UNIVERSITY OF VAASA
FACULTY OF TECHNOLOGY
DEPARTMENT OF PRODUCTION

Ta Minh Hieu

REAL-TIME LOGISTICS

Case: Development of a Shipment Status Display System for a Large Manufacturing Company

Master's Thesis in
Industrial Management

VAASA 2013

To Huong, Ha, and Thao

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATION

A-GPS	Assisted GPS
CON	Constraint
DM	Delivery Manager
DP	Design Parameter
ERP	Enterprise Resource Planning
ESB	Enterprise Service Bus
FR	Functional Requirement
GLONASS	Global Navigation Satellite System
GPRS	General Packet Radio Service
GPS	Global Positioning Service
HU	Handling Unit
IIS	Internet Information Services
LCT	Logistics Control Tower
MRP	Material Requirements Planning
MVC	Model-View-Controller
OD	Outbound Delivery
PM	Project Management
PO	Purchase Order
RFID	Radio Frequency Identification
SDK	Software Development Kit
SSCC	Serial Shipping Container Code
TM	Transport Manager
VBA	Visual Basic for Applications
VS	Visual Studio
WebPi	Web Platform Installer
WIP	Work in Process
WSP	Wärtsilä Ship Power

UNIVERSITY OF VAASA
Faculty of Technology

Author:	Hieu Ta
Topic of the Master's Thesis:	Real-time Logistics Case: Development of a Shipment Status Display System for a Large Manufacturing Company
Instructor:	Petri Helo
Degree:	Master of Science in Economics and Business Administration
Major subject:	Industrial Management
Year of Entering the University:	2008
Year of Completing the Master's Thesis:	2013 **Pages:** 99

ABSTRACT:

As traditional heavy industry businesses transform into global solutions providers, their business models change into project-based and their supply networks expand. Project business faces greater uncertainty within the supply chain than traditional business, thus requires greater need for data exchange within the supply chain. Numerous information systems have provided the organization with a wealth of data. However project management often faces great challenges to utilize it for better visibility on project delivery status, as well as to communicate that to stakeholders.

In response to the need for better usage and presentation of transactional project logistical data, a real-time shipment status display system has been developed. The complete system offers an intuitive, up-to-date, fast, and reliable display that is accessible through a wide range of devices. In this thesis, the system is customized to run on public displays.

In term of development methodology, spiral axiomatic design approach is adopted to ensure maximum independence of components. The end result is a system comprising of two independent sub-systems: one is for data collection and the second one is for presentation. Modern web technologies such as ASP.NET MVC4, HTML5, and CSS3 have been used to develop the presentation sub-system.

The thesis contributes a software artifact that complements information systems that are either too much focused on transactional data or unable to communicate project logistics data to stakeholders. It also demonstrates the use of axiomatic system design in developing modern web platforms.

KEYWORDS: real-time logistics display, axiomatic design, project logistics information system, LogTrack

1. INTRODUCTION

The last global financial crisis has contracted numerous economies, resulting in significantly lower demand from developed economies and shortage of capital for industries. This has badly affected international trade, leading to low demand for sea transportation and subsequently low demand for new ship buildings. Under harsh economic pressure, ship owners and shipyards struggle to maintain their operations, placing tougher requirements on price, service level, order lead time, and product technical performance. Furthermore intensified competition between engine manufacturers and pushed many to be leaner yet to further diversify their portfolios for higher returns and new areas of growth. Wärtsilä Ship Power expected the imminent recovery of world economy and had developed a new strategy to embrace it and become the leading systems integrator in the marine industry. The company plans to sell not only engines but also bundled, integrated, and solutions systems. Transformation from a traditional product seller to a solution provider means greater challenge for project delivery management because of increasing scope and amount of components to be tracked and traced.

This thesis belongs to project LogTrack, which is collaboration between Wärtsilä Ship Power with Vaasa University and Tekes – the Finnish Funding Agency for Technology and Innovation. While LogTrack focuses on providing outbound shipments data, the thesis focuses on presenting it, in the form of an airport-like display that is familiar to project people. The display should be capable of delivering shipment statuses to the right project people in real-time. This implies three major challenges:

- The right kind of information: Information displayed has to be as much concise and understandable as possible, so that people could quickly find it. This is analogous to importance of an airport screen to passengers during last-minute calls.
- The right time: at present, sometimes project people receive their project delivery status quite late, resulting in costly corrective actions. With this new

display system, logistics control tower can communicate shipments status straight to project management for more timely action.

- The right format: the layout and color convention have to be intuitive to users. It can be similar airport screens as project people tend to travel a lot.

In term of application architecture, data query and presentation are two independent processes and therefore are performed by two different sub-systems. While data is prepared by a VBA application, it is presented by a web server in the form of a web page. In order to develop the system, spiral axiomatic development approach is employed. The development process started from inception phase where business logic is analyzed and transformed into independent functional requirements. All functional requirements and design choices are either uncoupled or decoupled. Therefore the design satisfies the first axiom for good designs. Furthermore all of them are documented in a design matrix which can be found in Appendix 1.

Software design, implementation, and testing are developed through prototypes. After each prototype, user opinions are collected and functional requirements are updated. The development has two main iterations: the first iteration shapes system functionality and the second one enhances the system performance and usability. Interesting new web technologies have been explored while existing technology has found new ways of implementation. The final system is capable of running on a wide range of devices with different screen sizes. However for this thesis the system is further customized to run on public displays. Therefore a large industrial LCD display is purchased, to be mounted on a wall at the crossroad of main office streets.

2. BACKGOUND INFORMATION AND THEORY

2.1. Project logistics management through shipments at Wärtsilä

Unlike commodity manufacturers, most heavy industry manufacturers such as Wärtsilä Ship Power (WSP) follow make-to-order production process, meaning that products are only built after a contract is signed. In make-to-order production, stock keeping of assemblies is generally not applicable because of technical & classification variability and high working capital tied up in inventory. Two practical examples are preheating units and engines. Preheating units are generally not purchased as stock items due to unknown power, voltage, as well as classification society requirements. Engines in theory can be manufactured and kept in stock; but due to high unit value and high warehouse maintenance cost, this is not an option. In order to manage make-to-order process, WSP has organized itself into a project organization, whereas every customer delivery contract is a project, with standard gates and milestones served as decision points during project lifetime.

Project logistics is different from production logistics as it is not optimized for delivering steady flows of material through a network of processes and stock points. Instead, it coordinates delivery of finished or semi-finished assemblies from suppliers to consolidation points and ultimately to end-customer. Project logistics activities could start as early as in sales phase but it mainly occurs after completion of purchasing and when goods start to be delivered to consolidation warehouses. For transport management and business controlling reasons, gathering goods into consolidation warehouses is preferred for a number of reasons. First, it provides magnitude of order advantage when booking for intercontinental shipments. Second, it simplifies customer's good reception process through reduction of shipments. And lastly, export customs declaration and correct shipping marks can be executed at consolidation warehouses before goods leave for final destination. In practice, a typical customer delivery projects can involve as many as 20 suppliers from five countries, grouped in 30 inbound shipments with over 100 cases; all of them eventually leave for customer in only five shipments. Project logistics often involve large equipment which has to be treated separately from other equipment. Trucking a large engine will require road permit, while flying it will require a special cargo airplane. In most cases, shipments involve multimodal transportation, thus require one to five weeks for outbound transport preparation.

Picture 1. Trucking an engine. Image courtesy of Marco Iaccarino.

Picture 2. Special air transportation using cargo airplane. Image courtesy of Mari van Ee.

In order to manage project logistics information, Wärtsilä Ship Power uses SAP R/3. SAP is used extensively to manage the flow of material from suppliers to consolidation warehouses and final customers. Assemblies are purchased from international manufacturers. Upon good readiness, goods are packed into cases which are identified by Handling Unit Numbers (HU). Each delivery from supplier is given an Outbound Delivery Number (OD) that includes one or more HUs. In order to coordinate the matching of HUs to the physical cases, supplier informs purchaser the number of cases. Purchaser then creates in SAP the correct amount of HUs, adds them to packing list templates together with project markings such as final destination, contact person, and shipping mark. The templates are sent back to supplier. At supplier, the templates are then filled with further information about case contents, dimensions, weights, and case types. Non-Finnish suppliers also receive booking templates, to be filled in and informed to Wärtislä's inbound third party inbound service provider (3PL) for pickup. Final packing lists are marked on physical cases and sent back to Wärtsilä purchasers. Because goods leave supplier's depot with Wärtsilä's HUs, inbound transportation to consolidation warehouses can be tracked using HUs (see figure 1 for the complete process).

In order to manage the stream of inbound equipment, as soon as HUs are available in SAP, they are collected into Shipments through addition of Outbound Delivery numbers. From now on Shipment, Outbound Delivery (OD), and Handling Unit (HU) mean SAP-related documents. Decision about the number of shipments and departure times are contractually bound. Therefore Shipments are often created during project kick-off and their statuses are updated throughout project execution. When ready, shipment contents are printed out into a case specification. A case specification lists detailed dimensions and weights (gross and net) of all constituting HUs. The job of creating and updating Shipments belongs to Delivery Managers. Trade & Finance people also contribute to Shipments by updating payment statuses as well as letter of credit (L/C) statuses if applicable. Upon receiving green lights from Delivery Manager and from Trade & Finance, Transport Managers begin transport management process, starting from quotation request to cost settlement. In most cases a final case specification is required in order to start transportation planning. However preliminary case specifications are also used for large shipments that require permits and dedicated transportation means.

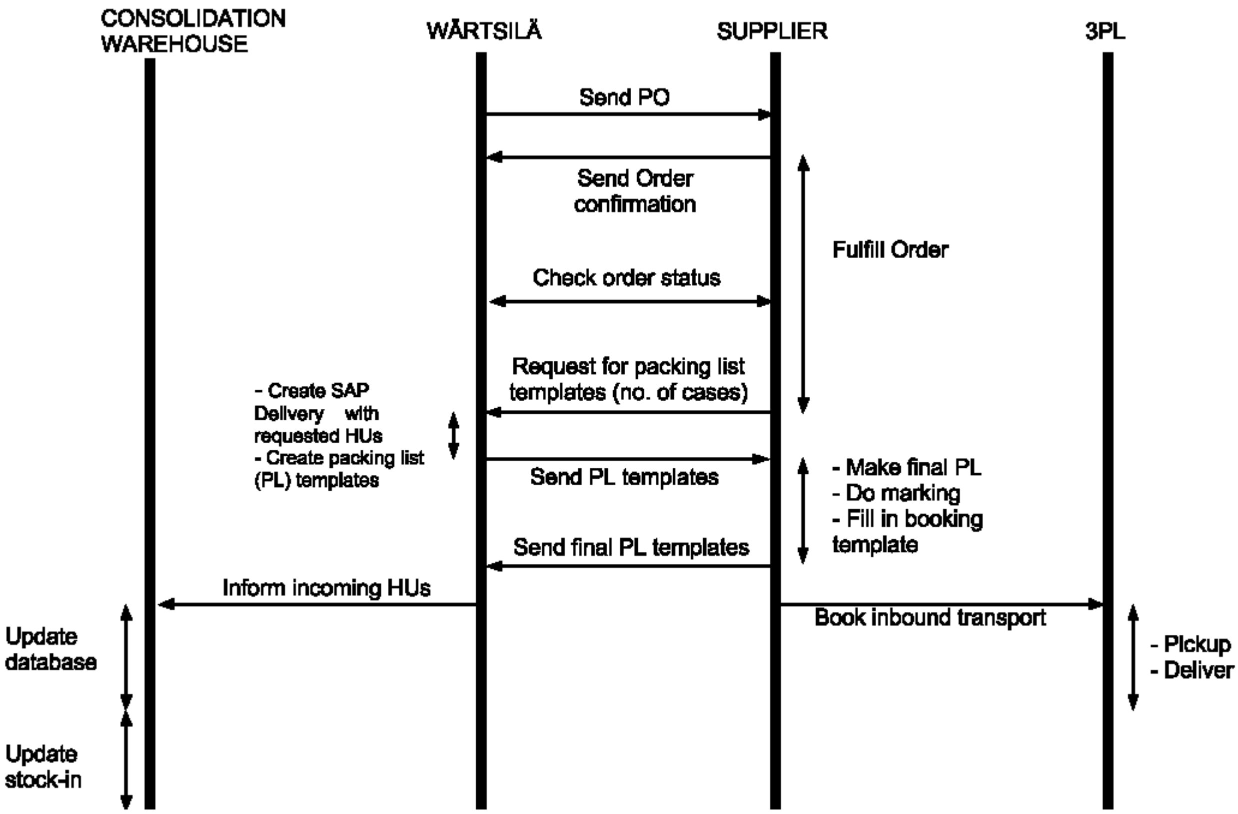


Figure 1. Materials management process in Wärtsilä Ship Power.

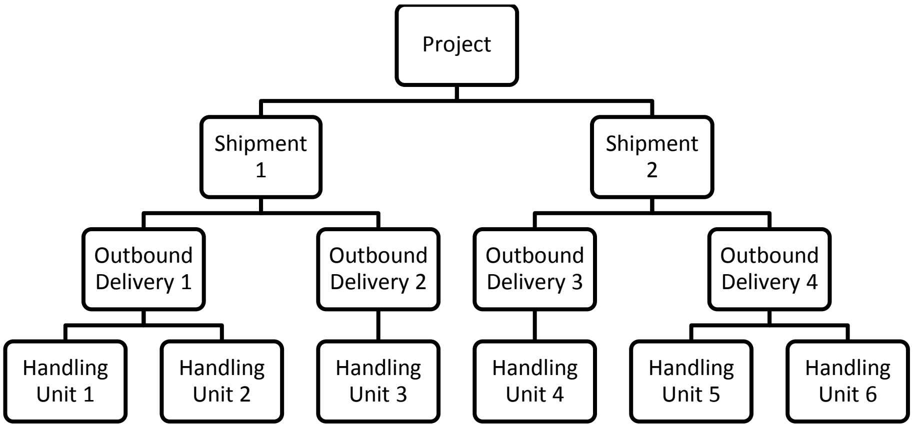


Figure 2. Organizing Handling Units into Shipments.

Overall roles and tasks of the transport control tower are as follow:

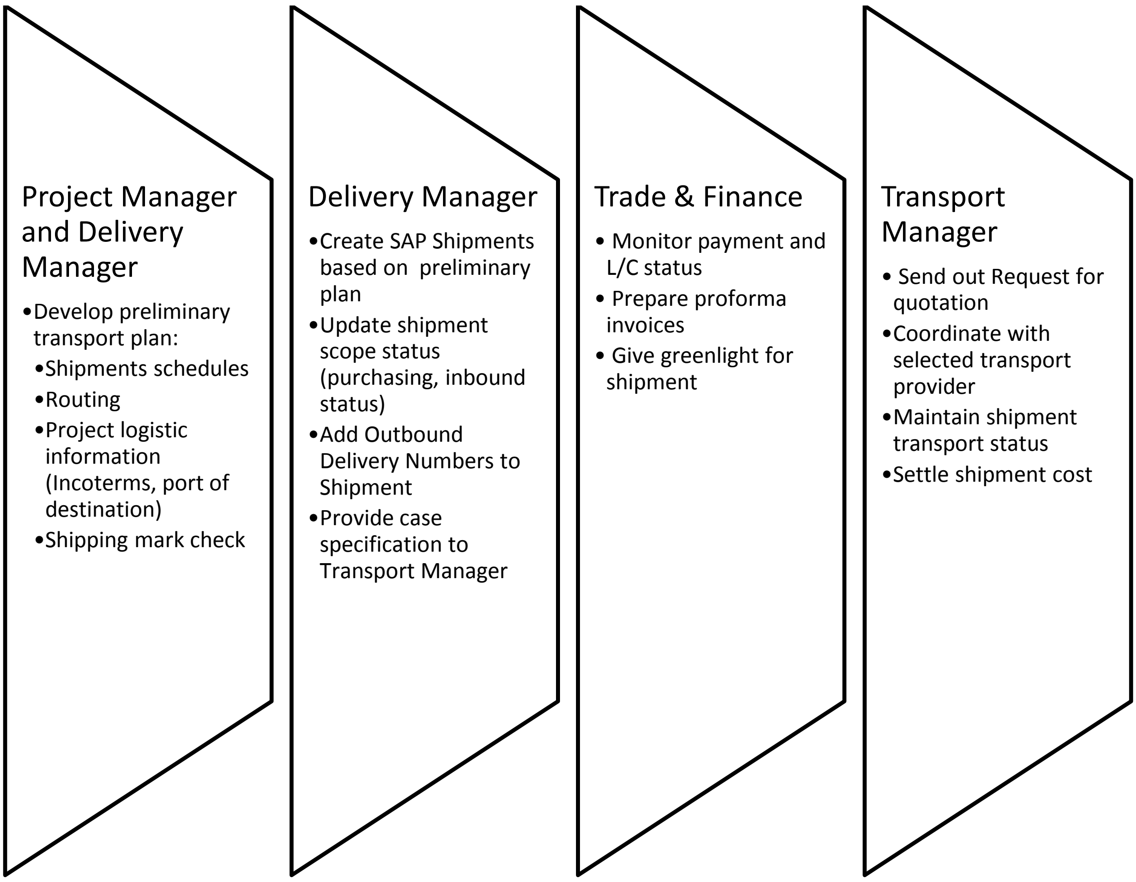


Figure 3. Wärtsilä Ship Power logistics execution.

As seen in figure 3, Delivery Manager, Trade & Finance, and Transport Manager together form a “logistics control tower” (LCT) that updates different aspects of shipments in SAP. During project execution, project management and LCT meet formally in one or more delivery check (PDC) meetings whereas the latest information about customer agreement, scope, payment, and contractual data are discussed and updated in SAP. Outside the PDCs, project people can also monitor shipments progress through a number of SAP transactions. However project management often contacts LCT directly for shipment-related questions. Increasing number of shipments (see figures 4 and 5) and scope complexity has created a desire to have a display that can automatically consolidate all the logistical details into more concise shipment statuses.

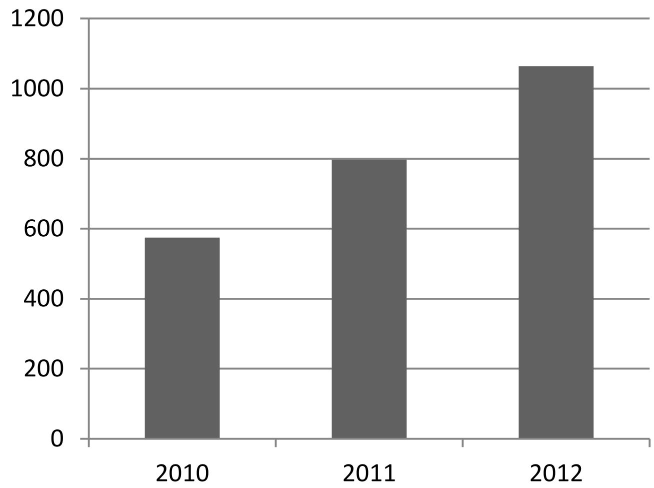


Figure 4. Wärtsilä Ship Power shipments during 2010-2012.

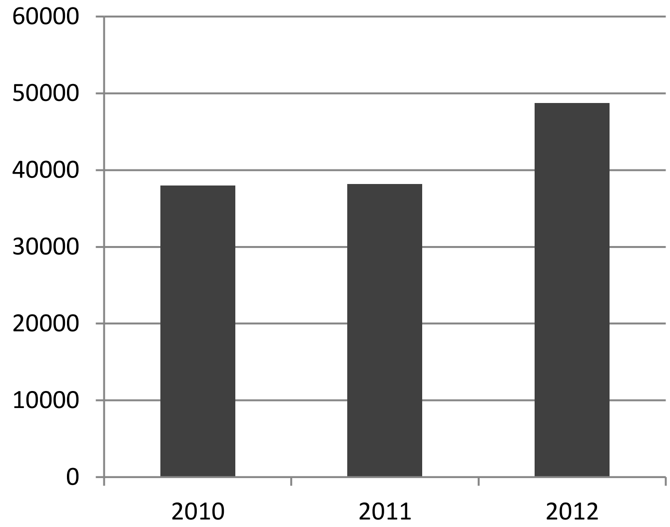


Figure 5. Wärtsilä Ship Power total amount of HUs shipped during 2010-2012.

2.2. Global project logistics and supply chain management information systems

2.2.1. Global project logistics and supply chain management

Projects are unique as they exist to accomplish definitive sets of goals and objectives that satisfy the need definitive groups of stakeholders (Heldman, Baca & Jansen 2007: 41). Projects are complex in term of technical, financial, political, and social factors (Sandhu & Ajmal 2012: 9). Finally, projects are temporary in economic relationship between stakeholders, dictated by their definitive starting dates and ending dates. Project-based business (PBO) deploys projects to achieve its major business objectives. This includes all firms which design and produce complex products and systems (Sandhu & Ajmal 2012: 8).

Supply chain is defined as a “goal-oriented network of processes and stockpoints used to deliver goods and services to customers” (Hopp 2003: 1). Each process represents an individual activity that adds value to the materials or services flow through it. A stockpoint represents a location in the chain where inventories are held. A network is an arrangement of processes and stockpoints into paths that materials and services flow through the supply chain (Hopp (2003: 2). In terms of project logistics, a supply chain network can be illustrated as in figure 6.

