

# Interoperability in self-organizing systems of multiple enterprises – a case on improving turnaround time prediction at logistics hubs

Wout Hofman  
TNO, Kampweg 5 3769 DE  
Soesterberg The Netherlands

Madan Rajagopal  
TNO, Kampweg 5 3769 DE  
Soesterberg The Netherlands

## Abstract

Besides supporting bilateral transactions, enterprise interoperability can also be applied to coordinate operational processes of stakeholders. With today's technology, each of these stakeholders can be equipped with actuators like applications on smart devices. Where applications support interaction of a user with its environment, smart devices can operate as sensors and allow user interaction for collaboration in complex, self-organizing systems. Interoperability is key in sharing data in such a dynamic environment. This paper presents an infrastructure supporting trucking companies, truck drivers and a container terminal to optimize their process by sharing data. Prediction of driving times towards a container terminal and turnaround times will be calculated by sharing state information and decisions. The paper will show how interactions between stakeholders can be modeled and present findings from practically applying the solution.

## 1. Introduction

The development of complex self-organizing systems has accelerated rapidly over the past decade, especially in military systems with autonomous unmanned air and ground systems [1] and is identified for smart cities [2]. There are different definitions of these systems with elements of self- and context awareness [3] as part of situational awareness [4]. The objective of these complex systems is to improve decisions for individual participants and the predicted impact of each decision [4]. A multi-agent platform with a build in communication layer [5] can represent these types of systems with different challenges that have to be addressed, namely their engineering, interoperability, and openness. There are different approaches to engineering multi-agent systems, but these are not satisfactory [6]. Each agent in a multi-agent system typically interacts by a message based predefined protocol with other agents, in contrast to distributed object architectures with a Service Oriented Architecture (SOA, [7]) applied in for instance software engineering [8]. Interoperability in these systems requires a common semantics expressed by for instance an ontology, although existing ontologies are either not used, extended and modified, or leave ambiguities even when used with no change, and a protocol specifying the behavior of each component or agent in the system [6]. Interoperability is a prerequisite for openness, whilst implementation of common semantics and protocol is required to participate in a multi-agent system. Openness of these types of systems refers to 'the ability of introducing additional agents of different suppliers into the system, and the capability of agents to leave the system and of the system to cope with such departures' [7]. Openness can be specified as dynamic openness, during runtime reconfiguring the system, static openness, all agents in the system are informed of changes in the system, or offline openness, the systems needs to be restarted.

---

*Copyright © 2015 by the paper's authors. Copying permitted only for private and academic purposes.*

In: M. Zelm (eds.): Proceedings of the 6<sup>th</sup> Workshop on Enterprise Interoperability, Nîmes, France, 27-05-2015, published at <http://ceur-ws.org>

In an enterprise setting, typically dynamic openness with common semantics and a protocol is required. In our case, there are always different on their way to and leaving a container terminal in a port over time, with a some 300 truck movements per day. The challenge is to develop an open system that allows for minimal turnaround times of each truck at a terminal and can easily be implemented by each participating carrier and terminal operator with open standards. This paper illustrates the development of such a system. Firstly, the problem is analyzed in more detail with reference to appropriate literature on development of these types of systems, and secondly the open system that is developed is presented. Findings of the actual application of the system in a practical setting are discussed. The paper ends with conclusions and future research.

This paper considers the development of an alpha version of practice inspired research [9]. This version of a new artefact is developed and validated by practitioners of both carriers and a terminal operator.

## **2. Analyzing the problem**

The type of systems is typically a self-organizing system in freight transport of multiple autonomous operating participants, partly specified by functionality of Collaborative Intelligent Transport Systems (C-ITS, [10]). Not only the protocol of agents in such a system is relevant, also its realization with information technology. C-ITS, interoperability in multi-agent systems, and required information technology for sharing data in a complex system are discussed in this section. First of all the problem is described in more detail.

