A synergy based method for designing ITS services

Shoaib Bakhtyar[1*], Johan Holmgren[1], and Jan A. Persson[1,2]

{shoaib.bakhtyar, johan.holmgren, jan.persson}@bth.se

[1]School of Computing, Blekinge Institute of Technology, SE-37424 Karlshamn, Sweden.
[2]School of Technology, Malmö University, SE-20506 Malmö, Sweden.

Abstract

In this paper, we propose a method for supporting the process of designing Intelligent Transportation System (ITS) services, which utilizes on primarily functional synergies with already existing services. Using synergies between services will enable sharing of resources, such as, information entities, functions and technical resources, which in turn may lead to reduced costs for implementing services. The method is built around an existing service description framework, which is used to describe both existing services and the service to be designed. In order to illustrate the usage of the suggested method, we have applied it for designing a new ITS service, i.e., the Liability Intelligent Transport System (LITS) service. The purpose of the LITS service is to support the process of identifying when, where and why freight has been damaged, and which actor was responsible when the freight was damaged. The LITS service may lead to better quality control of consignments and may also facilitate the identification of which actor was responsible for the freight damage, which is of particular interest in multi-modal transport. By applying our service design method we were able to identify that three out of four functions of the LITS service already exist in other existing ITS services. Therefore, the LITS service can be designed based on synergies with these services.

Keywords: freight damage, e-Waybill, ITS services, services synergy, service design method

1. Introduction

It has been suggested that the use of different types of Intelligent Transportation System (ITS) services might lead to positive effects on the execution of transport [1]. For example, ITS services might contribute to reduced congestion, decreased fuel consumption, and reduced amount of emissions, by enabling more efficient use of vehicles and infrastructure, as well as increased safety. However, it has been argued, that service usage is limited due to that service providers in general provide services only for their own terminals [2]. Therefore, high service utilization during transport will require that the vehicle is equipped with several terminals, unless services are developed for open platforms that are shared by different service providers. In addition, it has been suggested that the utilization of synergies between ITS services can lead to reduced implementation costs, which in turn might lead to a higher service utilization [3]. A possible way to benefit from synergies is to bundle multiple ITS services and distribute them as packages that are distributed for a shared platform. This would enable the same function to be implemented only once even though it is used by more than one service in the package.

We illustrate the concept of synergies between services by giving examples of functional, information, and technical synergies between the ITS services *Onboard Driver Monitoring* and *Onboard Safety and Security Monitoring*. The *Onboard Driver Monitoring* service monitors and reports driver conditions in real time, e.g., driver's actions and health. The *Onboard Safety and Security Monitoring* service monitors the vehicle and the freight, e.g., in order to detect theft. A shared function of these services is determining the speed of the vehicle. Information about the vehicle, i.e., type, id, and size, is needed as input by both

*Corresponding Author. Email: shoaib.bakhtyar@bth.se

services. The services share the technical resource *sensor to sense human presence*, which is needed in order to identify presence of human(s) in case of an accident.

There exist several studies [4], [5], [6] in which synergies between ITS services are considered. Van der Perre in [4] presented the idea of a platform, which is capable of accommodating different services and which can support relevant communication media (cellular broadcasts and dedicated short range communication). Talib et al. in [6] argued for the need for a service-packaging platform, which is able to offer service providers the ability to create reusable service packages. This is possible through organizing existing service delivery functions or creating new ones. The authors introduced the notion of service packages and a pattern-based approach to service packaging. In addition, they highlighted the requirements of a platform to enable service packaging. Although there are numerous possibilities for utilizing on the synergies between different ITS services, there exists, at least according to the best of our knowledge, no study that explicitly considers how functional synergies can be used in order to designing new ITS services.

