);
- robust planning (taking into account likely changes to a transportation plan);
- multi-party tracking and tracing based on a transportation plan; and
- automated reconfiguration of the tracking and tracing facilities based on changes to a transportation plan.

3 Architecture Overview

Considering the requirements that are explained in the previous section, we developed a software architecture for a transportation control tower. Fig. 1 shows the high level components that are provided by this architecture. The architecture is described in detail in [10].

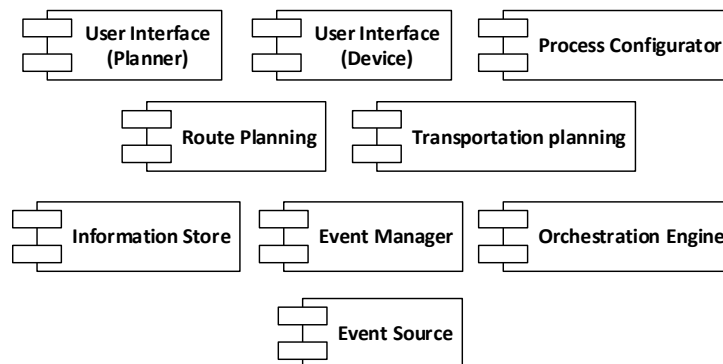


Fig. 1. Architecture for a Transportation Control Tower.

The architecture is layered and shows the following components. There are user interfaces for both the planner and for devices that drivers of transportation resources use. In addition, there is a process configurator that can be used by a designer, to describe the activities that must be executed for various types and legs of transportation routes in terms of processes. Through the user interface, the planner has access to various types of planning algorithms.

Both the planning algorithms and the user interfaces make use of information that is either accessible from an (external) information store such as a database or via an event manager. The event manager is a publish-subscribe mechanism that can be used to monitor events that are published by event sources, such as boardcomputers of trucks, road management systems, and AIS transponders of ships.

The tasks that must be executed and monitored during the execution of a transportation plan, are determined by composing them based on the transportation plan and the processes that are designed in the process configurator. These tasks are subsequently managed by the orchestration engine, which also takes care of subscribing (and unsubscribing) to events that are relevant in different stages of the transportation plan and must be shown on the various user interfaces.

4 Detailed Architecture Operations

In the usage scenarios, the architecture components work together as follows.

In order to plan the fulfilment of a transportation order, the *route planning* component is used to create a, possibly intermodal, routing for the transportation order or each consignment thereof. Such an intermodal routing is an alternating sequence of transportation legs and transshipments that connects the pick-up point with the delivery point of the consignment. Optionally, multiple transportation orders can be planned together, e.g. to benefit from optimal sharing of resources, by using the *transportation planning* component. The transportation plan must then be refined into a detailed actionable plan, which we also call a transportation process.

Fig. 2 shows a section of a transportation process. This section corresponds to two legs: first an empty container is procured by a truck and then the truck picks up the goods and drives them to a train terminal for further shipment.

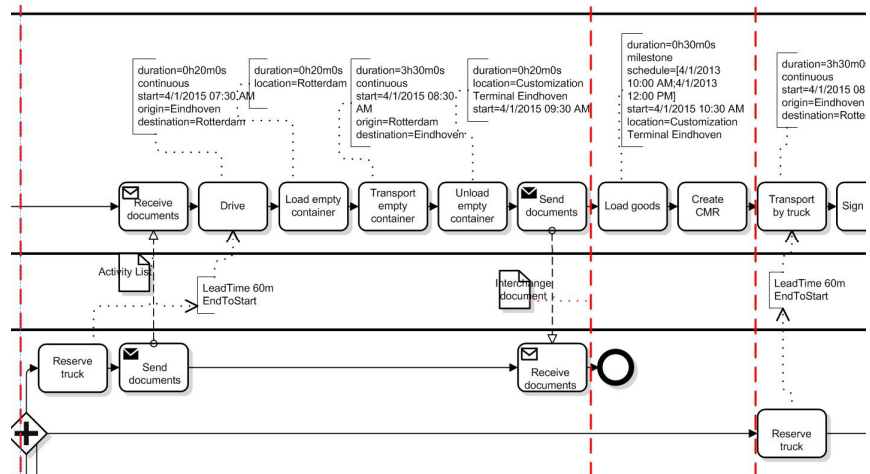


Fig. 2. Initial part of transportation process example.

To construct the transportation process from the transportation plan, we follow the idea that a process model is composed of process snippets, where we have essentially one snippet for each segment (i.e., leg or transshipment) of the transportation routing. For example, in Fig. 2, the vertical dashed red lines show the boundaries of the snippets. Functions to create models from snippets, i.e., creating a snippet from scratch, registering it to the repository using index attributes, searching for snippets with transportation routing segments and composing snippets together, are integrated into the *process configurator* component.

The *orchestration engine* provides a means to enact transportation processes. The orchestration engine has a task handler to monitor the state of transportation tasks that are derived from the transportation order and to enable different users, e.g. a planner, to perform the tasks that need to be done. To this end the orchestration engine also has access to the static *information store*. Optionally, it can activate external applications to perform certain tasks (not in the architecture).

The transportation resources that are being monitored may vary per process or per task. For example, during a truck leg, a truck is being monitored, while during a train leg, a train of a partner company is being monitored. In order to receive notifications from the *event monitoring* component about the status of the resources, the orchestration engine has to subscribe to predefined queries. The transportation process itself as well as all of its tasks can be annotated by these predefined queries as shown in Fig. 2. The orchestration then ensures that a subscription becomes active when the corresponding task or process becomes active.