### **2.1. Truck movements at a container terminal**

In a port like the port of Rotterdam, different container terminals operated by stevedores transship containers between different transport modes like road and sea, with sea transport as main transport mode. Containers are stacked at the terminal, either for loading onto a vessel or on-carriage to the hinterland by for instance barges, trains, and trucks. Most of the containers are delivered or picked up at a terminal by trucks, a smaller percentage is transported by train and barge. Container stacks are organized in terminal blocks at which trucks can pick up or drop off containers. The stevedore assigns a terminal block to truck drivers. Each truck can perform at maximum eight activities during one call at a terminal: drop off four containers and also pickup another four.

The current process of trucks arriving at the terminal gate is random in the sense that each truck driver decides when to call a terminal based on its scheduled activities, independent of other truck drivers. Trucking companies only provide a notification to the stevedore of their initial plan. Actual truck movements often differ from their plan, as plans are revised and updated frequently based on incomplete and unreliable data. For instance, truck drivers end up in traffic jams, causing them to arrive late at their destination or truck drivers have unexpected delays in a previous trip. Long waiting times at a terminal limit the number of trips for each truck at one day with impact on the turnover of the trucking company. Because handling seagoing vessels has priority over other modes, delays of handling trucks at terminal blocks will occur. At present the communication about delays between all participants is too late or not at all, resulting in a situation where the estimated time of arrival (ETA) and the turnaround time (TAT) of trucks at a container terminal are unknown to a transport planner and a stevedore.

The objective is to improve this situation by sharing TAT's of different trucks, both from a transport planner, a truck driver, and terminal planner perspective. For this purpose, data of activities has to be shared amongst stakeholders and advanced predictions ETA's and TAT's have to be shared.

### **2.2. Collaborative Intelligent Transport Systems**

The objective of C-ITS is to improve the use of existing road infrastructure and potentially allow for pay-per-use [10] by improving situational awareness [4] of all participants in road transport. Each participant has a better perception of its environment, comprehends its situation, and is able to make a projection of its future state based on predictions. A perception of the environment can be achieved in this particular case by allowing a planner of a carrier to receive ETA's and TAT's of its trucks calling a terminal. By interpreting these ETA's and TAT's, a planner is able to improve planning of its trucks and trips of those trucks, potentially increasing the number of trips per truck per day thus increasing its turnover and profit. A stevedore could use the dates of all activities of carriers to optimize block

planning and potentially stack planning. The latter can be an issue, since terminal operation is mainly focused on handling vessels, since shipping lines are the customers of stevedores.

Our particular case is interested in trucks move on roads to and from a container terminal, whereas C-ITS covers all types of vehicles and a road infrastructure. Therefore, we have to make a distinction between relevant trucks and other vehicles from a C-ITS perspective. C-ITS has the following relevant service domains for trucks:

- **Traveler information services** providing static and dynamic information about the infrastructure network to users. These services provide traffic jam, accident and incident information, and weather conditions.
- **Freight transport services** comprising the legal requirements for vehicles to participate in traffic and commercial functions like fleet and order management.

Other types of C-ITS services for drivers are for instance vehicle operation services providing warnings and assistance to users like truck drivers and transport related payment services. There are also various services domains for governing (road) infrastructure and its utilization like emergency -, traffic management and operation -, and national security services, that can affect traffic density and thus ETAs of truck drivers. These will be considered as part of traveler information services. Weather and environmental conditions services are specific services that impact travelling speed and thus the ETA's of trucks. These services are considered as an additional service domain that cannot be governed by organizations or users. C-ITS requires a collaborative environment for interoperability amongst all stakeholders [11].

C-ITS does not address any logistics activities like the ones for picking up and dropping off containers at a terminal. The concept of 'hub' like a container terminal is also not part of C-ITS. Freight transport services only cover management of a truck fleet for a carrier and do not address other modalities like inland waterways and rail.

