Modeling Transport in Service-Oriented Framework for Agent-Based Simulations of Supply Chains

Mariana Dorigatti[a], Omar Chiotti[b] and Hector E. Salomone[c]

[a,b,c] INGAR-CONICET, Avellaneda 3657, S3002GJC Santa Fe, Argentina

{marianadorigatti,chiotti,salomone}@santafe-conicet.gov.ar

Abstract.

This work presents an extension to the service-oriented framework for performing agent-based simulations to support the analysis of collaborative relationships in supply chain interactions already described in [5]. The extension includes a new service for simulating the planning of transportation, which is designed to allocate finite transportation resources to the transportation orders. The service includes the functionality that matches the required vehicle type, location and time to track the availability of a vehicle fleet as it is allocated to a plan.

The new service proposal is validated by simulating three different scenarios of a case study. The case study is based on an actual supply chain of dairy products covering the central region of Argentina.

Keywords: Transport service, Agent-based simulation; Collaborative supply chains; Service oriented framework

1 Introduction

Supply chain collaboration implies that two or more independent members work jointly in planning and executing supply chain operations for enhancing their performance. E-business environment enabled a series of collaboration mechanisms for information sharing, operation coordination, and joint decision making that convey the promise of improving competitive advantages for all members engaged in a common supply chain [1].

There are strong theoretical arguments about the improved overall performance attained through collaboration are not noticeable in practice since, to some known configurations, there is usually a dominant member of the supply chain that sets the pace for the collaboration extent. One of the main obstacles found in expanding collaborative models to supply chains without a dominant member is the difficulty in making a fair assessment of their benefits and required efforts and fairly distributing them among chain partners.

Even though analytical models have provided very valuable qualitative and quantitative insights, for a better understanding of collaborative models and their mechanisms, simulation-based approaches are recommended since they can afford the complexity of real scenarios [2], [3].

Building ad-hoc simulation models for studying complex supply chain interactions can be prohibitive in terms of both cost and time. Therefore, the availability of simulation frameworks, easy to use by business managers and facilitating the development of those models, has a strong incentive in the quest of current business efforts to increase their supply chain performance [4].

The objective of this work is to extend the service-oriented framework for performing agent-based simulations to support the analysis of collaborative relationships in supply chain interactions presented in [5] by adding a service component that performs transport planning. This service component includes the functions and logics needed for generating a transport orders plan and assigning vehicles to those orders.

The remainder of this work is structured as follows: Section 2 discusses related works . Section 3 presents a framework overview by describing its components and the proposed extension. Section 4 describes a validation by using a case study. Finally, a concluding discussion is presented in Section 5.

2 Related Works

Discrete Event simulations and System Dynamics are modeling approaches widely used to analyze supply chain behavior. An extensive literature review [7] records the usage of both approaches to deal with different supply chain problems. Despite the large list of contributions, the study of dynamic supply relationships among independent members appears as barely explored.

The proposal made by Umeda and Lee [8] perform a generic hybrid-modeling framework that combines discrete-event simulations with system-dynamics simulations. The operational processes within the supply chain are represented for discrete event models; and reactions in supply chain management circumstances are represented for system dynamic models. Conventionally discrete event approaches are focused on adopting a network perspective, therefore, supply chains become represented by a fixed topology and structure, while it is generally assumed an implicit representation of the control and decision process.

The works by [9] and [10] were the pioneers in having a vision of agents for the purpose of a flexible, reusable and generalized modeling and simulation framework. Models are made of components representing different entities in the supply chain; these components are reusable. The interaction protocols are submitted to support agents' interactions through message passing by regulating the flow of materials, information flow, and cash flow. In their library is classified as structural components and control elements, although the proposed controlling structures fall short in providing explicit representations of the relationship between decision making activities and their corresponding execution actions.

In work described by van der Zee & van der Vorst [11] recognize the need for an explicit definition of control policies and coordination mechanisms. They also highlight the need for the/an explicit definition of timing and execution of decision activities. About these requirements, they propose a modeling framework based in the key concepts: agents and jobs. The internal structure of agents is only defined at a very high level (jobs and resources) and the definitions of actual entities in supply chain are left to the specialization of this structure. And the planning and control concepts are explicitly represented through decision making agents carrying out control jobs.

