Tampere University

Ufuk Şıklar

WEB-BASED 3D VISUALIZATION SYSTEM FOR ROAD FREIGHT OPERATIONS

Faculty of Engineering and Natural Sciences
Master of Science Thesis
April 2020

ABSTRACT

Ufuk Şıklar: Web-Based 3D Visualization System For Road Freight Operations
Master of Science Thesis
Tampere University
Automation Engineering. Factory automation and industrial informatics
April 2020

Contemporary supply chains have become more complex with the globalization. In this complexity, shorter product life cycles, more demanding customers, and increased product variety create a volatile and uncertain environment, which is becoming more and more costly for organizations. In order to maintain competitiveness, organizations must provide high-quality services and meet customer expectations by creating a flexible and responsive structure in this dynamic environment while reducing the cost of the operations.

Information and communication technologies (ICT) have significant impact on supply chain performance in terms of collection, analysis, and transfer of information. Inter-organizational ICT and intra-organizational ICT are distinctly different roles of ICT within a supply chain. Intra-organizational ICT provides control and monitoring of internal processes, whereas inter-organizational ICT creates an environment to support collaboration and coordination among the organizations. Recently, the development of web technologies enhanced real-time information exchange in a cost-effective way leading to greater collaboration and rapid identification and efficient mitigation of potential risks in supply chain networks.

Aim of this thesis project is to develop a web-based, client-side rendered 3D visualization system for freight operations to enhance competitiveness of the organizations in the supply chain. This novel system provides a real-time collaborative workspace to work on cargo consolidation, which could also be used in training program of new employees. Visualization of container content and animation of cargo loading, unloading, and rearrangement tasks increase visibility, improve workforce effectiveness, reduce operation time, and therefore contribute to cost reduction strategies. In addition to this, dynamic packing sequence algorithm (DPSA) is developed to place items into partially loaded ongoing trucks in the most efficient way. This feasible loading sequence solution provides great advantages such as maximizing truck utilization rate, more efficient asset utilization, and rapid response to unexpected last-minute events with ongoing trucks, thus creates a more flexible and agile structure.

The visualization system has three-tier architecture. In the client tier, HTML5 and CSS are used to form the web page and Angular.js framework is used to extend capabilities of HTML5 for a more dynamic web page. Rich interaction experience is achieved by client-side rendering of the 3D graphics using Three.js framework. In the server tier, a web server is created by Express.js framework. The visualization logic is developed and implemented into the server along with the DPSA. Both for the visualization logic and the DPSA processes, the data is retrieved from the database, then sent to client-side for visualization. Data is stored in MongoDB in database tier which contains the vehicle fleet information such as truck routes, order details, item positioning and details of supply chain processes. Real-time client-client and server-clients communications are maintained by Socket.io framework.

A working prototype is developed and tested with a comprehensive sample scenario, and results are presented. The visualization system has performed well and smoothly with all features. DPSA may bring new insights into container loading problems and may be implemented into other container optimization algorithms, and the functionality and efficiency of this system could be improved with further developments in the future.

Keywords: Web-based visualization, supply chain management, container optimization

The originality of this thesis has been checked using the Turnitin Originality Check service.

PREFACE

I would like to thank my supervisor Dr. Borja Ramis Ferrer for his support and guidance on this thesis. I also would like to thank and express my gratitudes to Prof. Risto Ritala for reviewing and examining the thesis, and valuable feedback. In additon, I would like to thank Luis Gonzalez Moctezuma for examining the thesis. Also, special thanks to Eeva Castrén and Şeyma Avcılar for their valuable friendship and support.

Finally, I would like to thank my family for their endless support, love and encouragement.

Ufuk Şıklar
Tampere, April 2020

CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS AND ABBREVIATIONS

3PL	3rd Party Logistics
CLP	Container Loading Problem
CSCMP	Council of Supply Chain Management Professionals
CSS	Cascading Style Sheets
DPSA	Dynamic Packing Sequence Algorithm
FTA	Freight Transport Association
FTL	Full Truck Load
HTML	Hypertext Markup Language
HTTP	Hypertext Transfer Protocol
ICT	Information and Communication Technology
IIPP	Identical Item Packing Problem
JSON	JavaScript Object Notation
LIFO	Last In First Out
LTL	Less than Truck Load
MSTL	Multi Stop Truck Load
NOSQL	Non Relational Structured Query Language
OPENGL	Open Graphics Library
SCM	Supply Chain Management
UML	Unified Modeling Language
UP	Unified Process
VRML	Virtual Reality Modeling Language
WEBGL	Web Graphics Library

V_I	Volume of an item
V_O	Volume of an order
T_T	Total time spent for loading and unloading tasks
V_{IN}	Volume of initial orders
V_{NO}	Volume of new order
T_N	Delay caused by loading and unloading tasks
V_R	Total rearranged volume
V_{OR}	Volume of the rearranged order
T_R	Delay of total rearranged volume
T_c	Cumulative delay factor
T_T	Total packing time for initial condition
R	Delay rate

1. INTRODUCTION

The main aim of a supply chain is to deliver goods and services with minimum cost. As competition has intensified with the globalization over the last few decades, achieving high quality of services and meeting customer expectations with lower cost has become a challenge. Organizations have realized that not only maintaining close relationship with customers but also improving the efficiency of whole supply chain is required in order to remain competitive. Integration of information and communication technologies (ICT) into supply chain gained more attention since utilization of ICT is considered as a significant approach for improving supply chain. In fact, ICT has become a significant instrument for managing and improving the efficiency and performance of supply chains.

As goods and services in a supply chain flow from supplier to customer, demand information flows in the opposite direction. Information provides visibility and therefore supports managers in decision making process. Thorough collection and analysis of information is crucial for increasing supply chain performance. Accordingly, ICT systems provide necessary tools for collecting and analysing such information. In addition to decision support, ICT systems also support collaboration and coordination through information sharing [1]. Furthermore, ICT has impact on trust and commitment between customer and supplier [2]. More specifically, inter-organizational ICT and intra-organizational ICT are distinctly different roles of ICT within a supply chain. Inter-organizational ICT provides better supply chain integration that improves the performance through two aspects of integrative practices. First, information sharing allows partners of supply chain access to strategically significant information, such as inventory levels, forecasts, and future plans. Second, cooperation between supply chain partners creates the capability of joint problem solving and flexibility in mutual arrangements. Hence, inter- organizational ICT is an enabler for supply chain performance improvement via integration. On the other hand, intra-organizational ICT is practices or technologies to share information within an organization for improving its cost, return on invest, product quality and innovation performance. Thus, intra-organizational ICT is embedded into an organization in order to monitor and control internal processes, whereas inter-organizational ICT creates an environment for sharing information among the organizations within a supply chain. Although having different roles the two aspects are linked in that the effectiveness of inter-organizational information sharing and cooperative relationships depend on the quality of internal information. When intra-organizational ICT improves the quality of information, accuracy, timeliness, adequacy and credibility, it impacts the effectiveness of supply chain integration [3].

Since the internet is ubiquitous and cost effective environment for businesses, web technologies have played a crucial role in broadening the scope of inter-organizational and intra-organizational relationships. Web technologies makes it possible for supply chain networks (SCM) to share information in real time. Real-time data enhances the transparency of the supply chain, therefore it enhances collaboration and identification of potential risks. Frohlich [4] states that higher level of supply chain performance is directly linked with web-enabled supply chain activities. Furthermore, developments in high-speed network access increase the adoption rate of Web applications in SCM activities [5, 6].

1.1 Motivation for Thesis

Supply chain networks have become more complex as businesses compete in increasingly dynamic and globalized environment. In particular, short product life cycles, increased product variety and more demanding customers who expect to get high quality products/services creates volatile and uncertain environment. In order to remain competitive, companies need to be more flexible and responsive in this fast changing environment as well as reduce costs of operations. A web-based visualization system that visualizes truck contents can be implemented into supply chain operations to enhance an organization's competitiveness. Such a system can provide versatile benefits in logistics and supply chain activities by supporting cost reduction strategies and creating a responsive structure against uncertainties.

1.2 Problem Statement

In a supply chain network, suppliers prefer to deliver products more frequently in smaller quantities in order to meet fluctuating customer demand [7]. Often, these shipments are not large enough to fill a truck completely reducing the economic efficiency of transportation [8]. Carriers consolidate shipments from multiple suppliers into full-truckload shipments to maximize volume. To maximize volume efficiency, various container loading algorithms have developed [30]. However, these algorithms have limited practical value, since real-life problems comprise various hard and soft constraints that need to be taken into account simultaneously when loading trucks [72].

Hard constraints must be satisfied in order to create a feasible solution while soft constraints can be tolerated within limits [73]. Nevertheless, maximizing volume with hard constraint may lead to inappropriate solutions in that relaxing the constraint and studying volume/unloading time balance of a MSTL would lead to better economic performance. For instance, Last in First out (LIFO) loading policy is a hard constraint encountered in Container Loading Problem (CLP) for a multi-stop truckload (MSTL) shipment, which is a variant of Full Truckload (FTL) shipment. This constraint prevents unnecessary delay by re-handling of irrelevant cargo at pick-up or drop-off locations, and thus increases the efficiency of transportation operation. On the other hand, in a MSTL shipment, unused space of truck increase as the truck arrives to other stops in the route. Although there is empty space in the truck, no new items can be placed there as they would invalidate LIFO hard constraint (except destination of the additional cargo is next stop). Therefore, solution of CLP with LIFO hard constraint may maximize used volume for given set of items at the initial state, but the truck will have unused empty space at next stops. Creating a new dynamic packing sequence with extra items with minimum handling delay can improve current optimization solutions for MSTLs. This approach can also decrease the adverse effects of uncertainties caused by last minute changes or vehicle breakdown.

Consolidation processes are often carried out at cross-dock terminals. Cross-docking is the operation of transferring products from inbound vehicles to outbound vehicles with minimal (at most 48 hours) or no storage time. Therefore, cross-dock terminals are very busy places. Moreover, cross-dock operations are labor intensive and costly since they have to handle products rapidly [9]. Labor effectiveness is the key to success of cross-docking operations in a supply chain network. Complicated loading-unloading patterns or inexperienced workers may reduce operational efficiency.

A new dynamic packing sequence algorithm (DPSA) with a visualization system is required for efficient use of truck space while keeping the loading and unloading time small. The company objective is increased volume utilization rate and the capability of the vehicle fleet to quickly respond to sudden changes, such as last minute orders or vehicle breakdowns. Furthermore, the visualization system can support workers by visualizing loading-unloading tasks, monitoring fleet status and providing a real-time interactive environment to support collaboration between business partners.

Based on the problem statement, research questions can be defined as:

- What kind of uncertainties exists in transportation operations in a supply chain and how to deal with such uncertainties in a cost effective manner?

- What are the recent technologies used in web-based visualization and how web-based 3D visualization supports transportation operations?
- How to create real-time collaborative web environment and maintain real-time communication between collaborating supply chain partners?
- How visualizing container content supports labors in loading/unloading operations?

1.3 Contribution

The main goal of this research project is to develop a visualization system that enhances competitiveness of an organization by improving flexibility, responsiveness, collaboration and asset utilization efficiency in current dynamic and uncertain supply chain environment. To realize such features, this thesis contributes the following:

- A DPSA and an implementation that visualizes the result.
- An implementation that visualizes and animates loading, unloading and rearrangement tasks.
- A system with real-time communication of DPSA results between clients for collaborative operations.

1.4 Assumptions and limitations

The primary consideration of DPSA is to create feasible packing sequences to form a flexible structure which is able to respond rapidly to sudden changes in a dynamic environment. Visualizing the solution provides visibility of all processed items. All processed items are also reachable by workers or item handling equipment. All items of an order and orders having same drop-off locations are grouped together, respectively. In this process, problem type is limited to identical items (IIPP), since placing strongly heterogeneous items requires sophisticated optimization algorithms. However, solving for a dynamic packing sequence is independent from such optimization. Moreover, IIPP type allows to demonstrate solution explicitly and to reflect the absolute impact of the method. Nevertheless, this method can be combined with optimization algorithms for strongly heterogeneous items in multi-stop situations as a future work.

Item loading/unloading method (i.e. by forklift, or human) and item stacking style is same at all locations. Also, items which will be placed into front side of a container at the location, stay closer to the truck door than other items which will be placed into rear part of the container. Such way of stacking and using same loading method keeps loading and unloading time equal for all items. Therefore, total delay is proportional to volume and number of loading/unloading/rearrangement operations. Other considerations are:

- Vehicle fleet consists of identical trucks (ISO 40' high cube container) having single rear loading door.
- All subset items of received order must be placed into same truck (complete shipment a hard constraint).
- Total volume of the cargo must not exceed container volume.
- Items that belong to a same order must be placed together (relative positioning constraint).
- All trucks are assumed to have optimized route and orders initially. In order to keep the initial optimized delivery route and orders of a truck unchanged, the delivery point of any new order must be among truck's future unloading points. Matching process finds suitable truck before running the algorithm.

1.5 Methodology

This thesis project is planned as follows. First, relationship between supply chain and ICT is introduced. Then, current state of today's supply chains is analyzed and related problems identified. Web-based visualization system with specific features and DPSA is suggested as a solution. Afterwards, theoretical background is extended by scrutinizing visualization concept and CLP. Technologies, state of the art concepts and tools are researched and presented. Unified Process (UP) methodology is followed to develop the visualization system. UP is an adaptive approach in which requirements can change or evolve during the development. Since some of exact system requirements were not clear before preliminary development work, UP is a convenient methodology for this thesis project. Moreover, UP is an iterative and incremental development process. Each system component and functions are developed and tested iteratively. In initial iterations, key functions are implemented into the system then, whereas less crucial features are implemented in the later iterations. System model is represented by using Unified Modeling Language (UML) notation which is convenient for object-oriented development. Use Case Modeling Technique is utilized for explaining system capabilities.

1.6 Thesis Outline

This thesis is organized as follows: In Chapter 2 background information required for the research project is presented. Supply Chain management, Container Loading Problem and web-based visualization are scrutinized. Chapter 3 is concerned with the system development. System architecture, components, features, visualization logic and DPSA are explained. In Chapter 4, the visualization system is tested with an experimental scenario and the results are analyzed. Chapter 5 answers the research questions. Finally, Chapter 6 concludes the thesis project and proposes ideas for future development of the system.

2. SUPPLY CHAIN PERFORMANCE AND CARGO LOADING

This chapter presents background information required to understand importance of the visualization system, developed in this thesis project. SCM and CLP cover a wide range of research topics, therefore presentation is limited to topics most relevant for this thesis.

2.1 Supply Chain Management

2.1.1 Overview of Supply Chain Management

Supply chain is a network of various organizations such as suppliers, manufacturers, service providers, distributors, wholesalers and retailers where raw materials are manufactured into products and delivered to customers. In this network, monetary transactions and information flow from customer to supplier (upstream), whereas goods and services flow from supplier to customer (downstream). In addition to this, products may also flow backwards as a reverse logistics process because of unsatisfied customers who want to return the products. Smooth flow of information, products and monetary transactions is essential to supply chain's success in terms of maintaining customer satisfaction and high profitability [10].

Creation of supply chains started with trading of goods. In this simple form of supply chain, traders locate suppliers and deliver goods to customers. Since the trading process was between local buyers and sellers, these rudimentary supply chains faced little problems of communication, logistics, storage and payment. Explorer-merchants contributed to development of supply chains by traveling between continents and discovering different types of goods at different locations. Hence, goods in excess at a location flowed to other locations where the same goods were scarce. Moreover, technological developments such as invention of steam engine increased the speed and reduced the cost of transportation which extended supply chains from local to global.

Initially businesses had a strategy to be self-sufficient and highly profitable. This resulted in vertical integration: businesses had their own manufacturing facilities, distribution centers and retail stores, which gave them advantages through shorter cycle time, less work in process inventories and direct control over the operations. This strategy was efficient in local markets. However, product variety had to be increased in order to satisfy diversified customer needs as the markets expanded geographically. Strategy based on outsourcing became more effective to maintain profitability by manufacturing various products at lower costs, although it resulted in longer lead times and loss of direct control. Furthermore, with the impact of globalization, competition among the organizations in various areas has been evolving constantly. Volatility in customer demand and short product cycles necessitate collaboration between organizations and thus successful communication [11].

Today, supply chains are increasingly complex, dynamic and competitive networks involving various collaborating organizations. In such an environment, increased operational costs and uncertainties are inevitable. To succeed in this environment, organizations must satisfy customer needs while keeping the costs minimized and being responsive to uncertainties. Furthermore, in supply chains, a large portion of total cost is logistics costs [12]. Furthermore, transportation costs are a major portion of logistics cost [13] in supply chain. As indicated in Council of Supply Chain Management Professionals's (CSCMP) annual logistics report [14], motor carrier transportation costs 42% of total business logistics costs in 2016. Moreover, Freight Transport Association (FTA) states empty running in road freight transport is around 28% of total vehicle distance between 2012 and 2015 [15]. Therefore, inefficient transportation operations significantly deteriorate overall supply chain performance.

2.1.2 Cost Reduction Strategies in Transportation

Transport is a crucial function of a supply chain, since it enables the flow of goods by forming the physical connection between customers and suppliers. Road transportation can be separated into two types according to volume: Truckload (TL) or Less Than Truckload (LTL) transportation. Usually, shipments up to 2500-4500 kg are called LTL and preferred by shippers who continuously deliver goods in small quantities to meet customer demand [7]. Maximization of truckload volumes and reduction of empty traveled miles are crucial strategies to maintain cost savings and efficient asset management. Consolidating multiple LTL shipments with the same destination or multi-stop truckload shipment (MSTL) are typical methods for volume maximization. MSTL is also becoming more popular among shippers because of both potential for cost savings and reduced environmental impacts. According to 3PL company CH Robinson, business share of MSTL increased from 6.42% to 7.39% between 2013 and 2015 [16].

Another cost reduction strategy is cross-docking, which is a practice in logistics of transferring materials from inbound vehicles to outbound vehicles with little or no storage in between [74].FTL shipments require increased level of average inventory but cross-docking strategy enables reduction in transportation cost without increasing inventory levels as well as maintain quality of customer service [17]. Moreover, cross-docking strategy not only maximizes utilized volume of a truck but also decreases empty running miles.

Once inbound trailer arrived to the receiving dock, goods e.g. pallets, boxes, packages are removed. As LTL shipments vary in size and volume, different handling equipment, such as forklifts or pallet jack, are used. Unloaded goods are directly transferred to the shipping door, where they are loaded to an outbound trailer or stored in the facility for a very short period. As the success factor of cross-docking is short waiting time, cross-dock facilities are very busy places, where workers must handle goods quickly. Hence, cross-dock operations are labour intensive and costly. Almost 20% of LTL transportation costs are due to cross-docking operation [9].

Labour performance is significant in cross-docking operations. Increase in efficiency of labour can be explained with the learning effect. Labour gets familiar with the work and equipment at the first stage of learning process. Then, labour become more efficient as they are more experienced in the work. Simply, learning effect is the relationship between amount of repetition of work and time spent in completion of the work. It is crucial to support and train labour to speed up their learning process towards maximum effectiveness in order to complete a job in an optimal processing time [18].

2.1.3 Uncertainties in Transportation

Uncertainty is a significant factor that affects supply chain performance, leads to inefficient processes and decreases customer satisfaction. The impact has been researched for many years. Models identifying sources of uncertainties in supply chain have been developed. Beier [19] introduced the logistics triad, formed by consignor, carrier and consignee which is the basis of today’s uncertainty models. He claimed that at least these three units are needed in order to do analysis in supply chain. In 1993, Davis [20] developed an uncertainty model that defines uncertainty sources as suppliers, manufacturing and customers. In 1998, Mason and Towill [21] extended the uncertainty model to four main sources as supply side, control systems, demand side and manufacturing process. In 2003, Peck et al. [22] contributed to uncertainty model by adding external sources such as politics and natural disasters. However, all these early approaches were manufacturing oriented. In these studies, transportation was perceived as an insignificant activity within supply chain compared to manufacturing operations.