### 2.3. Interoperability in complex systems

A complex system like the one of trucks calling at a container terminal has five characteristics, namely operational independency, managerial independency, evolutionary development (openness), emergent behavior, and geographic distribution [12]. There are languages to model the interactions and state changes of stakeholders participating in such systems [6]. Such a language models send and receive actions of all stakeholders for either messages or recursive calls. Potential sequences of these send and receive actions can be represented by UML sequence diagrams, where each diagram depicts one particular use case in the system. Each agent in the system has its particular state diagram supporting an agreed protocol. Applying a formal language for modeling interoperability in these types of systems allows for model checking [6].

In the past, various formal languages for modeling distributed systems have been developed (see for instance [13]). Many of these languages were not often applied in practice. Furthermore, these languages and techniques were developed to model systems at construction time [14], whereas our system is open and its constellation changes over time. Extensions to include time have been made to represent behavior. By introducing a separate component in the system representing the constellation of the system at any time and protocols to assess that constellation [15], the languages could be suitable to construct a model of these systems. Thus, the issue would not be that of a design language, but extending the systems architecture.

### 2.4. Connectivity and platforms for data sharing

In general, connectivity is the ability to share data between any two IT systems [16]. An interoperability infrastructure is the set of software components of different solution providers implementing interoperability amongst participants in a complex system according an agreed protocol and providing services to participants [17]. An interface to the infrastructure is the local implementation of a service for a participant. Basically, such a interoperability infrastructure is stateless, meaning that data is stored by participants utilizing the infrastructure to share (future) state information. Such a stateless infrastructure meets data governance requirements of participants [18].

A service – and protocol specification between any two stakeholders or applications for data sharing in complex systems like logistics is outside scope of this paper. Basically, at least each participant should have a local implementation of a protocol and thus only the protocol should be specified, with a specific focus on our case. However, not all participants have IT facilities, although they will have smart devices, e.g. truck drivers will have

smart phones. A platform might be considered as software component providing services to applications of more than one participant like truck drivers. Basically, a platform has to support data sharing between heterogeneous computer systems of different organizations [19] with Application Programming Interfaces (APIs, [20]) providing a service implementation, although also other types of interfaces SPARQL endpoint [21] could be considered. Data semantics also needs to be clearly and concisely specified by for instance ontologies, with as less as possible different interpretations. A software application, for instance a Transport Management System of a carrier (TMS), a visualization platform or a truck driver app running on a smart device, are examples of IT functionality used by participants of the infrastructure. Such a platform should address the following requirements to function in an open, complex system [20]:

- **Data governance and security:** the ability to distinguish between open data, data with access restrictions for sharing with particular peers, and data that is not available outside a data source.
- **Data source management:** available sources with their APIs and quality features expressed by metadata.
- **Data preparation:** data enhancement and enrichment to improve data quality.
- **Temporary data storage:** a decoupling point between data providers and – users for storing to improve processing on behalf of a data user request.
- **Data driven action:** the ability to take action upon particular data changes based on an event driven architecture [8].
- **Data retrieval:** functionality to retrieve additional data, which might include integration of data from more than one data sources (also called: data combining [22]).
- **Analytics and support:** analysis of the shared data, including accountability.

### 3. The case: improve turnaround times for truck drivers

The participants and the issue to be addressed by the case have already been introduced in the previous section. We take an architectural approach [23] by first identifying the interactions between the various stakeholders and secondly the systems architecture in such a collaborative environment. This section finishes with the results of applying the solution in a practical setting.