A framework for building hierarchical simulation models that integrate discrete event simulation and spreadsheets is presented by Pundoor and Herrmann [4]. These process elements are standardized modules that can be reused and each process element is implemented as a separate submodel that represents a specific activity in the supply chain. The first level correspond the simulation model; the second level has submodels that correspond to supply chain participants (consumers, producers, and traders) and the third level has submodels that correspond to process elements performed by each participant. Each participant in the supply chain has its own set of modules. For building the supply chain simulation model, these modules are put together and connected using standard interfaces that represent the material, information, and cost flow. Each participant of the supply chain also has an Excel workbook associated with it. Planning activities are carried out by using Excel VBA. Execution is carried out in Arena. Activities are conceived as business processes by adopting SCOR reference model; and their implementation distinguishes between control and decision making activities.

In the work presented by Rosetti and Chan [12], propose a object-oriented framework, providing a list of reusable classes for modeling and simulating a supply chain. The structural aspects are understood by a *RelationshipNetwork*, composed of *Nodes* and *Relationships*. A *Relationship* represents a conceptual connection between two nodes and indicates the possible flow of information or material between nodes. A *Node* can represent a Facility, Region, or OrderGenerator. Control and decision making activities are not clearly decoupled from execution activities except for a class InventoryPolicy which encapsulates rules to control the associated inventory. Although the framework provides a class to represent the concept of an Order, transactions do not follow an explicit interaction protocol.

Labarthe et al. [12] present a framework for multi-agent simulations integrated by cognitive and reactive agents. Cognitive agents are based on decision models or procedures that require information interaction with other agents, are responsible for decision making or deliberative activities and they communicate through messages. In turn, reactive agents perform execution activities, in this work is introduced a clear separation between the two levels.

In the work presented by Chatfield et al. [9] divide the supply chain performance into five types of constructs: *node*, *arc*, *component*, *action*, and *policy*. Physical aspects of the supply chain are contained in node, arc, and component objects, whereas activities and logical aspects of the supply chain are described by using action and policy objects.

3 Framework Overview

A service-oriented framework for performing agent-based simulations to support the analysis of collaborative relationships in supply chain interactions was presented in [5]. That framework has been developed based on five main requirements: (i) provide reusable components that can be easily assembled to set a wide variety of supply chain scenarios among independent partners; (ii) allow for establishing dynamic links among partners without requiring a predefined network structure with the purpose to support collaborative interactions, support collaborative interactions; (iii) adopt a business process-oriented perspective to reflect the activities in the supply chain, following some established reference model known to most business analysts; (iv) provide for explicit representations of control and decision making activities separated from execution activities; and (v) support the interaction among members by using message-based and document-oriented protocols that resemble actual business interactions. Section 3.1 provides a brief overview of the framework.

This framework was extended by the functional component *PlanTransport_Service* that encapsulates internal activities and exposes public operations needed to support interactions for transport planning and vehicles assigning. Section 3.2 presents this extension.

3.1 The service-oriented framework

We explicitly adopt a business process perspective to capture all relevant activities in the supply chain, distinguishing between intra-organizational processes of each participating member and inter-organizational (collaborative) processes.

To organize the different concepts and entities that compose the framework, we distinguish among three aspects that need to be accounted for: *organizational*, *structural*, and *functional*.

Organizational aspect, refers to a business unit, which is represented as a member of the supply chain. Term member refers to an independent organizational unit able to make decisions based on its business goals. In this way, the supply chain will be modeled as a homogeneous agency of agents *SCMember*, each one representing a supply chain member.

Structural aspect, represents the physical logistic structure of each member. The structural aspect of the supply chain is modeled by adding components of type *Location* to a given *SCMember*. Each *Location* represents a storage point and will contain a list of *InventoryItems* to represent all goods being physically managed in this storage point. Entity *InventoryItem* represents the widely used concept of SKU (stock-keeping unit) (**Fig. 1**)

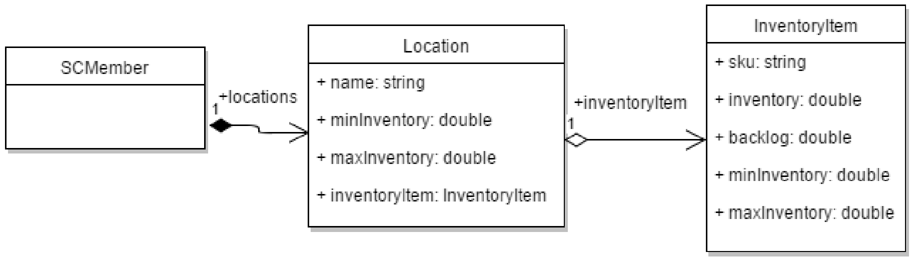


Fig. 1. Structural components

Functional aspects, are directly related to services. All activities in the supply chain are modeled as business processes organized in components representing services that can be offered by *SCMember*. In the proposed framework five generalized services were defined (**Fig. 2**). These services can be used as components to model the functional aspects of each member in the supply chain.