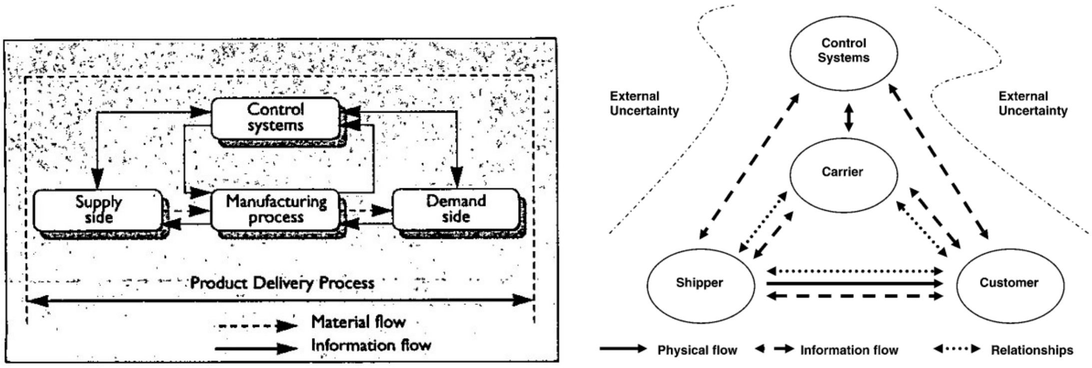


Figure 1. *Early (left) and current (right) structure of the uncertainty model [21, 23].*

Sanchez-Rodrigues et al. [23] proposed a new transport-oriented uncertainty model and evaluated uncertainty sources that affect transport operations within supply chains. In this work, there are four main themes, delays, variable demands, delivery constraints and lack of coordination, defined among the numerous issues causing uncertainty in transport operations. It is claimed that delays are the most significant reason for uncertainty in transport operations [24]. Mckinnon states that most of the delays are caused by traffic congestions and the most disruptive delays are vehicle/equipment breakdown [25]. Also last minute cancellations or changes of orders and problems at loading/unloading processes belong to the themes mentioned above.

Companies must have flexibility in order to deal with uncertainties and satisfy customer needs while keeping costs reduced. Flexibility as a logistics term is defined by Zhang et al. as the capability of adapting to sudden changes in situations quickly and efficiently [26]. It is crucial to strengthen operational flexibility via responsive structures against uncertainties. Increase of asset utilization efficiency, control over vehicle fleet are possible solutions to provide transportation flexibility to deal with such uncertainties.

2.2 Cargo Loading Problem (CLP)

Container loading is a process significantly affecting the efficiency of supply chain operations: improper container loading results in increased cost and poor customer service. In today's supply chains, achieving highest volume utilization of containers not only reduces overall transportation costs and increases asset utilization, but also reduces the ecological impact by reducing environmental effects such as CO2 emissions [27, 28].

CLP can be defined as fitting 3D objects (the cargo) into larger rectangular objects (the container) as maximizing the occupied space. Naturally, cargo items cannot overlap and total volume of cargos cannot exceed the volume of the container. There are various packing procedures and algorithms reported in the literature [29]. In the *wall-building* *approach*, vertical layers are created by boxes from rear to door. In the *horizontal layer building approach*, boxes are placed as horizontal layers from container floor to top. Boxes packed in stacks and arranged on the container floor is referred to as *stack building approach*. In block building similar boxes combined and placed into container as cuboid blocks. *Guillotine cutting approach* is one in which rectangular shapes/areas created by cutting process that starts from one edge and continue all the way across to opposite edge which is filled by items in the container (Figure 2).

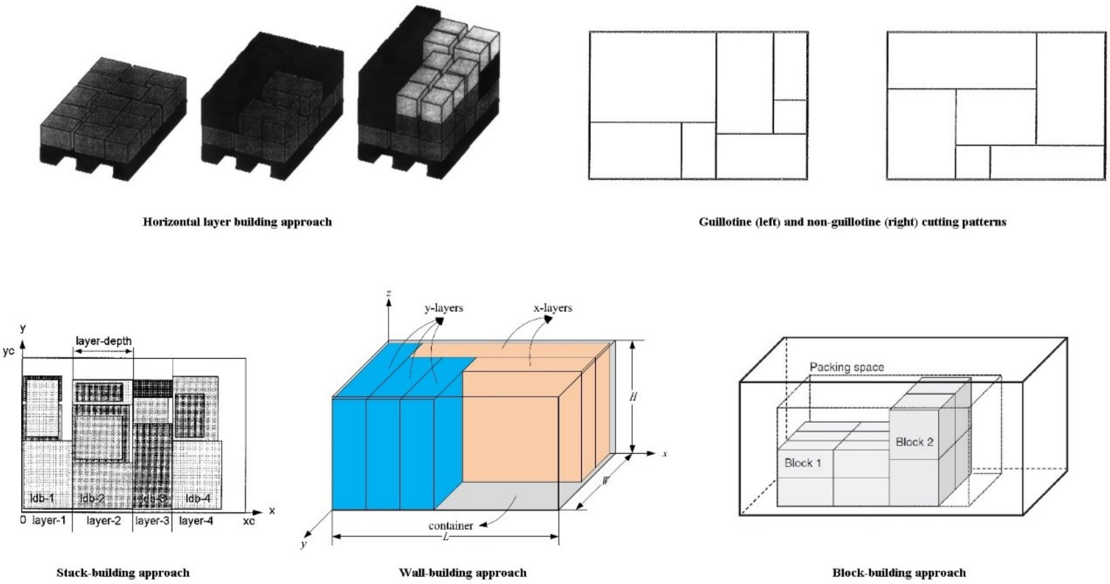


Figure 2. *Different packing approaches in literature [64-68].*

2.2.1 Problem Types

According to Wascher and Bortfeldt's comprehensive review on CLP, loading problems fall into 14 categories [30]. Packed items are considered *weakly homogeneous* cargo if most of those items' shape and size are similar, otherwise they are considered as *strongly heterogeneous* cargo. The aim of CLP is either maximizing value of loaded items when there is not enough amount of containers to carry all given set of items (maximizing output value), or minimizing cost of used containers when there is enough amount of containers to carry all given set of items (minimizing input value) [27].

2.2.2 Constraints

Loading solutions must confirm to a set of constraints to be applicable in real life. These constraints fall into five categories [30].

Container related constraints: Weight limit constraint requires that total weight of packed items may not exceed the weight limit set for the container. Weight distribution constraint for maintaining cargo balance determines that the weight of packed items is distributed on container's floor so that the cargo's center of gravity is as low as possible and close to container floor's mid-point.

Item related constraints: Loading priority constraint is set when the output value is maximized. Some of packed items must be left out as the container is not large enough to carry all items. Items can be allocated to have varying loading priorities. Orientation constraint defines acceptable orientations of a packed item. The unconstrained item has three vertical orientation and two horizontal orientations with respect to container walls, in total six different orientations. This constraint excludes some of the available orientations in order to maintain stability of the load and prevent possible damages. Stacking constraint restricts placement of boxes on top of other boxes due to insufficient load bearing strength of the boxes.

Cargo related constraints: Complete shipment constraint means that when an item placed into container, all other remaining items which belong to same order must be loaded too. The constraint is relevant in output value maximization problems with some items left behind because of limited available space. Allocation constraint in multiple container loading problems is similar to complete shipment constraint: if an item is loaded to a container, all other items that belong to same order must be loaded to the same container. In addition to these constraints, some specific item types such as chemicals and foods are not allowed to be loaded into the same container.

Positioning related constraints: Relative/absolute positioning restricts the location of items on a container either relative to one another or in absolute terms. For example, hazardous items must be placed (absolute) where they can be accessed easily and removed quickly when needed. Items that belong to same order/customer must be placed (relative) together or close to each other, which allows easier unloading of cargo.

Load related constraints: Stability constraint prevents items from being damaged when the container moves and labour from injuries during loading and unloading of cargo. Vertical stability prevents items falling on other items or container floor. Moreover, horizontal stability makes sure items do not shift when the container is moving. Complexity constraints exclude loading patterns too complicated to be understood by labour and time consuming to handle items.

In computational complexity theory, CLP is NP-hard (non-deterministic polynomial-time) problem which is very difficult to solve. In fact, very few exact solution methods exist in literature. Wascher and Bortfeldt state that most of the publications considered only 0-2 constraints simultaneously [30]. As the number of simultaneously considered constraints increases, amount of proposed solutions decreases dramatically. Figure 3 depicts the percentage of articles based on number of simultaneously considered constraint in literature.

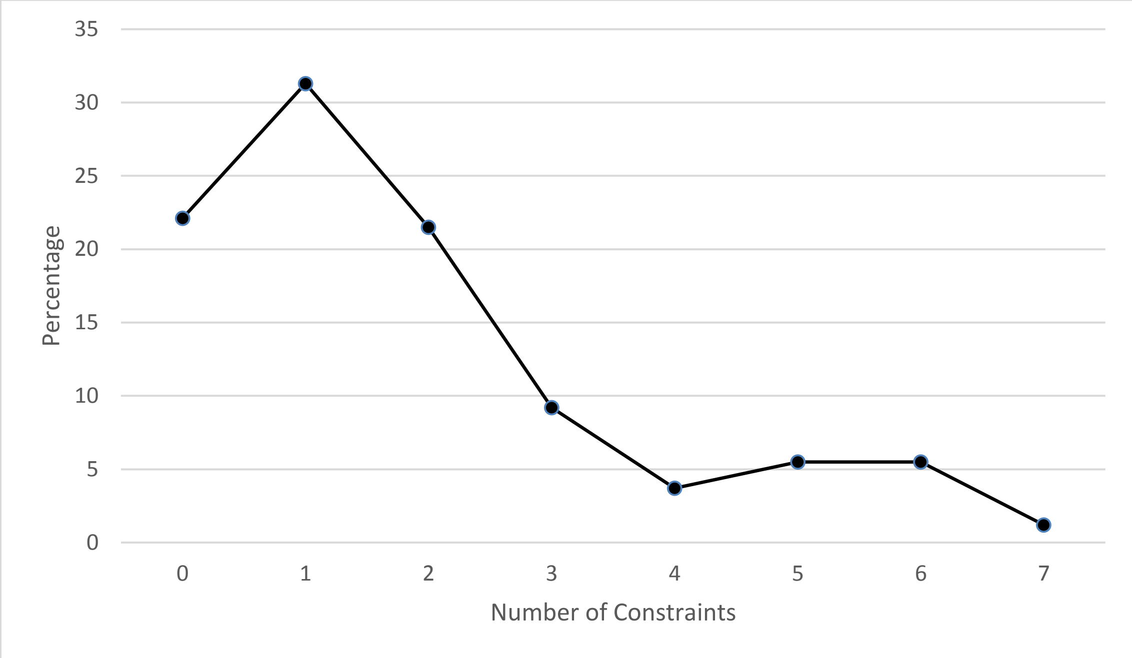


Figure 3. *Percentage of simultaneously considered number of constraints in CLP [30].*

2.2.3 Multi-stop CLP

Multi-stop CLP considers that the container has multiple orders that belong to different customers at different locations. The multi-stop loading problem combines absolute and relative positioning constraints. Items that have same destinations must be placed together in order to avoid rearranging. The delivery route poses absolute constraints: items are loaded in an order opposite to visiting locations. This is solved often with Last in First out (LIFO) policy. LIFO is implemented as a hard constraint for multi-stop CLP. Aim of LIFO is to avoid unnecessary rearrangement of items. In addition to LIFO constraint, Martinez identifies three type of constraints for multi-stop situation [31]:

- **Visibility:** An item must not be blocked by other items from the top of container and unloading door at a drop-off location. It is allowed to be blocked by other items if their destination is same with this item.
- **Reachability (touchable):** In addition to the visibility constraint, an item must be touchable by labor or handling equipment/machine at the destination.
- **Separated:** A virtual wall as a border, separates items by their drop-off location. Items are not allowed to cross the border, unless one of their sides contacts with the virtual wall.

2.2.4 Dynamic Packing Sequence

CLP aims to create complete LIFO structures for MSTL shipments to avoid the delays caused by rearrangements of irrelevant orders at each location. However, as the truck delivers the orders, the available space in the truck increases. Utilization of this available space with the addition of unplanned last minute orders may lead to better economic performance, although it also violates the LIFO policy. In this case, complete LIFO structures that generated by common CLP algorithms may not yield the optimized solution. A new algorithm can generate different partial LIFO structures as a sequencing approach and then selects the best option with the possible minimum delay.

3. WEB-BASED 3D VISUALIZATION

3.1 Overview of Visualization

Logged observations in a structured form is called data, and visualization is the process of generating visual representations of non-geometric data by images for more efficient communication and analysis [75]. A more comprehensive definition of visualization is stated by McCormick et al. [32] as

> ”*Visualization is a method of computing. It transforms the symbolic into the geometric, enabling researchers to observe their simulations and computations. Visualization offers a method for seeing the unseen. It enriches the process of scientific discovery and fosters profound and unexpected insights. In many fields it is already revolutionizing the way scientists do science*.”

The aim of the visualization process is to facilitate communication so that even non-experts understand and are able to extract the relevant information from data [33].

Visualizing massive data sets is hindered by such limitations low client computing power resources, bandwidth, latency and local storage capacity. Remote visualization approaches became a significant mitigation for these problems [34]. In this new concept, remote rendering servers take advantage of their powerful capabilities and compression algorithms to process visualization and sends the resulting images to clients. Moreover, such systems provide protection of original content against piracy or misuse by keeping the data private and by rendering at server-side [35]. However, remote visualization has limited interactivity.

Interactivity is a significant requirement for visualization technologies as the amount and complexity of data increases hugely. Users need to manipulate and explore data deeply, because the initial visualization may be too complicated for presenting and for reaching a clear and understandable conclusion. Interactivity enables user to perceive significant patterns, discover new patterns, focus on crucial parts of data, and communicate and discuss different aspects of results with other users. A simple model of interactions between users and visualization system is presented in Figure 4. Users should be able to modify visualization attributes, interact directly with visualized patterns and extract features. Furthermore, interactive visualization systems should also provide high quality user experience for non-experts [36].

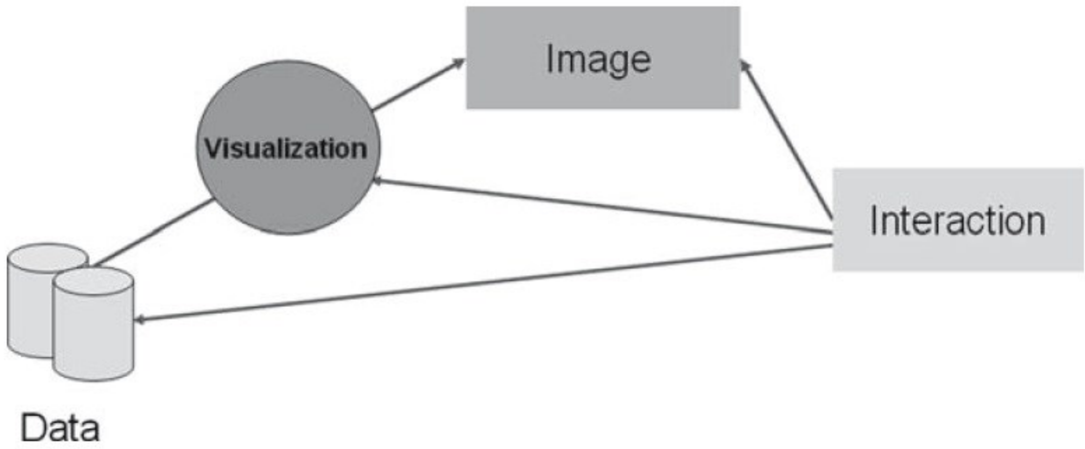


Figure 4. *Basic model of interaction between user and visualization system [36].*

Early stage visualization tools were designed for a single user. As real world problems became large and complicated, the complexity and size of data grew beyond the handling capacity of a single analyst. Feasible solution for such systems can only be attained by numerous experts working collaboratively in a multidisciplinary manner. Collaborative visualization, which is a form of interactive visualization, enhances co-located or distributed users working together on a problem. Moreover, it creates a collaborative learning environment where expert and non-expert users contribute to the same project and improve their understanding and skills. Another advantage of collaborative visualization is to pool remote computing resources together, so that an increased number of users benefit from the system [37].

Recently, due to the advent of browser technologies, numerous web-based visualization applications have emerged. Web-based visualization is an approach to remote visualization with significant advantages by providing collaborative, interactive and ubiquitous visualization platforms for various devices. Such an approach reduces the system maintenance work by allowing deployment of visualization tools from a single source code base and by removing the need for extensive software installations. Thus, users may focus on the main problem without being distracted by secondary system setup problems [34].

Initial techniques for web-based visualization were based browser plugins, such as VRML (Virtual Reality Modeling Language) and Java applets. VRML is a standard text file format to display 3D graphics by specifying polygonal shape components such as edges, faces and vertices as well as their attributes such as color or transparency [38]. Java is a programming language that allows user to run it on all Java supported platforms without compiling. Java applets can be downloaded via web and run at users’ browsers locally [39]. User interface that enables communication between client and visualization server can be designed by Java while VRML generates the visualization result for further use. First version of VRML allowed only viewing the visualization and navigating in a 3D world, but later versions allow also modification of data; rendering at the client-side. There were problems for both client and server-side rendering. Client-side rendering needs enough memory (which may not be available for all users) for storing data and has a long waiting time for both downloading and rendering large and complex 3D models. Therefore, usability was achieved by decreasing the level of details (LOD) of the 3D world and limiting the amount of data about the content sent to user [40]. These limitations in these web technologies led developers to adopt mostly server-side rendering approaches to overcome the poor performance. However, visualization processes continued to suffer from bandwidth and latency issues, which reduce the interactivity of visualizations [41].

The trend of server-side rendering in web-based visualization has changed with the emergence of WebGL and HTML5 technologies that exploit the power of GPU's. Furthermore, improved JavaScript performance, data encoding and transferring techniques supported high performance rendering at the client-side. One of the most significant advantages of these modern standards is that the rendering process at the client-side needs not any plugins. Moreover, improved interactivity with minimized latency is maintained by these contemporary technologies with the client-side rendering [34].

3.2 Web-based Visualization Application Domains

This sub-section reviews application domains in web-based visualization.

Particle Data Visualization: Particle based data has been for many years a significant subject of visualization research on how to present the multitude of metaphors by defining particles with attributes. Interactivity is preferred in such visualizations so that the user can freely explore data and visualization parameters. In addition to this, the interactivity requirement suggests GPUs for interactive visualizations as GPUs support algorithms enabling high performance rendering in real time on contemporary hardware. As GPU based rendering techniques have advanced, they have provided increasingly better image quality and rendering performance. Early developed tools such as AtomEye [42] and VMD [43] provided comprehensive functionality but they were designed before programmable GPUs became widely available, and thus the performance of these tools on contemporary GPUs is not optimal [44]. Another particle data visualization tool JSMol required installing a browser plugin [45]. NGL Viewer [46] was another tool to visualize molecular structures in browser by WebGL. On the other hand, a client-server approach for particle data visualization is presented by Mwalongo et al. [47] and realized visualization at both client and server-side.