To process the queries, we implemented an *event manager* responsible for collecting events from different sources and processing them. The event manager is based on ‘complex event processing’ [7, 3, 6] and builds on Esper [2, 5]. In this paper, we focus on the usage of event subscriptions during transportation execution and refer the interested reader for the technical details of the event manager to [1] and the corresponding tutorials².

The engine integrated in the event manager registers each event subscription via listeners. These listeners get informed if the subscription matches observed event types from relevant *event sources*. We assume that all required event types are registered in the event manager. The queries that activate the listeners look like SQL queries, but instead of tables, they query the pre-defined event types. For example, the query below, subscribes to events of the type ‘VehiclePositionUpdate’ for a specific truck that is given as a parameter and within a time interval that is given as a parameter.

```
SELECT * FROM VehiclePositionUpdate.win:time($timeInterval)
(operatorID= $truckID)
```

We mainly work with high-level and transportation-related events for which we provide subscription templates. To obtain such high-level events, aggregation rules are defined, which combine or select event information contained in events, possibly from different event sources. A simple example of such an aggregation rule is the

² <http://bpt.hpi.uni-potsdam.de/Public/EPP>

following rule, which generates events of type ‘TransportFinished’, when a truck with a given identifier arrives at a certain geographical position.

```
INSERT INTO TransportFinished SELECT * FROM VehicleLocation
(operatorID=$truckID,latitude=$lat,longitude=$lon)
```

The planning algorithms are made accessible to the end-users via *user interfaces*. The user interfaces also show information on the status of transportation orders, the tasks that must be performed for these transportation orders, the status of resources that are needed for executing the transportation orders, and related information such as the weather conditions, traffic congestion or low water levels that might affect container transport on barges.

5 Evaluation

We have done an initial evaluation of the architecture, using a questionnaire in which we asked practitioners about the use and usefulness of particular aspects of the transportation control tower. In particular we asked them to rank the importance of completeness, timeliness and accuracy of different types of information in their system on a five-point Likert scale. We also asked them whether their current systems provided the information in a complete, timely and accurate manner. The questionnaire was filled out by three practitioners from different logistics service providers during this initial evaluation. We are currently working to have it filled out by a much larger number of respondents. The practitioners ranked the importance of completeness, timeliness and accuracy of all types of information as high or very high. Interestingly, they also indicated that timeliness is a problem with many types of information. In a discussion with them, we found that this was because planners often need to resort to searching for information on, for example, delays of trains on a website and they need to ask for availability of partner vehicles by calling their partner organization. In a transportation control tower, that information is available in a more timely manner, because it is directly fed into the system through event aggregators. More detail on the setup of the evaluation, including the questionnaire can be found in [8].

6 Conclusion

This paper presented an architecture for software that transportation planners can use to manage and monitor their transportation orders. Using a publish-subscribe mechanism, information on transportation resources of the company itself, on the transportation resources of transportation partners, and on other transportation-related events can be made available in a flexible manner. Subscriptions are associated with tasks in the process model that is derived from the transportation planning. In that way, information subscriptions are highly configurable and will adapt to the context of the tasks that are being executed for the transportation order.

The architecture has been implemented in a prototype, of which a demo is available on-line³.

This paper focuses on introducing the overall architecture of a transportation control tower and the information aggregation component of the architecture. We acknowledge that there are other aspects that are important to ensure the collaboration of partners in such a platform, including security and technical and conceptual data translation. These aspects have, to an extent, been addressed in the work described in [1, 9, 10] of which this paper is a summary.

Acknowledgement. The research leading to these results is part of the GET Service project (<http://www.getservice-project.eu>) and has received funding from the European Commission under the 7th Framework Programme (FP7) for Research and Technological Development under grant agreement n^o2012-318275.

References

1. Baumgrass, A., et al.: Conceptual architecture specification of an information aggregation engine, Deliverable D6.2, GET Service, Service Platform for Green European Transportation (2014)
2. Bernhardt, T., Vasseur, A.: Esper: Event stream processing and correlation (2007)
3. Cabanillas, C., Baumgrass, A., Mendling, J., Rogetzer, P., Bellovoda, B.: Towards the Enhancement of Business Process Monitoring for Complex Logistics Chains. In: 11th International Conference on Business Process Management Workshop on "Process-Aware Logistics Systems", LNBIP 171. Springer (2013)
4. Demir, E., et al.: A Review of Transportation Planning Tools, Deliverable D5.1, GET Service, Service Platform for Green European Transportation (2013)
5. EsperTech: Esper - Complex Event Processing (as of July 2014)
6. Herzberg, N., Meyer, A., Weske, M.: An Event Processing Platform for Business Process Management. In: Enterprise Distributed Object Computing Conference (EDOC). pp. 107–116. IEEE (2013)
7. Luckham, D.C.: Event processing for business: organizing the real-time enterprise. John Wiley & Sons (2011)
8. Schmiele, J., et al.: Evaluation Plan, Deliverable D1.4.1, GET Service, Service Platform for Green European Transportation (2014)
9. Treitl, S., et al.: Use cases, success criteria and usage scenarios, deliverable d1.2, get service, service platform for green european transportation (2014)
10. van der Velde, M., Saraber, P., Grefen, P., Ernst, A.C.: Get architecture definition, deliverable d2.2.1, get service, service platform for green european transportation (2014)

³ <http://is.ieis.tue.nl/research/getservice/>