In this paper, we propose an ITS service design method for supporting the process of designing new ITS services by utilizing on synergies with existing services. By existing services, we here mean the services that were identified during different projects in [7][3]. The proposed method is built around the service description framework suggested in [8], which is used to generate uniform descriptions of both existing ITS services and the service to be designed. We illustrate the usage of the method by showing how it can be used to generate a design for a new ITS service, i.e., the *Liability Intelligent Transport System* (LITS) service. The purpose of the LITS service is to support the process of identifying why, when and where freight have been damaged, and in case of multi-actor transport, it should also support the identification of which actor should be held responsible for the damage. Since it is sometimes difficult to identify when freight damage has occurred, it might in multi-actor transport be difficult to identify who should be held responsible for the damage. Therefore, we believe that the LITS service might have an important role in future freight transport. By using a service to detect possible freight damage, based on the conditions for transport, it would be possible to identify the cause for damages that are not visible to a naked eye, e.g., a temperature deviation that has occurred for a short time.

In multi-modal transport, which typically involves multiple actors, there are two liability schemes that regulate the amount of penalty that should be paid by the actor(s) that is held responsible for freight damage, i.e., the *Network Liability Scheme* and the *Uniform Liability Scheme* [9], [10]. The major difference between these two schemes is how the amount of penalty is determined. In the Network Liability Scheme, the penalty is based on the transport mode (where the freight was damaged) while in Uniform Liability the penalty is identical for all modes of transport [10], [11]. Regardless of which scheme is adopted, there is a need to identify which actor was responsible for freight damage. We claim that the LITS service can contribute to addressing the problem of identifying the responsible actor for freight damages, which can make both of the liability schemes transparent.

The current paper is organized as follows. In Section 2, we present the service description framework that is used by our method to describe services. Then, in Section 3 we present our ITS service design method. In Section 4 we illustrate the method by showing how it can be used to design the LITS service. In Section 5, we finalize the paper by providing some concluding remarks and pointers to future work.

2. Service description framework

In this section, we briefly describe the service description framework [12], which suggests how to generate structured and uniform descriptions of ITS services. In our service design methodology, the framework is used to describe both existing services and the service to be designed. The framework defines how to decompose a service into different types of components (i.e., input entities, output entities, functions, triggers, and the locations where input and output entities are needed). These components describe what the service should do, what input it needs, and what output it produces.

A description that has been generated using the framework is based on various levels of detail and it is implementation (technology) independent. It uses two levels of specification, i.e., an abstract level (Level A) and a concrete level (level B), where level B is a further specification of level A. Level A represents what tasks should be performed by the service, by specifying input and output entities, and in particular, when and where these entities are needed. Level B builds on the level A description, by describing how the service should fulfil the tasks specified in level A. This is achieved by specifying functions and triggers.

3. ITS service design method

In this section, we present our method for designing new ITS services by explicitly utilizing on, mainly functional, synergies with existing ITS services. As mentioned above, the service description framework by Jevinger et al. [12]**,** plays an integral role in the method, since it is used to generate uniform descriptions of both the service to be designed and the existing services. A pre-step, i.e., Step 0, to the service design method identifies relevant ITS services (which might enable synergies with the service to be design). This pre-step is an input to the service design method. In Figure 1, we specify our service design method.

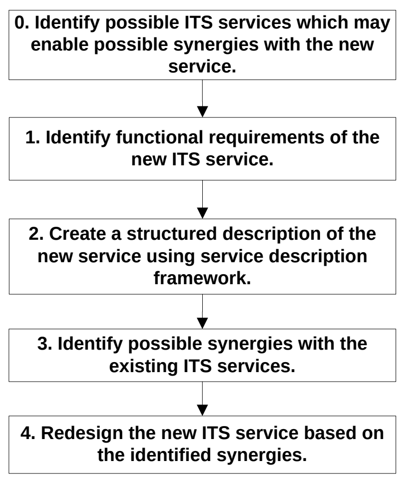

Figure 1. Service design method

Step 1 of the method concerns identifying the functional requirements of the “new” ITS service. This can be done in different ways, however, we have chosen not to specify any particular method for identifying functional requirements, because we want the service design method to be flexible enough to allow for different types of approaches. Step 2 concerns identifying the components of the new service, by applying the service description framework. The components specified when applying the service description framework represent an initial design of the new service, where all the processing is performed by the service itself without making use of synergies with existing services. The purpose of step 3 is to identify all possible synergies between the new service and existing services (identified in step 0), by searching for functions and information entities in the existing services that can replace functions in the new service. In Step 4, the initial description (or design) of the new service is updated by realizing synergies with the existing services. In particular, synergies are achieved when one or more of the functions of the new service are replaced by new components, as a consequence of "borrowing" functions from existing services. For example, a function in the new service can be replaced by an input information entity, generated as output by an existing service, and a trigger that reacts to the input.