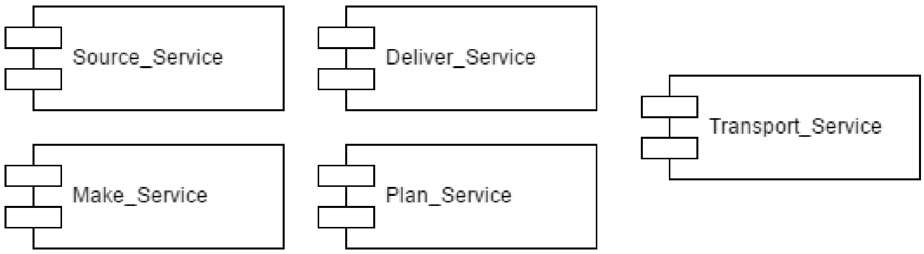


Fig. 2. Functional Components as Service

Table 1 depicts the operations that the interaction interface of each service exposes and its functional behavior, which adopts a business process perspective, based on the SCOR reference model. SCOR was chosen since it is the standard most widely used for supply chain modeling.

Service	Operation	Description
Source_Service	□ inform_DeliverNotice □ inform_TransportArrival □ inform_ModifyOrder □ inform_CancelTransferOrder	Provides the functionality to perform the business process for a client to receive materials from a supplier.
Deliver_Service	□ inform_CancelTransportOrder □ inform_DeliverReceipt □ inform_TransportArrival	Provides the functionality to perform the business process for a supplier to deliver materials to a client.
Make_Service		Provides functions for executing production orders transforming raw materials into products. Only define internal business processes, it has no public exposed operations

Plan_Service	□ inform_SourcingPlan □ inform_DemadRequirements	Provides functions to generate internal transfer and production orders and external delivery orders based on demand requirements, distribution lists, and product structure lists
Transport_Service	□ inform_EndLoad □ inform_EndUnload □ inform_CancelTransport □ request_TransportOrder	Provides functions for execution of transport operation of materials among different locations.

Table 2. Functional components as services

Each service corresponds to a business process that executes internal activities for process execution. These services contain operations that receive a call and return a response through a well-defined interface. The interface is the result of defining a standardized business process to coordinate supply chain activities.

The interaction among *SCMember* agents is achieved by delivering messages to the corresponding service port. Messages have attached a business document including all information needed to its interpretation. In the framework, these messages are based on the Agent Communication Language of FIPA (Foundation for Intelligent Physical Agent). These messages have been modeled by the *SCMessage* class, which includes attributes *sender* and *receiver*, denoting the identification of the corresponding *SCMember*; attribute *performative*, indicating the type of the communicative act; and attribute *content*, including business objects needed to understand the message.

Interactions are supported by business documents *TransferOrder* and *TransportOrder*. Document *TransferOrder*, composed of a list of order items where each one indicates SKU and quantity, specifies the origin and destination of a delivery. Document *TransportOrder* is composed of a list of destinations to visit in a trip and, for each location, a list of references to transfer orders to be processed for loading or unloading on the site.

In this way, the framework organizes functionality by encapsulating all internal processes as much as possible and reducing public interactions to a minimum set of message-based requests to exposed operations.

3.2 Extending the framework for modeling transport services

As mentioned earlier, all activities in the supply chain are modeled as business processes organized in components representing services that can be offered by SCMember. The service that performs transport planning is represented by *PlanTransport_Services*, which includes the functions and logics needed to generate a plan of transport orders and assign vehicles to those orders.

All interactions among roles are modeled as a service operation invoked through a standard message carrying the needed business information. For this process the document exchanged between roles (*Supplier* and *Carrier*) is *TransportPlan* which is a business document containing three lists of orders: required, accepted and rejected orders.