#### 3.1. Interaction modeling

Conceptually, a predictive analytics algorithm needs to produce ETA's and TAT's addressing both traveler – and freight services (section 2). Smart Forecasting services of Prime Data ([www.primedata.nl](http://www.primedata.nl)), based on a predictive analytics algorithm producing ETA's, provide C-ITS traveler services (section 2.2) to truck drivers. As concluded in section 2.2., C-ITS freight information services do not cover logistics activities at hubs like terminals, so a prediction algorithm is developed to produce TAT's based on ETA's and predicted waiting times at terminal blocks of a terminal. Thus, planning data of trucks, queue status and waiting times at terminal blocks of a terminal, any changes in the ETA of a truck at a terminal due to (un)foreseen changes in traffic, etc. has to be shared. This data is dynamic by nature, since it changes over time. Any change or modification of one of the participants of the system results in an event. For example change in the ETA of a truck heading towards a terminal is an event that should be notified to all the stakeholders interested in the ETA of that truck. These changed ETAs might give a change in the arrival sequence of trucks, producing changes to the TAT of each truck. Any changes in the planning of a truck should not only be notified to truck drivers and transport planners, but also to the prediction algorithm resulting in an updated TAT. Furthermore, any changes in processes at the terminal may affect the TAT of more than one truck driver. This requires a solution to be open [12] and capable of handling and distributing events.

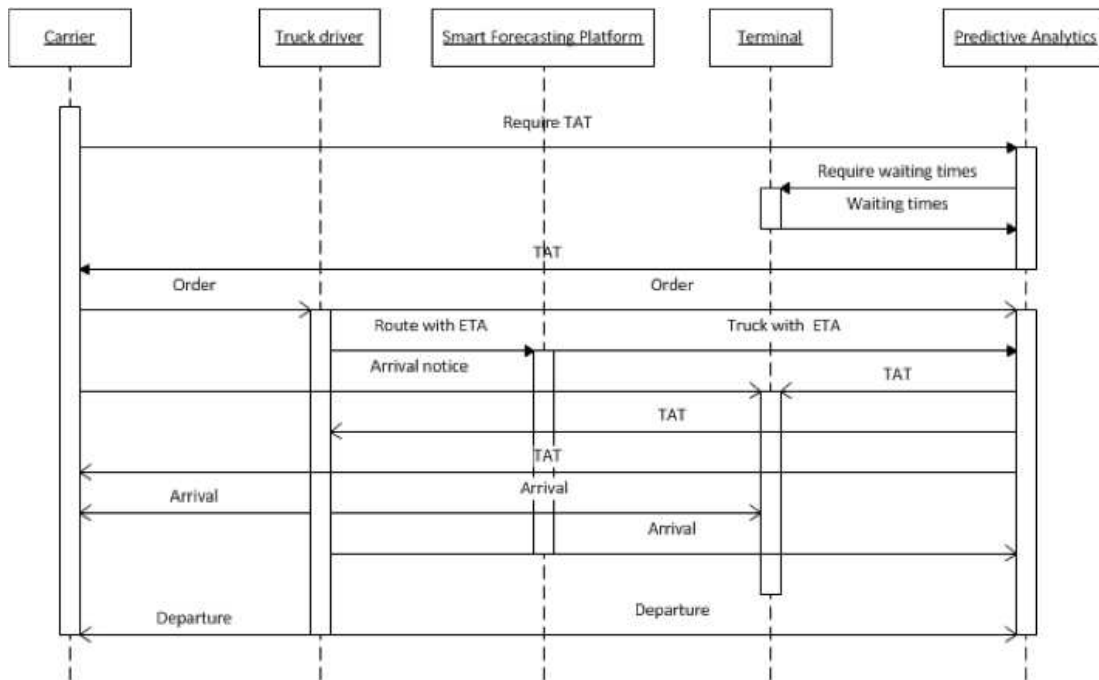


Figure 1: Sequence diagram of interactions between stakeholders and the algorithm

A UML sequence diagram visualizes the interaction between all stakeholders (figure 1). This sequence diagram only shows one truck driver and carrier, but there are of course several. It also shows a situation where ETAs and TATs do not change and completion of the process after departure of a truck driver at the terminal. Of course, the process of a truck driver continues by delivering a container to its destination. The sequence diagram consists of two parts. Firstly, a transport planner requires a TAT and ETA for assigning activities to trucks. Secondly, a truck driver receives an order and starts its trip to a terminal at a location at a time, to arrive at the terminal according to a planned ETA. During driving, several changes may occur as indicated.