4. Service design method applied on LITS service

In this section, we illustrate our ITS service design method by applying it in order to create a design of the so-called LITS service. The main purpose of this paper is to present an ITS service design method (based on synergies with existing services). Therefore, in rest of the paper, we limit scope of the LITS service. As discussed in Section 1, the LITS service should be able to detect various types of freight damage, which might be caused, e.g., by accidents, and temperature deviations. However, in the illustration of our method we limit the scope of the LITS service, to only consider freight damages related to temperature deviations.

The idea of monitoring the freight damage, which is an integral part of the LITS service, is not new. The literature contains several studies on freight damage detection during transport. Jedermann et al. [13] proposed a service for monitoring agricultural products, Abad et al. [14] introduced a service for monitoring fish products, Mahlknecht and Madani [15] proposed a service to monitor the inside environment of a container for tracking changes, and Carullo et al. [16] proposed a service for monitoring temperature-sensitive products. The LITS service, which is considered here, is different from the services discussed above, since they are not able to provide all functions of the LITS service.

Step 1. Identify functional requirements of the LITS service

We have used a scenario-based approach in order to identify the functional requirements of the LITS service, where we formulated and analysed two scenarios that in different ways involve temperature related freight damages.

We have used a scenario-based approach, since scenarios are considered to be an effective way of identifying requirements of a system [17]. In the literature, scenarios have been used for different purposes. Scenarios can be used to illustrate behaviour of a system in a certain environment. In the system design process, scenarios are used for design exploration, requirements elicitation and validation [18]. Scenarios can also be used to illustrate real world examples [18]. In this paper we have used scenarios to illustrate real world examples of the incidents concerning freight damage during transport.

In our scenarios, we assumed that a consigner is sending freight to a consignee, and the transport involves multiple carriers [c_1, c_2, c_3, ..., c_n]. The carrier may represent different operators. The freight is assumed to be in perfect condition at the origin and are loaded into the vehicle by a carrier "c_1". The scenarios S1-S2 represent some incidents that may occur during a traditional freight transport process.

S1: *While the vehicle is moving towards the destination, one package gets damaged due to rise in temperature in the storage environment of the freight. This deviation occurs for a short time causing freight damage, which is invisible to the naked eye. At a certain stop, c_1 transfers the freight to c_2 and since the damage is not visible, c_2 accepts the freight for further transportation. This will result in a damaged freight at the end of transport.*

S2: *The precondition is that the freight must be placed in a temperature-controlled environment during transport. The carrier c_1 reads the instructions from a waybill about the required storage temperature, which should be -20 degree centigrade, but c_1*

mistakenly sets the temperature to be 20 degrees. After travelling for some time, c_1 realizes the mistake and corrects the temperature without checking whether the freight has been damaged. Later on, c_1 transfers the freight to c_2, who manually checks the freight temperature. If the temperature is suitable, c_2 accepts the freight and proceeds with the transport until it eventually is handed over to the next carrier c_3. When c_3 arrives at the final destination, it is found out that the some of the freight has been damaged due to temperature deviations. It is unclear when and where the temperature deviation occurred and who was responsible for the damage.

From the incidents presented in the scenarios above, we derived the following functional requirements of the LITS service:

R1: The service should be able to detect possible freight damages caused by temperature deviations for a longer time than allowed.
R2: The service should be able to identify the timestamp for a possible freight damage.
R3: The service should be able to identify the current freight location at the time of a possible freight damage.
R4: The service should be able to identify the actor responsible for the transport at the time of possible freight damage.
R5: The service should be able to store the information specified in R1, R2, R3, and R4.
R6: The service should be able to create a report about the information specified in R1, R2, R3, and R4.