The collaborative process for planning the transport order is depicted in Business Process Model and Notation (BPMN) in **Fig. 3**. This business process shows the interaction between two SCMember – one in the role of Carrier and the other as the Supplier. The process begins when the *Supplier* prepares the requirements for transportation initializing a TransportPlan document with a list of required orders and informs the *Carrier* by sending a message for <*requestTransportPlan*>. The plan of transport orders is generated by process <*Generate Transport Orders*>, then for each transport order generated it is assigned a vehicle by business process <*Assign Vehicle*>. The assign vehicle method is described in more detail below.

After the transport order plan was generated and, the TransportPlan document is completed with the list of accepted and rejected order and sent back to the Supplier by the message <*inform_TransportPlan*>.

The Supplier analyzes the proposed orders and confirms the plan by <*confirm_TransportPlan*>. When are confirmed by the Supplier, the Carrier updates their status, thus each order is ready now to be executed. For those orders that were not confirmed, the supplier selects an alternative carrier and repeats the process.

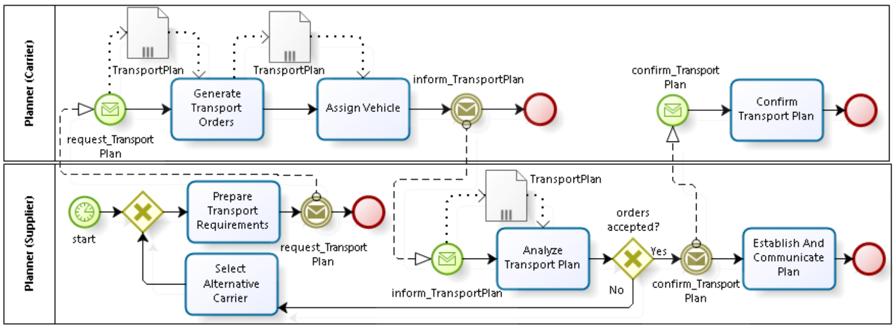


Fig. 3. Collaborative Business Process for executing a plan of transport

In accordance with this collaborative business process for planning of transport, a new functional component *Plan_TransportService* was defined to encapsulate internal activities and expose the public operations needed to support the interactions. Also the original *Plan_Service* used to generate the supply chain plan was extended to add the new interaction <*inform_TransportPlan*> and new internal processes: <bp_PrepareTransportPlan>, <bp_AnalyzeTransportPlan> and <selectAlternativeCarrier>. The extended services are depicted in **Fig. 4**.

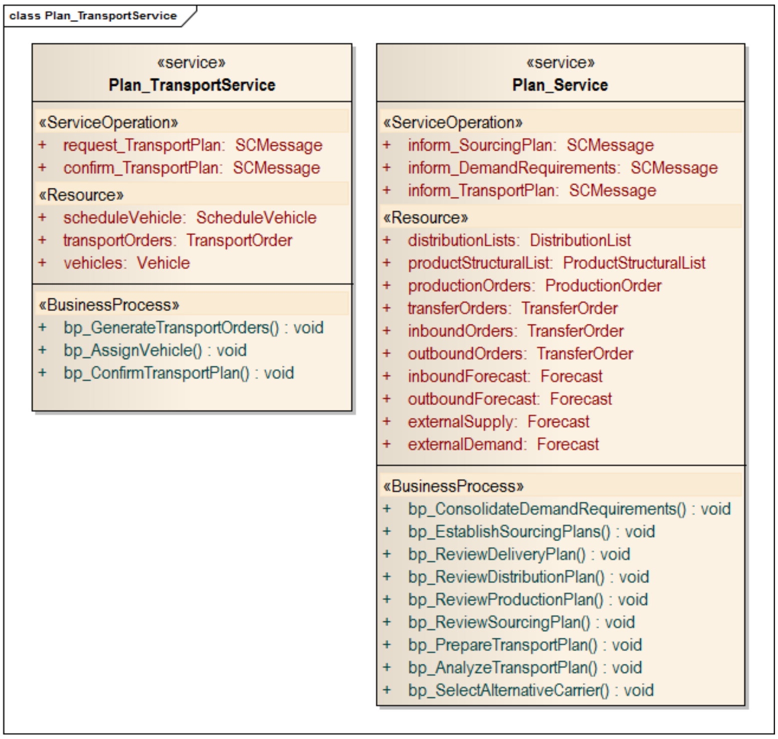


Fig. 4. Extended *Plan_Service* and new *Plan_TransportService*

In order to support the operation of the new *Plan_TransportService*, the following resources have been defined:

transportOrders, provided to represent a consolidated repository of transport orders.
- *vehicles*, a list of elements of class *Vehicle*, designed to represent any transportation resource that can move among entities *Location* carrying limited amount of items and account for transit times and in-transit inventory.
-*scheduleVehicle*, a structure designed to represent the occupation timetable for each vehicle once it is assigned with transport orders. This structure has been implemented as an *IntervalTree* that provide efficient methods for searching and detecting occupation conflicts in the order assignment process. This component is depicted in **Fig. 5**.