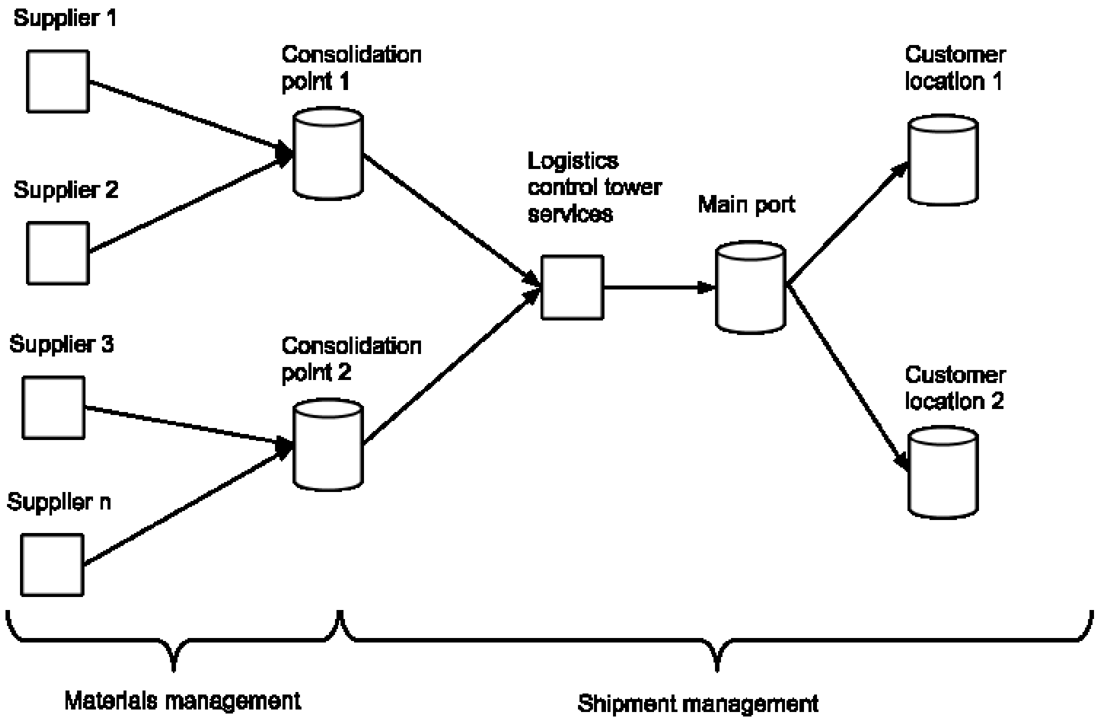


Figure 6. A project logistics network.

Supply chains differ by the amount of processes & stockpoints, the networks formed by these, as well as the choices of cost and service (quality, speed, and flexibility) to fit with the strategies. Supply chains can be categorized as either push or pull. According to Hopp (2003: 80) a pull supply chain releases work based on the status of the chain, while the push supply chain does not, resulting in uncontrollable amount of work in process (WIP) for the push system. It is found that a pull supply chain is more efficient than the push system. In fact, a pull supply chain achieves higher throughput at any level of WIP than a push supply chain (Hopp 2003: 86). Furthermore, a pull supply chain is also more robust than a push supply chain as it is less sensitive to errors in releasing work to the chain (Hopp 2003: 87). A pull supply chain is more likely to become lean than the push system. The reason is as follow: a supply chain is lean if it operates with the minimum amount of buffering costs (Hopp 2003: 75). Buffer is a combination of inventory, flexible capacity, and time. It exists to covers variability in the supply chains. In order to reduce buffers, variability has to be reduced. As a pull supply chain places explicit limit on the WIP, processes with high variability will be more likely to be identified than in a push supply chain. Therefore corrective actions can be more correctly done to reduce variability. The pull supply chain is a better supply chain than the push one, but the lack of transparent information coordination has pulled many supply chains into the side of push.

So far supply chains have been viewed as one entity with either push or pull characteristics. Supply chains are also viewed from the perspective of the chain master (the buyer). Managing the supply chain from the buyer point of view is generally divided into two areas: materials management (inbound) and shipment management (outbound). Lambert & Stock (1993: 449) defined the process of managing an influx of materials into the firm as materials management. Although this concerns mainly with purchasing tasks, materials management also includes inbound transportation. In the context of project logistics, most materials purchased are finished products from suppliers, to be consolidated in predefined consolidation points. Outbound transportation of equipment from consolidation points to end customer locations is managed through shipments and by a logistics control tower. The control tower offers centralized services such as transport quotation & booking, trade & finance control, and invoicing & documentations. Inbound transportation is generally less controlled than outbound transportation as "purchasing procedures tend to look at "total delivered cost"" (Lambert & Stock 1993: 454). Rather than having a dedicated logistics control tower as with outbound shipment, inbound logistics activities are outsourced to third party logistics (3PL) providers who offer more logistical competency and even dedicated facilities as found in some long-term business relationships (Farahani, Rezapour & Kardar 2011: 72). Most 3PLs do not physically transport goods but purchase transportation service from carriers. Besides being a pure freight forwarder, 3PLs also make profits by offering extra services such as customs clearance and packaging/labeling. Some 3PLs have their own fleets that operate on established routes to further achieve cost efficiency. At that time they are called forwarders acting as carriers. 3PLs optimize costs also by pooling loads from different shipments of different consignors (Lauterbach, Fritz, Gottlieb, Mosbrucker & Dengel 2009: 19-21). Because of this, according to Farahani et. al (2011: 76-77), companies (consignors) contracting for 3PL receive several benefits such as: 1) save time while receive better services, 2) focus more on competitive competencies while exploit external logistical expertise, 3) gain advantages of economies of scale and scope while reduce in-house inventory management costs. By contrast, poor information sharing, loss of customer feedback, loss of control and losing in-house logistics capabilities are among those disadvantages that firms may face when outsourcing logistics activities. Unlike inbound transportation, outbound transportation often requires in-house control due to following reasons: 1) payment has to be secured prior to release of good, 2) contractual Incoterms mandates how risks, costs, and customs formality are shared between buyers and sellers, 3) contractual maximum amount of shipments, 4) time & cost of global transportation encourages shippers to combine more shipments to similar destinations for economies of scales, and 5) contractual delivery time that dictates when the shipment must leave even though its scope may not be ready.

Shipper's objectives to consolidate shipments and end-customer's request for limited amount of shipments have formed shipment batches. Those batches are simultaneous batches as they are processed at the same time without causing further delay to process time. The process time here means transportation time. For example, transportation time of a container vessel from Europe to China is not affected by the amount of containers on the vessels, as long as the vessel can still accommodate those containers. Simultaneous shipment batches are further characterized by the fact that equipment arrives at consolidation points in different times. According to Hopp (2003: 33) shipment cycle time will increase proportional to its scope (batches sizes). Therefore although shipment scopes are contractually bound, sometimes it is not useful to wait for the full scope readiness. In this situation logistics control tower must have information of all equipment inbound statuses as well as payment status ready for project manager to make decision whether to release shipment. So far only physical flow of equipment in and out of organization has been discussed. Information flow is the second flow that is critical for accurate and timely decision making and reporting. Project based organizations conducting projects with several partners require higher level of inter-organizational communication than non-project organizations (Sandhu & Ajmal 2012: 24) to handle complexity.

2.2.2. Information management within supply chain

In order to effectively link physical flow to information flow in the supply chain, there must be a standard for identifying, capturing, and sharing logistics unit upstream and downstream of the supply chain. GS1 standards for transport & logistics (T&L) exist to provide an integrated system of global standards. They enable parties to identify, capture, and share items, locations, and events. The most important GS1 T&L identification key is the Serial Shipping Container Code (SSCC), which is a unique worldwide number for a logistic unit. An SSCC is an 18-digit number, composed of a prefix, a GS1 Company Prefix, a serial reference unique number, and a checksum. While the Company Prefix is uniquely and globally recognized by all supply chain partners and is therefore maintained by GS1, the serial reference numbers are maintained by the company. SSCC as part of ISO 15459 provides global unique identification to a logistic unit (GS1 2008: 41). SSCC uniqueness remains at least one year from the date of issue but can last as long as requested by the industry (GS1 2008: 33). When used in tracking and tracing logistic units throughout the supply chain, SSCCs do not only reduce complexity in logistic unit identification but also in relabeling. This due to a duo-nature of GS1 labels: readability by both computer systems and human. One example is an SSCC label for a pallet of good as shown in figure 12. The use of prefixes helps human readers understand the meaning of numbers: “(00)” indicates that trailing digits form an SSCC. Also as can be seen in figure 12, partners in the supply chain can retain SSCCs while relabeling logistic units with next destination information. In logistics scenarios whereas manufacturer requires tight control over labeling of cases deliver from material suppliers, relabeling at manufacturer consolidation warehouse will reduce time as a) no waiting needed from supplier for correct shipping mark from manufacturer purchaser (see figure 1 for the process) and b) total time for mass relabeling at consolidation warehouse is less than the sum of individual labeling time as supplier. While GS1 labels are readable by human beings, they are design for machine-captured mainly through barcodes and RFID (figure 10).

Besides manual input data recognition methods such as writing and typing, there are also technologies for machine reading, termed as Auto Identification and Data Collection (AIDC) (Symonds & Parry 2008:3). AIDC can include barcode, optical character recognition (OCR), vision systems, magnetic stripes and optical cards, synthesized voice, and radio frequency (RF) data communication. Farahani et. al (2011: 237) reported a study from US Department of Defense showing error rate for different input recognition methods. RF is found to have the least error rate and is the quickest and most robust method as show in table 2.

Table 1. Error Rates of different input recognition methods. Adapted from Farahani et. Al (2011: 237).

Written Entry	25000 in 3 million
Keyboard Entry	10000 in 3 million
OCR	100 in 3 million
Barcode (code 39)	1 in 3 million
Transponders (RFID tags)	1 in 30 million

Table 2. Time taken to read a Handling Unit. Adapted from Ehrs et.al (2011: 17).

Technology – tag combination	Time (m)	Note
RFID reader – RFID tags	2,5	N/A
Barcode reader – normal barcode	6	Integrated light-source
Mobile (iPhone) - normal barcode	11	Failed to read in dark
Mobile (Nokia 5800) - QR barccode	22,5	Failed to read in dark

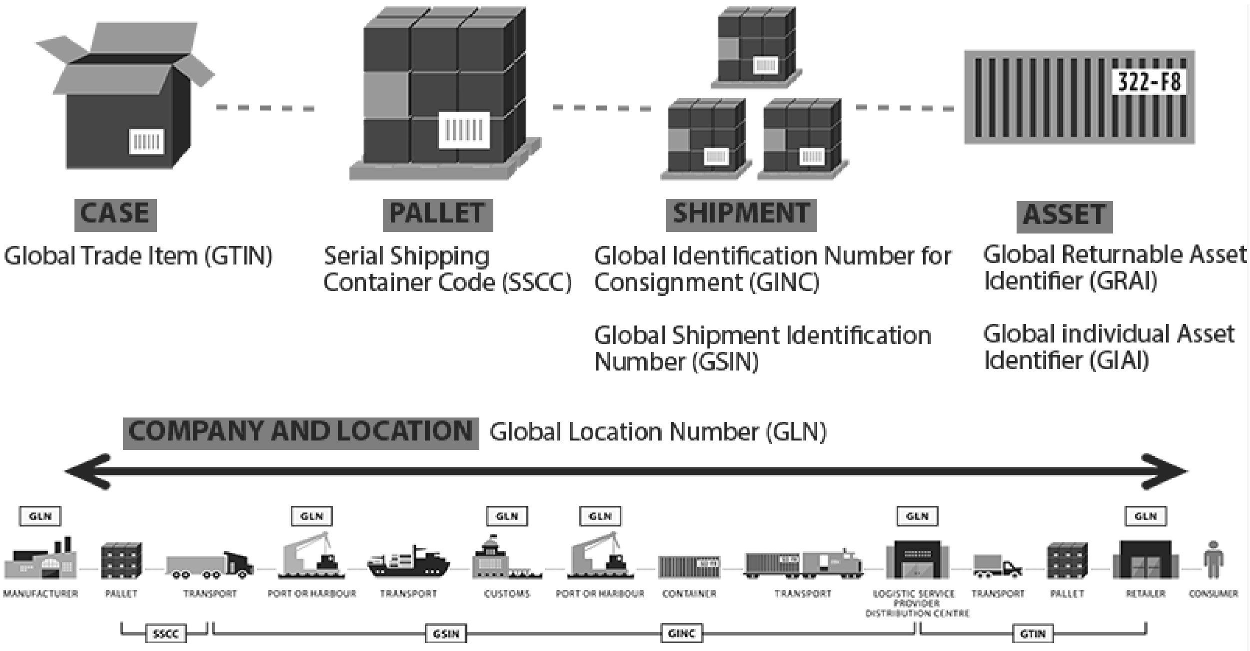


Figure 7. GS1 standard identifications in transport & logistics (Adapted from GS1 2013).

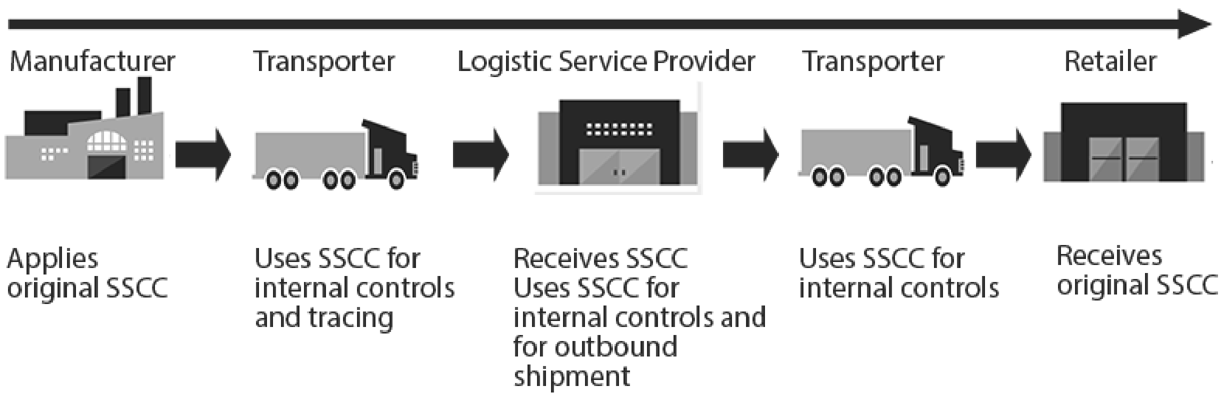


Figure 8. Track & Trace a SSCC unit in a supply chain (GS1 2013).

Figure 9. Example of a GS1's SSCC label (GS1 2013).

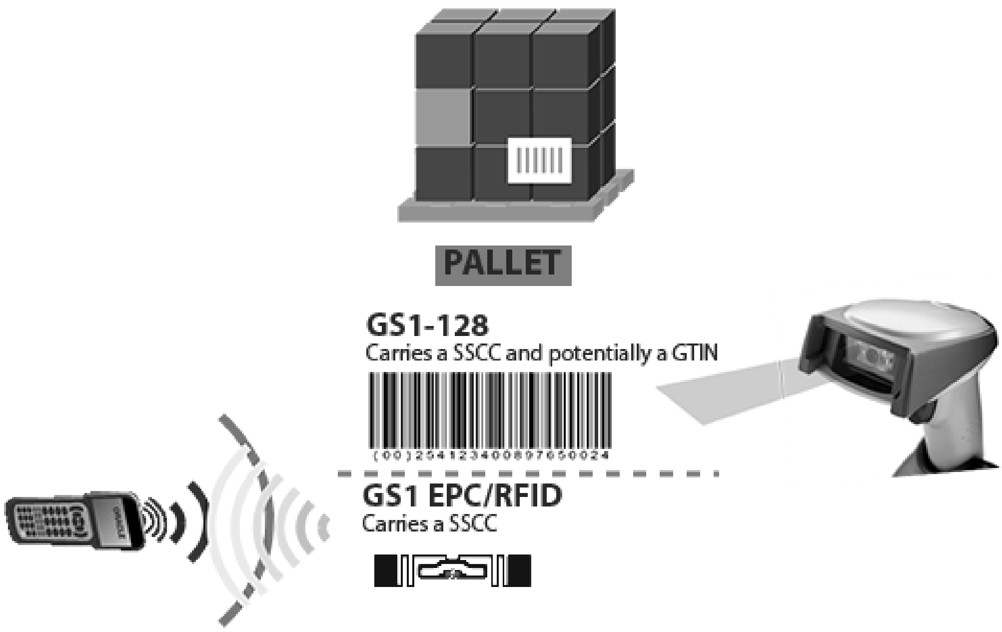


Figure 10. GS1's SSCC can be machine-captured by Bar Code or RFID (GS1 2013).

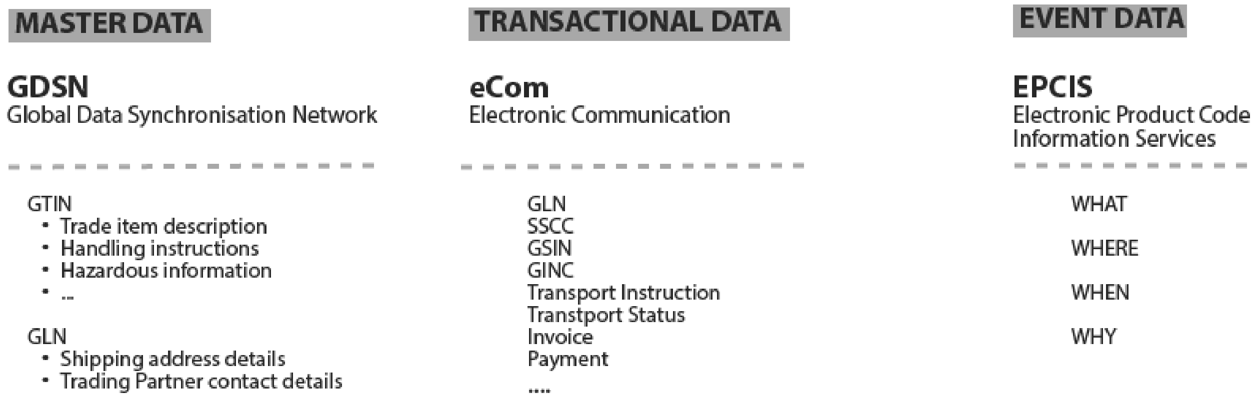


Figure 11. GS1 standard for electronic data sharing within supply chain (GS1 2013).

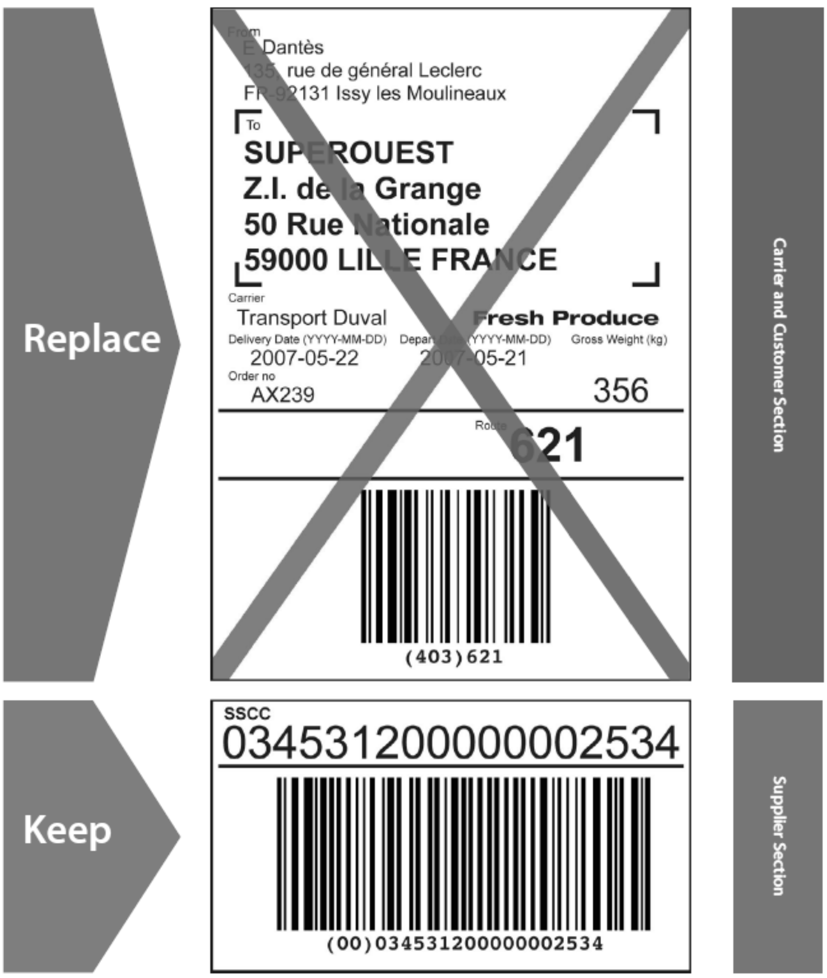


Figure 12. GS1 logistic label with ability to be relabeled (GS1 2008: 35).

2.2.3. Information management within a company

Advanced information systems are used in PBOs to manage resources and activities. Manufacturing tends to be more structured than other businesses (Corcoran 1998). In order to manage information, modern large manufacturing companies employ enterprise recourse planning systems (ERP) to consolidate data from scattered business units into a common data warehouse, from which user interacts through a set of standardized user interfaces. A common data warehouse links information together and ERP transactions present a global picture of operations for faster and more accurate decision making. ERP is a developed system with history dated back from the 1960s.

The development to ERP systems started in 1960s when MRP (Material Requirements Planning) was developed. MRP's purpose is to ensure materials availability for production while maintaining a low level of inventory and to plan manufacturing, delivery and purchasing activities. At the beginning MRP was conducted by pen and paper. During the 1970s MRPs were run by computers. In the 1980s MRPII (Manufacturing Recourse Planning) was developed. MRPII provided a more integrated system for the enterprises. This means a bigger part of enterprises functions are managed by a single system with a centralized database. However in the 1980s computers were not powerful enough to for tasks with large amounts of data. The term ERP (Enterprise Recourse Planning) was born in the 1990s. ERP further improved MRPII. ERP systems are designed to handle all types of data from all functions of the enterprise. In another word ERP unifies different information systems under one umbrella. ERP software typically has different modules for different functions of the enterprise. SAP AG is a German software company founded in 1972 that produces ERP software. SAP R/3 consists of modules that can be purchased and run separately. The most important modules are Project System (PS), Production Planning (PP), Materials Management (MM), Sales and Distribution (SD), and Financials & Controlling (FI-CO) (Williams 2008:21). SAP R/3 further offers country-specific modules which provide interchangeable currencies and language setting. In a fully integrated environment, different SAP modules interact to provide different services for materials management, as shown in figure below. Project management and logistics are managed in SAP R/3 as illustrated in figure below. Each of the stage is identified by a unique identification number, such as Sales Order number, Project number, Purchase Order number, or Handling Unit number.