Interoperability requires semantics of data shared between all stakeholders. The following concepts are specified:

- Activity: the actions performed by a truck a terminal with the relevant container numbers. Each truck can perform up to four activities: dropping of four and/or picking up four containers. An activity has several 'time' properties like ETA, estimated departure time, and actual arrival time.
- Vehicle: a truck with its license plate. Each truck can have a location, an ETA, and a TAT.
- Traveler data: the ETA of each truck produced by the Smart Forecasting services, including the location of a truck from where the ETA is produced.
- Terminal information: the terminal blocks, travel times between terminal blocks, number of trucks waiting at a block at a time, and waiting times at each terminal block.
- User: a user with a particular role participating in the system, each truck driver, terminal operator and transport planner of a carrier.

With each interaction, instances of these concepts are shared. These interactions are implemented by different technical solutions (see next section).

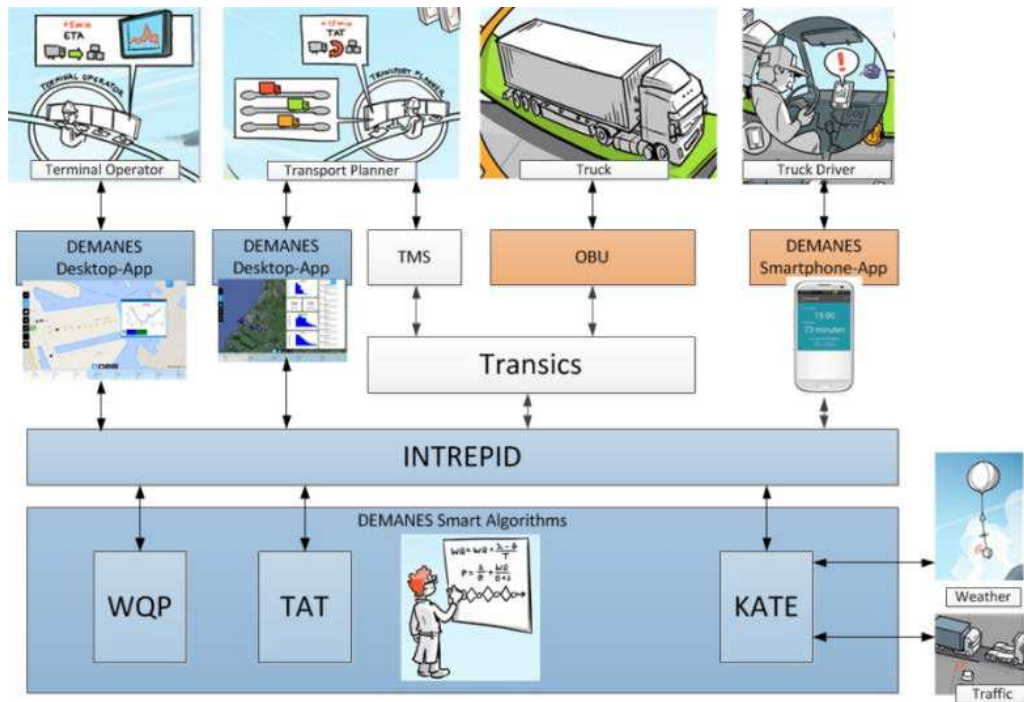


Figure 2: Application architecture and interfaces between components

### 3.2. Systems architecture

Whereas the previous section illustrates the interactions between individual components, this section maps these interactions to a systems architecture (figure 2). We have distinguished a carrier, truck drivers and a terminal operator. In our current situation, carriers have a Transport Management System (TMS) for planning the orders, On Board Units (OBUs) in trucks for providing activity data to truck drivers and support their logging of (completion of) activities. The company providing these OBUs, Transics operates a platform storing activity and location data of trucks for various carriers. They have a web service interface for data retrieval: activity and planning data of individual trucks retrieved from a TMS, real time location coordinates of trucks, and ETAs of those trucks at a destination.