The *IntervalTree* is an augmented version of a self-balancing Binary Search Tree (BST) to maintain a set of intervals so that all operations can be done in O(Log n) time. It is an organized set of nodes; and each *Node* is represented by an interval. An *Interval* is defined by a couple of dates: minimum date (low) and maximum date (high). These dates will refer to a period of time that represents vehicle occupation. Each time a transport order is assigned to a vehicle a new interval node is inserted in the tree to account for the occupation related to that order. When a tentative order is evaluated to be assigned the search method is used to efficiently detect overlaps with the current set of assigned orders. For a given fleet, the logic of the vehicle assignment can be configured to minimize the number of allocated units (leading to few used units with high occupation rate) by prioritizing the available vehicle with highest load or vice versa, to maximize the distribution of loads among the available fleet.

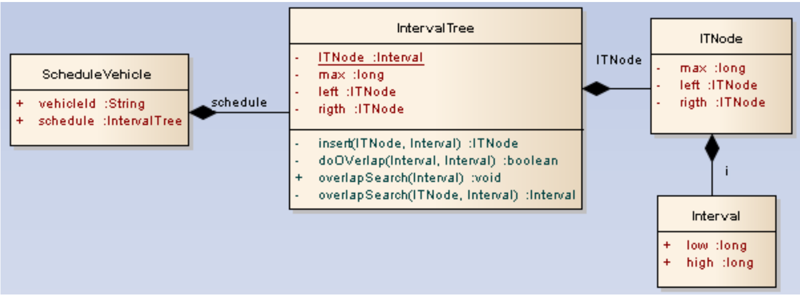


Fig. 5. Schedule vehicle as a resource of plan transport service

4 Validation of transport model using a case study

The proposed extension was added to the current Anylogic 6.8 implementation of the framework. A case study was utilized to perform a validation of the framework extension. The case is based on an actual supply chain of dairy products covering the central region of Argentina. It involves a manufacturing site for processing raw milk to produce a variety of dairy products and nine regional distribution centers that can store and deliver these products to satisfy local demand. For the purpose of this example, four main families of products are considered: Powder Milk (PM), Long-life Milk (UHT), Yogurt, and Fresh Cheese. The first two families, PM and UHT, do not require refrigeration during storage and transportation, and use different warehousing and transportation resources from those required by fresh products. For transporting these product families, there are six types of vehicles: with refrigeration (Fresh) and without refrigeration (Dry), which are combined in three different capacity sizes.

Type SKU		Capacity	Min (ton.)	Max (ton.)
		Small	15	19
DRY	FRESH	Medium	26	32
		Big	44	52

Table 3. Detail of vehicle and capacity types

Products are shipped by trucks from the factory; and transit time ranges from hours within the same day to two days for the most distant destinations. The transport path is always started at base location of the Carrier and finished at the same base when completing the roadmap.

This supply chain was modeled using the framework described in [5] with the extended functionality to plan the transport operations introduced here. Three scenarios were formulated to validate the ability of the proposed service to simulate the creation and modification of transport plans.

- Scenario 1: Dimensioning the fleet that supports the transportation requirements.

In this first scenario, the model was used to evaluate the fleet size required to support the supply chain distribution plan without affecting the service level; i.e., the fleet is dimensioned to never become a bottleneck to the distribution plan. A planning horizon of 180 days was considered, with plan revisions every 30 days and a weekly vehicle assignment to the transportation plan. A unique carrier was defined having a large number of vehicles of each type. The vehicle assignment method was selected to minimize the number of used vehicles. **Table 4** shows the results of simulating this scenario. For each vehicle type, the simulated usage profile records the number of simultaneous vehicles that were utilized at every moment during the execution of the plan. The maximum number of simultaneously allocated vehicles is considered as the size of the fleet required to fully support the distribution plan without constraining the service level.