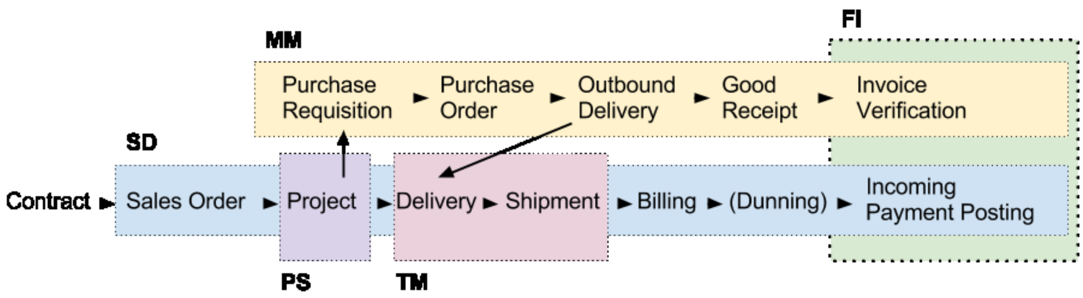


Figure 13. SAP R/3 materials management workflow.

Complexity and inflexibility have often been attributed to ERP systems (Hopp & Spearman 2000: 143). ERP systems such as SAP R/3 are organized into transactions. This sometimes makes simple tasks complicated. There is much data to be filled into different tabs in order to complete a transaction. This complexity distracts users from the core business and eventually may decrease work quality. Despite of the drawbacks, integration, integrity, and coherence make ERP the choice for large manufacturing firms. Furthermore, according to Hopp & Spearman (2000: 144), "many companies using SAP have changed their management procedures to conform to the software best practices". Although ERP software providers are adding more and more modules to existing platform to cover increasing aspect of business environments, they cannot cover all businesses needs. Furthermore, expensive license and high cost of implementation prevent companies from adopting new ERP modules. Another aspect is the complex integration between ERP systems, which ultimately prevents data exchange between companies. Considering project logistics business whereas inbound transportation lead time can last several weeks in multimodal transportation, status disconnection between systems adds uncertainty to outbound shipment transportation planning.

Similar to GS1's SSCC, SAP assign HU numbers to uniquely identify logistic cases. When an organization adopts GS1 standards, SAP then converts HU into SSCC formats, ready to be shared within the supply chain. Although possibilities exist for higher integration of information within supply chain, high difference within supply chain information infrastructure has led to a hybrid system whereas third-party web service is used for data integration. Figure 14 explains a scenario in which data in a company's materials process is managed through a third-party Internet cloud service. Through the service, parties in the supply chain contribute to different aspects of handling unit numbers. When an order is ready, supplier informs total amount of handling units, their dimensions, contents, and pickup details. Forwarding company then informs pickup schedule and even delivery status of handling units. Consolidation warehouse then informs stock-in/out of handling units. This setup ensures a minimal change to existing information systems, maximizes scalability, and also maintains higher data integrity throughout the supply chain.

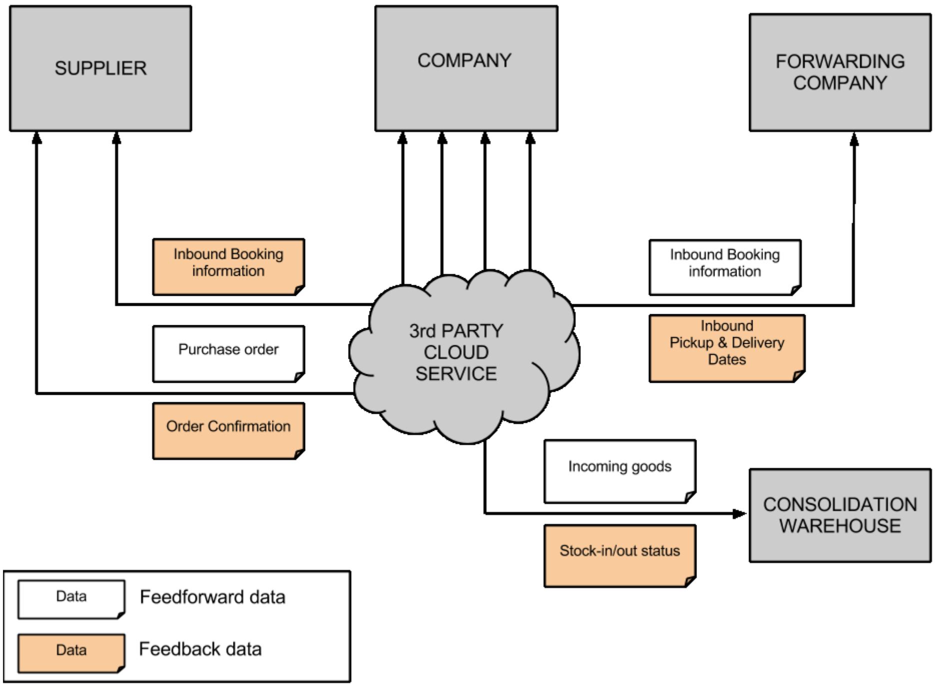


Figure 14. Information exchange between parties in materials management.

In the context of global project logistics, both inbound and outbound transportation lead times can be long, disparate, and complex in term of supplier readiness times and transportation modes. Therefore, being able to "track and trace" HUs would be highly desired especially when consolidating multi-origin HUs into multi-modal shipments (Kärkkäinen, Ala-Risku, and Främling 2004: 546). Shamuzzoha & Helo (2011: 243-244) defined tracking as the collection and management of present location of a physical unit, and tracing as the ability to keep and retrieve location history of a unit, including its origins and final destination. Track & trace happen in both intra- and extra-organization (Symonds & Parry 2008:3). Handling units (HUs) are used for tracking. HUs can be tracked through checkpoints in supply chain network using near-field input recognition methods, or tracked on the go using satellite positioning systems such as GPS or GLONASS, with transmission to tracking servers using GPRS. Thanks to its world-wide coverage, satellite positioning would be the method for real-time tracking but tracker's short self-sustaining time, trackers cost, and remote areas with no GPRS coverage have undermined usage of this method (Nguyen 2010: 64). Furthermore, the bulk size of trackers makes it difficult to position them inside cases (Shamsuzzoha, Tenkorang, Nguyen & Helo 2011: 9). In contrast, most near-field identifiers are very compact and long-lasting. However, relying solely on near-field recognition methods at checkpoints requires tremendous coordination efforts within the supply chain and is prone to mistakes due to different input recognition means at checkpoints. Furthermore, investing in system integration is generally not feasible for large, one-off supply networks such as mechanical engineering delivery projects (Kärkkäinen et.al 2004: 552). Therefore hybrid models are available. A HU therefore could be traced by both near-field trackers (e.g. plaintext identification number, barcode, RFID) and satellite trackers (e.g. GPS, A-GPS, and GLONASS). In this setup, near-field trackers are used at known checkpoints and satellite trackers are only initiated during transportation between checkpoints. For example, He (2009: 3) proposed a system with RFID for in-house and GPS tracker for between-house track & trace.

ERP system together with external track & trace data provide good sources of data for updating in-progress shipments statuses. Figure 15 presents a situation of a multi-scope, multi-shipment project that leaves from a consolidation warehouse. Shipments are created at the beginning of the project to capture project delivery status. In SAP, as a Shipment consists of HUs, its scope status is best determined by statuses of the individual HUs. Unfortunately there are two pitfalls: late creation of HU and unknown amount of HUs. Deliveries and HUs are created only when goods are ready at suppliers for pickup. In addition, the number and time of availability of Deliveries and HUs are unknown until the time of goods readiness as it depends on suppliers' order fulfillment progresses and ways of packing equipment. Therefore monitoring a Shipment status based on a complete set of packed HUs generally means tracking their inbound statuses. However, this period lasts only a few weeks before the shipment's planned start date. In order to monitor shipment statuses, other documents have to be also used. Prior to the availability of HUs, a Shipment status is monitored by its Purchaser Orders (PO) statuses. Prior to the availability POs, a Shipment status is monitored through Purchase Order Requisitions (POR). Unfortunately both PORs and POs are not directly linked to Shipments in SAP. This is why Shipments statuses are manually updated by Delivery Managers.

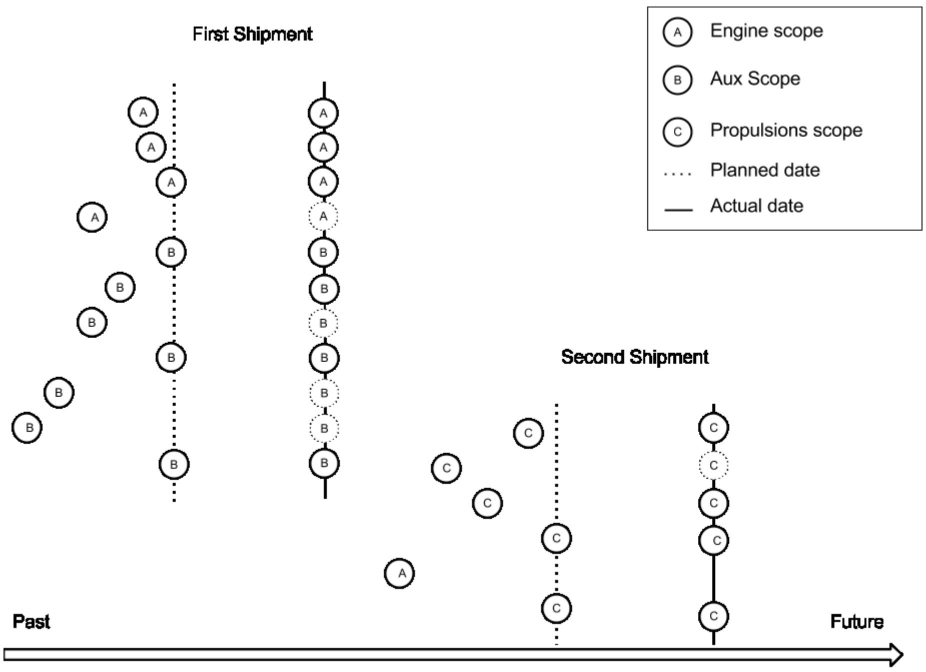


Figure 15. Arrival of Handling Units at a consolidation point for different shipments with different planned start dates.

2.3. Software Design and Development Approach

Building an information system takes four phases (Sage & Amstrong Jr 2000: 23; Giachetti 2010: 120; Sommerville 2006: 8): system definition, system development, system deployment, and operation & maintenance (Maciaszek & Long 2005: 6). After this, if the system is evolved then a new cycle is repeated; otherwise, it is phased out. These phases can be sequentially (Waterfall approach) or incrementally (Spiral approach). The Waterfall approach was introduced by Royce (1970) with an attempt to introduce moresteps to the traditional two-step software development method of analysis and coding. The approach emphasizes completion of a step before progressing to the next step, with low tolerance of change to preceding steps as the development progresses downstream. Unlike the name suggests, Waterfall approach does allow iteration. However, Royce (1970: 328) had limited it to only between immediate preceding and following steps. Waterfall also emphasizes simulation but with the assumption that requirements are known and will not be changed. Despites of drawbacks, the Waterfall approach is straightforward, honors the first-time-right way of working to minimize feedbacks, resulting in reduced total development time. Quite different to the Waterfall approach, Spiral approach promotes prototyping as a means to better capture requirements and risks. When reaching the last operational prototype, functional requirements are well captured. After that the Waterfall development approach can be applied (Boehm 1988: 65). Software architecture in Spiral approach is scaffolded from the most general structure, with more details added as the development proceeds. Therefore the approach can be considered as decomposition and zigzagging between functional requirements to technical design parameters and to physical process steps. Axiomatic design provides a process to decompose and zigzag between domains to maintain the highest synchronization between functional requirement and physical artifact (software). According to Suh (1997: 76) the process starts with identifying customer needs and attribute (customer domain). Then functional requirements with their constraints are identified. Next functional requirements (FR) are mapped into a physical domain, which determines the design embodiment and design parameters (DP) that satisfy both functional requirements and their constraints. Finally the process (PV) used to satisfy design parameters are chosen. In software development, the lowest-level DPs are the input variables to the system that determines the output of the function (process) (Suh 1997: 78). Axiomatic design is therefore iterative because designers deepen application details of the left domain based on the ideas generated from the upper level of the right domain (see figure below) (Suh 2006: 454). According to axiomatic design principles each FR is an independent requirement that completely characterizes a functional need of a product (2006: 455). Furthermore, each FR has to be maintained independent through a proper choice of DPs (Suh 2006: 456). In the best design (uncoupled design) a DP chosen for a FR will not affect performance of other FRs. Furthermore, the best designs always have equal amount of FRs and DPs, as they will be coupled if the amount of DPs is less than FRs and will be redundant if the amount of DPs is more than FRs. When there are multiple DPs that satisfy an FR, the one that has the least information content will be chosen. Therefore the solution that is most likely to succeed is the simplest solution (Do & Suh 1999: 121). Having independent system elements with explicit and known interfaces supports concurrent development and maintenance, thus also supports resources utilization.

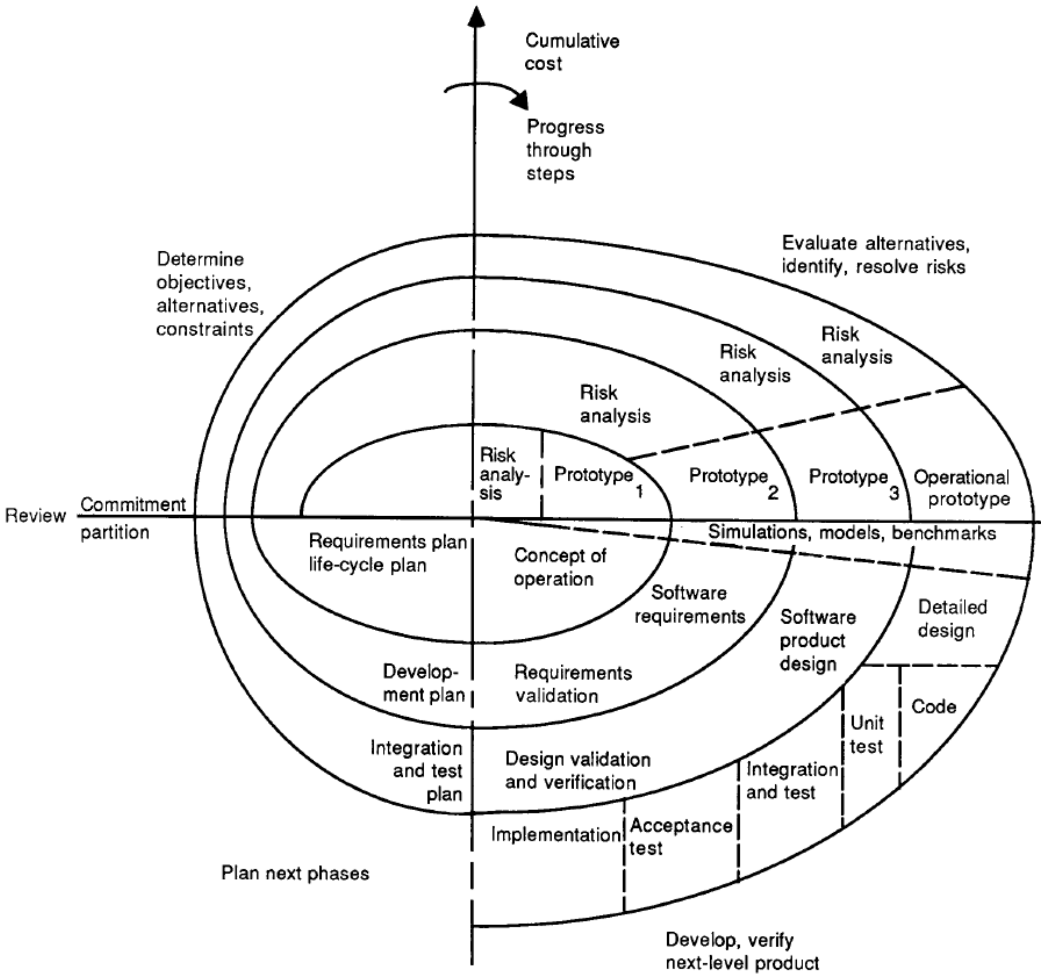


Figure 16. Spiral software development approach (Boehm 1988: 64).

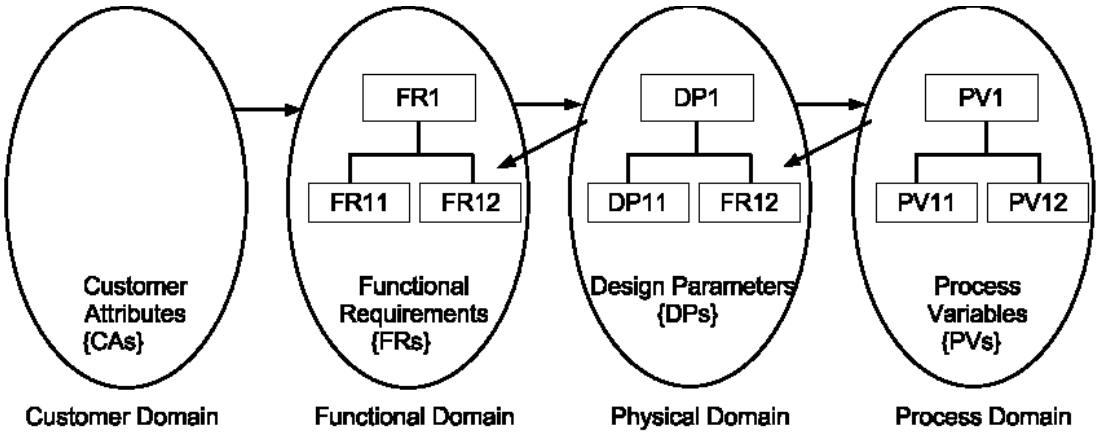


Figure 17. The concept of domains, mapping, zigzagging, and decomposition to develop system (Suh 1997:3).

3. DEVELOPMENT OF THE REAL-TIME SHIPMENT STATUS DISPLAY

3.1. First iteration

3.1.1. Voice of customer analysis

Analysis of existing systems and way of working shows that there is a gap in communication between logistics control tower (LCT) and project management (PM) regarding shipments status. While LCT is keener on managing shipments in SAP, PM mainly uses SAP for purchase requisitions, not for shipment execution status monitoring.

Table 3. Project management is managed through different information systems.

Group	Main systems in use and way of working
Project Management	- Use SAP for purchase requisition - Use another dedicated project and portfolio management tool for milestone/gate update - Use company-wide documents system to upload contractual and technical documents
Logistics Control Tower	- Use SAP to create and manage shipments - Use cloud service to manage inbound status - Use company-wide documents system to upload shipping and financial documents

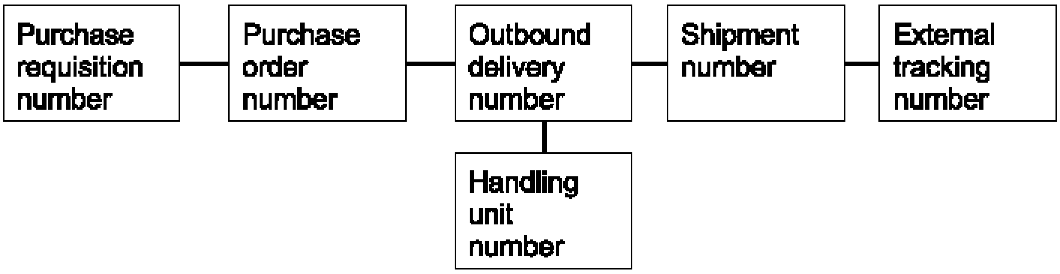


Figure 18. SAP documents involved in a typical SAP project logistics execution.

In order to support traceability, during project execution different SAP documents are created. Each document is created and maintained by a specific operational group whose tasks have been described in figure 1 and 3. There is a transaction in SAP which groups all these documents into a scope of supply report. Unfortunately complicated relationships between SAP documents and the lack of knowledge of different SAP document reduce the usefulness of the report.

Description	Item description	Purchas...	P	Order nu...	Vendors co...	Req. dat...	Order date	Req. dat...	Confirme...	GR Date
Other Equipment	Flexible pipe connections					22.11.2012				
	Flexible pipe connections spare set									
O&M Manuals & ELDOC	ELDOC									
	Engine manuals	18277211	10			15.07.2013				
Fuel system	MDF Suction filters	18047961		4502409057	VSM/3997703	28.06.2013	22.04.2013	20.06.2013	20.06.2013	17.06.2013
	Classification certificate: ABS									
	Cooler (MDF)	18047960		4502404276	2047356		16.04.2013	19.06.2013	19.06.2013	19.06.2013
	Classification certificate		20							
Cooling water system	Preheating unit	18047962	10	4502417585	VE13005661		30.04.2013	28.06.2013	28.06.2013	
	Classification Certificate: ABS							28.08.2013		
	Central cooler for W8L20	18047993		4502417662	27335258			28.06.2013	20.06.2013	20.06.2013
	Classification certificate : ABS		20							
	Motorised thermostatic valves	18047994	10	4502421188	PIT-380783		06.05.2013		23.06.2013	26.06.2013
	Seaworthy Packing									
Starting-air system	Starting air vessel	18047995		4502421225	NK 3.8036			20.06.2013	25.06.2013	03.07.2013
	Classification certificate: ABS		20						20.06.2013	
	Air filter (starting air inlet)	18047996	10	4502421248	13161328 S1			28.06.2013	28.06.2013	25.06.2013
	Seaworthy Packing									
Comb. air and Exhaust...	Exhaust gas silencer with spark arrestor	18047997		4502421401	72483					28.06.2013
	Turbocharger cleaning device	18213805		4502458045	40006094	08.07.2013	18.06.2013	26.06.2013	25.06.2013	26.06.2013
	Exhaust gas bellows	18213817		4502458058	40007077					27.06.2013
Engine Control and mo...	Power Unit	18047998		4502421467	800040	28.06.2013	06.05.2013	28.06.2013	28.06.2013	01.07.2013
	Starters for electric motor driven pumps	18047999								
Electric Power Generati...	Emergency genset -440HS19L6RC	17737672		4502314575	A16848	15.05.2013	08.01.2013	15.05.2013	19.06.2013	27.06.2013
	Spares and tools									
	Classification ABS		20							
Power Transmission S...	CENTAMAX CM-8000-SDE-72-21	17878197	10	4502346649		01.04.2013	11.02.2013	27.03.2013	10.04.2013	10.04.2013
	Classification ABS		20							18.04.2013
	Generator AvK DSG99L1/8	17737355	10	4502301590	81 31505 A0...	31.03.2013	17.12.2012		24.05.2013	22.05.2013
	Spare parts								01.05.2013	
	Generator Loose Parts									

Figure 19. Screenshot of a SAP project scope of supply report.