Volume Visualization: Volume data is a 3D structure without clear geometrical shape such as surfaces or edges. Volume visualization aims to extract such information by modeling, rendering and manipulating the data [48]. It is mostly used in medical applications to create 2D sections from 3D data. Congote et al. [49], visualize weather radar scans and medical imaging data by GPU-based ray marching technique in WebGL. This research helped to extend X3D standard in web-based volume visualization. Hou et al. [50], visualized medical volumes with an image-based technique that accelerated rendering at server-side with a GPU, but they could not maintain smooth interaction because of network latency. Jacinto et al. [51], proposed a client-server framework for a web-accessible image visualization of medical applications. HTML5 and WebGL technologies were used at the client-side and image processing algorithms were deployed at the server-side.

Geospatial Visualization: Contemporary web-based visualization of geospatial and spatio-temporal data, including raster images is based on HTML5 and WebGL technologies. Figueiredo et al. [52] developed a system to visualize high-resolution mesh data of geomorphological structures. Jenny et al. [53] presented a new technique for raster images that with a hardware accelerated projection enables users to scale a map and control its center.

3.3 Technologies Used in 3D Web-Based Visualization

Many technologies exist for web-based visualization. The main technologies for visualization on web browsers are reviewed in this section.

HTML5 is the latest major version of the standard markup language for creating web pages and web applications. It combines the set of techniques consisting of JavaScript, CSS and HTML5 and aims at a rich web experience independent of plug-ins [54]. HTML5 has a new graphic container element ("Canvas") with JavaScript based on API for drawing images. Another aim of HTML5 is to decrease the workload of developers and to provide human readable content as well as improving multimedia and graphics [55].

WebSocket is a technology that enables bi-directional, full-duplex communication between client and server. Initially, it was referenced as a part of HTML5 specification but later has been transferred to a separate standard's document. Unlike other solutions, such as unscalable polling and long-polling for full-duplex communication, WebSocket has reduced latency and unnecessary network traffic [56]. Implementation of

WebSockets enables real-time communication between client/s-server and supporting Canvas for developing interactive graphics applications [57].

WebGL was initially designed by a non-profit technology consortium Khronos Group in 2009. It is a cross-platform, royalty-free and graphics related JavaScript API based on OpenGL ES 2.0 which is a simplified version of the standard graphics library OpenGL with the purpose of creating accelerated 3D graphics within compatible web browsers [58].

WebGL graphics is rendered with Graphical Processing Unit (GPU) on client-side. This hardware accelerated operation moves some of the central processing unit (CPU) load to GPU, and therefore the system performance is not affected by other applications, and the rendering performance is limited only by GPU memory bandwidth. Hence, this technology has that significant benefit that the rendering processes on browsers do not require plugins or additional software [59, 60].

Various JavaScript 3D libraries that have been built upon HTML5 canvas and scalable vector graphics (SVG), abstract WebGL to enable developers to work on higher code level. Developers have two popular ways of working with WebGL. First, they may directly use WebGL underlying language with HTML5 canvas tag and insert the rendered graphics into canvas tag by calling the property in JavaScript. Second, they may use WebGL frameworks/libraries to develop WebGL content quickly and easily. For instance, Three.js is one of the most popular 3D JavaScript libraries for WebGL and it’s built-in functions can be called directly in JavaScript [61]. Therefore, the performance of js engines and OpenGL driver dependencies with GPU capacity are the limitations of WebGL technology.

Three.js has major advantages compared to many other 3D libraries (Pex, ClayGl, AwayJs, Litescene etc. [62]). Three.js contains many features, such as effects, scenes, cameras, animation, lights, materials, shaders, objects, geometry, export and import files. In addition to this, it has been actively developed, and support to large and active community by developers is provided. The documentation of the library has clear explanations with examples. It supports most of industry standard file formats such as obj, mtl, fbx, 3ds, gltf, collada, babylon, playcanvas, stl, vrml, and draco.

3.4 3D Visualization

Visualization graphics in Three.js requires a scene, camera and renderer. In Three.js library, the scene is an object where 3D graphs, lights and cameras are placed into. Camera is defined after the scene is set up. The camera object has several properties that creates viewing frustum which is the visible region (pyramid) of the modeled world (Figure 5). The Field of View (FOV) -property is an angle in y direction of the pyramid. Aspect ratio -property is the width divided by the height. It is used to determine the width and height of the near, and far planes of the frustum. The last two properties define distances of near and far clipping plane. Objects are rendered between these two planes. A point light source is also added to the scene in order to illuminate the environment and cast shadows. Then, renderer (WebGL) is added and size of the output canvas is set. It determines whether a full screen or a smaller area is rendered. To start rendering process, a rendering loop must be created. First, the method requestAnimationFrame() creates an infinite loop of updating objects, then renderer is called to render using the scene and the camera defined earlier (Program 1).

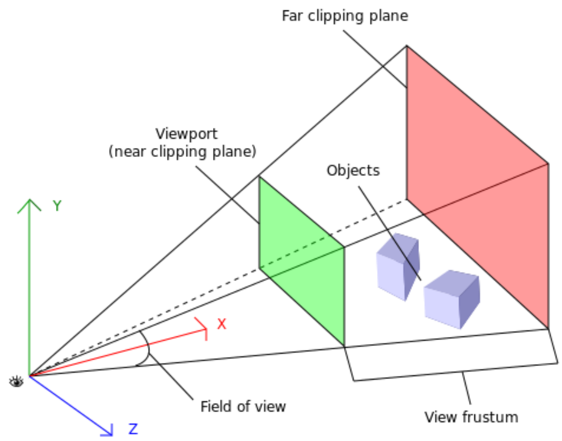


Figure 5. *View frustrum [69].*

```
var scene = new THREE.Scene()
var camera = new THREE.PerspectiveCamera( FOV, aspect ratio,
near distance, far distance)
var renderer = new THREE.WebGLRenderer()
renderer.setSize(Width, Height);
var light= new THREE.PointLight(0xffffff, 1, 0);
scene.add(light);
function animate() {
      requestAnimationFrame( animate );
      renderer.render( scene, camera );
}
animate();
```

Program 1. *Setting up a scene and animation in Three.js.*

4. SYSTEM DEVELOPMENT

This thesis develops a visualization system for cargo loading and unloading, for CLP. The system is mainly designed for suppliers to handle last minute events, container loading, test loading patterns, training, and to support collaboration with partners. It can also be used to present relevant loading plans to customers to enhance customer satisfaction in terms of service quality and transparency. The visualization system has three main capabilities: managing orders, animating loading and unloading tasks, and providing a real-time collaborative work environment. User can add or remove an order from a selected truck, animate the resulting loading, unloading and reconfiguration tasks, and use collaborative work area for practicing consolidation plans or loading/unloading cargo manually. The browser user interface of the visualization system is shown in Figure 6. The user selects a function from menu at the upper left side (Manage Order, Loading Plans or Training) and opens related sub menus in the green area below. Information area at the lower left side at the web page displays information (customer name, order origin, order destination, order ID, item ID, and status) of highlighted object. The use case diagram is depicted in Figure 7.

HTML5 is selected to create a web-page since it combines the set of techniques consisting of JavaScript and CSS. WebSocket is selected to achieve real-time communication between clients and the server. WebGL is selected to enable client-side rendering which improves the interactivity of the application.

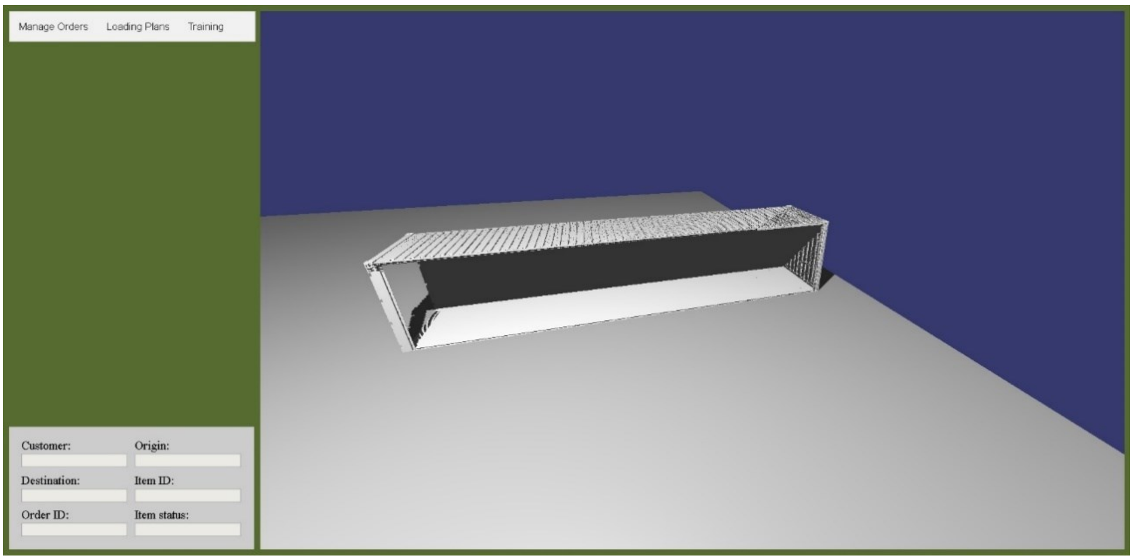


Figure 6. *Browser user interface of the visualization system*

4.1 System Architecture

Simple client-server architecture is commonly called as a two-tier architecture. Tier is a physical structuring mechanism with comprising group of technologies that provides services to a requestor. Client-side contains presentation layer which displays information, whereas server-side contains the application layer, which has the functional business logic, and the data layer, where the data is stored and managed. In this thesis, web-based visualization system is designed based on a three-tier architecture with a client, server and database model. In the three-tier architecture, business and data layer are separated. Figure 8 shows the system architecture.

Client-side (presentation layer) contains the components required to create the user interface, to communicate with server and to display retrieved information. HTML5 and CSS are applied to create a web page. A JavaScript framework Angular.js extends HTML5 features to create a dynamic web page. When user changes the input data, the view is automatically updated by the data binding feature of the framework. For example, the user may view a truck list, which contains the available trucks for transportation depending on input pick-up and drop-off locations or incoming trucks of a selected location. Three.js is used for visualizing 3D graphics on the browser. Tween.js is used for creating animations.

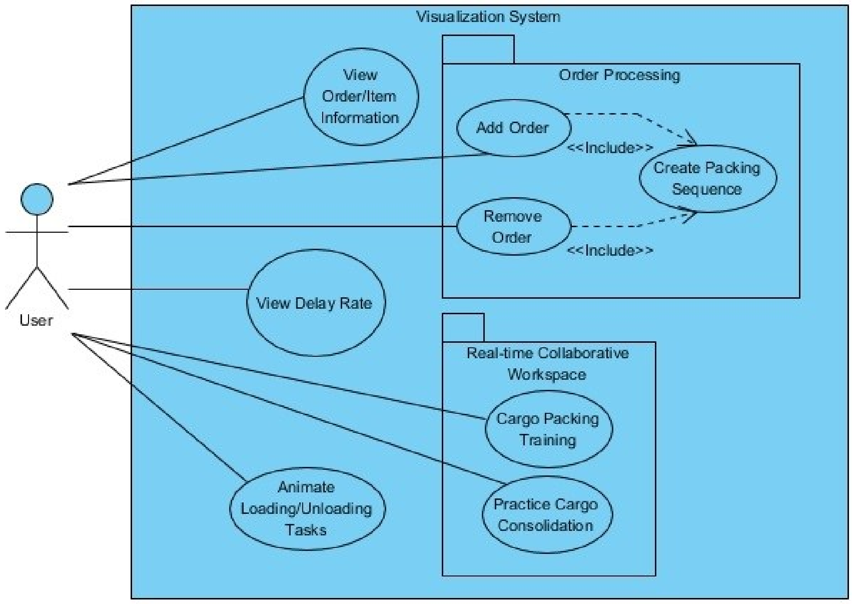


Figure 7. *Use Case Diagram*

Communication between the server and clients is maintained by Socket.io. Socket.io is a JavaScript library that abstracts WebSocket connection. However, HTTP protocol provides unidirectional communication in which request is initiated by a client and response is processed by the server. On the other hand, broadcasting feature of

Socket.io allows all the connected clients to receive the message that was sent by a client. Therefore, clients can collaboratively work on same problem in real-time.

Web server is created by using Express.js framework at the server-side. The server contains system visualization logic and the DPSA. Requested data is retrieved from database and processed, then send to related clients for visualization.

MongoDB stores the database file which contains vehicle fleet information at the database tier.

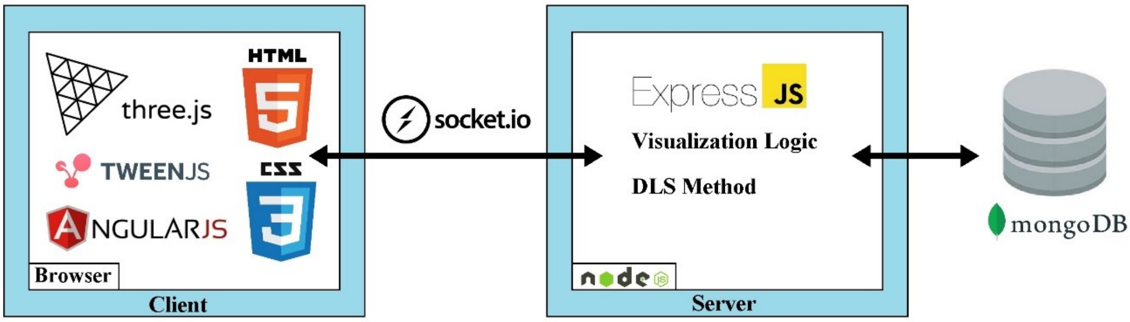


Figure 8. *System Architecture.*

4.2 Database

A database is implemented to the visualization system in order to store and manipulate the visualized data. MongoDB is an object oriented NOSQL database that is simple to use, dynamic and provides high performance, availability and scalability. It uses JSON (JavaScript Object Notation) human and machine-readable files to store data. JSON is a text format for storing and transferring data. The syntax is derived from JavaScript object notation syntax. In this syntax, each data (value) is associated to a key (name) and it is stored as key-value pairs.

The database stores the vehicle fleet information. Both vehicle fleet information and data produced by DPSA are in JSON format which is convenient to work in JavaScript. Fleet has a set of trucks with optimized routes and orders as a predefined condition. This information is the basis for enabling utilization of ongoing trucks for last minute calls so that sending a new truck is avoided. Each truck has a truck ID, an item placing coordinates both inside the truck and at locations, route properties, order and item information (Program 2).

The properties “positions_in_truck” and “positions_at_location” define the item placement positions inside a truck and at locations, respectively. These positions are created according to the item's dimension, and the 3D Cartesian coordinates are stored as a set of arrays.

The property "route" contains the locations that a truck visits during the delivery. Orders have specific positions inside the truck when arriving at a location ("initial_pos") and leaving ("final_pos") the location. Other orders which are at a location are kept in "out" property.

The property "options" is an empty array which stores the alternative packing sequence plans during the search for the best packing sequence. Array elements are deleted once the best option has been found.

In the visualization process, 3D objects with related information within selected truck/container are displayed. Each customer may own one or more orders, an order owns item/s and each 3D object corresponds to an item. The relationship between customer, order and item is shown in Figure 9. Orders are placed into their drop-off/pick-up locations in "route" property in the database. The object constructor function creates object types for item and order objects (Program 3-4).

```
{
  "truck1":{
    "truckid":1,
//item placing positions inside a truck
    "positions_in_truck":[ [x1,y1,z1], [x2,y2,z2], ... ],
//item placing positions outside a truck
    "positions_at_location":[ [x4,y4,z4], ... ],
    "route":{
      "1st_stop":{
//Orders' initial positions inside the truck
        "initial_pos":[{Order1}, {Order2} ],
//Orders' final positions inside the truck
        "final_pos":[ {Order1}, {Order2}, {Order3} ],
//Orders' positions at the location
        "out":[ {Order3} ],
// Order placement sequence variants
        "options":[ ]
      },
      "2nd_stop":{
        "in_initial":[ ... ],
        "in_final":[ ... ],
        "out":[ ... ],
        "options":[ ]
      },
      ...
    }
  }
  "truck2":{
    ...
  }
}
```

Program 2. *Database structure.*

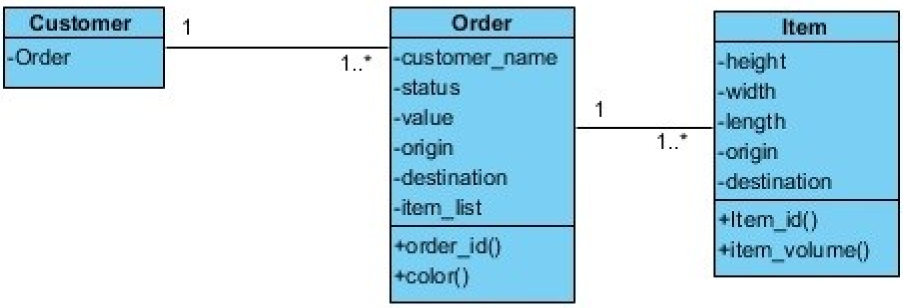


Figure 9. *Class diagram of customer, order and item.*

```
function Item(height,width,length,pick_up,drop_off){
    this.id = new Date().getTime() + Math.random();
    this.x = height;
    this.y = width;
    this.z = length;
    this.origin=pick_up;
    this.destination=drop_off;
    this.volume = height*width*length;
}
```

Program 3. *Item object constructor.*

```
function Order(pick_up, drop_off, customer) {
   this.order_id = new Date().getTime() + Math.random();
   this.origin = pick_up; //pick-up location
   this.destination = drop_off;  //drop-off location
   this.status = "out";          //order status
   this.value = 0; // order reconfiguration amount
   this.customer = customer;
   this.items_list = [];  //has Item objects
   this.color='#'+Math.floor(Math.random()*16777216).toString(16);
}
```

Program 4. *Order object constructor.*

4.3 Dynamic Packing Sequence Algorithm

This Section describes an algorithm developed in this thesis to improve on the common LIFO packaging approach.

LIFO policy is considered as a hard constraint in MSTL shipments. A set of given items are placed into a container according to their drop-off locations which prevents rearrangement of the cargo at stops. Although it is an optimized packing sequence for a given set of items, volume of unused space of the container increases as the number of stops increases, see Figure 10a. Utilizing this space may improve the competitiveness, if the total travel time including the delay resulting from cargo additions and rearrangement stays within the initial planned travel time. However, addition of new items is not considered as a degree of freedom in common optimization approaches as it violates LIFO policy. Depending on the volume of a new order (purple), a different sequencing approach, such as partial LIFO structures instead of overall LIFO policy, may yield better result (Figure 10b-c). Figure 10c displays a complete LIFO structure in which order “yellow” and “blue” are rearranged. On the other hand, Figure 10b displays the two different partial LIFO structures where the new order and the order “blue” are rearranged during the transportation. The packing sequence which has the minimum rearranged volume is the best solution for the new transportation plan.