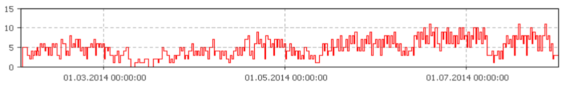
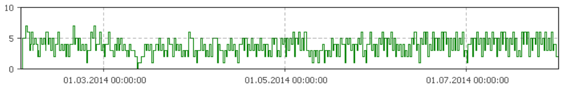
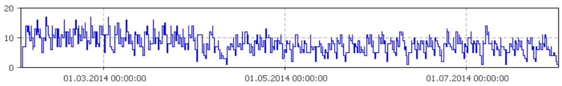
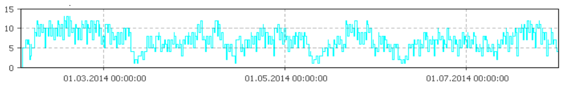
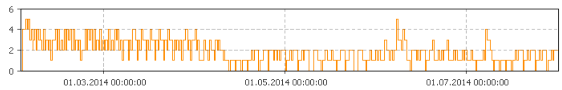
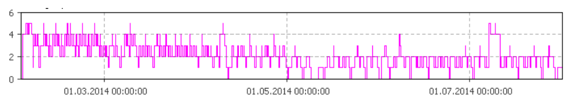
Vehicle type	*Max #*	*Avg. Occupation %*	*Usage profile*
Small-Dry	11	40.3	
Medium-Dry	7	52.7	
Big-Dry	17	46.8	
Small-Fresh	13	46.3	
Medium-Fresh	5	38.9	
Big-Fresh	5	40.4	

Table 4. Scenario 1: Dimensioning the required fleet

Note as well that given the current policies for loading/unloading and the given routing scheme the average occupation rate of this fleet is rather low. To increase the occupation rate, different routing schemes and more flexible loading/unloading periods should be considered illustrating how the model can be used to support with quantitative assessment a typical negotiation of a collaborative agreement between supplier and carrier.

- Scenario 2: Effect of the fleet size on occupation rate and service level.

In a second scenario, the number of available trucks of every type is reduced to show the impact of this resource limitation in the overall performance of the supply chain. Starting for a fleet size dimensioned in the previous scenario, the number of available trucks of the single carrier were reduced 10%, 20% and 40%. For each case, a simulation experiment including 15 repetitions with different values of random parameters was conducted for collecting the expected values of performance indicators over 180 days of horizon. **Table 5** summarizes the average values of the most relevant indicators.

	Ideal fleet	-10%	-20%	-40%
Total Planned Orders	2292	2302	2328	2363
Canceled Orders	28	89	169	553
Delayed Orders	31	46	48	25
% Orders Delivered In Full	97.4	94.1	90.7	75.6
Vehicle occupation rate %	46.9	51.1	55.1	61.7

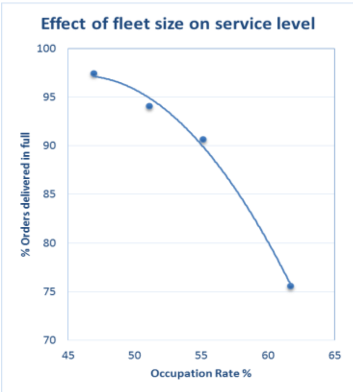


Table 5. Fleet size, occupation rate and service level

It can be observed that as the fleet is reduced, the vehicle occupation rate is increased but the overall supply chain service level, measured as % of orders delivered in full, decreases affected by the limitation in transportation resources.

- Scenario 3: Alternative carriers and dynamic shifting of allocation

In a third scenario, an alternative carrier is made available for the supplier to consider when creating the transport plan. In addition, a disruptive event is scheduled to simulate a reduction of the service level of preferred carrier and illustrate how the proposed model is able to dynamically modify the transport plan shifting the allocation of orders from one carrier to the other in response of the event. During a window of time of 45 days, the availability of resources for Carrier-1 (which is the preferred one) is randomly reduced affecting its ability to confirm requested transportation orders. **Fig. 6** shows the number of daily trucks allocated to each carrier. The arrow indicates the period of time where the availability of Carrier-1 is reduced in correspondence with the increase of allocation to the alternative Carrier-2.