One of logistics control tower (LCT) tasks is to check scope of supply reports and summarize them into shipments. LCT also have a SAP report for their shipments. As LCT constantly update SAP shipment statuses, PM is expected to check SAP shipment report by themselves. In practice PM seldom checks this list because of complexity in user interface with poor semantics support. For example in shipment search view, project numbers are to be filled in the field labeled “External ID 1”. Another reason for not using this report is because of redundant shipments view which includes irrelevant columns. Although views are configurable and layouts can be saved, their configuration is a trial-and-error process. For example the field labeled “Addit. text 1” is used for updating shipment status.

△	Shipme	Su	Ext. ID 1	Addit.text 3	PlanShipSt	Descrip.of Shipment	Addit.text 1
	127902	R…	SP/00117	SHI 1946		Lamp cover for stand	Scope complete - input is given
	58230	R…	SP/90908	Niestern Sander 843		flex side elements	Shipment to be checked (Scope in VCK ?)
	57672	R…	SP/00150	Ulstein 293	05.01.2012	REST/hoses, service	Transport booked
	58713	R…	SP/00083	Kleven Verft 346	13.02.2012	REST/O-rings	
	58444	R…	SP/00168	STX NOS 771	20.02.2012	REST/flex. hoses	GR expected (FCA) 20.01.2012 Stenflex
	58493	R…	SP/00083	Kleven Verft 346	23.02.2012	Additional spares	
	58906	R…	SP/91111	ESM 80	01.03.2012	SPARE PART SET 2	Scope complete - Input will be given
	58576	R…	SP/00083	Kleven Verft 346	05.03.2012	REST/Isolator	
	58856	R…	M/02939.M3	BACH DANG EG4500-B…	08.03.2012	Spares	Scope complete - Input will be given
	58798	R…	SP/00346	Kleven Verft 353	08.03.2012	REST/conn. rod	
	58960	R…	SP/91663	SHI 1813	08.03.2012	Temerature feeler	Scope complete - Input will be given
	58964	A…	SP/91570	Fincantieri 6201	09.03.2012	Zinc anodes	Scope complete - Input will be given
	59044	A…	SP/91601	U 32	15.03.2012	AO 8	Scope complete - Input will be given

Figure 20. Screenshot of a SAP shipment report.

Discouraged by the complexity of checking for project execution status, PM often try to get the information through email or direct talk to LCT. This is not encouraged from LCT point of view as they work hard to update the system so “all the data is in there (SAP)”.

Having all reports accessible only from SAP is disadvantageous for project people because they do not always have access to their office computer when travelling.

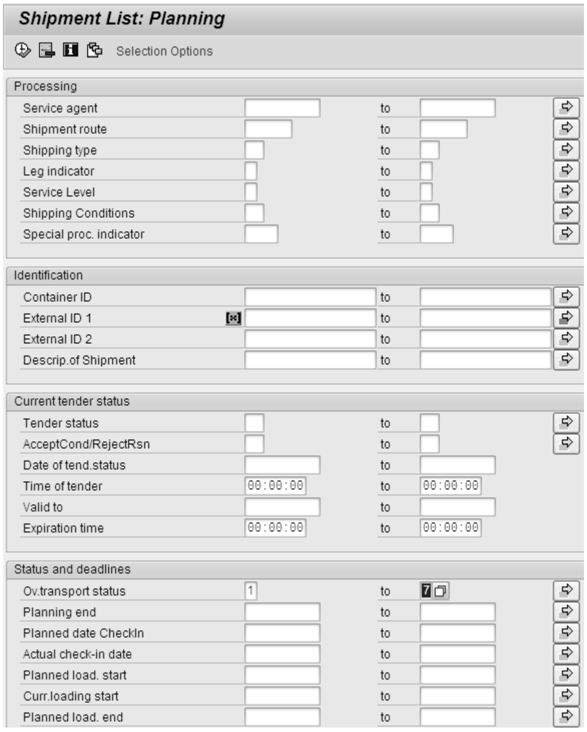


Figure 21. Complex criteria list prior to running a shipment report.

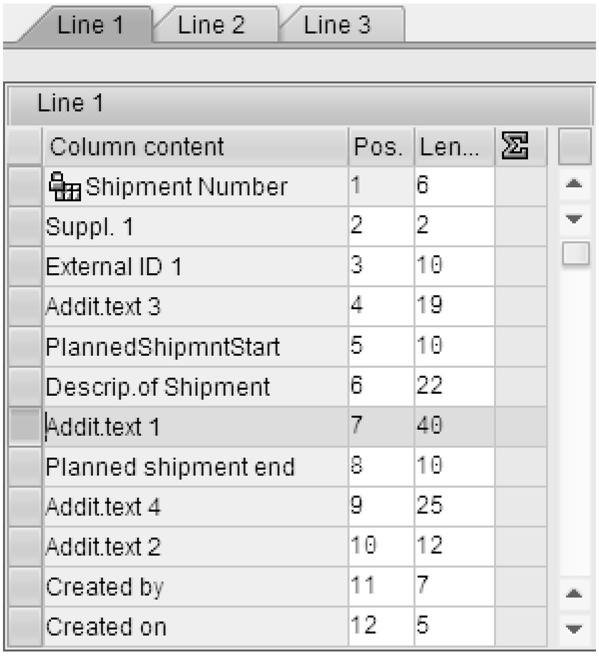


Figure 22. Poor semantics support in configuring a shipment view.

Analysis shows that the main user groups are project people who have below independent needs for the display:

- Right information, meaning that shipment statuses are communicated to readers with the least amount of data.
- Good data quality, meaning that the data is both accurate and up-to-date, with no change to existing infrastructure.
- Intuitive layout which is familiar to reader and at best requires very little time learn the layout.
- Easily accessible, meaning that the display can be accessed through different devices, with no additional software installation to users.

The application mission statement is stated as follow:
“To make a display system that help mobile people quickly informed about their shipments statuses”.

These customer needs (CN) are then mapped to independent functional requirements (FR) as shown in the figure below. Note that this is the axiomatic design matrix whereas an “X” denotes relationship between a pair of FR and DP while an “O” denotes no relationship between them.

	FR0: Functional Requirements (What)	FR1: Provide good data model	FR2: Provide efficient data I/O	FR3: Provide intuitive layout	FR4: Provide high accessibility
CN0: Shipments Status Display	X	O	O	O	O
CN1: Right information	O	X	O	O	O
CN2: Good data quality	O	X	X	O	O
CN3: Intuitive layout	O	X	O	X	O
CN4: Easily accessible	O	X	O	X	X

Figure 23. Top-level CN-FR mapping.

Together with CN-FR mapping, two systems constraints (CON) are also recognized and mapped to FRs.

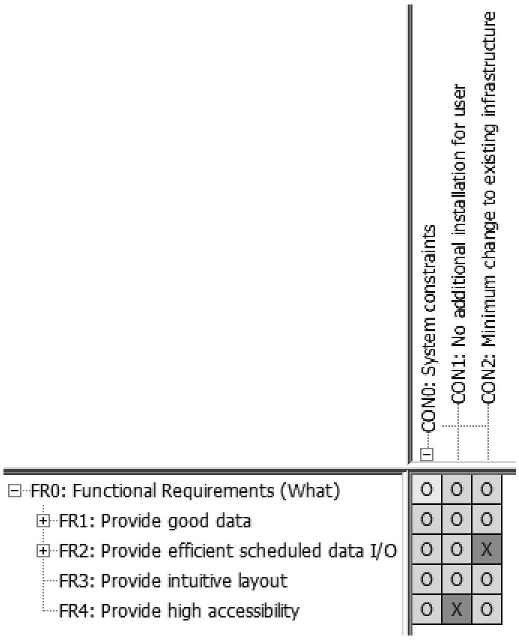


Figure 24. Top-level FR-CON mapping.

DPs are mapped to FRs as shown in figure 25. According to the first axiom this is a decoupled design and therefore is an acceptable design. The matrix is transformed into a development flow diagram as shown in figure 26. After database design, data gathering technology and layout design can be developed concurrently. However, as the development is made by one person, the two concurrent stages are cascaded sequentially, with preference given to developing data gathering technology.

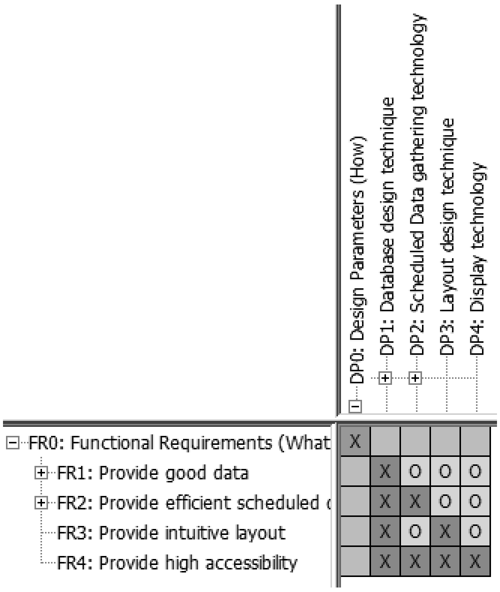


Figure 25. Top-level FR-DP mapping.

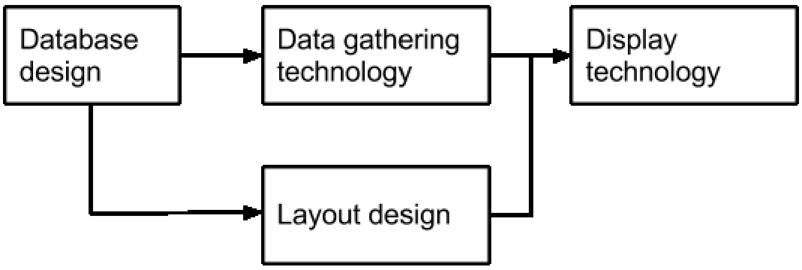


Figure 26. Top-level development flow.

3.1.2. Develop data model (FR1)

To provide good data (FR1) is to be able to provide a concise set of models (FR1.1), to limit the amount of data exchanged between systems (FR1.2), and to determine the correct frequency of update (FR1.3). Figure 27 shows the FR-DP mapping.

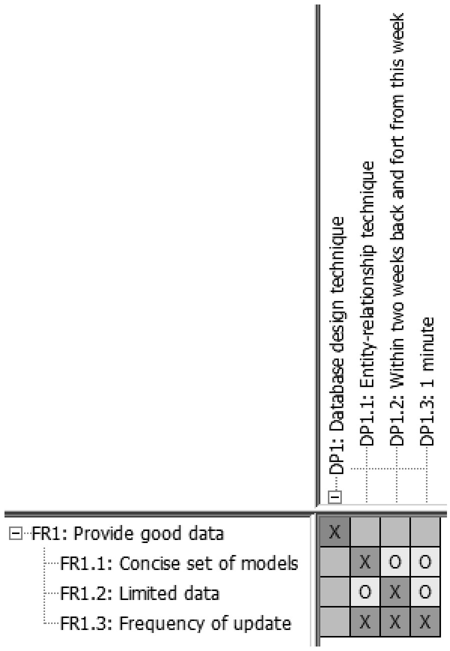


Figure 27. First-level decomposition of FR1 and its matching DP1s.

In order to have a concise set of model, each data attribute was examined. Below table describe all the data attributes needed and their mapping to existing SAP fields. For each data attribute, data type and length are also considered.

Table 4. First analysis of data fields needed for the display

SAP Data Field	Data attributes Name	Data type	Length	Description
Shipment	Id	Integer	4	SAP identification of a shipment
Descrip.of Shipment	Name	String	32	Shipment description, for example: Engine and Aux
Type	Type	String	6	Shipment type, for example MAIN for main shipment
ST	Transport mode	String	16	Means of transportation
IncTm	Incoterms term	String	3	For example DAP, CIP, …
Destination:Add.info	Incoterms port	String	40	Port specified by incoterms
Ext. ID 1	Project Id	String	16	Project number
Addit.text 3	Project Name	String	40	Project name
Addit.text 1	Scope status	String	40	Scope status, maintained by Delivery Manager / Transport manager
Addit.text 2	Financial status	String	40	Financial status, maintained by Trade & Finance
PlanShipSt	ETS	Date	N/A	Planned shipment start date
PlanShipEn	ETA	Date	N/A	Planned shipment arrival date
CurrShipSt	ATS	Date	N/A	Actual shipment start date
ActShipEnd	ATA	Date	N/A	Actual shipment arrival date
Planning	Input given date	Date	N/A	Date when green light is given for transport booking
City	Ship to city	String	32	Final destination city
Point of departure	Ship from city	String	32	City of departure

After having all data attributes identified, they are grouped into three main entities: Shipment, Project, and Destination in order to satisfy database normalization rule.

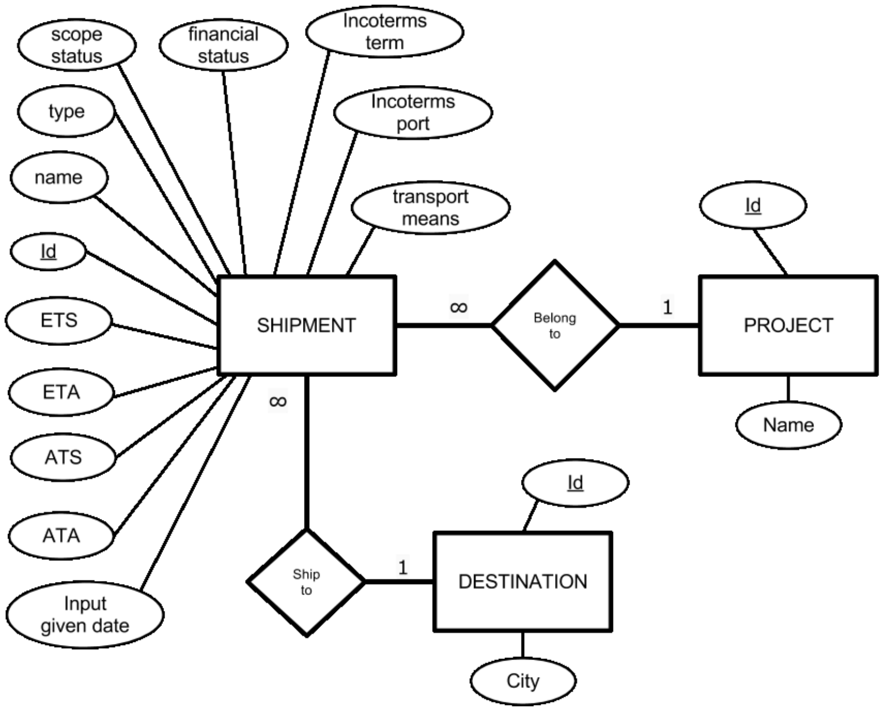


Figure 28. First ER Diagram for the shipment status display.

Provided that technology stays unchanged, less amount of data exchanged means better I/O performance. Planned shipment start time is used as a criterion to consider the amount of data to be included. Week is chosen as unit of time measurement because day is too detailed and month is too general. Now the question of how many data is quantified by a more specific question of how many weeks surrounding this week (both backward and forward) should be considered. In order to answer this question a sensitivity analysis of all SP shipments of 2012 were conducted.

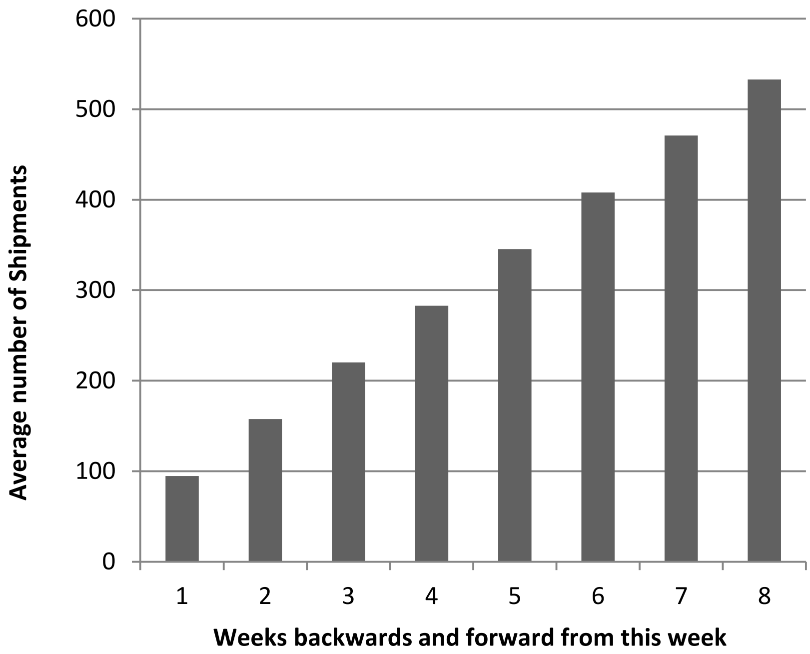


Figure 29. Average number of shipments based on this week $\pm x$ weeks.

From this data, the relationship between number of weeks included around a week and average number of shipments can be characterized by below function:

$$y = 62\,x + 32$$

Whereas y is the average number of shipments and x is the amount of weeks to be included in any given week.

Note that the above number of shipments includes both shipments that are not departed and those that are departed. Layout designer will want to split shipments into "Departure" and "Arrival" views. Supposed that there are equal numbers of shipments to be departed and to be arrived, and a physical screen can accommodate 30 shipments, then adding one more week will add one more page to each view. A $\pm$ 4-week option for example, provides enough data to fill five pages of each view, which will decrease usefulness of the screen. The balance is chosen to be $\pm$ 2 weeks.

In order to have good data quality, frequency of data refreshment (FR1.3) also has to be determined. This FR is affected by the two preceding FR as leaner data structure and data quantity means less data and therefore will allow for more frequent update. Although the purpose is to develop a real-time shipment status screen, it is wasteful to update data more frequently than the rate of update of the data itself. Therefore it is useful to study the frequency of shipment update in SAP. An analysis of 1617 Shipments change log since 2010 was conducted. Its statistics (in second) are presented in below table. The average time it takes before any shipment gets updated is 605s – about 10 min. Unfortunately the skew statistic is 6,5 – indicating a positive skewed distribution. With this type of distribution, estimation based on normal distribution assumption is invalid. Therefore the mean value cannot be used. This requires further examination of the data in order to identify the satisfactory frequency.

Table 5. Statistics of time of change (in second) of shipments during 2010-2013.

Mean	605
Mode	10
Median	163
Standard deviation	1417,5
Skew	6,54

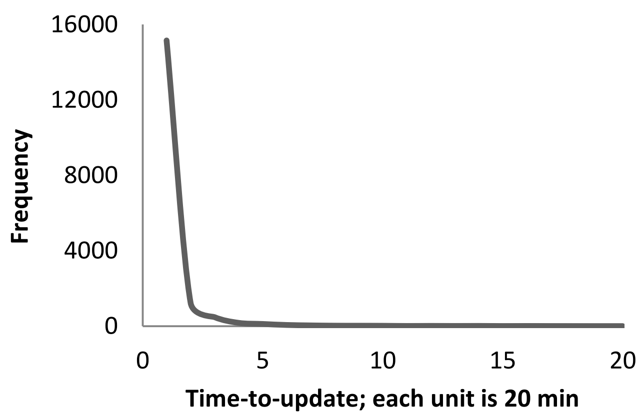


Figure 30. Time-to-update distribution is positive skewed.

Table 6. Ten most frequent shipment update frequencies.

Minutes	Percentage %	Cumulative %
1	31,20	31,20
2	12,64	43,85
3	8,39	52,24
4	5,94	58,18
5	4,99	63,17
6	3,81	66,97
7	33,0	70,00
8	2,46	72,46
9	2,37	74,82
10	1,82	76,64

Above table shows that using the mean time (10min) will result in high chance of getting out-dated data because chances that over three-third amount of shipments will be updated earlier than that. In fact, in order to have a real-time update, update frequency has to be set to one minute.

The last consideration about good data is the time of the day when it should be queried. With a high frequency of update, it will conserve energy and may reduce down time if the system is allowed to "sleep". The standby time is set to be between 21:00 today until 6:00 the next day.

3.1.3. Develop efficient scheduled data gathering technology (FR2)

In order to have good data I/O, the amount of data stored, database management system (DBMS), storage location, and mechanism of data synchronization have to be determined. In order to have an efficient scheduled data I/O; also a timer mechanism also needs to be determined. DBMS and storage location choice are affected by the constraint of minimum change to existing infrastructure (CON2). As data is from SAP, the best solution is to have a direct connection to SAP table data and present it immediately to the display. Unfortunately investigation shows that this is infeasible as 1) external consultants and internal IM people need to be involved each time there is a change to direct SAP connection while 2) data model will change during software development. A more suitable solution is to set up a database and build a middleware to sync data between SAP and the database. Therefore there are four FRs to be satisfied for FR2, as illustrated in figure 31.