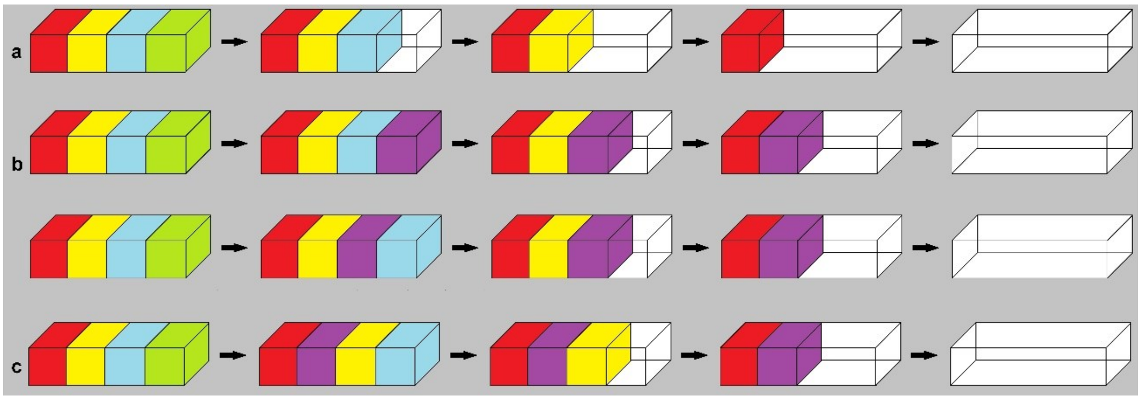


Figure 10. *a) Initial transportation plan. b) Two different partial LIFO structures. c) Complete LIFO structure.*

The algorithm generates different packing sequences and determines the optimal sequence dynamically for each location in truck’s route when a new order is received. Unlike common LIFO approach, the algorithm creates partial LIFO structures by using permutation of selected subset of orders and appending it to non-selected orders.

The procedure (Program 5) starts with matching the new order with a suitable truck. Then, the algorithm enters its iterative phase that starts from pick-up location of the new order to final stop of the truck. In each iteration (location), orders are processed and all possible packing sequences for that location are generated, then next iteration is independently repeated for each generated packing sequence option. Order processing direction is from front side of the container to rear side. Orders to be delivered at that destination point and blocking items in the way are removed from the truck. As a result, some orders left in the truck, some orders are removed for rearrangement and orders to be picked-up are waiting for loading outside. Packing sequence generation starts at this point. Since the orders removed for rearrangement and the new orders are outside, they can be loaded into the container in LIFO order. The loading sequence is created by appending this structure to the other structure inside the truck. Depending on the new order’s drop-off location the resulting structure may not be a LIFO order. In this case, an order closest to the door is removed from the container and another packing sequence is generated. This process continues until the structure is a LIFO order. As a result, one or several loading sequences are generated for this location.

The algorithm estimates the delay for each generated packing sequence. Packing time is affected by many factors such as the distance between the truck and orders at the location, payload of material handling equipment and item positioning inside the truck. In order to estimate the delay caused by packing of a new order and other necessary rearrangements, effect of the other factors are minimized by applying these conditions: it is assumed that locations have order pick-up, drop-off and rearrangement zones in which each zone has same distance from the docked truck. At each zone, items that will be located into front side of a truck, stay closer to the truck than other items that will be located into rear side of the truck. Total volume of set of items carried by a worker or material handling equipment for each transportation (loading and unloading) process is same. Once these conditions eliminate the effect of distance, speed and payload parameters, changes in time of packing process is mainly affected by handled volume. As a result, the delay is proportional to the handled volume and it can be easily compared with total packing time by observing the volume of processed orders.

```
for location
    for orders
        Flag=0
        if item is at destination, remove and flag=1
        if item is not at destination and flag=0, no action
        if item is not at destination and flag=1, remove
        for reconfiguration
    for packing sequence generation
        Create LIFO structures for orders outside, then
        append it to orders inside truck
            if whole structure != LIFO
            Remove one order from truck and repeat
            sequence generation
```

Program 5. *Pseudo code of generating packing sequences.*

Volume of an item V_I is simply calculated by multiplying its dimensions. The volume of an order $V_O$is the sum of the volumes of its items

$$V_I = WxHxL \quad (3.1)$$

$$V_O = \sum_{j=1}^{J} V_{I\,j} \quad (3.2)$$

where $j = 1,2,\dots,\mathrm{J}$ is number of items.

The transportation operation starts with loading of orders at the first stop and finishes after all the orders are removed from the truck at the last stop. The total time T_T spent in loading and unloading tasks in the initial plan is estimated be proportional to twice the volume of initial orders V_{IN}

$$V_{IN} = \sum_{l=1}^{L} \sum_{n=1}^{N} V_{O\,ln} \quad (3.3)$$

$$T_T \propto 2V_{IN} \quad (3.4)$$

where $l = 1,2,\dots,\mathrm{L}$ is indexing the stops and $n = 1,2,\dots,\mathrm{N}$ is indexing the orders.

The sum of volume of all orders inside the container must not exceed the container's usable volume at each stop

$$\sum_{k=1}^{K} V_{O_k} \leq V_{Container} \quad (3.5)$$

where $k = 1,2, \dots, \mathrm{K}$ is number of orders in container.

When a new order V_{NO} added to the system, the delay T_N caused by both loading and unloading tasks

$$V_N = \sum_{m=1}^{M} V_{NO_m} \quad (3.6)$$

$$T_N \propto 2V_N \quad (3.7)$$

where $m = 1,2, \dots, \mathrm{M}$ is number of new orders accepted during the transportation .

The total rearranged volume V_R of a packing sequence is calculated by multiplying the rearrangement amount k during the transportation with volume of the rearranged order V_{OR} then summing the results of all rearranged orders. In a rearrangement task, an order is moved out and then put back into container, which means the rearranged volume is twice of the actual volume of the order, and therefore the result must be multiplied by two in order to obtain the delay of total rearranged volume of a packing sequence option T_R. The algorithm selects the best option which cause the minimum delay among all the generated packing sequence options

$$min(V_R) = \sum_{p=1}^{P} k_p * V_{OR_p} \quad (3.8)$$

$$T_R \propto 2min(V_R) \quad (3.9)$$

where $p = 1,2, \dots, \mathrm{P}$ is number of rearranged orders.

In the case of existence of several minimum delay results, the packing sequence options are compared by cumulative delay factor T_c. For example, one packing sequence causes a half an hour delay at the second stop and another half an hour delay at the fourth stop because of rearrangement of the orders. The truck arrives to the third and then the fourth stop a half an hour late. The truck arrives to fifth and other remaining stops an hour late. The total delay is an hour at the end of the transportation operation. Another packing sequence causes an hour delay at the second stop only. This time, the truck arrives to the third and other remaining stops an hour late. Although the two packing sequences cause a total one hour delay at the end of transportation operation, the third and fourth stops suffer less delay in the first packing sequence option. Therefore, estimating the cumulative delay allows to find better packing sequence option.

$$T_c \propto \sum_{l=1}^{L} \sum_{p=1}^{P} (L-l) x V_{OR_p} \quad (3.10)$$

From (3.3), (3.6) and (3.8) the ratio of total delay of a selected packing sequence option to initial packing time is defined in (3.11). Assuming the total packing time for initial condition (T_T) is already known, additional delay of new packing sequence to the whole transportation operation can be estimated. Delay Rate R can be written as:

$$R = \frac{\sum_{l=1}^{L}\sum_{n=1}^{N} V_{O_{ln}} + \min(\sum_{p=1}^{P} k_p * V_{OR_p}) + \sum_{m=1}^{M} V_{NO_m} - (\sum_{l=1}^{L}\sum_{n=1}^{N} V_{O_{ln}} + \min(\sum_{p=1}^{P} k_p * V_{OR_p})_{initial})}{\sum_{l=1}^{L}\sum_{n=1}^{N} V_{O_{ln}} + \min(\sum_{p=1}^{P} k_p * V_{OR_p})_{initial}}$$

$$\propto \frac{T_R + T_N - T_{R_{initial}}}{T_T + T_{R_{initial}}} \qquad (3.11)$$

Delay rate is used to determine the total packing time of the selected sequence.

4.4 Visualization Process

The visualization system visualizes the contents of the selected truck. This section describes the interactive functions, object drawing and object positioning processes of the visualization system.

4.4.1 3D Object Drawing

The 3D rectangular objects of the cargo are created with the “BoxGeometry” object. This object contains all vertices and faces of a rectangular. After the geometry is set, material is defined by the “MeshBasicMaterial” object. This is a basic material property. Many material and geometry types exist in Three.js. The “Mesh” object applies the defined material onto the created geometry. Furthermore, 3D objects must contain other information such as IDs, positions at each location etc. Additional properties can be defined to this mesh. When the 3D object is created, it must be added to scene by calling scene.add(), in order to render it (Program 6).

```
Object3D(itemid, height, width, length, status, position_x ){
    var geometry = new THREE.BoxGeometry(height, width,length);
    var material1 = new THREE.MeshBasicMaterial({color});
    var cube = new THREE.Mesh(geometry, material);
    cube.cubeid = itemid;
    cube.position.x = position_x;
    scene.add(cube);
}
```

Program 6. *Adding 3D object into a scene.*

4.4.2 Object Positioning

Visualization system displays items at pick-up (red), drop-off (green), rearrange (yellow) and container (blue) zones (Figure 11). In order to create realistic animations such as non-floating items, algorithms are prepared for the zones and a container for item placements points.

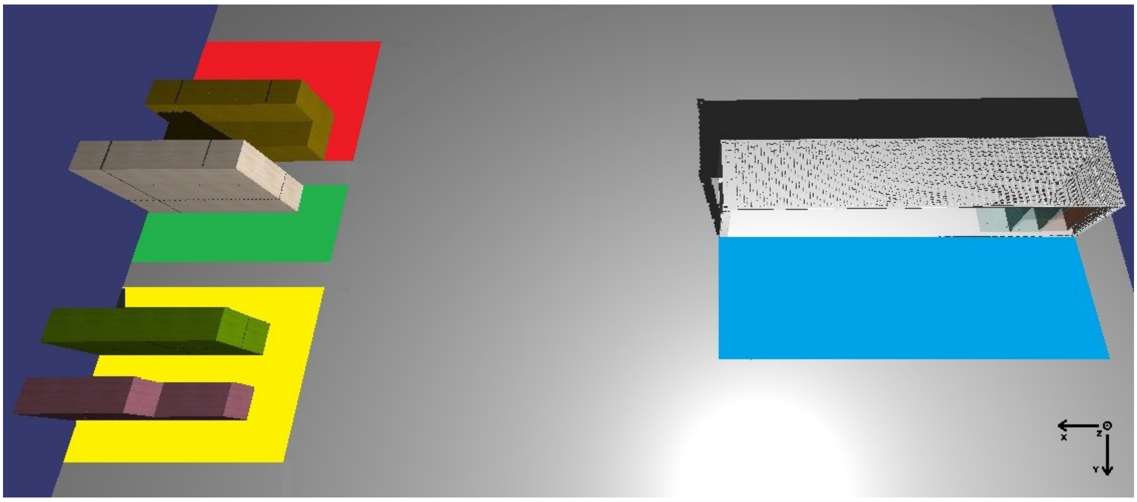


Figure 11. *Visualization area zones.*

Since CLP considered here is IIPP, object placement points can be identified easily after the object length is determined. Wall building is applied for generating placement points. Placement starts from lower back left corner of the container and other items are placed next to one another, along the width of the container. After the first layer is created, the next item is placed on the first item and the same procedure is repeated until stacked items have reached the top. As a result, the first vertical wall is created, and the next item’s location is in front of the first item. Other vertical walls are generated along the length of the container. The following example code generates the placement points inside a truck (Program 7).

```
var positions=[ ];
for(var i = (item_length/2) ; i <= (container_length –
   (item_length/2) ) ; i+= item_length)
    for(var k = (item_length/2) ; k <= (container_height –
       (item_length/2) ) ; k+= item_length){
        for(var j = (item_length/2) ; j <= (container_width –
            (item_length/2) ) ; j+= item_length){
            positions.push([i,j,k]);
        }
    }
}
```

Program 7. *Generating positions in truck.*

When an order arrives at a destination, it is removed from the truck and placed to the drop-off zone. Placement points follow -x, +z and +y axes directions respectively. Similarly, the placement points for rearranged items follow the same directions in the rearrange zone. However, the algorithm is called separately for each rearranged order and therefore each order is grouped in itself (Program 8).

```
var out= [ ];
for( var j = item_length/2 ; j <= max_value_y ; j+= item_length){
    for( var k = item_length/2 ; k <= max_value_z ; k+=
    item_length){
        for(var i = max_value_x ; i >= min_value_x ; i-=
        item_length  ){
            out.unshift([i+offset,j+offset,k]);
        }
    }
}
```

Program 8. *Generating positions in drop-off zone.*

Placement of orders at the pick-up zone follows different directions. Each order is grouped separately, and items are loaded to truck by following +x, -y and –z axes directions, respectively (Program 9).

```
var wait=[ ];
for k =(item_length/2 ; k <= max_value_z  ; k+= item_length ){
    for( var j = max_value_y ; j >= item_length/2  ; j-=
    item_length ){
        for(var i = max_value_x ; i >= min_value_x ; i-=
        item_length){
            wait.push([i+offset,-j-offset,k]);
        }
    }
}
```

Program 9. *Generating positions pick-up zone.*

4.4.3 Interactive Functions

The visualization system contains several interactive functions (Figure 12).

Item move: User can drag an object by first clicking on it. Assigned axis helps users to move an item along specific axes or planes. The axes can be toggled on or off.

Item highlight: Whenever mouse cursor hovers on an object, that object is highlighted.

Display information: Information box displays highlighted object’s information.

Switch color (green, red, yellow): In order to distinguish the orders, a unique color is assigned to each order when they are created. This function switches color to green for orders that arrived to destination, yellow for rearranged orders and red for orders at pick-up zone. It helps workers to distinguish orders easily for different tasks.

Transparency & Opacity: Items which have not any task (i.e. load, unload or rearrange) at a location are transparent in “Loading Plans”. Moreover, when a user adds a last minute order into a truck at “New Order”, visualization system displays items of the last minute order as opaque objects and other items as transparent to present placement of the last minute order clearly.

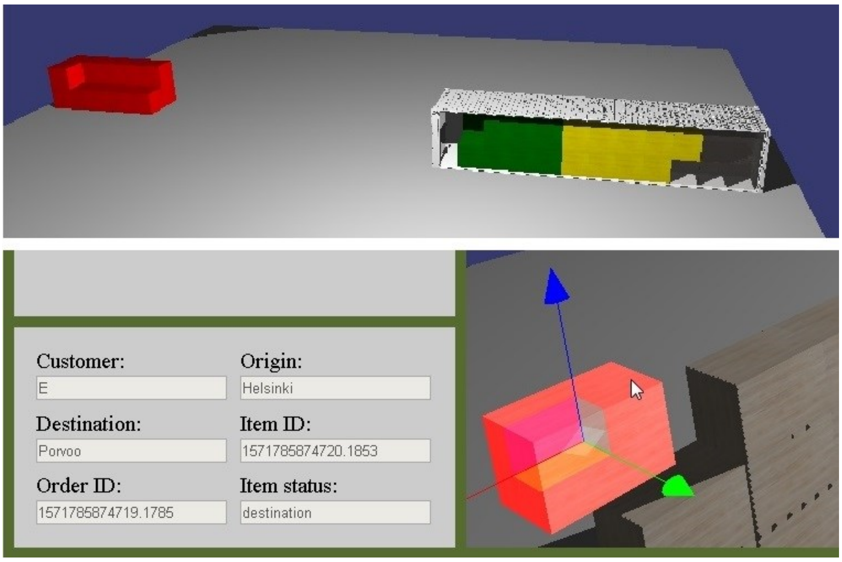


Figure 12. *Visualization features.*

4.5 System Functions/Capabilities

This section describes system capabilities. UML Sequence diagrams are used for modeling the interactions between server and clients.

New Order: When a user adds a new (last minute) order, customer name, order origin and destination locations information are sent to the server. The server analyzes the input and then creates a list of available trucks. Once the client-side receives the list, the user selects a truck and the input order details, such as the amount and item dimensions. The server runs the DPSA to place the new order into the truck and create the best loading patterns for each locations of truck’s route. The resulting data is sent to the client-side to be rendered and visualized. The sequence diagram for adding the new order is shown in Figure 13.

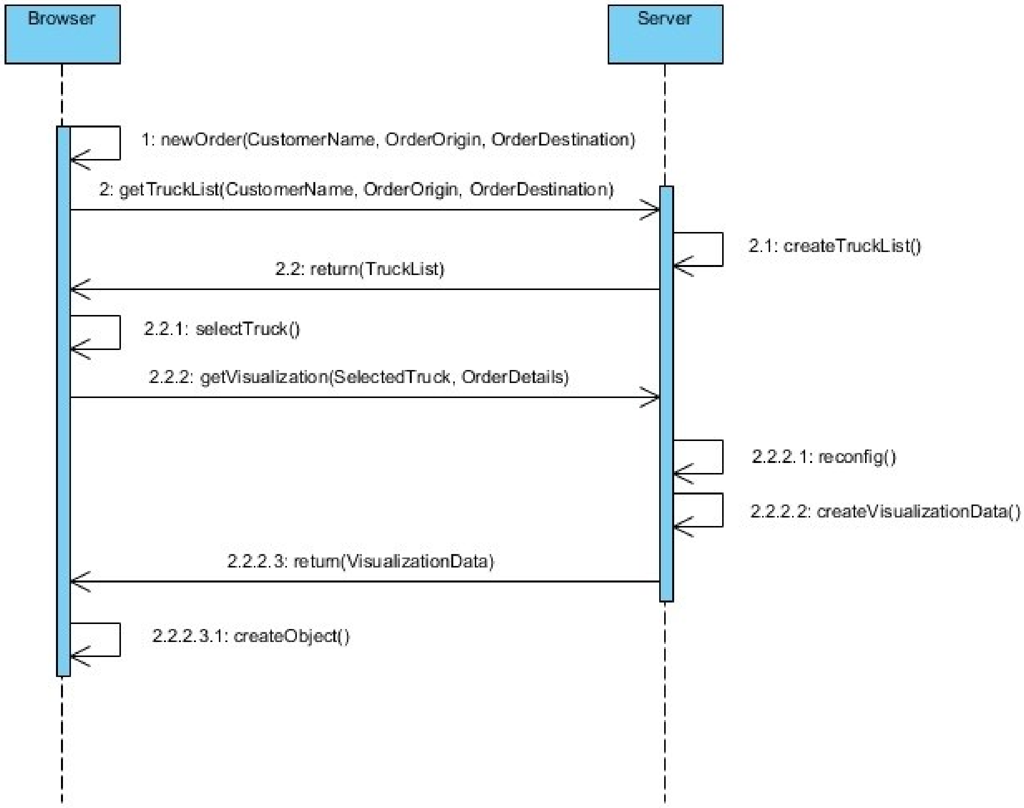


Figure 13. *Sequence diagram of adding a new order.*

Remove Order: As a last-minute event, an order might be cancelled and need to be removed from the transportation plan. User selects his/her own location to view the list of orders and finds the order which is to be removed. After the remove request is sent, the server removes the order and runs the DPSA to update loading plans. The sequence diagram for removing an order is shown in Figure 14.