From the requirements above, we conclude that the service should have four categories of functions, i.e., *monitoring, tracking, tracing,* and *reporting* freight damages. Monitoring concerns detecting a freight damage. The service should be able to track when and where a damage occurred. It should be able to trace information related to the consignment and the actor involved at the time of damage. Finally the service should be able to store and report the output from the monitor, track, and trace functions to a storage location and user respectively.

Step 2. Applying service description framework on the LITS service

We applied the service description framework (see Section 2) on the LITS service in order to describe how to achieve the requirements R1, R2, R3, R4, R5, and R6. This will specify the relevant information entities (see Table 1), and the abstract level (see Table 2) and concrete level (see Table 3) descriptions of the LITS service.

Table 1. Information Entities (IEs) used in the abstract and concrete level service description of the LITS service.

Information Entity (IE)	Description
IE1	Unique id of the goods/packages
IE2	Current temperature of the environment in which goods are being stored
IE3	Ideal temperature for storing the goods
IE4	Maximum time goods can resist temperature deviations
IE5	Current location of the goods
IE6	Time when an actor was responsible for the transport
IE7	Actor id at the time of damage
IE8	Status of the goods, i.e., damaged or normal

IE9	Damage level
IE10	High/low temperature change
IE11	Timestamp of the possible damage occurrence

At the abstract level, the relevant input entities to the service are IE1 and IE6. The service can operate in two ways. It can either acquire all the IEs by itself, which we term *EPull*, or the IEs can be provided by the user, which we term *EPush*. Output IEs generated by the service are IE1, IE5, IE7, IE9, and IE11, which can be either reported by the service, i.e., EPush, or acquired by the user, i.e., EPull. The output IEs can be generated in the vehicle or at the back-office.

Table 2. Abstract level service description of the LITS service

Inputs information entities (IEs)	Outputs information entities (IEs)
What ∈ {IE1, IE6 }	What ∈ {IE1, IE5, IE7, IE9, IE11}
When ∈ {EPush, EPull}	When ∈ { EPush, EPull }
Where ∈ {back-office, vehicle}	Where ∈ {in vehicle, at back-office}

The concrete level specifies the functions (i.e. the processes) that the LITS service should execute in order to achieve the abstract level output. In Table 3, it can be seen that the output IEs generated by each of the functions of the LITS service can be used by other functions of the service. The output entities generated by the B_1 function can be all used by B_2 and B_3. All the input entities required by B_2 are generated by B_1, and B_3 gets some input IEs from the output IEs generated by B_1 (IE1, IE8, and IE11) and some IEs from B_2 (IE5). Some of the input entities required by B_4 (IE1 and IE11) can be acquired from the output entities generated by B_1, B_2, or B_3.

Table 3. Concrete level description of the LITS service

Functions	Input entities	Output entities	Description
B_1	IE1, IE2, IE3, IE4	IE1, IE8, IE11	This function identifies and reports the goods status by monitoring the goods environment for any unwanted change in temperature.
B_2	IE1. IE8, IE11	IE1, IE5, IE11	This function is used to identify the location of the goods when the damage occurred.
B_3	IE1, IE4, IE5, IE8, IE10, IE11	IE1, IE9, IE11	This function determines the damage level (e.g. to what extent the goods may be possible damaged with temperature increase or decrease) and the timestamp for the damage.
B_4	IE1, IE6, IE11	IE1, IE7, IE11	This function identifies and reports the actor responsible for the goods at the time of possible freight damage.

In this initial design of the LITS service, it executes by first detecting the goods status using function B_1. If a possible damage is detected, then the timestamp of the damage occurrence and the current location of the freight are identified using B_2. The function B_3, identifies the damage level. Finally, the actor responsible for the freight at the time of freight damage occurrence is identified using B_4. The abstract level then generates the final output to a user to a storage location, such as, a database.