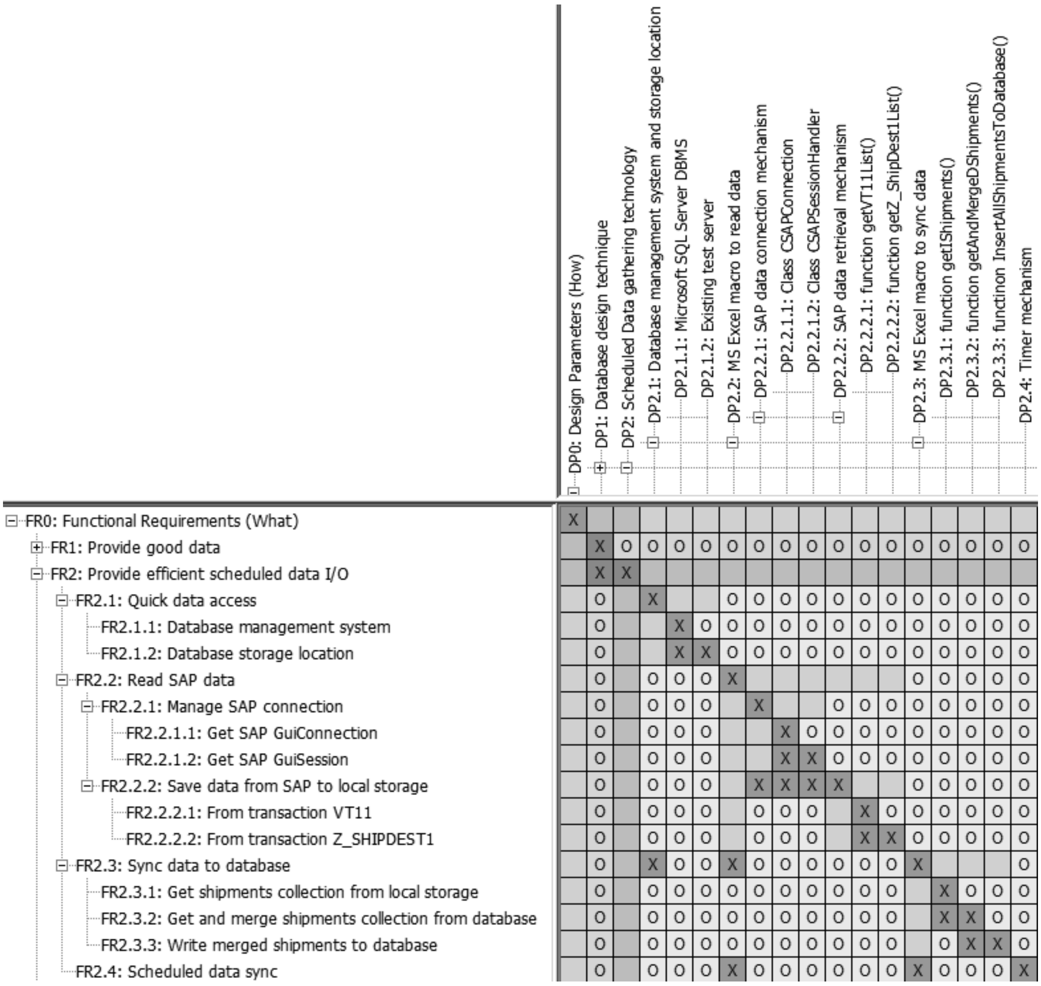


Figure 31. FR2's Second-level mapping of FR-DP.

Considering the relatively small size of data to be stored for this application and the constraint for minimum change of infrastructure (CON2), there are two systems that fit for the purpose: MS Excel or Microsoft Access files stored in a file server. Those file systems are installed as standard office application so no further installation required. When the file is stored in a file server, a macro can be built to access data remotely. This setup has been used extensively in small applications within Wärtsilä. Working experience shows that data connection is good when accessed locally, while being very slow when connected from remote areas such as China. Therefore this setup is not chosen. The other option is to use a dedicated SQL database management system run in a database server. Normally setting up a database in database server requires investigation and permission from information management organization (IM). There were a number of meetings held between the developer and global & local IM people to explain the software purpose and usage intention for the database. Different DBMS and storage locations were examined. One test server running MS SQL Server was found to be able to host the application database with no change to existing infrastructure and no extra cost to setup. However there is concern about data I/O speed. Therefore an experiment was developed. In order to test data query speed, a table was created in the server and fetched with over 2000 data rows, then an MS Excel macro was created to query and fetch the table data into an excel worksheet. The experiment was conducted through a remote VPN connection in Vaasa and each cycle on average took 1,4s – indicating a good I/O speed. Therefore selecting MS SQL Server as DBMS and Test Server as database storage location is the most suitable design choice. This is documented in the design matrix as in figure 32.

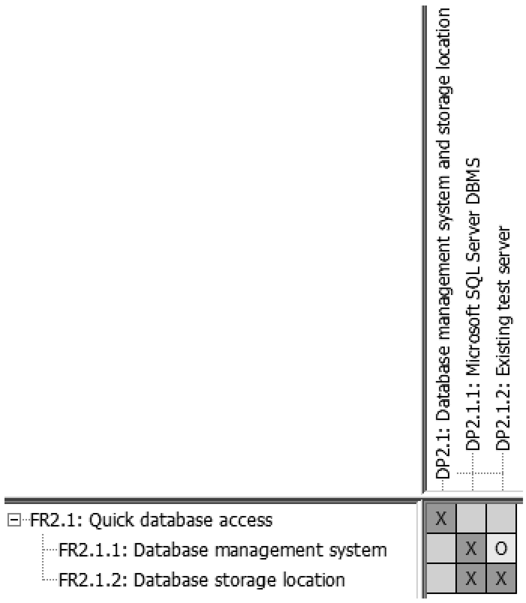


Figure 32. FR2.1's mapping of FR-DP.

The next development path is to query data from SAP (FR2.2). This includes two steps: maintain SAP connection (FR2.2.1) and data query (FR2.2.2). As mentioned earlier having direct connection to SAP tables requires involvement of external consultants and IM, which is not favorable when the number of tables and fields will need to change as the software development proceeds. One approach is utilizes SAP Gui Scripting, which is an automation interface that interacts with SAP graphical user interface to facilitate automation of repetitive tasks. SAP Gui Scripting language is very similar and compatible to Visual Basic for Application (VBA). Because of this many SAP automation tasks are developed with VBA. Moreover, this approach also satisfies CON1 and CON2. Below is the process for utilizing SAP Gui Scripts into VBA application.

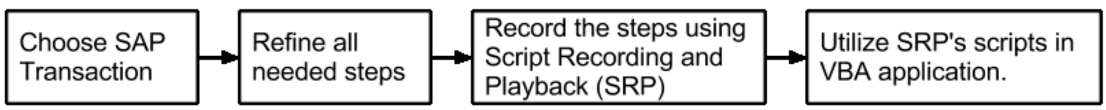


Figure 33. Process of utilizing SAP Gui scripts in VBA.

The main reason for using SAP Gui scripts is to simplify navigation to the correct element in the screen. Just as with HTML DOM navigation, SAP Gui elements are navigated through a complex hierarchy of objects. For example, writing a transaction name (e.g. VT11) in the transaction launcher is done by two methods, as shown in figure 34.

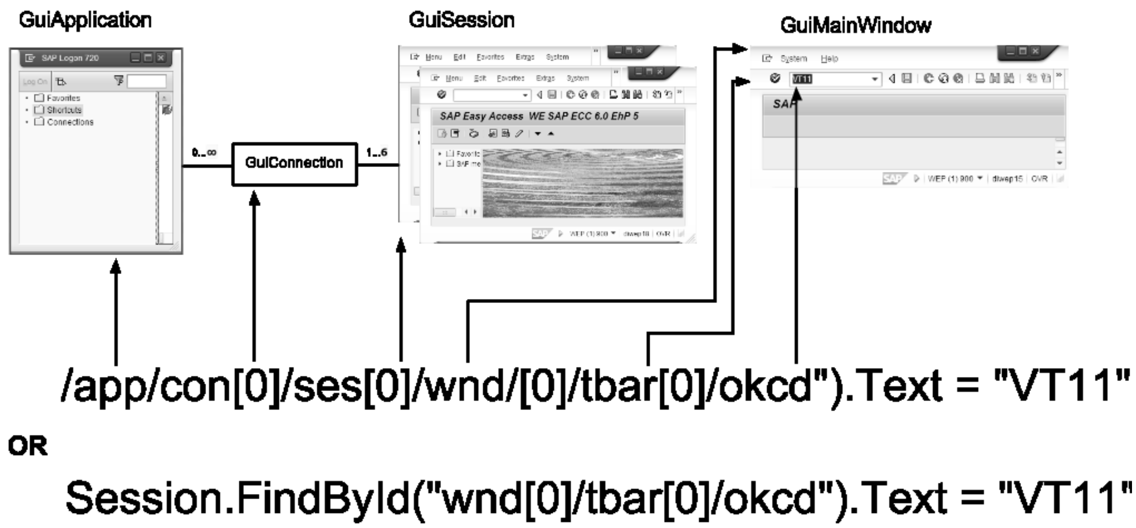


Figure 34. SAP Gui Scripting syntax explained.

As seen from the figure referring to a screen element by using the second syntax is simpler. This is called relative Id referencing (Cohrs & Loff 2003: 24). By having SAP Gui Script Recording and Playback record the screen hierarchy, developer only needs to copy the script to his VBA application code. While referring to a SAP Gui element may be challenging, referring to the correct free session is also challenging. This is important as the automation application must not terminate user work in any occupied session. For performance and security issues, the application must terminate the session it created before exiting. The above tasks must be secured prior SAP any transaction manipulation task. Therefore in order to satisfy FR2.2, there are two sub-FRs: manage SAP connection (FR2.2.1) which include creating and terminating session, and get data from SAP (FR2.2.2). FR2.2.1 outputs a SAPGuiObject that is ready for further use in the application. The subsequent step is to get data from different SAP transactions which contain the data fields specified in table 4. There are two transactions that together contain all the data required: VT11 and Z_SHIPDEST1. Although the majority of data fields are found in VT11's report, details geographical data such as destination's city is found in Z_SHIPDEST1's report. The two transaction data can be exported to two spreadsheet files which are stored in a local drive. These two files are the physical output required for FR2.2.

As input and output are clearly determined at this level of decomposition for FR2.2, each DP represent an actual software function (see figure 35). Matching a process variable (PV) to DP now means developing and optimizing a function. For example, figure 36 describes the algorithm for function `getVT11List()` (DP2.2.2.1) inside module M2.2.2 which is a process implementation (PV) of DP2.2.2.

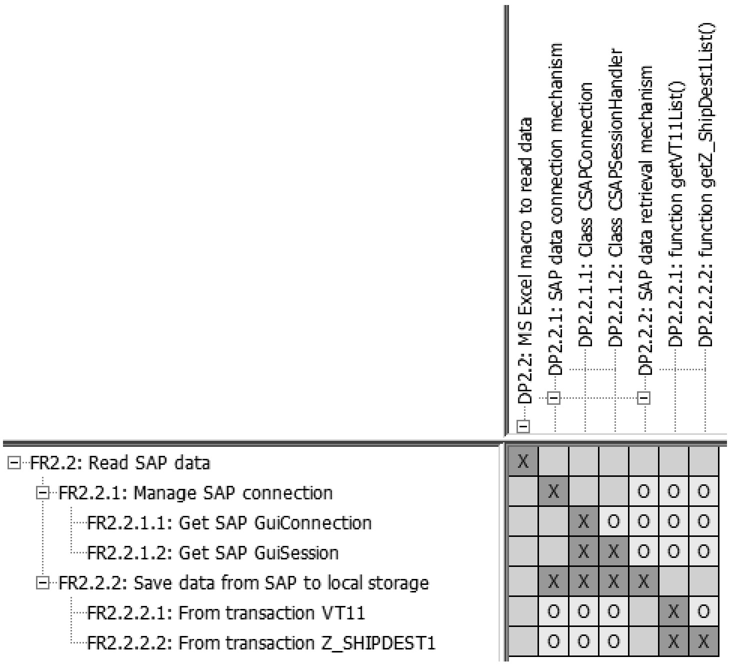


Figure 35. FR2.2's mapping of FR-DP.

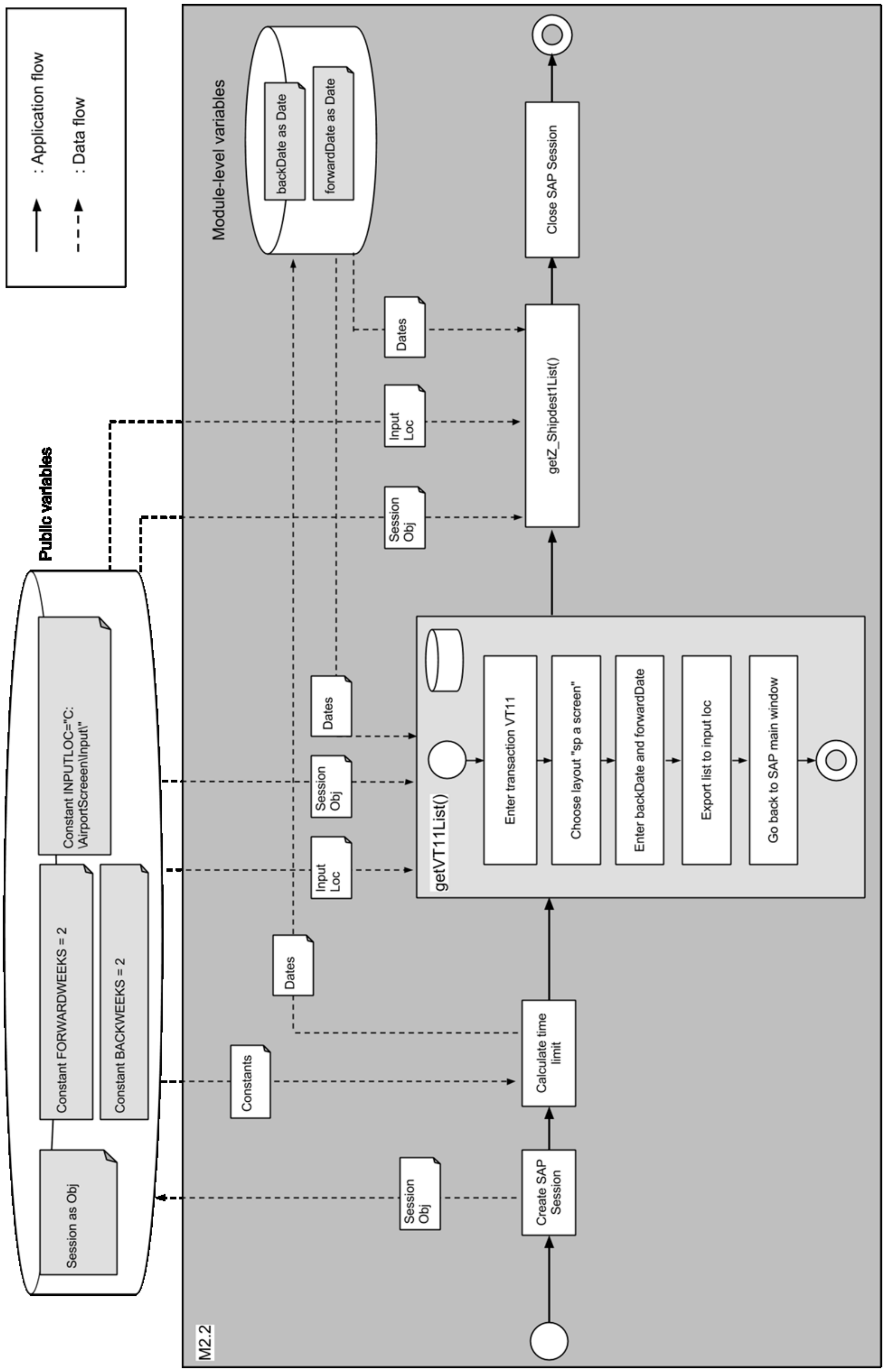


Figure 36. Module-level architecture for saving data from SAP to local storage.

After securing raw data into a local storage, a method to write data to the database needs to be developed (FR2.3). In the first development iteration, there is no two-way data synchronization between database and local storage; all table data in the database is deleted and replaced with a new batch of data. Development therefore has three main steps: 1) merge data from the two lists (VT11 and Z_SHIPDEST1) into a shipments collection, 2) delete table data from database, and 3) copy the shipments collection data to database.

After being able to complete one cycle of data synchronization, the application must be able to complete n-times more cycles. One functional requirement is to be able to repeat that cycle n-times after every d-duration that was specified in FR1.3. One way to do this is to trigger two timers at the end of the first call. One timer is for timing the next run and one timer is for timing the ultimate stop. Also the requirement for running hours (FR1.4) is satisfied by placing a logical check at the beginning of module M2 (PV of the FR2). If running time is not in the specified hours, then the application jumps directly to calling a new timer, which schedules a next run based on the interval specified by DP1.4 (see figure 37).

During the design and development of FR2 it is noticed that many constituting modules share common data and functions. Therefore the modules are physically assembled together under one Excel macro to utilize shared resources such as variables and functions. This grouping improves performance and code clarity noticeably. One practical example: by sharing a public Session object, `getVT11List()` (DP2.2.2.1) and `getZ_Shipdest1List()` save time establishing one more Session object, which takes up to five seconds. Centralization of public variables makes garbage cleaning more manageable as a function can be written to unset all variable upon application exit. However using public variables increases traceability challenge as sometimes it is unknown which function has changed the variable value. Therefore the practice is to 1) limit the amount of public variables, 2) implement a copy of the public variable locally, as well as 3) limit the amount of functions that can modify public variables. Adding a utilities module has reduced total amount of codes and improved readability. One example was the creation of function `getPIDFromString()` in the utilities module. This function uses regular expression to separate project ID from a raw string. The function is widely used in many modules so it is more manageable to move the codes to a separate function and call it when needed.

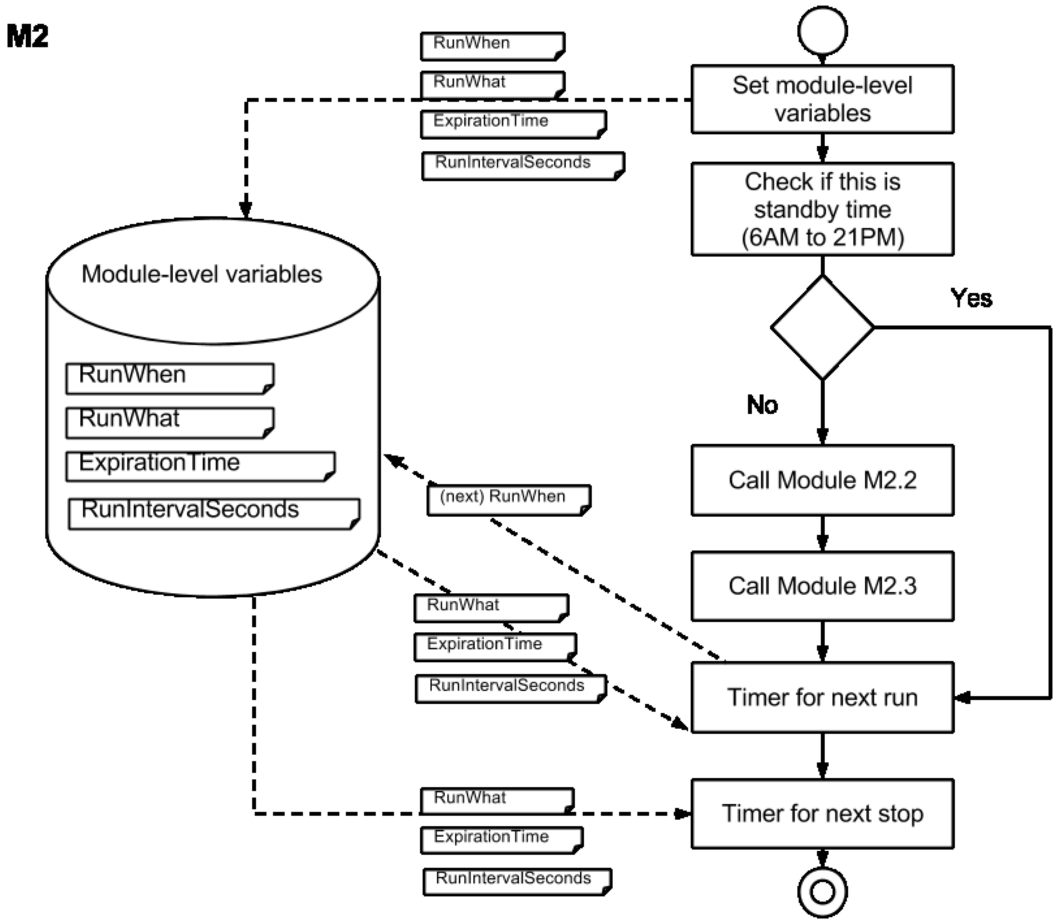


Figure 37. Logic for scheduling data synchronization.

At the end of this stage, there are two physical artifacts: the SQL Server DBMS installed on a database server and the Excel macro on a desktop computer. Although each design parameter can be satisfied by one physical software application (Excel module), performance initiatives have integrated them into one application to share common resources. This integration is useful as there is now only one application responsible for maintaining data, and therefore improving its performance means improving data quality for the whole system.

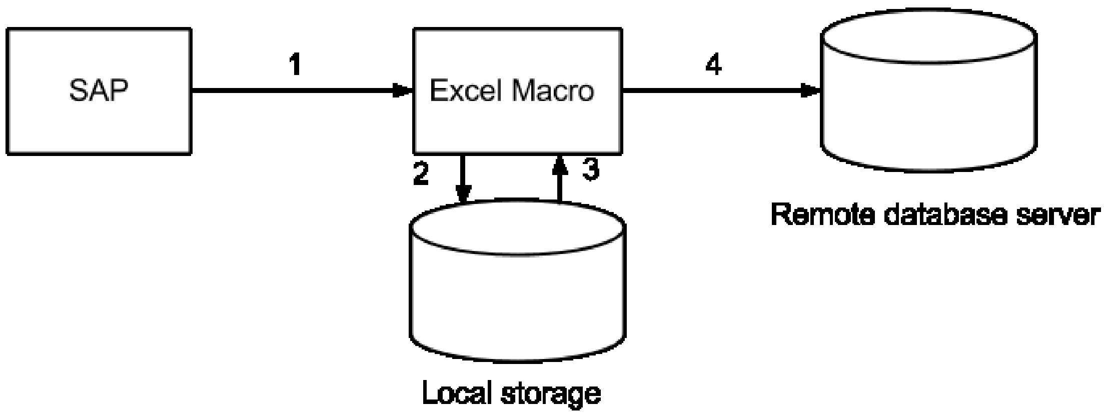


Figure 38. System architecture for syncing SAP data to database.

Unit testing were conducted to compare the Excel macro performance against design parameters specified in FR1. Data of time taken to complete a full data synchronization cycle were collect as shown in figure 39. The data was also further decomposed into two parts: time from SAP to local storage and time from local storage to database.