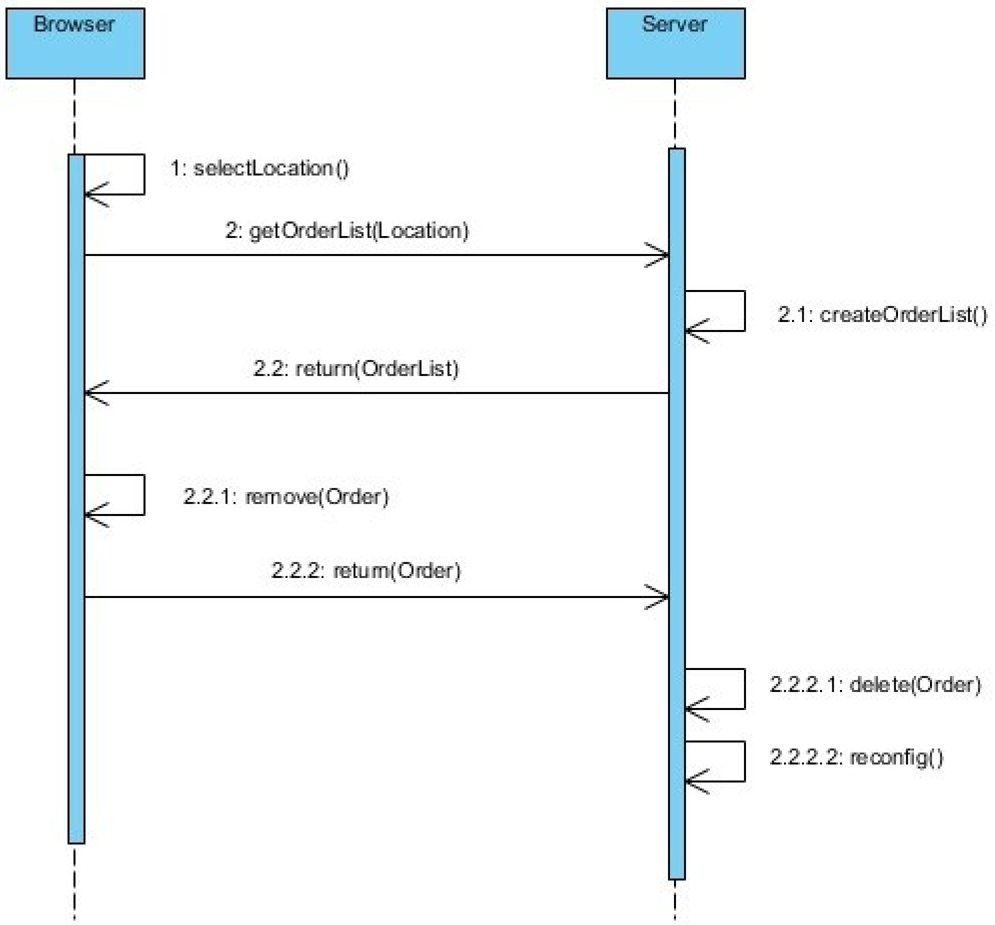


Figure 14. *Sequence diagram of removing an order.*

Animation: The visualization system animates loading, unloading and rearrangement tasks at each location. Each order has a unique color and all items in the same order have same color. Orders move between pick-up, drop-off, rearrange and container zones, depending on the tasks. User selects a location and receives a truck list from the server. Then, the user selects a truck from the list and the server sends visualization data, which displays the orders at their zones. Animate button starts the animation and shows how items should be loaded, unloaded and rearranged. The sequence diagram for animation is shown in Figure 15.

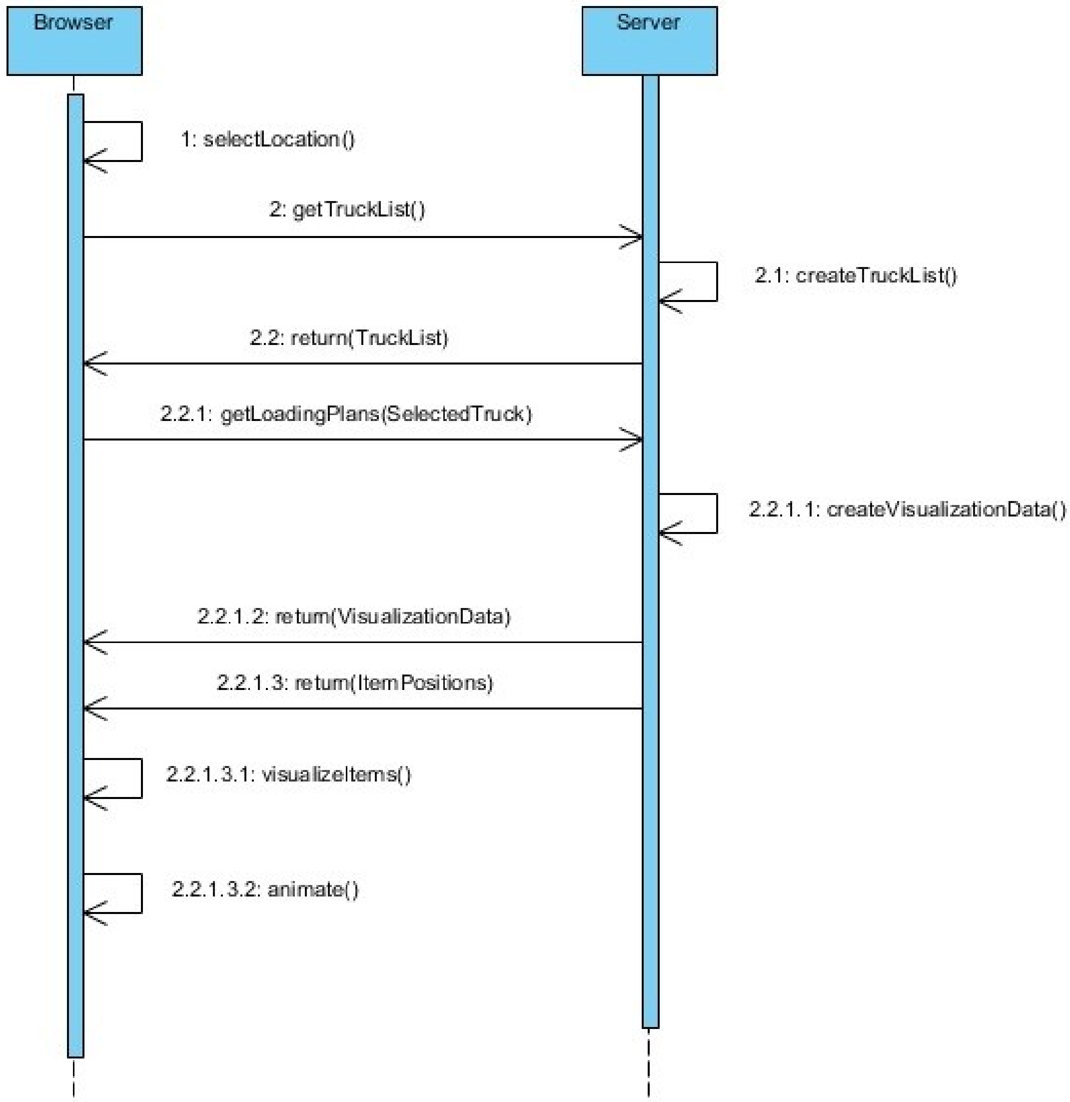


Figure 15. *Sequence diagram of animating tasks.*

Real-time Collaborative Workspace: Several users can interact in same workspace and actions are displayed in real-time. In this thesis, only the add object feature is implemented as a test feature for collaborative workspace. A user defines a 3D object and adds it to the workspace. As soon as the object is added, all other users can view it. Each user can interact and move this object and other users can view the changes. Figures 17 and 18 show the sequence diagrams for adding an object and moving an object for collaborative use, respectively. To prevent conflicts, red and green flags are added to the system. A user becomes active by checking the checkbox and a red flag is displayed on the workspace to inform other users. Red flag also disables all functions in the menu. Therefore, inactive users must wait the active user to return inactive status before they can become active. Unchecking the checkbox returns a green flag and enables checkbox for state transition (Figure 16).

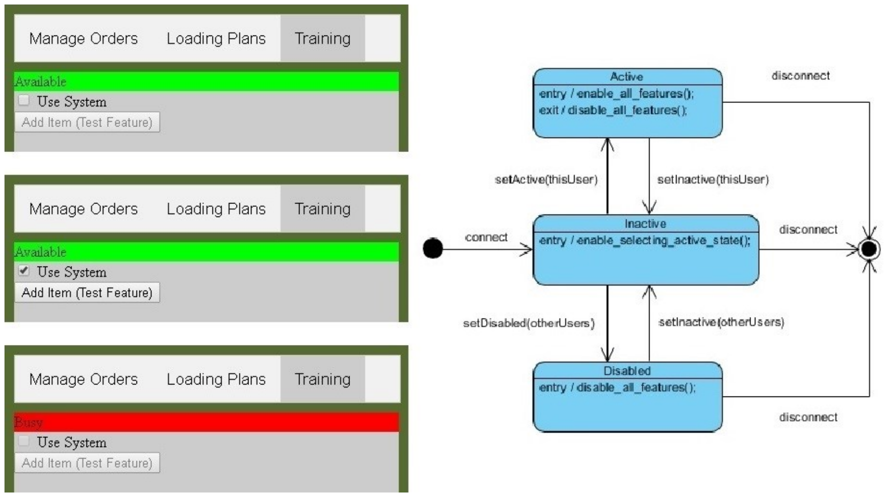


Figure 16. *State machine and appearance of menu for different states (Inactive state (up), active state (middle) and disabled state (down)).*

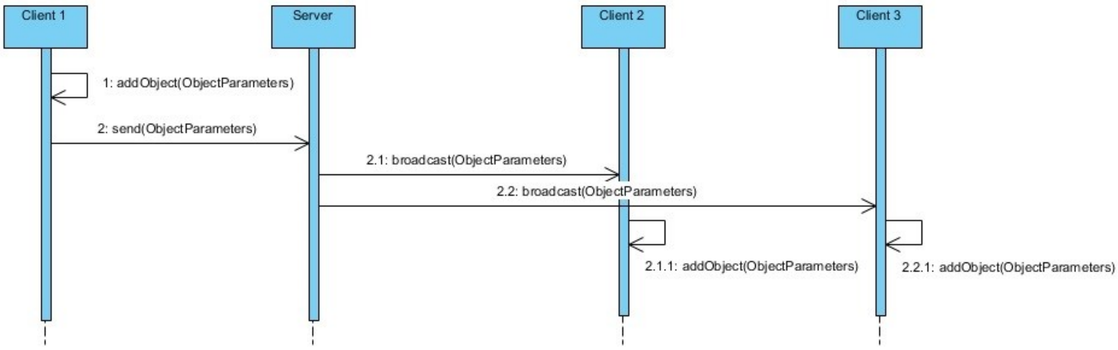


Figure 17. *Sequence diagram of add an object.*

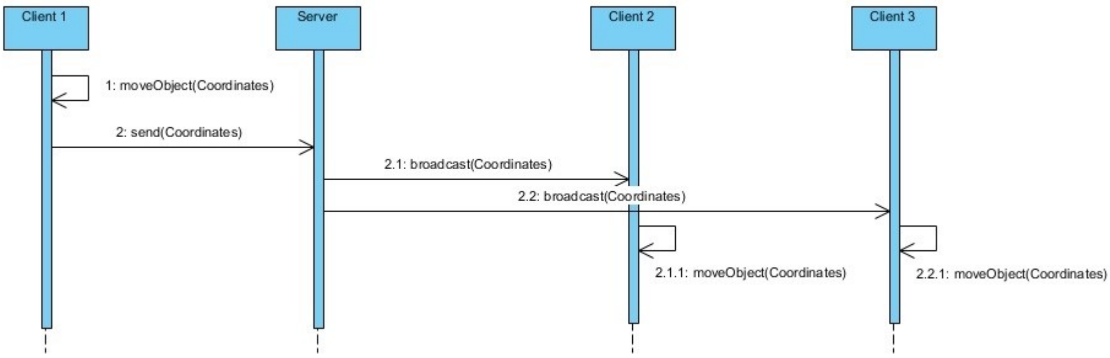


Figure 18. *Sequence diagram of moving an object*

5. EXPERIMENTAL SCENARIO AND RESULTS

In this chapter, an experimental last-minute order scenario is tested to evaluate the impact of DPSA and to analyze the performance of the visualization system.

In this example, it is assumed that a logistics company creates transportation plans by using its optimization instruments. Hence, the resulting data of three trucks with optimized and designated routes and cargos has been inserted into the visualization system's database. These trucks have their own optimized routes and cargos. The initial optimized plans are described (as follows) in Table 1.

Table 1. *Optimized truck routes in the database.*

Name	Id	Route
Truck1	1	Helsinki-Porvoo-Lahti-Tampere-Pori-Kuopio-Vaasa-Oulu-Rovaniemi
Truck2	2	Helsinki-Porvoo-Lahti-Tampere-Pori-Kuopio
Truck3	3	Tampere-Pori-Kuopio-Vaasa-Oulu-Rovaniemi

The process of adding a last-minute order into a suitable truck is described as follows: the user selects ”New Order” tab from “Manage Order” section and sets the customer name (F), the origin (Porvoo) and the destination (Pori) of the last minute order, and then clicks “Search Available Trucks” button. The system returns a list of available trucks. Selecting a truck (Truck-2) from the list opens “Add Item” menu as a continuation of “New Order” tab. Truck-2's cargo information in database is displayed in Table 2.

Table 2. *Initial optimized plan of Truck-2.*

Customer Name	Order Origin	Order Destination	Unit Volume
A	Helsinki	Kuopio	4
B	Helsinki	Pori	6
C	Helsinki	Tampere	8
D	Helsinki	Lahti	10
E	Helsinki	Porvoo	15

The user inputs the amount of objects (pallet, box etc.) and the unit size. Unit size of 1 equals approximately 40 cm in the visualization. Therefore, setting the unit size as 3, creates an object having 1200mm width and 1200mm length which is similar to the standard pallets such as EPAL CP3 (1140mm x 1140mm), EUR-2 (1200mm x 1000 mm), EUR-3 (1000mm x 1200 mm) or ISO 1100mm x 1100mm [70, 71]. Then, the user clicks “Visualize Order” button to see the result. System returns volume exceed alert and resets the menu, if the volume of the last-minute order exceeds the container's unused space.

The user must select another truck from the list in this case. Otherwise, the systems returns a visualization of the loaded container and displays the delay rate. The added order is opaque and other orders are transparent in the container visualization. The user can interact with the visualization by moving objects, viewing information related to the orders and hide the container frame to display orders easier. If the user approves that the resulting delay rate does not affect the entire transportation operation, he saves the resulting data into the database by clicking the “Save Order” button (Figure 19). Other users receive an alert “database updated”.

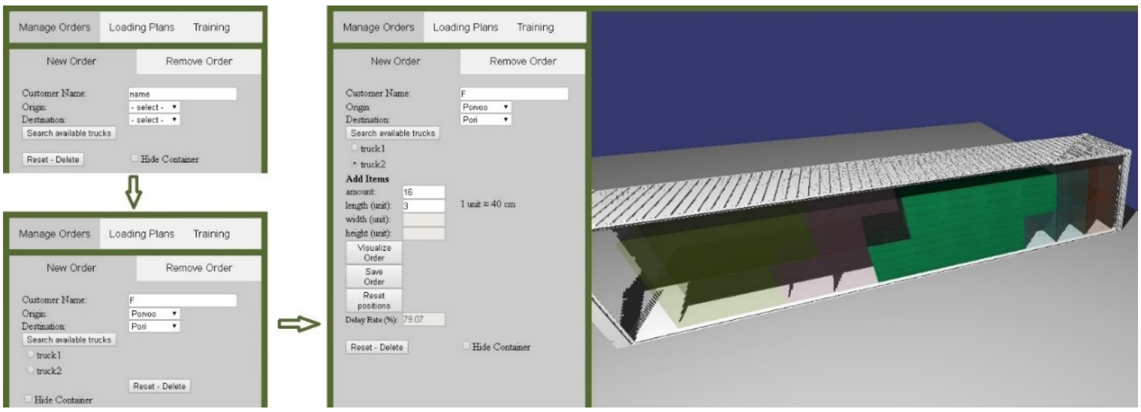


Figure 19. *Adding last minute order into a truck.*

“Remove Order” tab displays the order information at a selected location. The user can delete a cancelled order from the database. In Figure 20 orders originating from Helsinki and Porvoo locations are displayed (In this example, Truck-1 has a dummy order (origin: Helsinki) in order to display different orders at different trucks.).

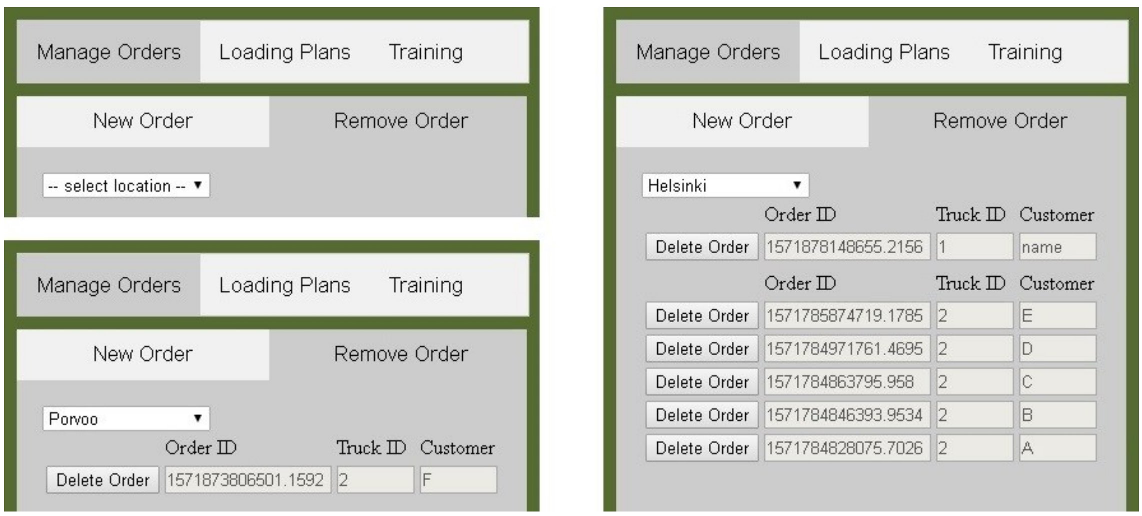
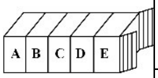


Figure 20. *Removing orders from trucks.*

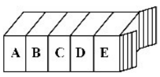
The loading sequence in the visualization is created by DPSA. The algorithm generates a permutation of loading sequences and selects the best option among them. The best option directly depends on the volume of the last-minute order. In Table 3, the generated loading sequences and changes in the total amount of rearranged items by an amount of the last minute order is described. Orders are denoted with customer name and displayed as a row in which an order closest to the container door is placed to rightmost side of the row. Number of rearrangements of an order is denoted with number of dots over the order. ($\ddot{A}$: Order A is rearranged two times during the transportation). In Figure 21, the loaded container is displayed for two last minute orders of different size (2 and 16) at the pick-up location. The top image belongs to the visualization of sequence 4 (F=16), while the bottom image represents sequence 1 (F=2) at Porvoo location.

Table 3. *Generated sequences and the total amount of rearranged items.*

	Truck-2 Route						Total Rearrangement	
	Helsinki	**Porvoo**	**Lahti**	**Tampere**	**Pori**	**Kuopio**	**F = 2**	**F = 16**
Initial Plan	ABCDE	ABCD	ABC	AB	A			
Sequence 1	ABCDE	ABCDF	ABC$\dot{F}$	AB$\ddot{F}$	A		4	32
Sequence 2	ABCDE	ABCDF	AB$\dot{F}\dot{C}$	AB$\dot{F}$	A		10	24
Sequence 3	ABCDE	ABCF$\dot{D}$	ABCF	AB$\dot{F}$	A		12	26
Sequence 4	ABCDE	ABF$\dot{C}\dot{D}$	ABF$\dot{C}$	ABF	A		18	18

Rearranged volume at each stop is shown in Table 4. Volume of one item is 27 cubic-unit or approximately 1.728 m^3. A loading sequence which gives minimum total rearranged volume is selected for the solution.