Step 3. Identifying synergies with existing ITS services

We conducted a search for functions (of the LITS service), and input to these functions, that can be (partially or completely) fulfilled through synergies with existing ITS services. Since some of the functions of the LITS service can have synergies with more than one service, there should be some sort of criteria for selecting the relevant services to consider. Services that are used to provide synergies with the new service can be selected based on, e.g., their costs or power consumption. In this paper, our criterion for selecting service(s) is based on the core function of the service. A core function is the main task performed by the service, i.e., the main purpose of a service. For example, the core function of a service *Estimated time of arrival* (ETT) is to predict the estimated arrival time of a vehicle. In addition to the core function, there may be other functions of the ETT service, such as, data storage. Hence, we have identified a number of services that can have synergies with the LITS service. However, we have selected only those services whose main function include any of the functions in the LITS service.

Function B_1, of the LITS service, already exists in the *Notify goods physical status* (GS) service. Therefore, the LITS service can use the B_1 function from the GS service. The GS service needs freight related information as input, such as, id, type, amount, and temperature. It generates output about the status of the freight based on temperature deviations. We found that the GS service can get some freight-related information from the e-Waybill service, i.e., goods id (IE1) and the required temperature for storing the freight (IE3) and the maximum time the freight can resist temperature deviations (IE4).

Function B_2 determines the current location of the damaged freight. This function can be provided by the *Real-time track and trace of goods* (RTT) service, which needs freight related input, such as, global positioning and freight identifier. The service generates real-time freight location. Instead of implementing a separate B_2 function for the LITS service, it is possible use the function from the RTT service. The B_2 function will execute only if the status of the goods is reported to be damaged by the B_1 function. The output entities generated by B_1 are input to the B_2 function.

Function B_3 can use the output generated by B_1 and B_2 functions. The function will execute only if the status of the goods is reported to be damaged by the B_1 function. Therefore, the output entities by the B_1 function are input to this function. The function also gets input based on the output information about the goods location from the B_2 function. The B_3 can also make use of information entities from the e-Waybill. Since the e-Waybill has information about the conditions for storing the freight, it can provide the input information entity IE4 to the B_3 function.

Function B_4 has synergies with the e-Waybill service, since the e-Waybill has information about the freight as well as when a particular actor was responsible for handling the freight, i.e., IE1 and IE6. The information entities IE1 and IE11 can be sent from the LITS service to the e-Waybill and the e-Waybill can return the information entity IE7to the LITS service.

Step 4. Redesigning the LITS service

Based on the possible synergies identified in Step 3, we redesigned the LITS service as shown in Figure 2.