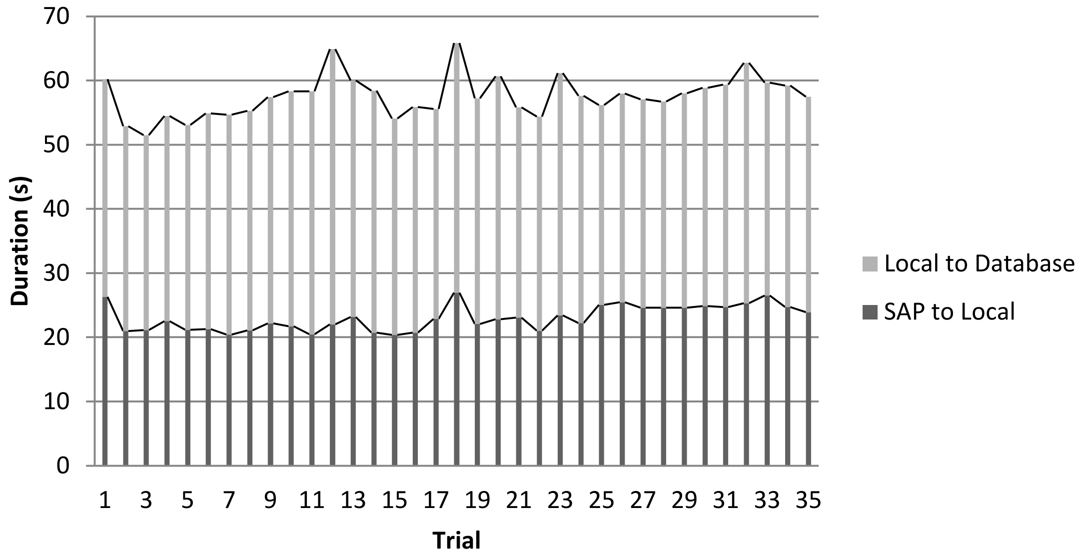


Figure 39. Time (in seconds) taken to complete a full data sync cycle.

There are some remarks about the test results: a) in most cases DP for one-minute data refresh is achievable as most sync circles last under one minutes, and b) getting local data to database take almost twice more time as getting data from SAP to local storage. The result in b) was not expected as data retrieval from SAP to local storage should take longer time due to all the necessary steps to set up connection and open reports. This implies that there is room for improvement in the technology for local to database data input.

3.1.4. Design an intuitive layout (FR3)

When data gathering technology is developed, layout design is the next critical task to successfully communicate data to user. There are two important questions that must be answered in this design task: 1) who are going to use and 2) how will they use this layout? The main persona of this layout is named Sami, a project manager who has background education in mechanical engineering and has extensive project management experience. He travels often to meet customers, contractors, and colleagues. He often expresses that too much time is spent on monitoring project execution while he would like to also dedicate his attention to other aspects of project management. Although not being a rude type of person, Sami wants to keep conversation short; if he needs to get an answer, it should better be short and to the point. If there were a shipments display available, then he wishes it to be familiar and simple, and accessible even when he is not at his desk.

Figure 40. The persona for shipment status display: a mobile project manager.

A layout that fits with this type of persona constitutes a transient posture. According to Cooper, Reimann, & Cronin (2007:169), a layout with transient posture must be “simple, clear, and to the point”. The layout must be “built-in” to user as they often drop in to view the status and often do not have time to learn how to read it. Also the layout is intended to direct user, not to discuss with them, i.e. information shown helps reader proceed with concrete actions – whether to continue with his ongoing task or to contact responsible people if he spots some nonconformity.

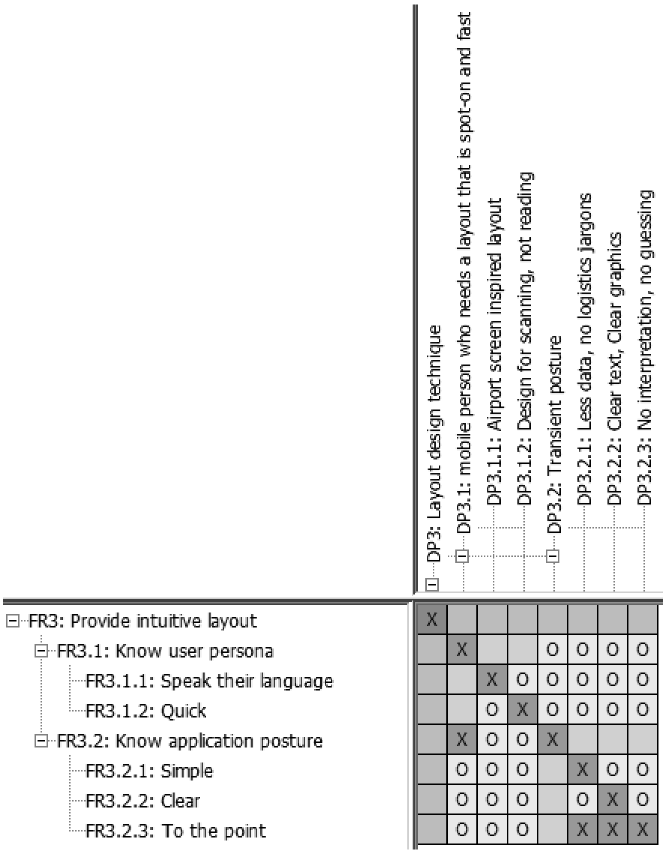


Figure 41. FR3's mapping of FR-DP.

In order to satisfy above design parameters, the design approach is to create a layout that readers are familiar with, use good texts and colors to aid reading, and to use icons to reduce the amount of textual data.

A project manager may not know about SAP VT11 report but he may know an airport screen. Further analysis of airport screen layouts in Vantaa and Vaasa airports provides insights that can be used in creating shipment status display.

Figure 42. Picture of an airport screen taken in Vantaa airport.

Firstly airport screens have bright header color with plain background that promotes readability of clear-type font texts. In addition, the use of traffic light colored circles at the outermost corner helps readers identify ongoing activities. Secondly, information presented is very concise but not too concise that it becomes obscure. For example flight number alone can include destination city information, but the flight number is not user-friendly and therefore does not aid user in searching for his/her flight in the screen; that is why destination city is also presented to help reader focus his/her eyes more quickly. Also some of the data is self-descriptive, i.e. readers can understand without looking at the header. Thirdly, the screen instructs readers with the next course of action; the message is accompanied by traffic light background colors to indicate level of urgency.

Inspired by the airport screen layout, the first shipment layout prototype was developed.

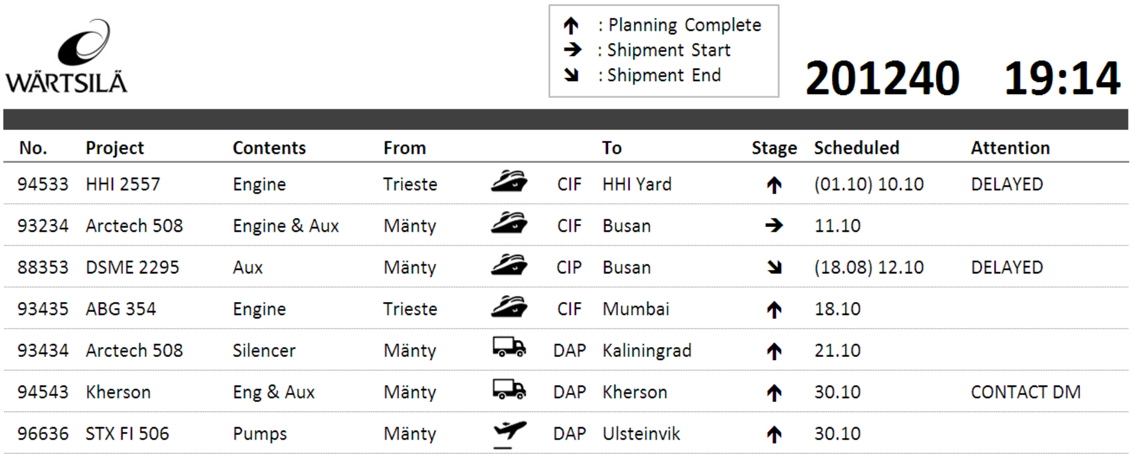


Figure 43. The first layout prototype, with mixed views for Departures and Arrivals.

Although inspired by the airport screen layout, a shipment status layout still retains some distinct characteristics. Firstly, it is because of higher uncertainty of planned shipment start date that readers demand to know more detailed about shipment status. Secondly, project managers do not keep shipment numbers with them the same way as they keep their flight tickets. Subsequently, the layout must provide more information to help readers. Thirdly, shipments are multi-modal so additional data of mode of transport may be also needed. Lastly, while unit of time measurement is measured in minutes in airport screen, it should be measured in weeks in shipment status screen. The first prototype however still lacks some important statuses.

Prior to transport booking, there are two aspects concern a shipment status: (physical) scope of supply and financial issues. In SAP they are maintained in two different text fields – one is updated by Delivery Managers, and one by Trade & Finance people. Trying to incorporate plain texts of those two fields just worsen the already crowded screen real estate. One approach is to transform the two textual statuses into one Boolean value: whether shipment is open or closed (for transport booking), and then iconify the value.

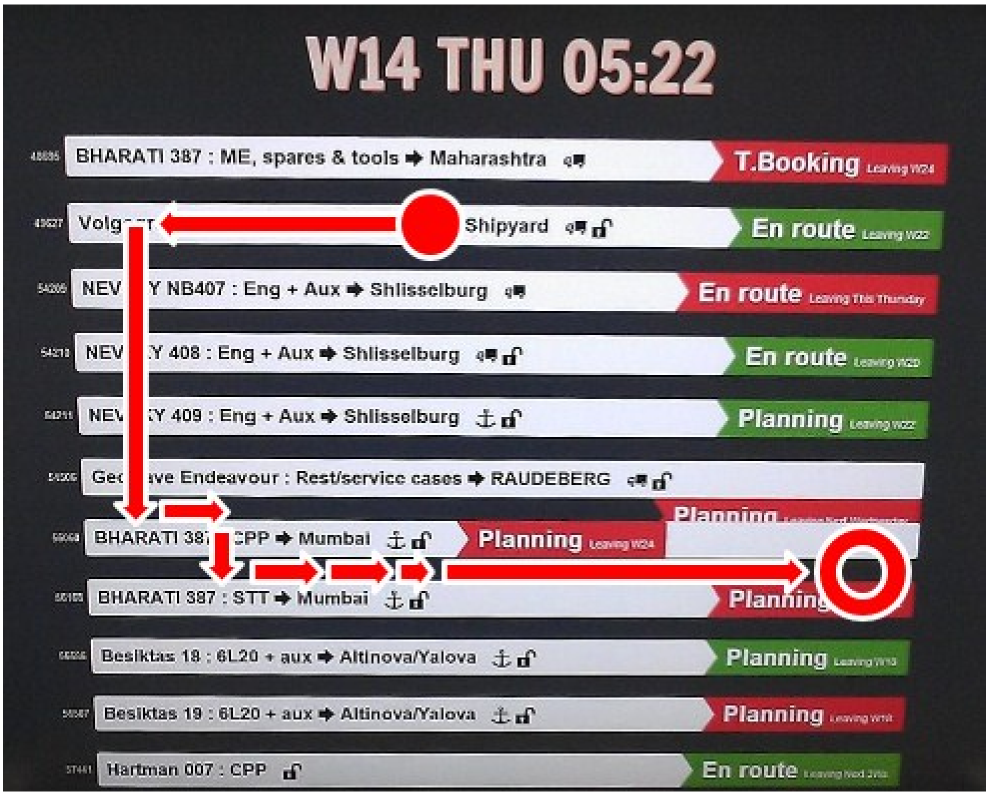


Figure 44. The second prototype of Departures view.

As seen from figure 42 the layout now delivers a more contrast, simpler, and more discretional at showing shipments stages. Each shipment is presented in a white thread which is clearly separated from other shipments by wide dark gaps. Instead of presented in tabular form, each shipment is presented as a sentence. Transportation means is presented as icons, shipment status is also iconified by a lock / unlock icon, which describes whether the shipment is ready or not ready for transportation preparation. As there is still much information to display in a thread, same-direction arrows are used at every one-third the thread’s length to help guiding reader eyes to the end. Information is arranged so that for details shall emerge as readers look from left to right. Figure 44 describes reader eyes movement. Each arrowhead represents a reader's eyes stop to comprehend data. Shipment numbers are now presented in smaller font size and placed outside shipment threads due to their lessen importance. They are however not eliminated from the layout because they still provide an efficient token to communicate with logistics control people. For example, a Delivery Manager can immediately check a shipment status when given a Shipment number; if he instead receives other a project name and shipment description, then he has to use different SAP transactions to extract the project number and then use the description to manually navigate to the correct shipment.

The second shipment status layout is designed to be clearer and more informative than the first prototype. Unfortunately it has some drawbacks that eventually led to the third design prototype. Firstly it is not intuitive because shipments are sorted by project name instead of due time; furthermore planned shipment start dates are displayed in too small texts. Secondly, it lacks directional property, i.e. it does not tell reader the next course of action for problematic shipments; this is due to the lack of "attention" texts and the vague meaning of the lock / unlock icon. Lastly considerable gaps between shipment threads limit the amount of shipments on the screen, increasing the amount of pages per view.

The third layout design addresses all the above mentioned shortfalls of the second prototype. Shipments data is now rearranged in tabular format and sorted by planned shipment start ascending. There are four most notable changes in this layout 1) decomposition of shipments status into scope and payment statuses: a green open box icon denotes complete scope while a red open box denotes incomplete, a green dollar icon denotes financial OK while a red dollar icon denotes not financial not OK; 2) addition of attention message, and 3) traffic lights icons added at the beginning of the shipment thread to help guiding reader eyes to only problematic shipments (that have ETS approaching or passed), lastly 4) Indication of updated planned by a backward pointing arrow 🡐.

As there is no dedicated field in SAP where attention message to project management is maintained, those messages have to be filtered from the operational messages maintained by Delivery Managers and Transport Managers. In order to do that FR2.3 (Sync data to database) has to be modified. In fact one additional module is added to interpret the field `Addit.text 1` so that the `Attention` field will contain only messages that matter to project management are retained. Also because of decomposition of shipment status, the data model also has to be updated to include two more Boolean fields for scope status and financial status. The result is a display as shown in figure 45.

Week	Destination	Shipment	Scope	Pay	Attention
18		58276 : Akhtubinsk 52 / NB3 : Engine and Aux			
19		81579 : Armon G002 : GB + CPP+Cntrl			ETS closing in
18		141929 : ASL 1052 : AUX			ETS closing in
18	DAP, Talcahuano	110312 : ASMAR 110 : Controls			ETS closing in
18	CIP, Daewo-Mangalia	103733 : BG BMV 169 : Silencers, start.air			ETS closing in
18		144755 : BG BMV 169 : Catalyst elements			ETS closing in
18		58352 : BP Quad 204 at HHI : 8L26 (2 pcs)			Departed 20.04.13
18	DAP, Ulsan	58353 : BP Quad 204 at HHI : Auxiliaries			Departed 22.04.13
18	DAP, HHI Ulsan Yard	58354 : BP Quad 204 at HHI : Outboard parts			
17	FCA, Hamburg	140197 : Concordia 674 : 8L20, Vaasa			
18		153581 : COSCO 418 : Safety valves			ETS crossed
18	CIP, Shanghai	148723 : COSCO 419 : E trans dmg repair 1			ETS closing in
18		133929 : Cotecmar : Gear			ETS closing in
18	CFR, Cartagena, Colombia	58938 : Cotecmar : CPP, contro(from NL)			ETS closing in
17		138023 : CSBC Navy H1025 : rest service and aux			Departed 24.04.13
19		105074 : CSBC Navy H1025 : GB			ETS closing in
17		150953 : CSBC Navy H1025 : Geislinger and Servi			
19		150954 : CSBC Navy H1025 : 2 pc Intermediate			ETS closing in
18		58823 : CSSC H3039 : 4x8L32 - 2x6L32- Aux			
17		155813 : DSME 2288 : kpa pressure gauges			
17	CIP, Busan	155819 : DSME 2289 : loop isolators			
18		158194 : DSME 2290 : Seal guard			
19	DAP, Busan	150959 : DSME 2296 : service rest			ETS closing in
17	FOB, Mäntyluoto	130530 : DSME 2296 : Aux 3 - Hoses			
17		149201 : DSME 3031 : Gauge board for stan			
18		158197 : DSME 6049 CLOV FPSO : SW pump battery cabi			ETS closing in

Figure 45. The third prototype of Departures view.

3.1.5. Develop display technology (FR4)

The completions of data gathering technology and layout design have provided a foundation for developing technology display that is universally accessible. There are two systems that satisfy this functional requirement as well as constraints of no additional installation to end-user (CON1) and minimum change to existing infrastructure (CON2): 1) a display system built on an Excel macro and 2) a display system built on web technology. While Excel macro and a web page are both familiar concept for most users within this organization, an Excel macro is not the chose option because 1) it requires user intervention to bypass security check when opened; 2) furthermore it still retains traits of an application such as menu bar, status bar, etc. despite efforts to turn those off; 3) lastly it does not support advanced UI technology. Web technology in contrast compensates all above mentioned shortfall of an Excel macro. There are two main sub functional requirements that need to be satisfied: to build a layout that can support multiple reading devices (FR4.1) and to build a system that can provide that layout (FR4.2).

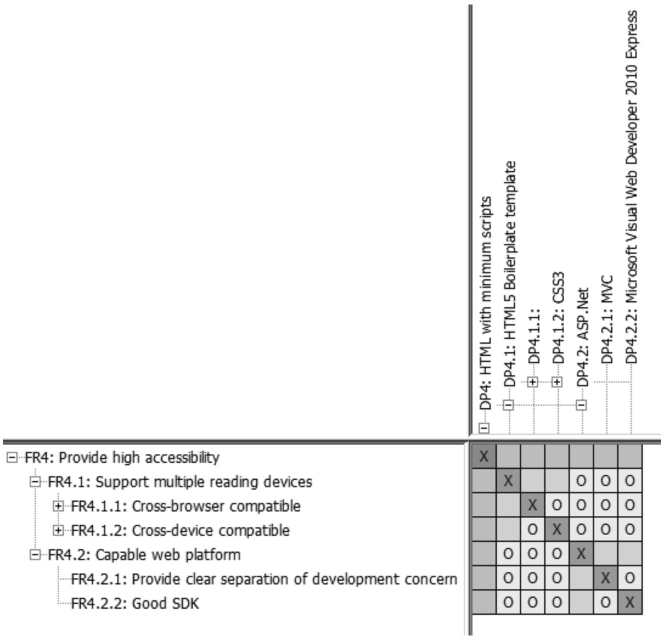


Figure 46. FR4's mapping of FR-DP.

In order to build a page that is readable by multiple devices, standardization is important. The rule to web standardization is to separate contents from appearance (Sikos 2011: 140). In order to correctly transform the layout designed in FR3 into a web page, HTML codes are used to define the structure of the page, while CSS (cascade style sheet) are used to define how each element specified inside the page is displayed. One major concern is about cross-browser, cross-device compatibility. Prior to HTML5 and CSS3 software developers have to use a combination of server-side and client-side programming to determine the correct browser and device in order to prepare a suitable page. Nowadays one view can be built to suit the majority of device based on client-side programming with HTML5 and CSS3. Support for new web technology has been a concern in the past as previously Wärtsilä computers were running older versions of Internet Explorer which could not render the newest versions of HTML and CSS. Recently the majority of Wärtsilä computers have been upgraded to Internet Explorer 10, which supports HTML5 and CSS3. In addition for mobile devices, the majority of project managers are equipped with devices running iOS Safari web browser and Windows Phone Internet Explorer; both of them can render HTML5 and CSS3. With supportive information technology landscape, it is decided that the newest HTML and

CSS technology will be used to utilize their capability of multi-platform, multi-device support. Figure 47 summarizes functional requirements and design choices made for this task.

	DP4.1: Client-side adaptive screen layout	DP4.1.1: HTML5, CSS3, and JavaScript	DP4.1.1.1: <!DOCTYPE html>	DP4.1.1.2: normalize.css	DP4.1.1.3: Modernizr.js	DP4.1.1.4: JQuery	DP4.1.2: CSS3	DP4.1.2.1: rem unit	DP4.1.2.2: @media	DP4.1.2.3: webfont and SVG
FR4.1: Support multiple reading devices	X									
FR4.1.1: Cross-browser compatible		X					O	O	O	O
FR4.1.1.1: Avoid browser's quirks mode			X	O	O	O	O	O	O	O
FR4.1.1.2: Consistent elements rendering			O	X	O	O	O	O	O	O
FR4.1.1.3: Browser feature dection			O	O	X	O	O	O	O	O
FR4.1.1.4: Consistent JavaScript behavior			O	O	O	X	O	O	O	O
FR4.1.2: Cross-device compatible		O	O	O	O	O	X			
FR4.1.2.1: Relative unit of measurement		O	O	O	O	O		X	O	O
FR4.1.2.2: Media queries		O	O	O	O	O		X	X	O
FR4.1.2.3: Undistorted graphics		O	O	O	O	O		O	O	X

Figure 47. FR4.1's mapping of FR-DP.