Table 4. *Rearranged volume at each stop for two different amount of order.*

		Truck-2 Route						
		Helsinki	**Porvoo**	**Lahti**	**Tampere**	**Pori**	**Kuopio**	**Total**
Sequence 1	**F=2**	0	0	54	54	0	0	108
	F=16	0	0	432	432	0	0	864
Sequence 2	**F=2**	0	0	270	0	0	0	270
	F=16	0	0	648	0	0	0	648
Sequence 3	**F=2**	0	270	0	54	0	0	324
	F=16	0	270	0	432	0	0	702
Sequence 4	**F=2**	0	486	0	0	0	0	486
	F=16	0	486	0	0	0	0	486

Delay rate for each case is calculated by using (3.11). Delay rate is 79.07% when the size of last-minute order is 16 and 13.95% for the size of 2. Since the total initial item loading and unloading (and rearranging, if available) time is known (original optimized plan), 79.07% of this time is added to the estimate of total item handling time if the amount of last order is 16 and 13.95% if the amount of last order is 2.

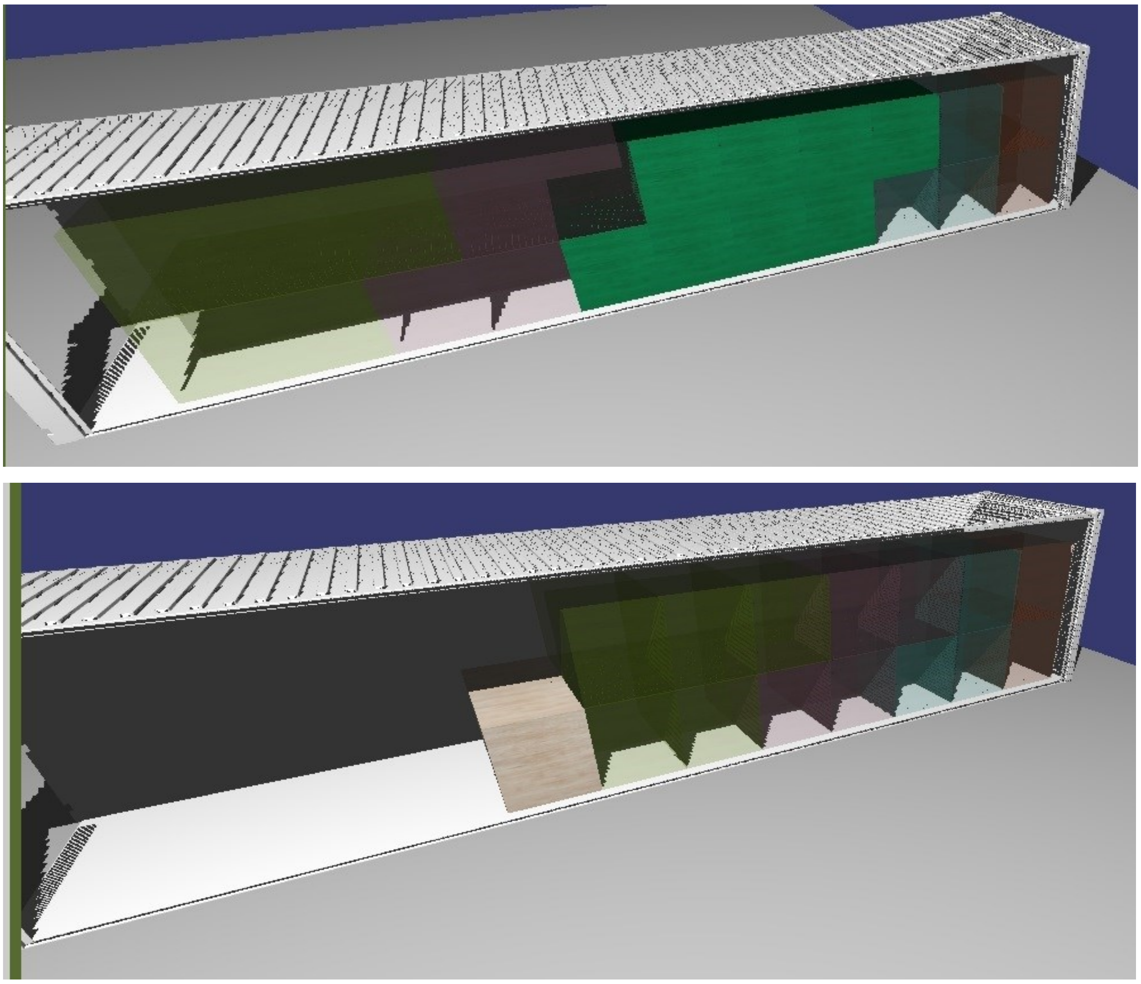
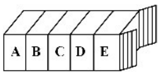


Figure 21. *Visualization of packing sequence changes by amount of items in new order. 16 items in upper image and 2 items in lower image.*

Adding another last-minute order (customer: G, origin: Lahti, destination: Kuopio) into the same truck generates new loading sequences. Four cases displayed in Table 5. For instance, sequence 11 creates the minimum delay when the size of F is 16 and G is 2, and sequence 1 is the best when the sizes of F and G are 2. The delay rate is calculated to be 7.79% for the size of F is 16 and G is 2.The delay rate is 12.24% for the size of F is 2 and G is 2.

Table 5. *Generated sequences for two last minute orders with four variations.*

	Truck-2 Route						Total Rearrangement			
	Helsinki	**Porvoo**	**Lahti**	**Tampere**	**Pori**	**Kuopio**	**F=16 G=2**	**F=16 G=10**	**F=2 G=2**	**F=2 G=24**
Initial Plan	ABCDE	ABCD	ABC	AB	A					
Sequence 1	ABCDE	ABCDF	ABCG$\dot{F}$	AB$\dot{G}\ddot{F}$	A$\ddot{G}$		36	52	8	52
Sequence 2	ABCDE	ABCDF	ABCG$\dot{F}$	A$\dot{G}\ddot{F}\dot{B}$	A$\dot{G}$		40	48	12	34
Sequence 3	ABCDE	ABCDF	ABG$\dot{F}\dot{C}$	ABG$\dot{F}$	A$\dot{G}$		28	34	12	34
Sequence 4	ABCDE	ABCDF	AG$\dot{F}\dot{B}\dot{C}$	AG$\dot{F}\dot{B}$	AG		30	30	16	16
Sequence 5	ABCDE	ABCF$\dot{D}$	ABCFG	AB$\dot{F}\dot{G}$	A$\ddot{G}$		30	46	16	60
Sequence 6	ABCDE	ABCF$\dot{D}$	ABCFG	A$\dot{G}\dot{F}\dot{B}$	A$\dot{G}$		34	42	20	42
Sequence 7	ABCDE	ABCF$\dot{D}$	ABCG$\dot{F}$	AB$\dot{G}\ddot{F}$	A$\ddot{G}$		46	62	18	62
Sequence 8	ABCDE	ABCF$\dot{D}$	ABCG$\dot{F}$	A$\dot{G}\ddot{F}\dot{B}$	A$\dot{G}$		50	58	22	44
Sequence 9	ABCDE	ABCF$\dot{D}$	ABG$\dot{F}\dot{C}$	ABG$\dot{F}$	A$\dot{G}$		36	44	22	44
Sequence 10	ABCDE	ABCF$\dot{D}$	AG$\dot{F}\dot{B}\dot{C}$	AG$\dot{F}\dot{B}$	AG		40	40	26	26
Sequence 11	ABCDE	ABF$\dot{C}\dot{D}$	ABF$\dot{C}$G	ABF$\dot{G}$	A$\ddot{G}$		22	38	22	66
Sequence 12	ABCDE	ABF$\dot{C}\dot{D}$	ABF$\dot{C}$G	AB$\dot{G}\dot{F}$	A$\ddot{G}$		38	54	24	68
Sequence 13	ABCDE	ABF$\dot{C}\dot{D}$	ABF$\dot{C}$G	A$\dot{G}\dot{F}\dot{B}$	A$\dot{G}$		42	50	28	50
Sequence 14	ABCDE	ABF$\dot{C}\dot{D}$	ABFG$\ddot{C}$	ABFG	A$\dot{G}$		28	36	28	50
Sequence 15	ABCDE	ABF$\dot{C}\dot{D}$	ABG$\dot{F}\ddot{C}$	ABG$\dot{F}$	A$\dot{G}$		44	52	30	52
Sequence 16	ABCDE	ABF$\dot{C}\dot{D}$	AG$\dot{B}\dot{F}\ddot{C}$	AG$\dot{B}\dot{F}$	AG		48	48	34	34

In the loading plan area, initial and final status of the truck are animated at each location. After the last-minute order is added to the system, the user enters loading plan area and selects location. Then, he selects the truck from the truck list. The visualization system displays the initial status of the arrived truck. By clicking the animate button unloading, loading and rearrange tasks are animated. For the example above, visualizations of the truck's initial and final status at each location after the last-minute orders F and G added can be seen at the Appendix A.

In some cases, the minimum of total rearranged volume may be degenerate. If in the example above F=2 and G=6 fourfold minimum is found, as shown in Table 6. As all the sequences have the same amount of total delay, the truck completes the transportation operation with the same delay in all these cases. However, the distribution of delay at intermediate stops varies. Cumulative delay is measured from cumulative rearranged volumes by using (3.10). Rearranged volume and cumulative rearranged volume displayed in Table 7 and 8 respectively. According to Table 8, sequence 1 provides better result.

Table 6. *Generated sequences for two last minute orders with four minimum value.*

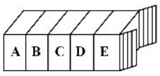
	Truck-2 Route						Total
	Helsinki	**Porvoo**	**Lahti**	**Tampere**	**Pori**	**Kuopio**	**F=2** **G=6**
Sequence 1	ABCDE	ABCDF	ABCG$\dot{F}$	AB$\dot{G}\ddot{F}$	A$\ddot{G}$		16
Sequence 2	ABCDE	ABCDF	ABCG$\dot{F}$	A$\dot{G}\ddot{F}\dot{B}$	A$\dot{G}$		16
Sequence 3	ABCDE	ABCDF	ABG$\dot{F}\dot{C}$	ABG$\dot{F}$	A$\dot{G}$		16
Sequence 4	ABCDE	ABCDF	AG$\dot{F}\dot{B}\dot{C}$	AG$\dot{F}\dot{B}$	AG		16

Table 7. *Rearranged volume at each stop for four best packing sequences.*

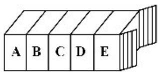
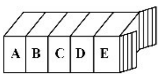
	Truck-2 Route						Total
	Helsinki	**Porvoo**	**Lahti**	**Tampere**	**Pori**	**Kuopio**	**F=2** **G=6**
Sequence 1	0	0	54	216	162	0	432
Sequence 2	0	0	54	378	0	0	432
Sequence 3	0	0	270	0	162	0	432
Sequence 4	0	0	432	0	0	0	432

Table 8. *Cumulative rearranged volume for four best packing sequences.*

	Truck-2 Route						Total
	Helsinki	**Porvoo**	**Lahti**	**Tampere**	**Pori**	**Kuopio**	**F=2** **G=6**
Sequence 1	0	0	54	270	432	432	1188
Sequence 2	0	0	54	432	432	432	1350
Sequence 3	0	0	270	270	432	432	1404
Sequence 4	0	0	432	432	432	432	1728

Finally, several clients can work on same truck to create consolidated shipment or exercise different loading patterns manually. Because it is a real-time environment, only a screenshot is provided to present collaborative work of two clients (Figure 22). Client at above is active and has control on the left panel to use features. The other client can only interact with added objects in the visualization area.

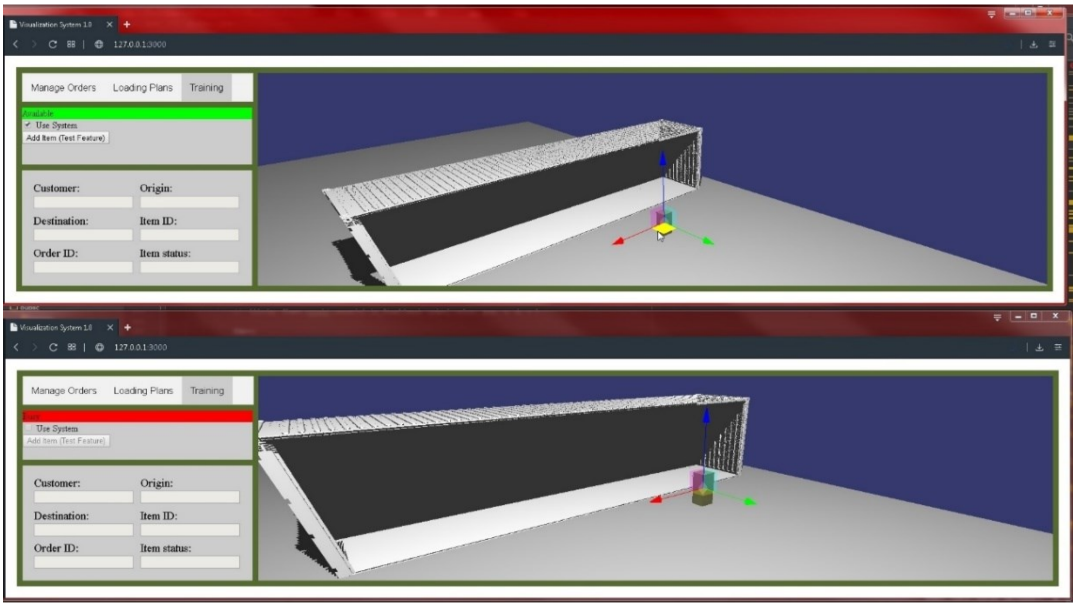


Figure 22. *Browser user interfaces of two clients that collaboratively work in real-time.*

6. DISCUSSION

In this chapter, the research questions are answered briefly based on the background study and the contribution of this work.

- *What kind of uncertainties exists in transportation operations in a supply chain and how to deal with such uncertainties in a cost-effective manner?*

Supply chain is a dynamic, complex and broad network that suffers from various sources of uncertainty, such as product characteristics, manufacturing, organization structure, customer, supplier and external factors. In this thesis project, uncertainties related to transportation operations were scrutinized. There are several sources of transportation uncertainties. Uncertainties are caused by traffic congestions, vehicle breakdowns, lack of drivers and operational problems in loading and unloading at delivery points, all resulting in delays. Furthermore, uncertainties due to poor demand information, delivery constraints, poor coordination, rigid infrastructure, cost and inventory management, and have a negative impact on efficiency of transportation operations. . Flexibility is required to deal with uncertainties. In this thesis project, DPSA implemented into a visualization system can create flexibility by increasing asset utilization efficiency to rapidly respond to last minute unexpected events.

- *What are the recent technologies used in web-based visualization and how web-based 3D visualization supports transportation operations?*

Initial web-based visualization technologies such as VRML and Java suffered from various disadvantages. Plugins were required to create visuals on web browsers and only server-side rendering techniques suitable because of weak client machines. Low bandwidth, storage and latency issues led to low interactivity.

Emergence of powerful hardware and new technologies enables client-side rendering with interactive and collaborative visualizations. HTML5 contains several features to support graphics such as 2D Canvas, WebGL and SVG, and can be used to form user interface at Front-end of the web-based visualization systems. JavaScript frameworks such as Angular.js extends the capability of HTML5 to create dynamic web page.

The rendering process is moved to GPUs on the client-side by WebGL technology, which allows to implement accelerated 3D graphics in web browsers. Various JavaScript 3D libraries (Three.js, Pex, ClayGl, AwayJs, Litescene etc.) abstract WebGL to allow developers to work on a higher level code. As a result, interactivity is improved in web-based visualization systems.

WebSocket technology enhances collaboration by providing real-time data update. HTTP requires full round trip in typical client-server system communication, but WebSocket provides efficient bi-directional, full-duplex communication channels when client and server have a handshake. Therefore, WebSocket has reduced the latency and unnecessary network traffic.

Web-based visualization system can support transportation operations in many ways. Workers can easily inspect cargo loading tasks with 3D animations, locate items in containers and view related information. Implementing algorithms into visualization logic can extend the visualization system capabilities further. For example, a rearrangement algorithm can be used to deal with dynamic situations. Web-based platform provides collaborative environment for training workers or freight consolidation service which is one of the most efficient solutions for organizations to reduce costs.

- *How to create a real-time collaborative web environment and maintain real-time communication between collaborating supply chain partners?*

Real-time communication between clients is significant for smooth interaction and efficient collaboration. As with the second research question, WebSocket is the effective technology for low latency and efficient communication. WebSocket is a low-level protocol, and thus difficult to work with. A JavaScript library, Socket.io, provides a high-level abstraction of WebSocket protocol. Socket.io’s broadcasting feature allows sending messages between clients.

- *How visualizing a container content supports workers in loading/unloading operations?*

The visualization system displays container content in 3D, item placement positions, other item information and explicitly animates the proper way of handling the items. Inexperienced workers will be able to complete faster loading, unloading and rearrangement tasks when supported by these features. Therefore, reduction of delivery time and an increase of the service level of transportation operation is possible. Interacting with the 3D visualization enables working on complex loading patterns. Workers can be trained for loading, unloading and rearrangement tasks. As a result, productivity and effectiveness of workers increase.

7. CONCLUSION AND FUTURE WORK

The focus of this thesis is to develop a DPSA and a visualization logic to create a visualization system for transportation operations to enhance competitiveness of organizations in a supply chain. Therefore, Supply Chain management, Container Loading Problem and web-based visualization are scrutinized. This chapter briefly presents the conclusions of this work and introduces possible further developments.

7.1 Conclusion

This thesis project proposes a web-based, client-side rendered visualization system to enhance an organization’s competitiveness in a dynamic and complex supply chain environment. The system utilizes client-side rendering and real-time communication technologies in order to create an interactive, collaborative and ubiquitous visualization platform. Furthermore, a DPSA is developed and implemented into the visualization logic to respond rapidly to last minute orders by creating different packing sequences.

Today, supply chains are becoming more and more dynamic and complex environments with globalization, shorter product life cycles, increased product varieties and demanding customers. As a result, uncertainties and costs are highly increasing. Organizations develop cost reduction strategies such as consolidating LTL shipments into FTL shipments to improve asset utilization efficiency and cross-docking to reduce empty traveling miles. Furthermore, organizations need flexibility to deal with uncertainties and customer complaints caused by vehicle breakdown, congestion and last minute changes in orders.

The visualization system is designed in a way that could address these problems rapidly and efficiently. A client-server-database architecture is employed in the design of the visualization system. The client tier is formed by HTML5 and CSS where HTML is supported by Angular.js so that a more dynamic webpage could be created. 3D computer graphics in the web browser is created with Three.js framework. The server tier contains the visualization logic and the DPSA. The visualization logic processes the data gathered from the database and creates visualization data that will be used in the client-side. Socket.io framework enables real-time communication between the server and the client. Finally, the database tier is created by MongoDB. The vehicle fleet information is stored in the database as JSON format.

Addition of last minute orders into partially loaded ongoing trucks with minimum delay is achieved by implementation of the DPSA into the visualization system. The system also provides the user opportunity to remove cancelled orders. Animation of loading, unloading and rearrangement processes provides handling of complex loading patterns more efficiently and with less workforce, and thus improves utilization of resources. Furthermore, a real-time collaborative work environment is created by Socket.io framework. In this environment, users can work on the consolidation plans simultaneously. In addition to this, new employees can be trained by practicing various loading patterns.

Addition of several features such as hiding container frame, displaying information of highlighted objects, moving objects, change of object color according to assigned task and different transparency to distinguish objects without any assigned tasks, provided better visualization and enhanced the interactivity.