These software components are not sufficient to support all interactions and to improve the TAT of trucks at terminals. Furthermore, the terminal did not have sufficient data for predicting waiting queues of terminal blocks (see section 2). As figure 3 shows, the following additional components have been developed, where the KATE platform provides the Smart Forecasting services:

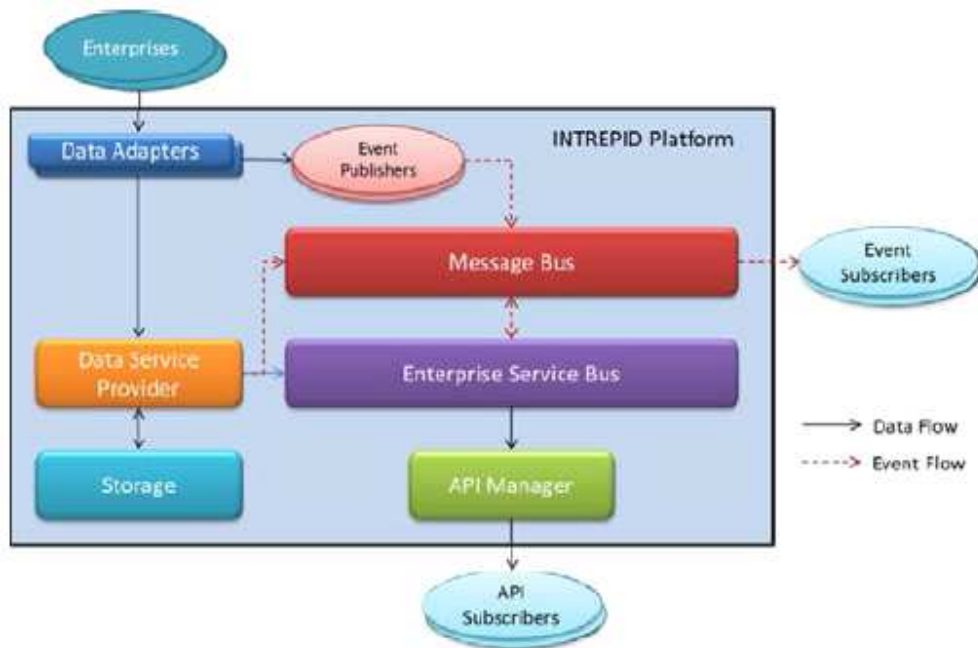


Figure 3: Platform architecture of the case

- **Intrepid.** An existing data integration - and communication platform configured for sharing and temporary storage of data shared by events and web services between various components. The platform consists of a Enterprise Service Bus, API Manager, Message Bus, Business Activity Monitor, and Storage space (figure 3). WSO2 API Management is used for realizing Intrepid. Intrepid polls Transics regularly to retrieve any location changes by a web service, calls KATE to calculate the ETA by an API, and calls the TAT and WQP algorithm via an API in case ETAs are changed. The terminal publishes waiting times at terminal blocks as HTML pages that have to be parsed.
- The platform has a topic table for receiving and producing **events**. The table has topics like 'create/update/delete of a new activity' produced by Transics or a smart phone application of a truck driver. This topic is the key to all others like an update of a vehicle location, ETA, and TAT. The topic table can be considered as the state information of the system specifying the objects that are of interest with their changes.
- **Smartphone Application.** An app that can be used by freelance truck drivers or truck drivers of carriers without back office or OBU functionality, and truck drivers of carriers that have this functionality. A truck driver can enter planning of activities, log the execution of that plan a existing plan, and interact via Intrepid with the TAT application. An app interfaces via an API with Intrepid.
- **TAT and WQP algorithms.** Based planning and location data of either the Transics platform or the app, terminal block status of a terminal, an improved prediction of the ETA retrieved from KATE, the estimated TAT for trucks and waiting queues for terminal blocks are calculated using a Markov Chain Process. The TAT algorithms uses queuing data of the blocks and traffic density and travel times of trucks between the various blocks. The WQP algorithm improves the waiting times of each terminal block based on the ETAs, activity data, and container stacking data.
- **Dashboards.** Additionally, a dashboard is developed for a transport planner and a terminal operator for TATs of trucks and waiting queues of terminal blocks respectively. These dashboards interface with an API with Intrepid.