Cross-browser compatibility means that the desired layout is rendered correctly regardless of the web browser. In order to account for cross-browser compatibility, first a browser has to be explicitly instructed not to use its quirks mode to render page, which interprets CSS and HTML documents differently (Manian 2009). This is done by declaring `<!DOCTYPE html>` at the beginning of the code. The next step is to use CSS to provide better cross-browser consistency. The reason is because browsers have different default style render engine and therefore some HTML elements are rendered differently. One approach is to reset the CSS styles to explicitly command browsers to render the desired HTML elements. Another approach is to normalize those browser behaviors so that standardization is achieved while preserving browser useful default styles. Gallagher (2012) introduced `normalize.css` – a small CSS file that contains additional styles that fix the above mentioned issues. In order to maximize browser compatibility, feature-detection mechanism is also needed. This means the ability to identify (modern) features that the calling browser does not support; with that knowledge developer can customize the CSS styles to provide backward compatibility. One technical solution to this is to use `modernizr.js` – a JavaScript module that is included in the `<head>` tag of the HTML document; on page load the script is triggered to detect and output all tags that are both supported and not supported by the browser. For example, below figure is a copy of the HTML codes output for Chrome web browser version 28. Before `mondernizr.js` is executed, the `class` attribute of `<html>` tag is empty. After page load, the `<html class=””>` element is filled with supported and unsupported HTML5 elements. The unsupported elements are preceded with “no-*” prefix, such as “no-touch”.

```
<!DOCTYPE html>
▼<html lang="en" class=" js no-flexbox flexbox-legacy canvas canvastext webgl no-touch
geolocation postmessage websqldatabase indexeddb hashchange history draganddrop websockets
rgba hsla multiplebgs backgroundsize borderimage borderradius boxshadow textshadow opacity
cssanimations csscolumns cssgradients cssreflections csstransforms no-csstransforms3d
csstransitions fontface generatedcontent video audio localstorage sessionstorage webworkers
applicationcache svg inlinesvg smil svgclippaths">
  ▼<head>
     <meta charset="utf-8">
     <meta content="IE=edge,chrome=1" http-equiv="X-UA-Compatible">
     <title>Index - All Shipments</title>
     <meta name="viewport" content="width=device-width, initial-scale=1.0">
     <link href="/favicon.ico" rel="shortcut icon" type="image/x-icon">
     <link href="http://fonts.googleapis.com/css?family=Francois+One|Open+Sans:
     400italic,400,800" rel="stylesheet">
     <script src="http://code.jquery.com/jquery-1.9.1.js"></script>
     <style type="text/css"></style>
     <link href="/Content/Heydings/Heydings.css" rel="stylesheet">
     <link href="/Content/site.css" rel="stylesheet">
     <script src="/Scripts/modernizr-2.5.3.js"></script>
   </head>
  ▶<body id="gfs">…</body>
</html>
```

Figure 48. Output of `modernizer.js` is a list of supported & non-supported HTML5 elements by the calling web browser.

Not only HTML elements but also JavaScript is interpreted differently between browsers. Therefore one more functional requirement for cross-browser compatibility is to have consistent JavaScript behavior (FR4.1.1.4). In order to avoid browser quirks mode for JavaScript, JQuery is used to act as an abstraction layer (Chadwick 2012: 69). By providing an extensive set APIs for DOM manipulation, JQuery does not only provide consistent behavior but also simplify the code. Also due to some useful animation functions were also used to swipe between pages of the same view.

When foundation technologies for cross-browser compatibility are identified, cross-device compatibility needs to be developed (FR4.1.2). Cross-device compatibility means the ability to maintain uniform layout across different device screen sizes. This task used to be server-side centric. When a request is sent to server, the server checks web browser version to decide appropriate view (full or mobile). Normally mobile pages contain basic HMLT markups with limited amount of script and stylesheet. This method has been increasingly proven to be inaccurate as modern mobile web browsers are far more capable of rendering complex markups but are still treated as mobile old-generation mobile browsers and thus receiving plain HTML pages. Modern web markups have provided more powerful screen layout at client side. One approach is to let web browser determine the device screen size in order to decide the contents layout. Responsive web design proposes using new set of elements in CSS3 for designing adaptive view ports as the screen size changes Marcotte (2010). The approach used to develop adaptive shipment status screen is to 1) to use `rem` unit elements sizes and location, and 2) use `@media` to change the root font size based on the screen size. A `rem` unit is a relative unit of measurement based on value of font-size attribute of the root element (W3C 2013); the root element in this case is the `<html>` element.

During the development of shipment status display, following steps are taken to develop a responsive layout: 1) determine root font-size, 2) set up relative layout, 3) determine different `@media`, and 4) vector-based graphics.

The root font size is determined as the default font-size, i.e. 16px. As all elements are scaled based on the root font size, the use of `rem` unit helps maintain the correct ratio in all scales.

When designing HTML layout, the HTML body does not contain page contents directly but through a division `<div>` in order to maintain margins. The division is divided into three sub-divisions: header, content, and footer. Their dimensions are determined by CSS classes using `rem` units (see figure 49).

Prior to CSS3, in order to maintain unbroken graphics, software developers can a) prepare different version for the same graphics to accommodate different screen sizes; or b) prepare one high resolution graphics and resize it accordingly. While the first approach conserves bandwidth, it is inflexible to change, i.e. a new set of different sizes have to be prepared when the graphics changes. CSS3 provides two alternatives to overcome this challenge. The first option is web fonts. Using webfont dingbats (icons) is similar to using `wingdings` font in MS Word. The icon will not get distorted even in high zoom level. With CSS3 adaptive graphics is accomplished two alternatives, which are both used in developing this layout. The first is to declare `@font-face` in the CSS file and then refer to the font keyword in the HTML file (Irish 2013), (Google 2013) (see figure 51). The second is to use scalable vector graphics SVG (W3C 2011). This is done by providing a drawing data to the `path` attribute of a `<svg>` tag, as shown in figure 52.

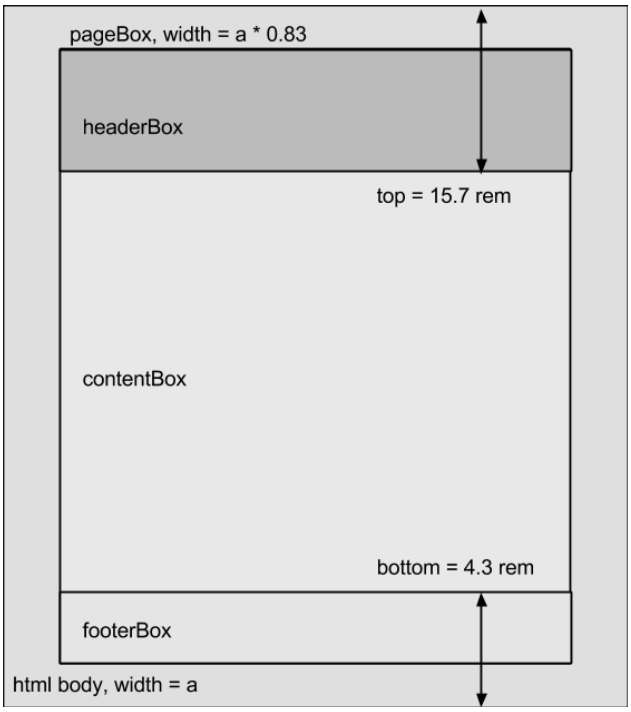


Figure 49. Tablet/computer screen CSS layout.

There are three types of display screen sizes based on their width resolutions: 1) mobile phones with width no larger than 680 px, 2) large display minimum width of 1400 px, and 3) tablet or computer screen with width between 680 px and 1400 px. Each of these screen sizes will have a customized set of relative measurements to define the page and columns layout, as seen in the below figure.

```
@media only screen and (max-width: 748px) { /*680 x 1.1*/
  html {font-size: 10px; }
  body {max-width: 90%; }
  img {height: 2rem; }
 }

@media only screen and (min-width: 1540px) { /*1400 x 1.1*/
    html{font-size: 24px;}
    .contentBox{top: 15rem;}
    img { height: 2rem ; }
    .week{ width: 6rem;}
    .destination{width: 12rem;}
    .project{width: 12rem;}
    .shipment{width: 12rem;}
    .scope{width: 3rem;}
    .pay{width: 3rem;}
    .attention{width:22rem;}
    .snumber {width: 4rem;}
}
```

Figure 50. CSS3 code for adaptive screen sizes.

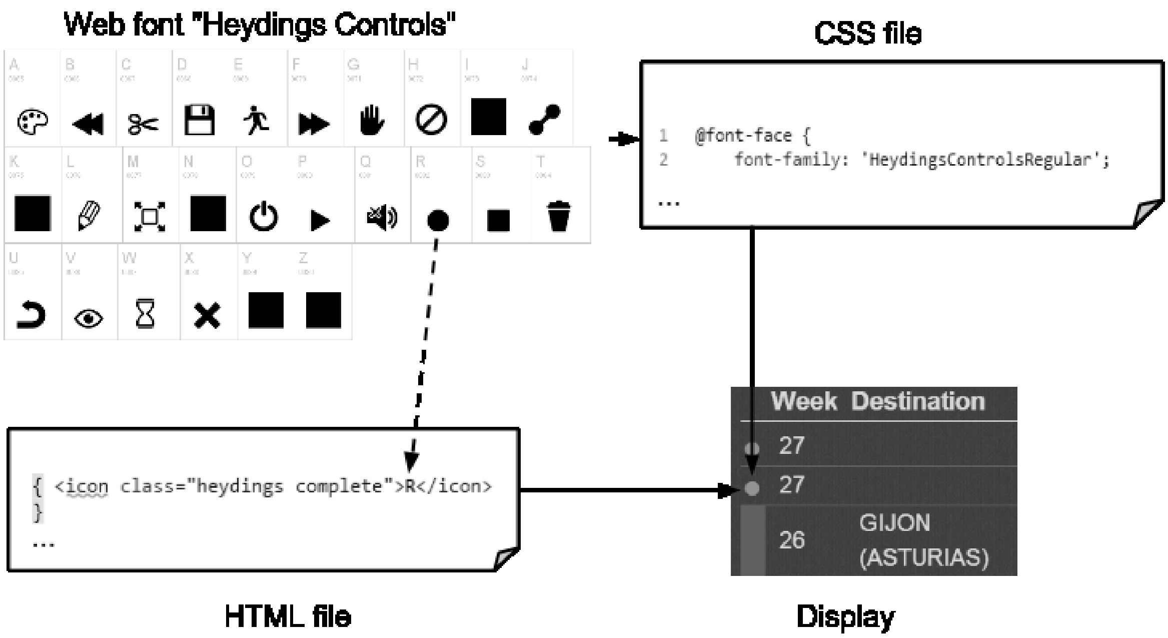


Figure 51. Adaptive graphics using web fonts.

```
▼<svg style="vertical-align:middle;" height="40" version="1.1" width="36" xmlns=
"http://www.w3.org/2000/svg">
   <path style="-webkit-tap-highlight-color: rgba(0, 0, 0, 0); fill-opacity:1;" fill=
   "#60ac39" d="M15.5,3.029l-10.8,6.235L4.7,21.735L15.5,27.971l10.8-
   6.235V9.265L15.5,3.029zM24.988,10.599L16,15.789v10.378c0,0.275-0.225,0.5-0.5,0.5s-
   0.5-0.225-0.5-0.5V15.786l-8.987-5.188c-0.239-0.138-0.321-0.444-0.183-0.683c0.138-
   0.238,0.444-0.321,0.683-0.183l8.988,5.189l8.988-5.189c0.238-0.138,0.545-
   0.055,0.684,0.184C25.309,10.155,25.227,10.461,24.988,10.599z" stroke-width="0"
    transform="matrix(1.2,0,0,1.2,1.2,1.2)" fill-opacity="1"></path>
 </svg>
```

Figure 52. Adaptive graphics using SVG.

One task in layout design is to develop navigation within a multi-page view and between views. Even though data input has been limited to within four weeks around the current week, there are still more shipments than the amount a view can accommodate. Furthermore, development of multi-page navigation has to consider absence of peripheral input. There will be no keyboard or mouse in public corner so the display has to be able to switch pages automatically and when reaching page end, it shall switch back to the first page. After a number of rounds, the view should also be able to switch to another view, i.e. between Arrivals and Departures views. This animation should last perpetually. View-switching in the same display is both space-saving and cost-saving. In fact it is inspired from airport displays on which both Departures and Arrivals views appear routinely. In order to implement these two functional requirements a set of JQuery codes is written at the end of each view. The script algorithm is described as in figure 53.

Front-end layout design using CSS3 has separated presentation from contents (HTML), making code maintenance and improvement faster and easier. This results in uniform user experience across web browsers and device. Figure 54 exhibits screenshots of the same Departure view in different web browsers with different screen sizes. This design has satisfied FR4.1 for supporting multiple reading devices. In addition, JQuery codes complete the display system by providing seamless animation between pages of the same view and between views. So far development work has concentrated solely on front-end development using imaginative data. The next development task focuses on server-side web development to connect the layout to production data.

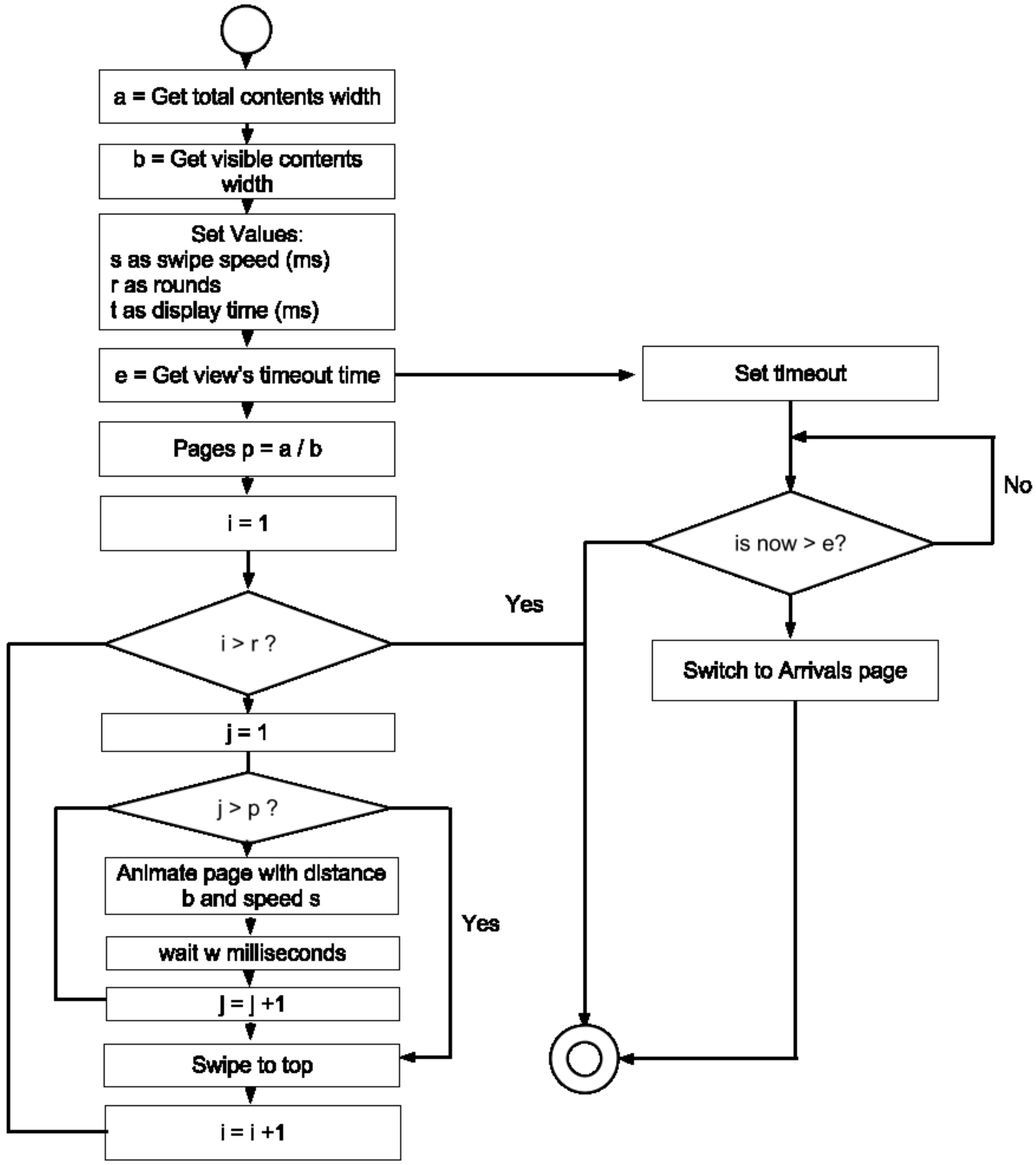


Figure 53. Algorithm for navigation between pages and between views.

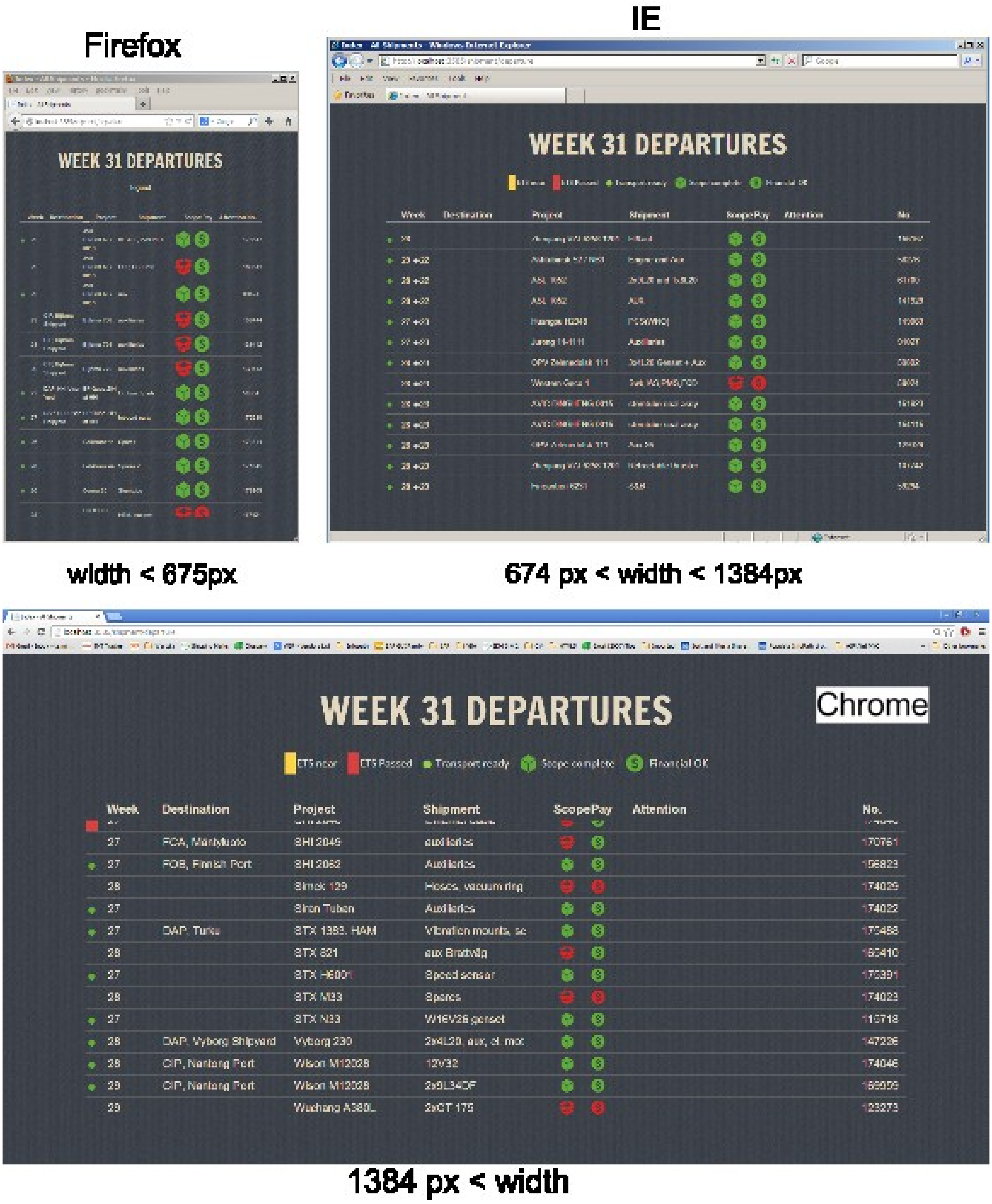


Figure 54. Front-end adaptive layout design results in cross-browser, cross-device uniformity.

The choice of server-side web development is dependent on the existing web server technology in the test server that will run this application (DP2.1.2). The server runs IIS which is a Microsoft web server software application. Therefore the technology chosen must be highly integrated to IIS, as well as having good foundation for some overhead tasks normally encountered in web development such as database access and session management. Because of this ASP.NET web application framework is chosen. There are three different ways to build an application based on ASP.NET: web pages, web forms, and MVC, with MVC having the clearest separation concerns. MVC stands for Model-View-Controller. A model presents business logic, a View displays the Model data, and a Controller interprets user request to instruct which Model will be fetched to which View for display. The MVC approach fits well with the development approach so far so it was chosen. Model, view, and controller are connected through a naming convention. MVC for this application is simple with only one controller, one model and two views. The controller is named `Shipment.` The `Shipment` controller has two views: `Departures` and `Arrivals`. Each view shares the same model named `Shipment`. When a request is sent to web server (IIS) from a URL `http://www.domain.fi/shipment/departure`, it is sent first to ASP.NET's routing engine. This is a pattern-matching system which matches the URL against the URL patterns registered. In this case it determines that method `Departures()` inside `Shipment` controller must be invoked. When invoked, `Departures()` initializes a `Shipments` collection and passes it to `Departures` view. The view integrates `Shipment` model data into html tag and returns the plain HTML codes, ready for display on a web browser.

In order to develop ASP.NET MVC application, developers have a variety of web software development kits (SDK). Microsoft provides a matrix of SDKs that suit different development needs and skills. Available SDKs range from the most end-user development friendly tool (Sharepoint Designer) to the most structured tool (Visual Studio Ultimate). SDK costs from nothing to tens of thousands of US dollars. Among all the SDKs, Visual Studio (VS) Express provides the most value for this development task as it is both free and feature-rich. It shares core features and project format and as with more premium VS packages (Ciliberti 2013:22). Investigation on premium VS packages show that those features missing from VS Express is not critical and can be substituted by separate tools. For example load testing and web performance testing are available only in VS Ultimate; however there are free alternatives available. Chrome web browser's element inspection tool provides front-end load tests and code audit by default and free of charge.

The VS Express SDK is free and downloadable from `http://www.asp.net` web site. One notable remark is that all software used for development work can be centrally managed by Microsoft Web Platform Installer (WebPi). This simplifies significantly SDK setup process as it detects all needed dependencies and installs them together with the SDK. For a fresh installation of development environment, WebPi can install also SQL Server Express as database management system and IIS Express as web server (Allen 2013). However, as both SQL Server and IIS instances are available in the test server, these pieces of software were not installed. The SDK provides many templates for MVC web application. When a template is chosen, the SDK prepares overhead tasks to prepare corresponding application architecture. MVC 4 also includes by default many practices that maintain cross-browser and cross-device compatibility (see figure 49). This has increased the speed of development as the main tasks are only to 1) prepare the shipment model, 2) prepare the controller, and 3) integrate the designs in FR3 to the views.