The DPSA creates packing sequences in partial LIFO order. CLP aims to create complete LIFO structures for MSTL shipments to avoid the delays caused by rearrangements of irrelevant orders at each location. Although it is the best approach for an optimized solution, unused space of a truck increases as the truck visits and drops orders at each intermediate stop. Considering the dynamic structure of a supply chain network, these partially loaded ongoing trucks can be deployed for last minute orders with possible minimum delay within transportation schedules. Thus, vehicle fleet becomes more flexible and can rapidly respond to unexpected last minute orders. On the other hand, bringing a new empty truck from a depot for last minute orders decreases asset utilization, increases costs and response time. The DPSA calculates a delay ratio, which is the ratio of total handled volume of final cargo consisting the last minute orders to total handled volume of initial cargo. The total delay can be estimated with the delay rate and the initial loading time.

7.2 Future Work

In this thesis project, a working prototype of a visualization system is developed. After completing the prototype, capabilities of the system could be improved with further iterative development processes.

The visualization system does not allow users to create or modify the optimized plans of vehicles in the database, such as creating or modifying the initial cargo or vehicle routes. The optimized plans must be processed by a system administrator exclusively. A new user interface can be developed for the system administrator to process optimized plans. Furthermore, different levels of access rights could be defined for different users according to their roles. For example, customers could only view loading plans of their own last-minute orders, workers could view the loading plans related to their own tasks whereas partners could work in collaborative visualization area and administrator users could have authorization to handle and modify optimization plans. In the DPSA's current version, items are removed from a truck for the rearrangement task. The algorithm can be developed in a way that the rearranged items can be relocated inside the container, which may decrease the delay caused by rearrangement. Moreover, the DPSA can be developed further or combined with CLP algorithms to work with strongly heterogeneous objects.

REFERENCES

[1] J. Auramo, J. Kauremaa and K. Tanskanen, Benefits of IT in supply chain management: an explorative study of progressive companies, International Journal of Physical Distribution & Logistics Management, vol. 35, no. 2, pp. 82-100, 2005.

[2] X. Zhang and Y. Liu, The Impact of ICT on Supplier-Buyer Relationship in Different Types of Supply Chain, 2007 International Conference on Wireless Communications, Networking and Mobile Computing, 2007.

[3] X. Zhang, D. Van Donk and T. van der Vaart, The different impact of inter-organizational and intra-organizational ICT on supply chain performance, International Journal of Operations & Production Management, vol. 36, no. 7, pp. 803-824, 2016.

[4] M. Frohlich, e-Integration in the Supply Chain: Barriers and Performance, Decision Sciences, vol. 33, no. 4, pp. 537-556, 2002.

[5] A. Barlow, Web Technologies and Supply Chains, Supply Chain Management - New Perspectives, pp. 461-480, 2011.

[6] D. Chou, X. Tan and D. Yen, Web technology and supply chain management, Information Management & Computer Security, vol. 12, no. 4, pp. 338-349, 2004.

[7] Bo Dai and H. Chen, Mathematical model and solution approach for collaborative logistics in less than truckload (LTL) transportation, 2009 International Conference on Computers & Industrial Engineering, pp. 767-772, 2009.

[8] P. Benda, S. LaVoie, W. Osborn, G. Davison and P. Rocha, Transport vehicle capacity maximization logistics system and method of same, US7840429B2, 2010.

[9] V. Yu, D. Sharma and K. Murty, Door Allocations to Origins and Destinations at Less-than-Truckload Trucking Terminals, Journal of Industrial and Systems Engineering, 2019.

[10] J. Mentzer, W. DeWitt, J. Keebler, S. Min and N. Nix, C. Smith and Z. Zacharia, Defining Supply Chain Management, Journal of Business Logistics, vol. 22, no. 2, pp. 1-25, 2001.

[11] R. Crandall, W. Crandall and C. Chen, Principles of supply chain management, 2nd ed. Boca Raton: CRC Press, Taylor & Francis Group, 2015, pp. 6-12.

[12] Q. Yan and Q. Zhang, The Optimization of Transportation Costs in Logistics Enterprises with Time-Window Constraints, Discrete Dynamics in Nature and Society, vol. 2015, pp. 1-10, 2015.

[13] Y. TSENG and W. YUE, THE ROLE OF TRANSPORTATION IN LOGISTICS CHAIN, in Proceedings of the Eastern Asia Society for Transportation Studies, vol. 5, pp. 1657-1672, 2005.

[14] A. Kearney, State of Logistics Report, Council of Supply Chain Management Professionals, 2017.

[15] Logistics Report, Freight Transport Association, 2017.

[16] P. Parsa, M. Rossetti and S. Zhang, Multi Stop Truckload Planning, in Industrial and Systems Engineering Conference, 2017.

[17] U. Apte and S. Viswanathan, Effective Cross Docking for Improving Distribution Efficiencies, International Journal of Logistics Research and Applications, vol. 3, no. 3, pp. 291-302, 2000.

[18] A. Amini, R. Tavakkoli-Moghaddam and A. Omidvar, Cross-docking truck scheduling with the arrival times for inbound trucks and the learning effect for unloading/loading processes, Production & Manufacturing Research, vol. 2, no. 1, pp. 784-804, 2014.

[19] F. Beier, Transportation Contracts and The Experience Effect: A Framework For Future Research, Journal of Business Logistics, vol. 10, no. 2, pp. 73-89, 1989.

[20] T. Davis, Effective supply chain management, Sloan Management Review Summer, pp. 35-46, 1993.

[21] R. Mason-Jones and D. Towill, Shrinking the supply chain uncertainty circle, IOM control, vol. 24, no. 7, pp. 17-22, 1998.

[22] H. Peck, J. Abley, M. Christopher, M. Haywood, R. Saw, C. Rutherford, M. Strathern, Creating Resilient Supply Chains: A Practical Guide, The Centre for Logistics and Supply Chain Management, Cranfield School of Management, pp. 1-100, 2003.

[23] V. Sanchez Rodrigues, D. Stantchev, A. Potter, M. Naim and A. Whiteing, Establishing a transport operation focused uncertainty model for the supply chain, International Journal of Physical Distribution & Logistics Management, vol. 38, no. 5, pp. 388-411, 2008.

[24] M. Sanchez-Rodrigues, D. Potter and P. Naim, Evaluating The Causes Of Uncertainty in Logistics Operations, The International Journal of Logistics Management, vol. 21, no. 1, pp. 45-64, 2010.

[25] A. McKinnon and Y. Ge, Use of a synchronised vehicle audit to determine opportunities for improving transport efficiency in a supply chain, International Journal of Logistics Research and Applications, vol. 7, no. 3, pp. 219-238, 2004.

[26] Q. Zhang, M. Vonderembse and J. Lim, Logistics flexibility and its impact on customer satisfaction, The International Journal of Logistics Management, vol. 16, no. 1, pp. 71-95, 2005.

[27] A. Ramos, J. Oliveira and M. Lopes, A physical packing sequence algorithm for the container loading problem with static mechanical equilibrium conditions, International Transactions in Operational Research, vol. 23, no. 1-2, pp. 215-238, 2014.

[28] B. Domingo, S. Ponnambalam and G. Kanagaraj, A Differential Evolution Based Algorithm for Single Container Loading Problem, in IEEE Symposium on Differential Evolution, 2013.

[29] D. Pisinger, Heuristics for the container loading problem, European Journal of Operational Research, vol. 141, no. 2, pp. 382-392, 2002.

[30] A. Bortfeldt and G. Wäscher, Constraints in container loading – A state-of-the-art review, European Journal of Operational Research, vol. 229, no. 1, pp. 1-20, 2013.

[31] D. Martínez, R. Alvarez-Valdes and F. Parreño, A Grasp Algorithm For The Container Loading Problem With Multi-Drop Constraints, Pesquisa Operacional, vol. 35, no. 1, pp. 1-24, 2015.

[32] B. McCormick, T. DeFanti and M. Brown, Definition of visualization, ACM SIGGRAPH Computer Graphics, vol. 21, no. 6, pp. 3-3, 1987.

[33] Visualization, University of Stuttgart, 2002. Available: https://cgl.ethz.ch/teaching/former/scivis_07/Notes/stuff/StuttgartCourse/VIS-Modules-01-Preface-Introduction.pdf. [Accessed: 5 September 2019].

[34] F. Mwalongo, M. Krone, G. Reina and T. Ertl, State-of-the-Art Report in Web-based Visualization, Computer Graphics Forum, vol. 35, no. 3, pp. 553-575, 2016.

[35] D. Koller, M. Turitzin, M. Levoy, M. Tarini, G. Croccia, P. Cignoni and R. Scopigno, Protected interactive 3D graphics via remote rendering, ACM Transactions on Graphics, vol. 23, no. 3, pp. 695-703, 2004.

[36] E. Zudilova-Seinstra, T. Adriaansen and R. Liere, Overview of Interactive Visualization, Trends in Interactive Visualization, pp. 3-15, 2009.

[37] P. Isenberg, N. Elmqvist, J. Scholtz, D. Cernea, K. Ma and H. Hagen, Collaborative visualization: Definition, challenges, and research agenda, Information Visualization, vol. 10, no. 4, pp. 310-326, 2011.

[38] The Virtual Reality Modeling Language. [Online]. Available: http://www.web3d.org/documents/specifications/14772/V2.0/index.html. [Accessed: 05- Sep- 2019].

[39] Java Software | Oracle. [Online]. Available: https://www.oracle.com/java/. [Accessed: 05- Sep- 2019].

[40] T. Rischbeck and P. Watson, A scalable, multi-user VRML server, in Proceedings IEEE Virtual Reality, 2002.

[41] K. Engel and T. Ertl, Texture-based Volume Visualization for Multiple Users on the World Wide Web, Eurographics, pp. 115-124, 1999.

[42] J. Li, AtomEye: an efficient atomistic configuration viewer, Modelling and Simulation in Materials Science and Engineering, vol. 11, no. 2, pp. 173-177, 2003.

[43] W. Humphrey, A. Dalke and K. Schulten, VMD: Visual molecular dynamics, Journal of Molecular Graphics, vol. 14, no. 1, pp. 33-38, 1996.

[44] M. Falk, S. Grottel, M. Krone and G. Reina, Interactive GPU-based visualization of large dynamic particle data. 2016, pp. 1-10.

[45] Jmol: an open-source browser-based HTML5 viewer and stand-alone Java viewer for chemical structures in 3D. [Online]. Available: http://jmol.sourceforge.net. [Accessed: 06- Sep- 2019].

[46] A. Rose and P. Hildebrand, NGL Viewer: a web application for molecular visualization, Nucleic Acids Research, vol. 43, no. 1, pp. 576-579, 2015.

[47] F. Mwalongo, M. Krone, G. Karch, M. Becher, G. Reina and T. Ertl, Visualization of molecular structures using state-of-the-art techniques in WebGL, Proceedings of the Nineteenth International ACM Conference on 3D Web Technologies - Web3D '14, pp. 133-141, 2014.

[48] Volume Visualization: Principles and Advances. [Online]. Available: https://pdfs.semanticscholar.org/de5b/c8f6768fe08001ba644e7710dffe9fc6abd5.pdf. [Accessed: 06- Sep- 2019].

[49] J. Congote, A. Segura, L. Kabongo, A. Moreno, J. Posada and O. Ruiz, Interactive visualization of volumetric data with WebGL in real-time, Proceedings of the 16th International Conference on 3D Web Technology - Web3D '11, pp. 137-146, 2011.

[50] X. Hou, J. Sun and J. Zhang, A web-based solution for 3D medical image visualization, Medical Imaging 2015: PACS and Imaging Informatics: Next Generation and Innovations, vol. 9418, pp. 1-8, 2015.

[51] H. Jacinto, R. Kéchichian, M. Desvignes, R. Prost and S. Valette, A web interface for 3D visualization and interactive segmentation of medical images, Proceedings of the 17th International Conference on 3D Web Technology - Web3D '12, pp. 51-58, 2012.

[52] M. Figueiredo, J. Rodrigues, I. Silvestre and C. Veiga-Pires, Web3D Visualization of High Detail and Complex 3D-mesh Caves Models, 2014 18th International Conference on Information Visualisation, pp. 275-280, 2014.

[53] B. Jenny, B. Šavrič and J. Liem, Real-time raster projection for web maps, International Journal of Digital Earth, vol. 9, no. 3, pp. 215-229, 2015.

[54] C. Peng, The research and design Of 3D Web guide system based On WebGL, The 26th Chinese Control and Decision Conference (2014 CCDC), pp. 4052-4054, 2014.

[55] B. Chen and Z. Xu, A framework for browser-based Multiplayer Online Games using WebGL and WebSocket, 2011 International Conference on Multimedia Technology, pp. 471-474, 2011.

[56] About HTML5 WebSocket - Powered by Kaazing. [Online]. Available: http://www.websocket.org/aboutwebsocket.html. [Accessed: 06- Sep- 2019].

[57] H. Levkowitz and C. Kelleher, Cloud and Mobile Web-Based Graphics and Visualization, 2012 25th SIBGRAPI Conference on Graphics, Patterns and Images Tutorials, pp. 21-35, 2012.

[58] WebGL Specification. [Online]. Available: https://www.khronos.org/registry/webgl/specs/latest/1.0/. [Accessed: 06- Sep- 2019].

[59] M. Hennig, D. Gaspers and B. Mertsching, Interactive WebGL-based 3D visualizations for situated mathematics teaching, 2013 12th International Conference on Information Technology Based Higher Education and Training (ITHET), 2013.

[60] C. Gocmenoglu, T. Acarman and B. Levrat, WGL-VANET: A web-based visualization tool for VANET simulations, 2015 IEEE International Conference on Vehicular Electronics and Safety (ICVES), pp. 62-63, 2015.

[61] Y. Yikang, L. Xinxing and L. Lei, A WebGL-Based Method for Visualization of Space Environment, 2012 Fifth International Symposium on Computational Intelligence and Design, pp. 331-334, 2012.

[62] A collection of WebGL frameworks and libraries. [Online]. Available: https://gist.github.com/dmnsgn/76878ba6903cf15789b712464875cfdc. [Accessed: 06- Sep- 2019].

[63] A. Malahova and A. Tiwari, IT for Product design, in KTP Associates Conference, 2009.

[64] J. Terno, G. Scheithauer, U. Sommerweiß and J. Riehme, An efficient approach for the multi-pallet loading problem, European Journal of Operational Research, vol. 123, no. 2, p. 378, 2000.

[65] A. El-Bouri, A Search-Based Approach for Solving Regular Two-Dimensional Cutting Stock Problems, University of Manitoba, p. 6, 1993.

[66] A. Bortfeldt and H. Gehring, A hybrid genetic algorithm for the container loading problem, European Journal of Operational Research, vol. 131, no. 1, p. 145, 2001.

[67] L. Sheng, S. Xiuqin, C. Changjian, Z. Hongxia, S. Dayong and W. Feiyue, Heuristic algorithm for the container loading problem with multiple constraints, Computers & Industrial Engineering, vol. 108, p. 153, 2017.

[68] D. Mack, A. Bortfeldt and H. Gehring, A parallel hybrid local search algorithm for the container loading problem, International Transactions in Operational Research, vol. 11, no. 5, p. 515, 2004.

[69] V. Forest, Loading .obj | Web Art, Judisdaid.com, 2019. [Online]. Available: http://judisdaid.com/webart/?p=292. [Accessed: 06- Sep- 2019].

[70] ISO 6780:2003, ISO, 2003. [Online]. Available: https://www.iso.org/standard/30524.html. [Accessed: 06- Sep- 2019].

[71] Load Carriers Overview, Epal-pallets.org, 2019. [Online]. Available: https://www.epal-pallets.org/eu-en/load-carriers/overview/. [Accessed: 06- Sep- 2019].

[72] A. Duenas, Y. Tutuncu and J. Chilocott, A Genetic Algorithm Approach to the Nurse Scheduling Problem with Fuzzy Preferences, IMA Journal of Management Mathematics, vol. 20, no. 4, pp. 369-383, 2009.

[73] N. Sabar, M. Ayob, G. Kendall and R. Qu, A Honey-bee Mating Optimization Algorithm for Educational Timetabling Problems, vol. 216, no. 3, pp. 533-543, 2012.

[74] J. Bartholdi and K. Gue, The Best Shape for a Crossdock, Transportation Science, vol. 38, no.2, pp. 235-244, 2004.

[75] R. Donolo, Contributions to Geovisualization for Territorial Intelligence. [Online]. Available: https://tel.archives-ouvertes.fr/tel-01371535/document. [Accessed: 06- Sep-2019].

APPENDIX A: ANIMATION IN LOADING PLAN AREA

Transportation plan of Truck-2 with two last minute orders (F=16, G=2) is animated (Sequence 11, in Table 5). Container is hidden to display items clearly. Truck route is: Helsinki, Porvoo, Lahti, Tampere, Pori and Kuopio.