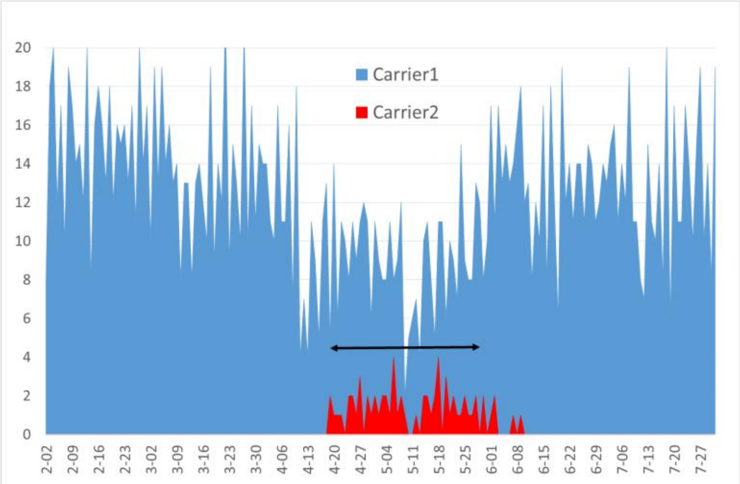


Fig. 6. Allocated trucks per day

5 Conclusions

In this work we have presented an extension to the service-oriented framework for performing agent-based simulations to support the analysis of collaborative relationships in supply chain interactions already described in [5].

The extension includes a new service for simulating the planning of transportation. This service is invoked by the previous supply chain plan service to provide with finite transportation resources (the vehicles) that are allocated to the transportation orders.

The service includes the functionality to track the availability of a fleet of vehicles as it is allocated to a plan. The allocation logic matches the required vehicle type, location and time.

The new service proposal was validated by simulating three different scenarios of a case study. The first scenario demonstrated the application of the model to evaluate the size of the fleet that is required to support a distribution plan without affecting the overall service level. The second scenario illustrated how the reduction of the fleet affects the service level and the occupation rate. And the third scenario included a second, non-preferred alternative carrier to demonstrate the ability of the model to dynamically change the transportation plan in response to an unexpected change (event) in the availability of the main preferred carrier.

6 References

[1] J. M. Swaminathan and S. R. Tayur, "Tactical Planning Models for Supply Chain Management," pp. 1–44, 2003.

[2] H. Chan and F. Chan, "A Review Of Coordination Studies In The Context Of Supply Chain Dynamics," *Int. J. Prod. Res.*, vol. 48, no. 10, pp. 2793–2819, 2010.

[3] U. Ramanathan, "Performance of supply chain collaboration - A simulation study," *Expert Syst. Appl.*, vol. 41, no. 1, pp. 210–220, 2014.

[4] G. Pundoor and J. W. Herrmann, "A hierarchical approach to supply chain simulation modeling using the Supply Chain Operations Reference model," *Int. J. Simul. Process Model.*, vol. 2, no. 3–4, pp. 124–132, 2006.

[5] M. Dorigatti, A. Guarnaschelli, O. Chiotti, and H. E. Salomone, "A Service-Oriented Framework for Agent-Based Simulations of Collaborative Supply Chains,". *Computer in Ind.* 2016 *(in press)*

[6] Supply Chain Council, *Supply-Chain Operations Reference-model SCOR Version 6.1 Overview*. 2004.

[7] A. a. Tako and S. Robinson, "The application of discrete event simulation and system dynamics in the logistics and supply chain context," *Decis. Support Syst.*, vol. 52, no. 4, pp. 802–815, Mar. 2012.

[8] S. Umeda and Y. T. Lee, "Design Specifications of a Generic Supply Chain Simulator," *Proc. 2004 Winter Simul. Conf. 2004.*, vol. 2, pp. 103–111, 2004.

[9] D. C. Chatfield, J. C. Hayya, and T. P. Harrison, "A multi-formalism architecture for agent-based, order-centric supply chain simulation," *Simul. Model. Pract. Theory*, vol. 15, no. 2, pp. 153–174, Feb. 2007.

[10] J. M. Swaminathan, S. F. Smith, and N. M. Sadeh, "Modeling supply chain dynamics: A multiagent approach," *Decis. Sci.*, vol. 29, no. 3, pp. 607–631, 1998.

[11] D. J. van der Zee and J. G. a. J. van der Vorst, "A Modeling Framework for Supply Chain Simulation: Opportunities for Improved Decision Making*," *Decis. Sci.*, vol. 36, no. 1, pp. 65–95, Feb. 2005.

[12] M. D. Rossetti and H. Chan, "A Prototype Object-Oriented Supply Chain Simulation Framework," no. 1991, pp. 1612–1620, 2003.