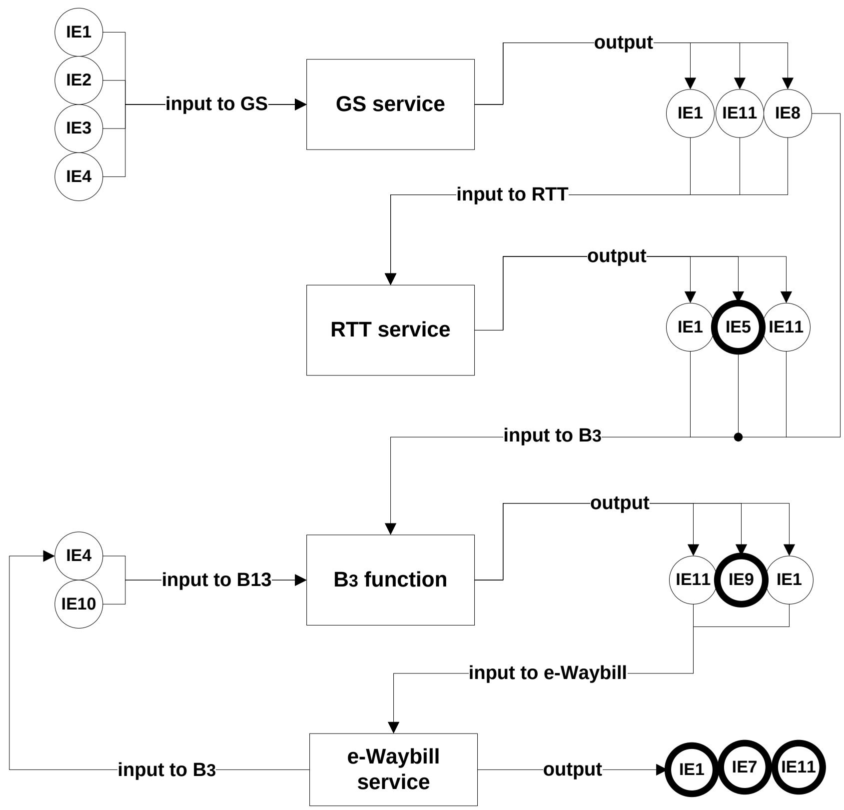

Figure 2. Concrete level design of the LITS service

Figure 2 represents how the LITS service executes based on synergies with already existing ITS services. The first two functions of the *LITS service*, i.e., B_1 and B_2, can be provided by the GS and RTT services, since these are the core functions of these services respective. The third function B_3, i.e., identifying the fright damage level, gets input entities from the GS and RTT services, as well as, from the e-Waybill service. Finally, the B_4 function can be replaced with the e-Waybill service. The e-Waybill service gets input information entities IE1 and IE11 from the B_3 function of the LITS service and it generates output information entity IE7 that can be used by the LITS service as an output. The highlighted entities belong to the abstract level of the LITS service as the final output from the service. The final output generated by the service may be reported to the e-Waybill (to store) for later use or the output may be reported to the user of the service.

5. Concluding remarks

The main contribution in this paper is an ITS service design method, which can be used to design ITS services by explicitly utilizing on synergies with existing services. Since the service design method makes use of synergies with other services, a service designed using our service method could possibly lead to a reduced cost for implementing that service. Although the purpose of the proposed method is to design new ITS services, we believe, it can also be used to redesign existing services.

By applying the service design method in order to create a design of the LITS service, we verified that there exist components in other existing ITS services, such as, information entities and functions, which can be used by the LITS service. A general conclusion is that the LITS service can be designed using synergies with other ITS services, i.e., the e-Waybill, Real time track and trace (RTT), and Notify goods physical status (GS). Two of the functions of the LITS service, i.e., the function to identify damage and the function to determine the current location of the goods can be provided by the GS service and the RTT service respectively. The third function, i.e., identifying who is responsible for the freight at the time of a possible damage, exists in the e-Waybill service and it can be provided by the e-Waybill service. Finally, the output information about the freight status can be stored by an e-Waybill service, therefore the e-Waybill can provide storage functionality to the LITS service for storing freight status information. Therefore the LITS service has synergies with the GS, RTT, and the e-Waybill service. The LITS service may have positive impacts in a transport process by detecting freight damage and the actor responsible for the damage. The LITS service may lead to better quality control of the freight being transported by monitoring and reporting the freight status in real-time and will also contribute in making the Network and Uniform liability schemes more transparent in case of a multi-modal transport.

In this paper, we have limited the scope of the LITS service by only considering freight damages related to temperature deviations. However, we believe this limitation will not affect our work, since it will be possible at a later stage to redesign and add more functions to the LITS service using the proposed service design method. A possible direction for the future development of the proposed method is develop criteria for selecting which services should be use to provide synergies with the service to be design (in Step 4).

References

[1] G. Mbiydzenyuy, J. A. Persson, P. Davidsson, and P. O. Clemedtson, “Method for quantitative valuation of road freight transport telematic services”, IET Intelligent Transport Systems, vol. 6, no. 4, pp. 388–396, 2012.

[2] C.-S. Kim, J.-I. Kim, D.-S. Yoon, M.-J. Lee, H. Kim, and K.-S. Kwon, “Service Bundle Providing System in Open Telematics Environment,” in the 9th International Conference on Advanced Communication Technology, 2007, vol. 3, pp. 2188–2191.