### 3.2. Findings

Validation of the solution with practitioners and actual data resulted in a number of findings. An average of 25 to 30 trips of three carriers utilizing Transics were observed to a total of 200 movements. The observed sample is less than 10 percent of the actual population. This relative low population influences the quality of the predictions made by the TAT and WQP algorithms.

Besides the fact that the system needs to be extended with more carriers, two issues influence the quality of a predicted TAT, namely data quality and truck driver behavior. Not all participating carriers were able to provide all data and data of these carriers was inconsistent, meaning that each carrier applied the interface between their TMS and the Transics platform differently. Inconsistent use of interfaces was also due to different use of TMSs and different planning procedures. Therefore, only 50 of the 200 truck movements of these three carriers were useful. The data of different carriers needs cleaning before utilizing the solution. Secondly, truck drivers ignored planning, leading to different ETAs than originally planned based on predicted TATs used by a planner. These types of deviations lead to different waiting queue predictions, impacting TATs of other truck drivers. Truck drivers also did not log their activities consistently, which made the validation of TATs difficult.

### 4. Conclusions and further work

We have developed and validated an alpha version of an artefact [9] for collaboration amongst a large number of stakeholders in an open environment [12] with 200 truck movements of three carriers to a terminal, 50 of which were useful.

We have been able to model and construct the solution, but practical application of the solution requires, besides more participants, a cleansing algorithm for data preparation (section 2.4) and incentives to change behavior of truck drivers. Examples of cleansing algorithms are already available for Linked Open Data [24]. Another approach to increase data quality could be to provide feedback to carriers, thus enabling them to improve data quality. The latter also includes a change of utilizing IT systems by planners and thus potentially a change in behavior of those planners. Changing behavior of truck drivers requires incentives, e.g. the ability of each driver to contribute to profit increase by increasing the number of daily trips or contribute to cost reduction. Typically, interoperability is not only required at operational and construction level, but also on social and cultural level [14].

From a technical perspective, structuring all data in for instance XML formats with a shared semantics, e.g. replacing the HTML data provided by the terminal, will improve stability of the system. Scalability will improve by adding a directory service of participants [15]. The Intrepid platform already stores 'users' like transport planners and truck, but currently only an interface to one OBU provider is supported. Adding other OBU providers and other constellations of back office interfaces of carriers requires more flexibility [15]. From a C-ITS perspective [10], the freight service domain does not cover the activities of truck drivers at hubs like terminals. There are other frameworks of standards that business transactions in the logistics domain [25], but do not cover events and activities required in a complex system.

System validation is performed by processing actual data and with practitioners. Another approach would be to simulate the like an open system, which can be a next step.

### References

- [1] S. H. Young, T. A. Mazzuchi and S. Sarkani, "A Model-based Framework for Predicting Performance in Self-adaptive =Systems," in Conference on Systems Engineering Research, 2014.
- [2] N. Komninos, M. Pallot and H. Schaffers, "Special Issue on Smart Cities and the Future Internet in Europe," Journal of the Knowledge Economy, volume 4, issue 2, pp. 119-134, 2013.
- [3] F. D. Macias-Escriva, R. Haber, R. d. Toro and V. Hernandez, "Self-adaptive systems: A survey of current approaches, research challenges and applications," Expert Systems with Applications, vol. 40, no. 18, pp. 7267-7279, December 2013.