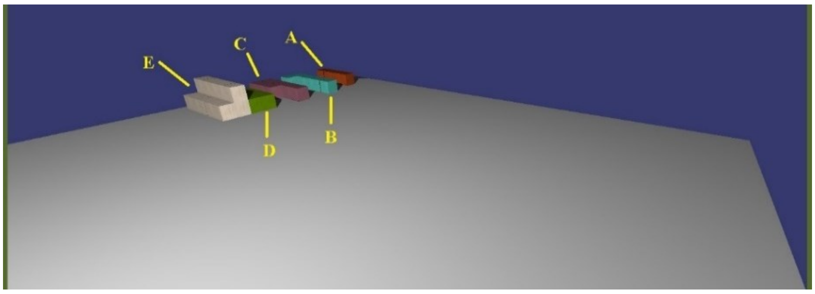


Figure 23. *Helsinki, initial state.*

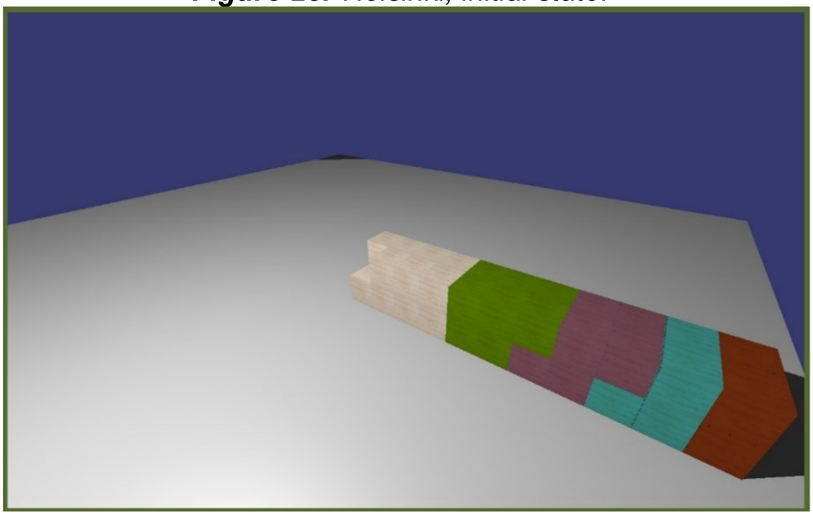


Figure 24. *Helsinki, final state.*

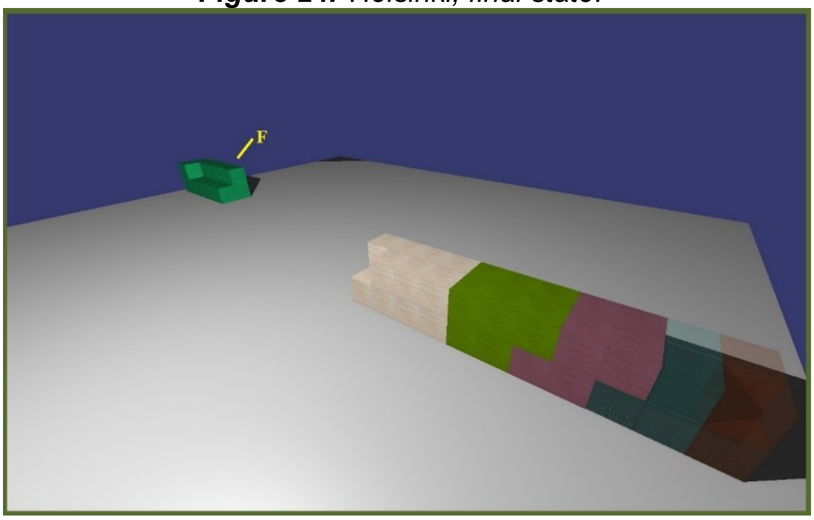


Figure 25. *Porvoo, initial state.*

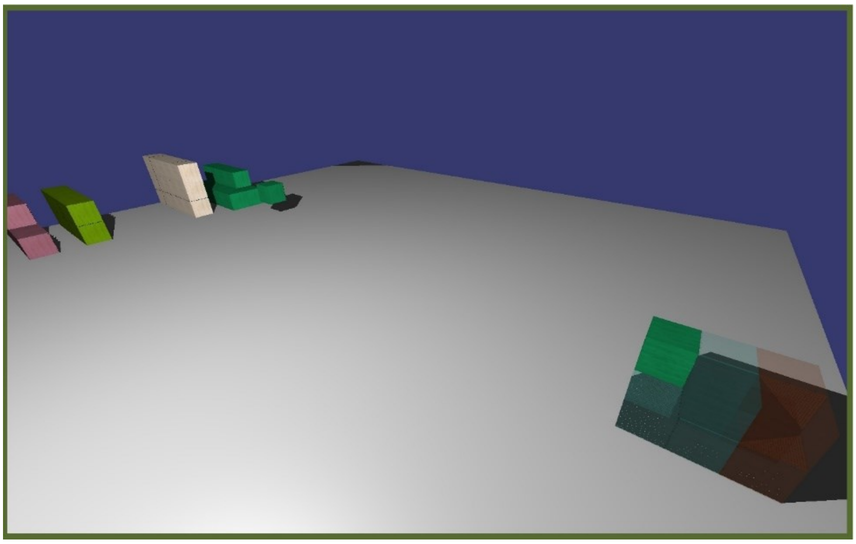


Figure 26. *Porvoo, reconfiguration state.*

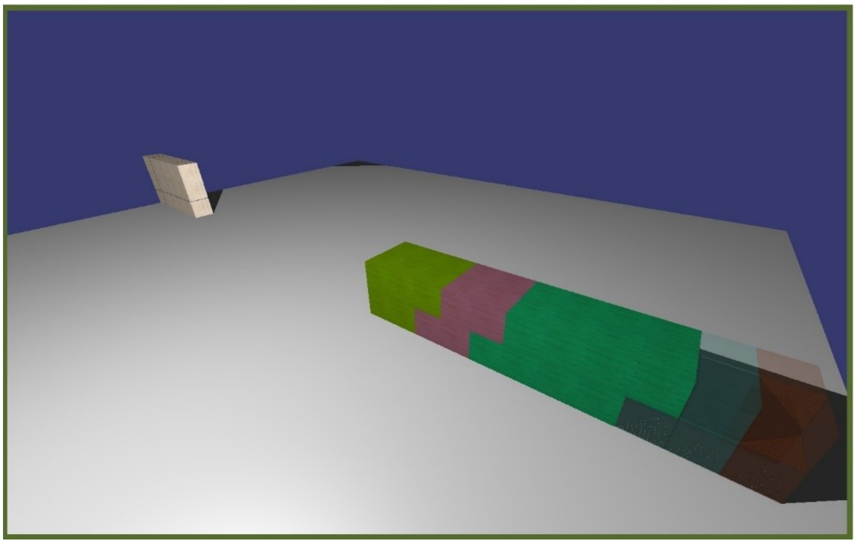


Figure 27. *Porvoo, final state.*

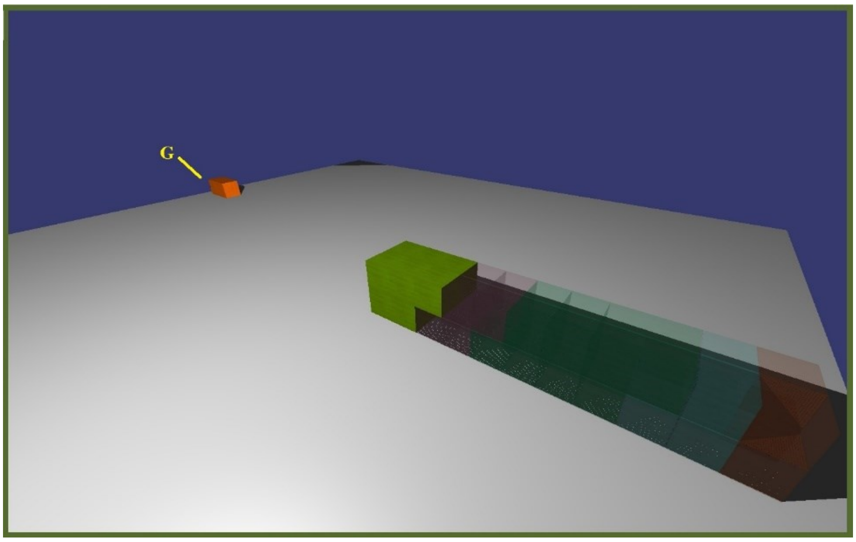


Figure 28. *Lahti, initial state.*

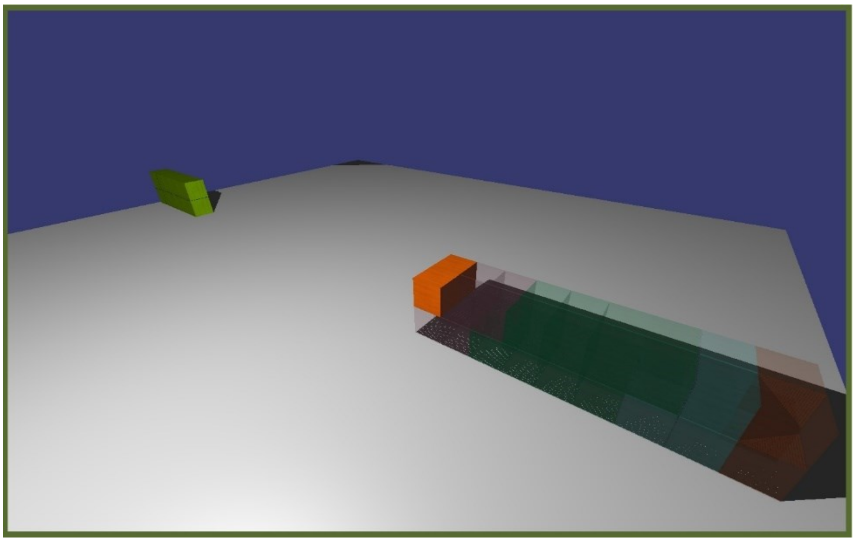


Figure 29. *Lahti, final state.*

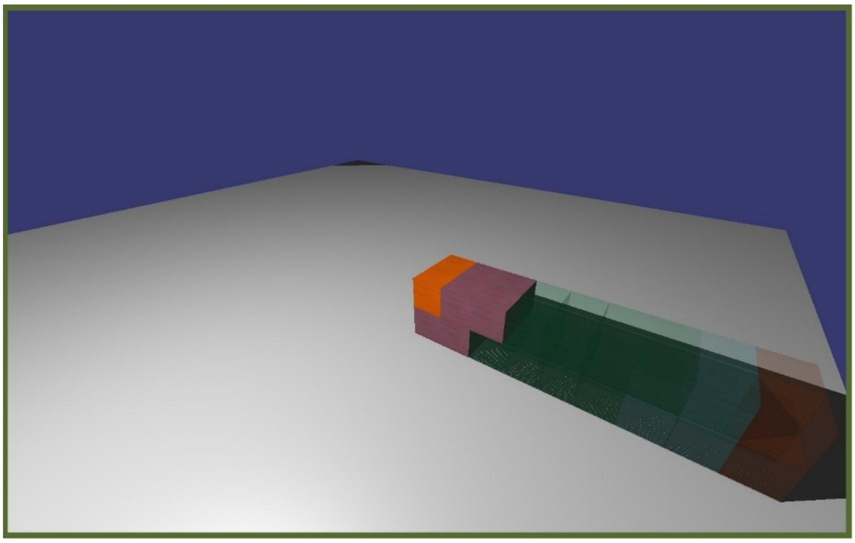


Figure 30. *Tampere, initial state.*

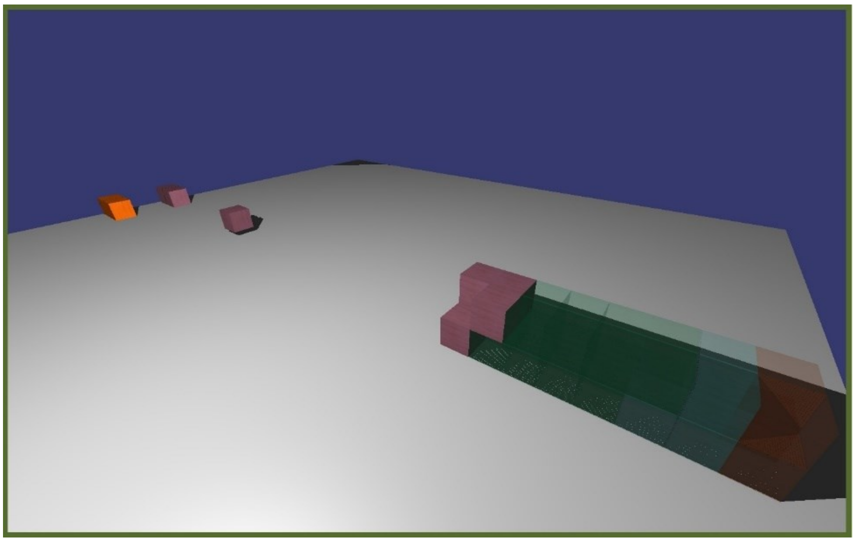


Figure 31. *Tampere, reconfiguration state.*

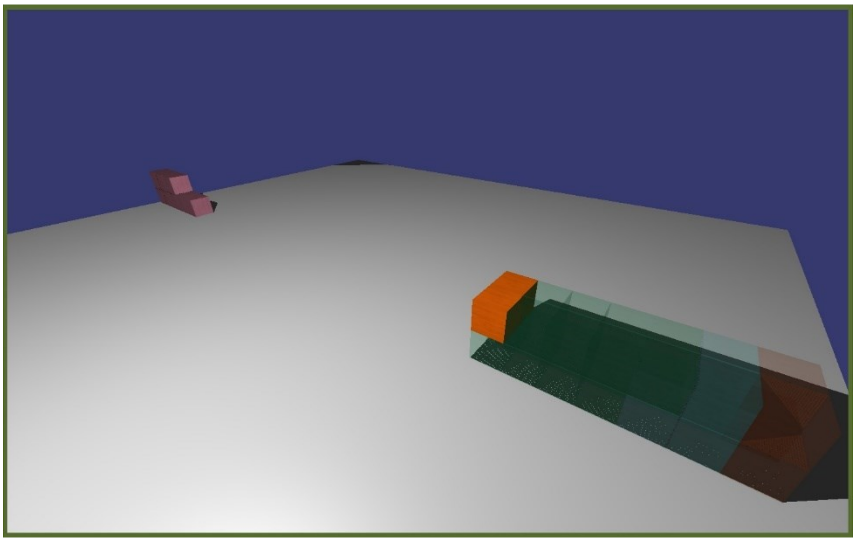


Figure 32. *Tampere, final state.*

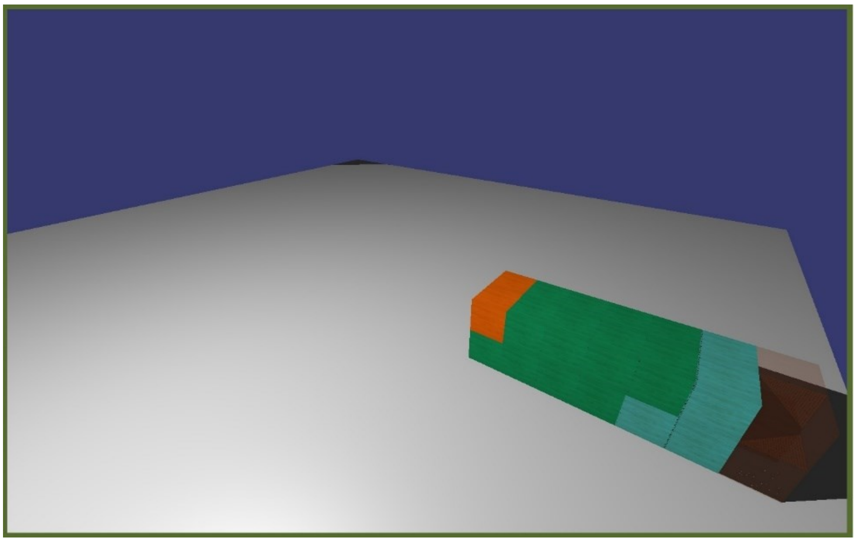


Figure 33. *Pori, initial state*

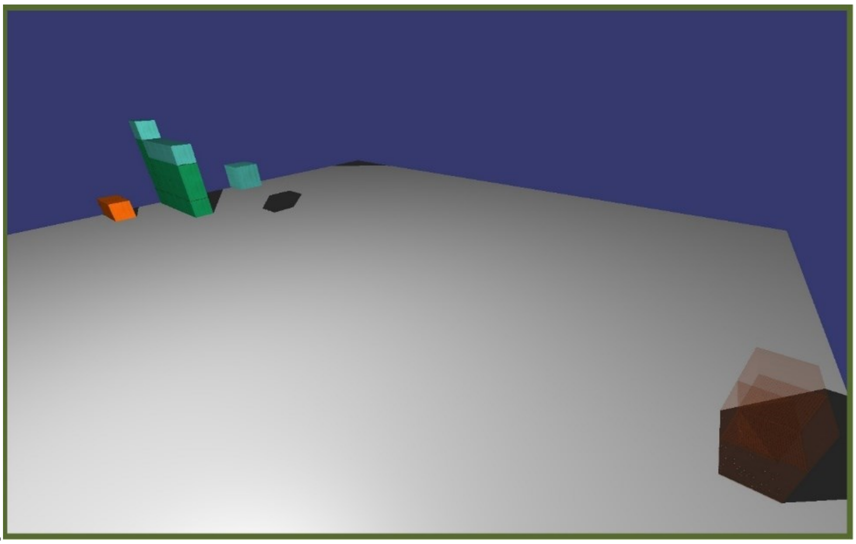


Figure 34. *Pori, reconfiguration state.*

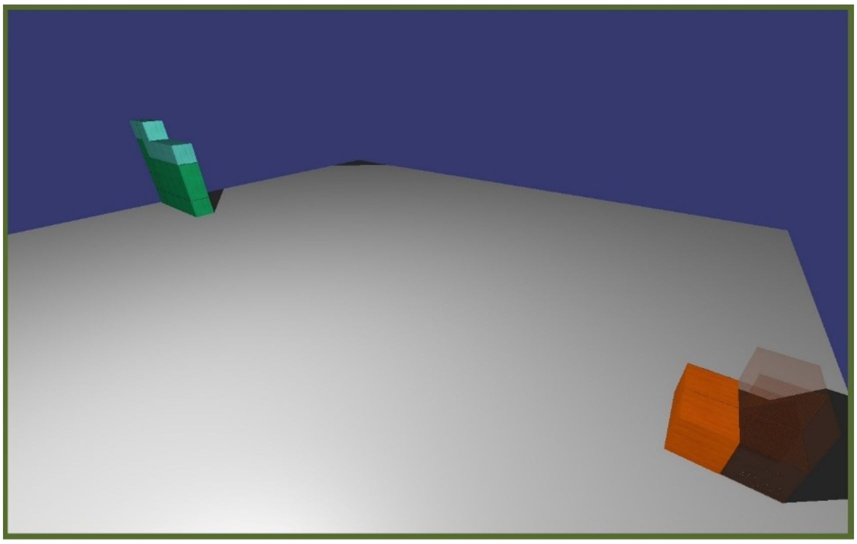


Figure 35. *Pori, final state.*

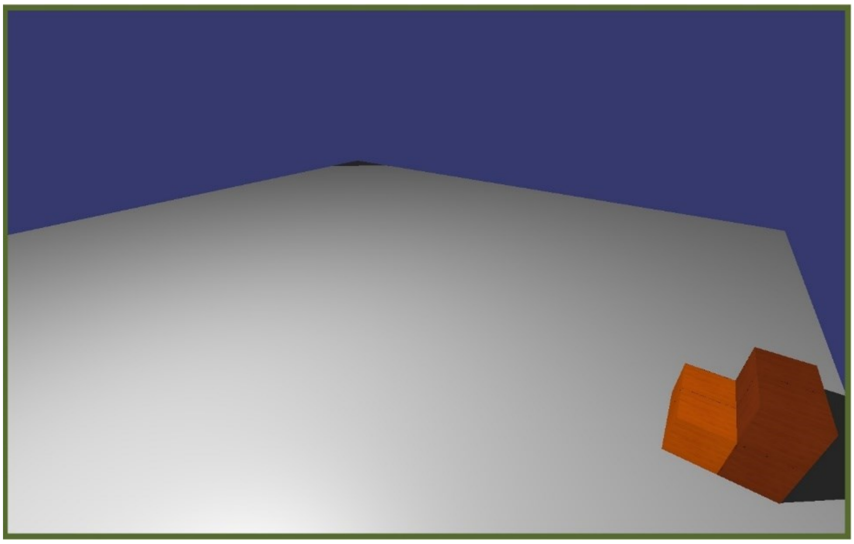


Figure 36. *Kuopio, initial state.*

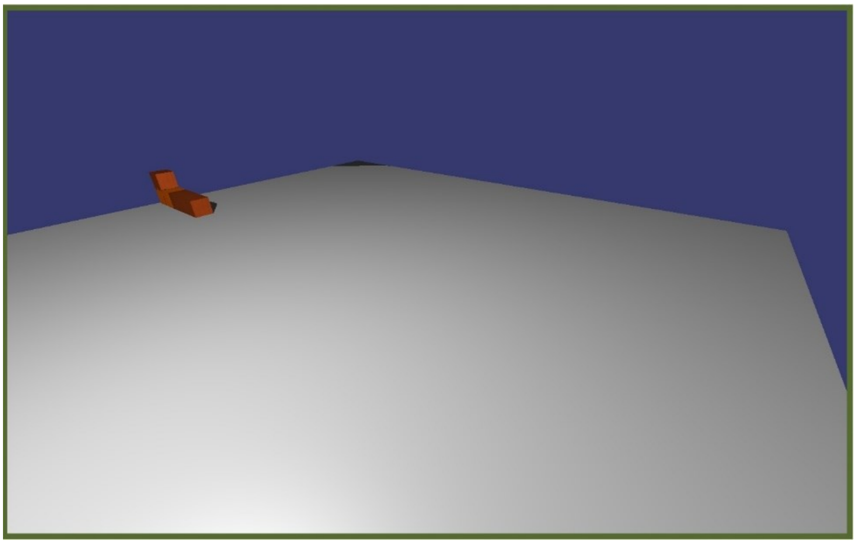


Figure 37. *Kuopio, final state.*