[3] G. Mbiydzenyuy, J. A. Persson, and P. Davidsson, “Analysis of added-value services in telematic systems and service in road-based vehicle freight transport,” in 14th World Congress on Intelligent Transport Systems, 2008.

[4] P. Van der Perre, “Common protocols and APIs for the remote installation, operation, upgrading and removal of automotive services,” in First International Conference on Communications and Networking in China, 2006, pp. 1–5.

[5] S. Chandrasekaran, G. Silver, J. A. Miller, J. Cardoso, and A. P. Sheth, “XML-based modeling and simulation: web service technologies and their synergy with simulation,” in Proceedings of the 34th conference on Winter simulation: exploring new frontiers, 2002, pp. 606–615.

[6] M. A. Talib, A. Colman, J. Han, J. King, and M. Kapuruge, “A Service Packaging Platform for Delivering Services,” in IEEE International Conference on Services Computing (SCC), 2010, pp. 202–209.

[7] Å. Jevinger, J. A. Persson, and P. Davidsson, “Analysis of transport services based on intelligent goods,” *in the 22nd Nordic Logistics Research Network (NOFOMA)*, 2010.

[8] Å. Jevinger, “Intelligent Goods - Characteristics and Architectures,” Licentiate Thesis, Blekinge Institute of Technology (BTH), Karlskrona, Sweden, 2012.

[9] R. Daujotas, “Justification of Liability Limitation in International Carriage of Goods”, Law & Society: International & Comparative Law eJournal, vol. 6, no. 71, 2011.

[10] E. Eftestöl-Wilhelmsson, “EU Intermodal Transport and Carrier Liability - Content and Context”, Helsinki Legal Studies Research Paper No. 15, pp. 133–166, 2007.

[11] M. Zanardi, “Multimodal Transport of Goods. Some examples to enlighten the differences in the applicable liability regime,” in IUMI Conference Berlin Marine Insurance - Reinventing our Business, 1999.

[12] Å. Jevinger, P. Davidsson, J. A. Persson, G. Mbiydzenyuy, and S. Bakhtyar, “A service description framework for intelligent transport systems: applied to intelligent goods,” in 5th European Conference on ICT for Transport Logistics (ECITL 12), 2012.

[13] R. Jedermann, C. Behrens, D. Westphal, and W. Lang, “Applying autonomous sensor systems in logistics—Combining sensor networks, RFIDs and software agents”, Sensors and Actuators A: Physical, vol. 132, no. 1, pp. 370–375, Nov. 2006.

[14] E. Abad, S. Zampolli, S. Marco, A. Scorzoni, B. Mazzolai, A. Juarros, D. Gómez, I. Elmi, G. C. Cardinali, J. M. Gómez, F. Palacio, M. Cicioni, A. Mondini, T. Becker, and I. Sayhan, “Flexible tag microlab development: Gas sensors integration in RFID flexible tags for food logistic”, Sensors and Actuators B: Chemical, vol. 127, no. 1, pp. 2–7, October 2007.

[15] S. Mahlknecht and S. A. Madani, “On Architecture of Low Power Wireless Sensor Networks for Container Tracking and Monitoring Applications,” in 5th IEEE International Conference on Industrial Informatics, 2007, vol. 1, pp. 353–358.

[16] A. Carullo, S. Corbellini, M. Parvis, and A. Vallan, “A Wireless Sensor Network for Cold-Chain Monitoring”, IEEE Transactions on Instrumentation and Measurement, vol. 58, no. 5, pp. 1405–1411, 2009.

[17] A. Van Lamsweerde and L. Willemet, “Inferring declarative requirements specifications from operational scenarios”, IEEE Transactions on Software Engineering, vol. 24, no. 12, pp. 1089–1114, 1998.

[18] A. Sutcliffe, “Scenario-based requirements engineering,” in Proceedings of the 11th IEEE International conference on Requirements Engineering, 2003, pp. 320–329.