- [4] M. R. Endsley, "Toward a theory of situation awareness in dynamic systems," *Human Factors*, 37(1), pp. 32-64, 1995.
- [5] D. Mitrovic, M. Ivanovic, Z. Budimac and M. Vidakovic, "Radigost: Interoperable web-based multi-agent platform," *Journal of Systems and Software*, vol. 90, pp. 167-178, April 2014.
- [6] C. D. Walton, "Verifiable agent dialogues," *Journal of Applied Logic*, vol. 5, pp. 197-213, 2007
- [7] O. Shehory and A. Sturm, "Multi-agent systems: a Software Architecture Viewpoint," in *Agent-Oriented Software Engineering, Reflections on Architectures, Methodologies, Languages, and Frameworks*, Berlin, Springer Verlag, 2014.
- [8] T. Erl, *Service Oriented Architecture - concepts, technology and design*, Prentice-Hall, 2005.
- [9] M. K. Sein, O. Henfridsson, S. Purao, M. Rossi and R. Lindgren, "Action Design Research," *MIS Quarterly*, vol. 35, no. 1, pp. 37-56, 2011.
- [10] B. Williams, *Intelligent Transport System standards*, Artech House Inc., 2008.
- [11] A. L. Osorio, H. Afsarmanesh and L. M. Camarinha-Matos, "Towards a Reference Architecture for a Collaborative Intelligent Transport System Infrastructure," in *Collaborative Networks for a Sustainable World*, St. Etienne, IFIP WG 5.5, 2010, pp. 469-477.
- [12] M. W. Maier, "Architecting Principles of Systems-of-Systems," *Systems Engineering*, vol. 1, no. 4, pp. 267-284, 1998.
- [13] K. J. Turner, *Using formal description techniques: an introduction to Estelle, Lotos, and SDL*, New York: John Wiley & Sons, 1993.
- [14] S. S. Ostadzadeh and F. Shams, "Towards a software architecture maturity model for improving Ultra-Large scale systems interoperability," *The International Journal of Soft Computing and Software Engineering*, vol. 3, no. 3, pp. 69-74, March 2013.
- [15] M. Lorenz, J. Muller, M.-P. Schapranow, A. Zeier and H. Plattner, "Discovery services in the EPC network," *Designing and Deploying RFID Applications*, Intech, pp. 109-130, 2011.
- [16] European Commission, "European Interoperability Framework for Public Services (version 2.0)," Brussels, 2009.
- [17] A. S. Tanenbaum, *Computer Networks (Third Edition)*, Prentice Hall, 1996.
- [18] S. Eckartz, W. Hofman and A. F. v. Veenstra, "A decision model for data sharing," in *eGov2014*, Dublin, 2014.
- [19] C.-M. Chituc, A. Azevedo and C. Toscano, "A framework proposal for seamless interoperability in a collaborative networked environment," *Computers in industry*, vol. 60, pp. 317-338, 2009.
- [20] W. Hofman and M. Rajagopal, "Constructing a Data Ecosystem - an overview of available software solutions," *Journal of Theoretical and Applied Electronic Commerce Research*, p. provisionally accepted, 2014.
- [21] J. Davies, R. Studer and P. Warren, *Semantic web technologies - trends and research in ontology-based systems*, Wiley, 2006.
- [22] A. Zuiderwijk, N. Helbig, J. Gil-Garcia and M. Janssen, "Guest Editors' Introduction. Innovation through open data: a review of the state-of-the-art and an emerging research agenda," *Journal of Theoretical and Applied Electronic Commerce Research*, vol. 9, no. 2, 2014.
- [23] M. e. a. Lankhorst, *Enterprise Architecture at work - Modelling, communication, and analysis*, Springer-Verlag, 2005.
- [24] W. Beek, L. Rietveld, H. R. Bazoobandi, J. Wielemaker and S. Scholbach, "LOD Laundromat: a uniform way of publishing other people's dirty data," in *The Semantic Web - ISWC 2014*, 2014.

[25] J. T. Pedersen, "One Common Framework for Information and Communication Systems in Transport and Logistics: Facilitating Interoperability," in Sustainable Transport,