The software development iteration was closed with the completion of server-side development. The system architecture is illustrated in figure 55. There are three physical devices: the user/public monitor to display real-time shipment statuses, the server to host web application and database, and a desktop computer to synchronize data between SAP and database. Strict separation of display from data input sub-systems has enabled the system to run two independent routines: one for data input and one for display shipment statuses. Because the system updates data as frequently as the rate of shipment change, it can be considered as a real-time system.

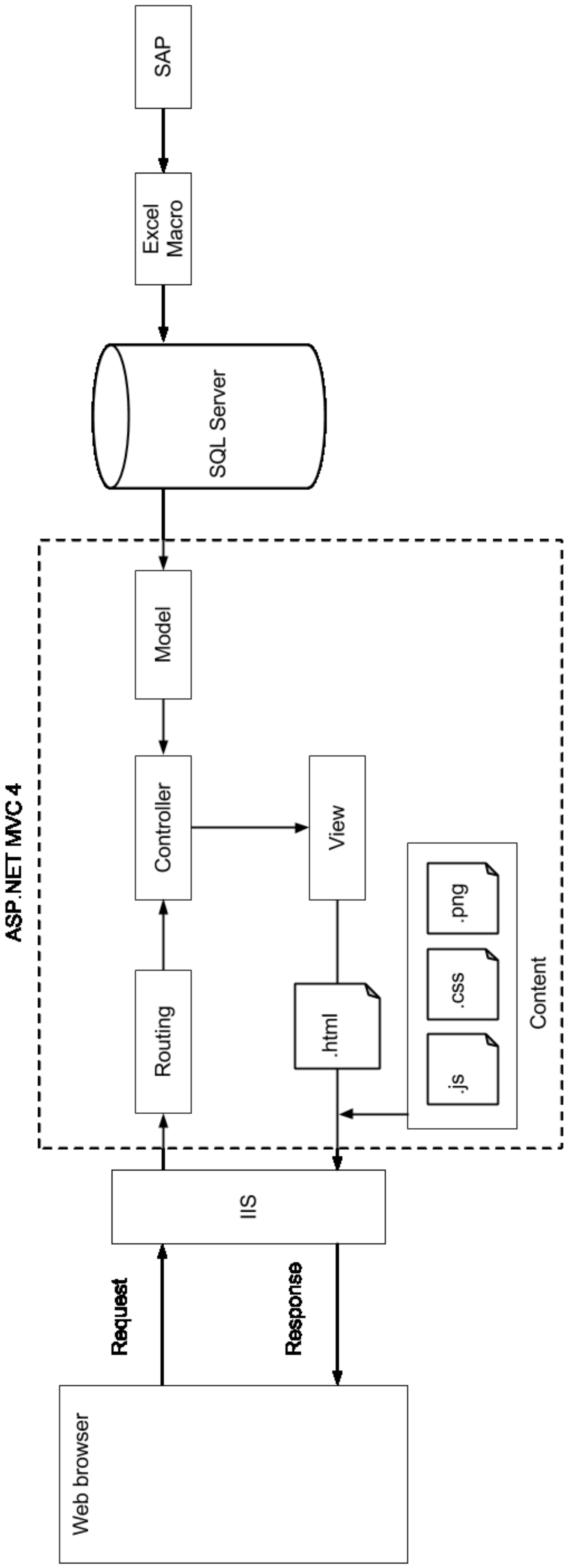


Figure 55. Shipment status's application architecture.

3.2. Second iteration

While the first iteration is about functional development, the second iteration is about performance improvement and layout refinement.

3.2.1. Improve writing to database

During test run of the sub-system (Excel macro) to retrieve data from SAP it was notice that the writing time to database is almost twice more the retrieval time from SAP (see figure 39). This behavior is against developer's assumption as there is more data transferred between SAP and local storage than between local storage to database; also SAP retrieval logic requires time buffers to avoid `overspeeding`. By placing timers after every function call, developer is able to trace the function with the longest processing time. `InsertAllShipmentsToDatabase()` is the most time consuming function. By inserting timer after every major chunk of codes, the developer is able to discover the part of codes affecting performance. It is found that inserting shipments to database is the most time consuming task. Each `SQL Insert` opreation includes setup and execution operations. Execution time improvement is not always feasible as it depends on physical network bandwidth. Therefore improving setup time is crucial to improving performance. There are two measures taken: 1) to reduce setup time, and 2) to reduce total amount of setups.

In the first iteration a VBA class was written to manage database connection and operations. The reason to use the class was code simplicity. It is discovered that calling a method from this class takes more time than calling a similar function from the same module. Therefore the class was removed and its methods moved to the same module where they are needed the most. This integration helps reduce setup time. In addition, it is found that by not opening the record set as dynamic (meaning ability to navigate backward and forward between elements in the record set), the query speed improves. In order to reduce the number of setups, shipments data is combined into one `Insert` operation instead of having an `Insert` operation for each shipment. The insert algorithm is therefore revised, as shown in figure 50.

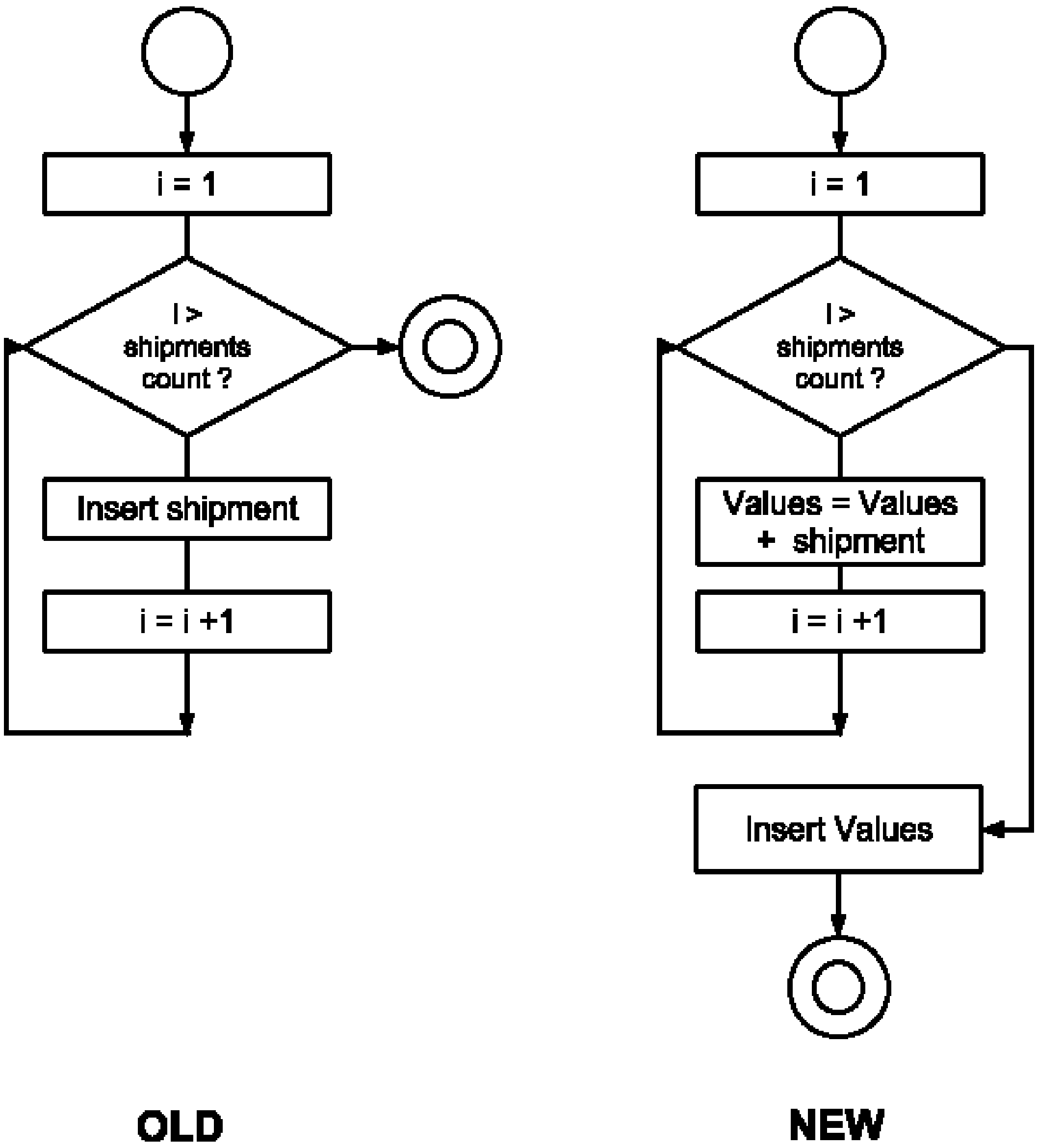


Figure 56. Revised algorithm for better Insert operation.

Test data provides promising result with total cycle time drops by half, thanks to a significant reduction of the `Insert` operation time. Total cycle time now takes on average 25,6 s

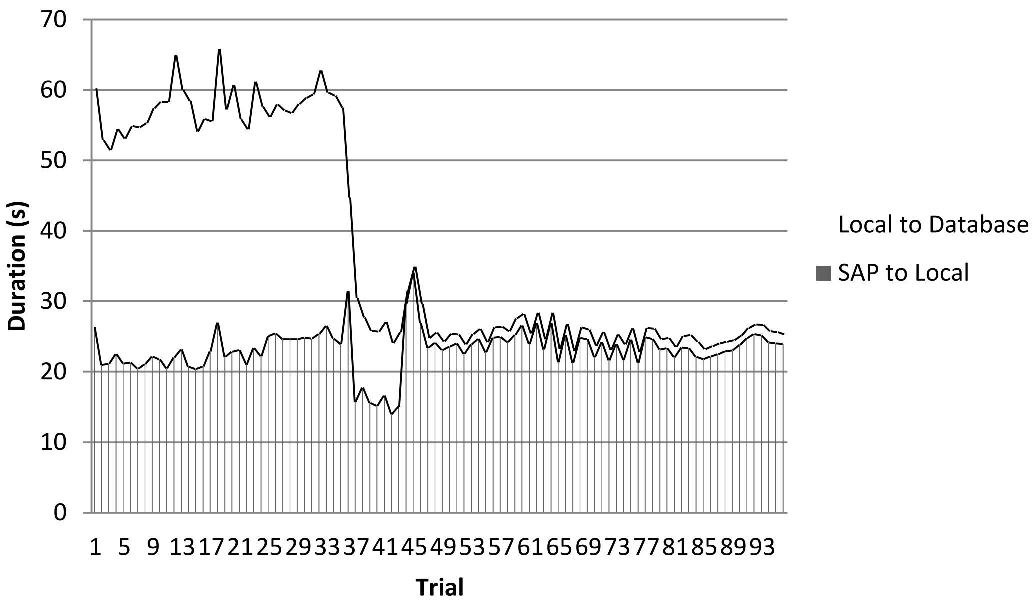


Figure 57. Improved SAP-database synchronization cycle time.

3.2.2. Improve SAP connection reliability

While speed has been improved, connection reliability still fluctuates. After speed improvement, the Excel macro was test run and result can be seen in figure 52. From figure 52 it can be seen that there reliability is now in region A as there are many downtimes. Investigation shows that this is caused by a symptom called VBA "overspeeding" in opening a new SAP session. Overspeeding refers to a situation when the next VBA command is executed while previous command is still being executed in SAP. This is because of inconsistency in SAP Gui API that results in some functions demand VBA to wait until control is passed back from SAP, while some do not require. The SAP Gui function for creating a new SAP session does not require such control. Therefore error occurs in situations where there is a lag in the network that delays creation of a Session object while VBA code has already moved on to the next command that uses the object. In order to fix this problem, a timer and a time limit are set up. After creating a session VBA shall wait until the time specified by the timer before checking if a Session object available. It can wait for some time if the Session object is not yet available but the waiting time does not exceed the time limit. In order to maintain system performance, the developer tries to set the times so that the timer is short enough while time limit is long enough. In practice the timer is found to be 0,4 seconds and time limit is 20 seconds. The result is positive, as can be seen in figure 53, region B. The system manages to deliver consistent performance with virtually no downtime for a long period of time.

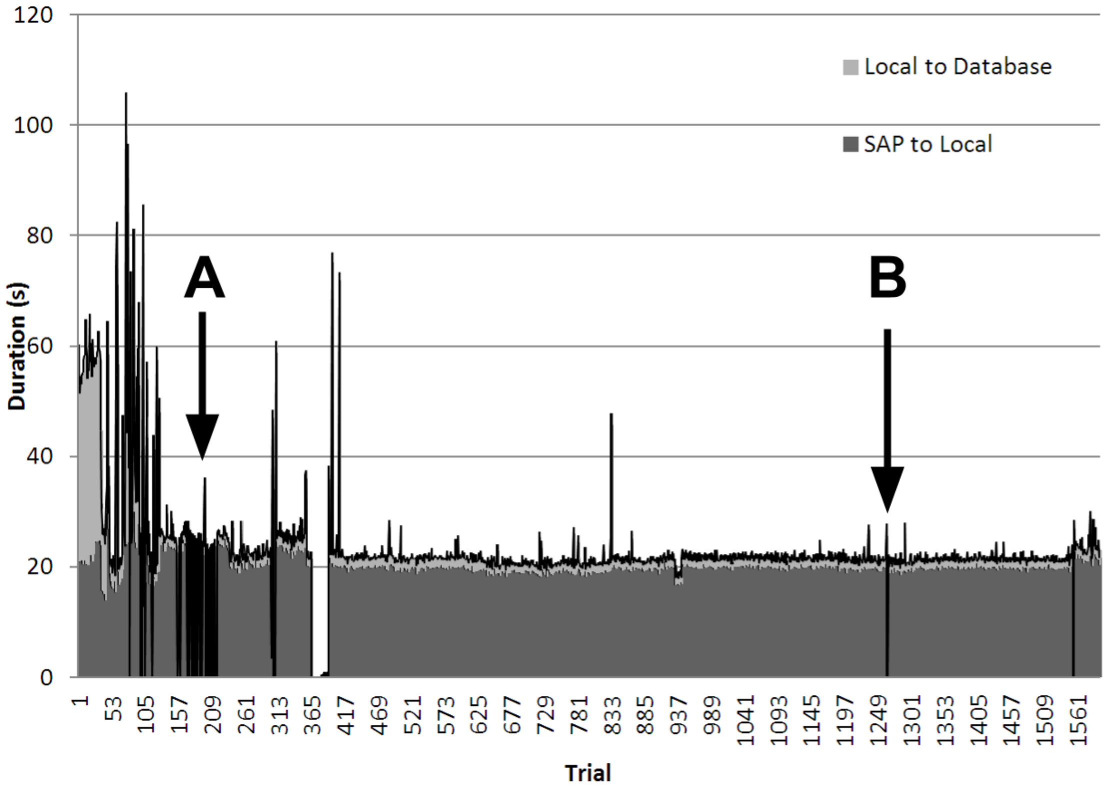


Figure 58. Extended trial run for Excel macro.

Although the system now exhibits stable performance, some of its occasional downtimes are critical and requires human intervention to restart (see figure 52 point B).

There were few times when SAP crashed during a VBA call, leaving the VBA application jammed. Once again this is due to the inconsistency of SAP Gui API which requires passing back of control for some functions. Because of this the VBA application will wait until infinity if SAP is crashed. There have been attempts to trigger a timer within the VBA application that routinely checks for availability of SAP connection and session. Unfortunately the timer cannot be triggered while VBA is waiting for SAP response. Another approach is to concurrently open an independent instant of Excel macro to routinely close SAP application. Timing between the two macros is critical as to avoid collision, as illustrated in figure below. In the below scenario the SAP check macro was timed inappropriately, leading it to terminate SAP while the main macro is still working with SAP.

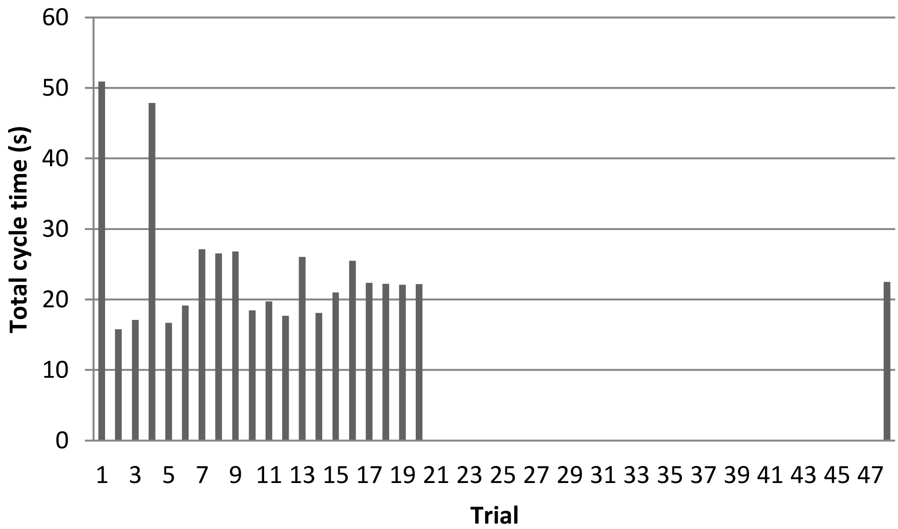


Figure 59. Macros collision leading to prolonged downtime.

In order to have better timing, developer first studies the behavior of SAP synchronization cycle times. It is found that SAP sync cycle time follows a normal distribution, as can be seen from figure 54, illustrating a bell-shape curve. Table 7 further confirms that this is a normal distribution because mean and median are almost identical, and skew value is close to zero.

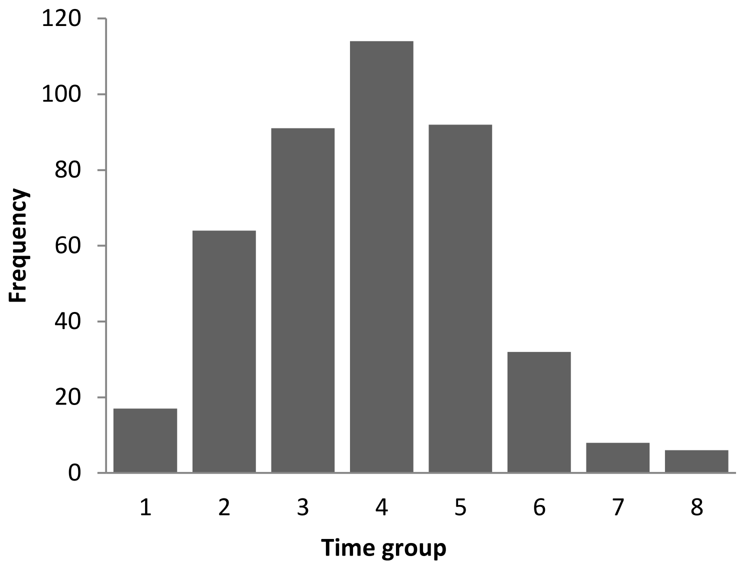


Figure 60. SAP synchronization cycle time follow a normal distribution

Table 7. Statistics of SAP synchronization cycle time distribution.

Statistics	Time (s)
Average	21,24
Median	21,28
Std	0,84
Min	19,28
Max	23,98
Skew	0,28

Based on the statistics it can be estimated that 99,7 % cycle times will be between 18,7 to 23,7 seconds. Therefore within a 60-second timeframe, setting the SAP closure time to be between the 45th and 59th second will likely avoid collision. In practice the SAP synchronization macro is set to run at the beginning of every new minute, while the SAP closing macro is set to run at the 53rd second of a minute. As it takes on average five seconds to open SAP and get a session, the SAP closing macro is set to run at the

53^{rd} second in every five minutes. A three-hour test results in good data with no downtime and no unexpected termination of SAP.

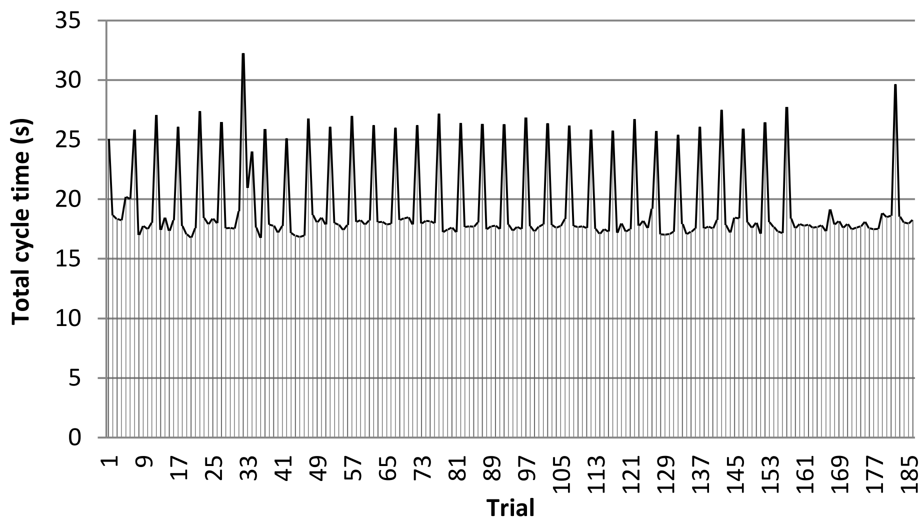


Figure 61. Total cycle time when two macros simultaneously run.

While forcefully terminating SAP has created some periodic peaks in total cycle time, it has created a more resilient, auto-recovery SAP synchronization sub system.

3.2.3. Improve layout design

In the second iteration of layout design, focus is placed on further improving clarity and find-ability. Previously data was set to be queried within four weeks from the current week. This amount of data results in a large number of pages for each screen. Therefore the period of time is further reduced to within two week from the current week.. Data are now sorted so that related shipments are close to each others. Also data columns and columns width are rearranged so that most the important data gathers around an imaginative central vertical line. The purpose of sorting and rearranging is to let reader eyes catch as much useful information as possible without much movement. In practice data is sorted ascending with this order time – project name – shipment description.

Project column is positioned in the middle of the page, with shipment description, statuses, and attention message adjacent to it. For example a reader looking for a problematic shipment of his project will have eyes movement as illustrated in figure 62. First the reader looks to find his project. When he has found the shipment series, his eyes will be attracted to those with red colors. He decides to know more about the second problematic shipment so he looks to see its planned start date. He sees that the shipment is planned to be sent out this week but the scope is not complete. He decides to contact Delivery Manager (DM) so he looks at shipment number, writes it down and sends a question to DM with the shipment number as reference.

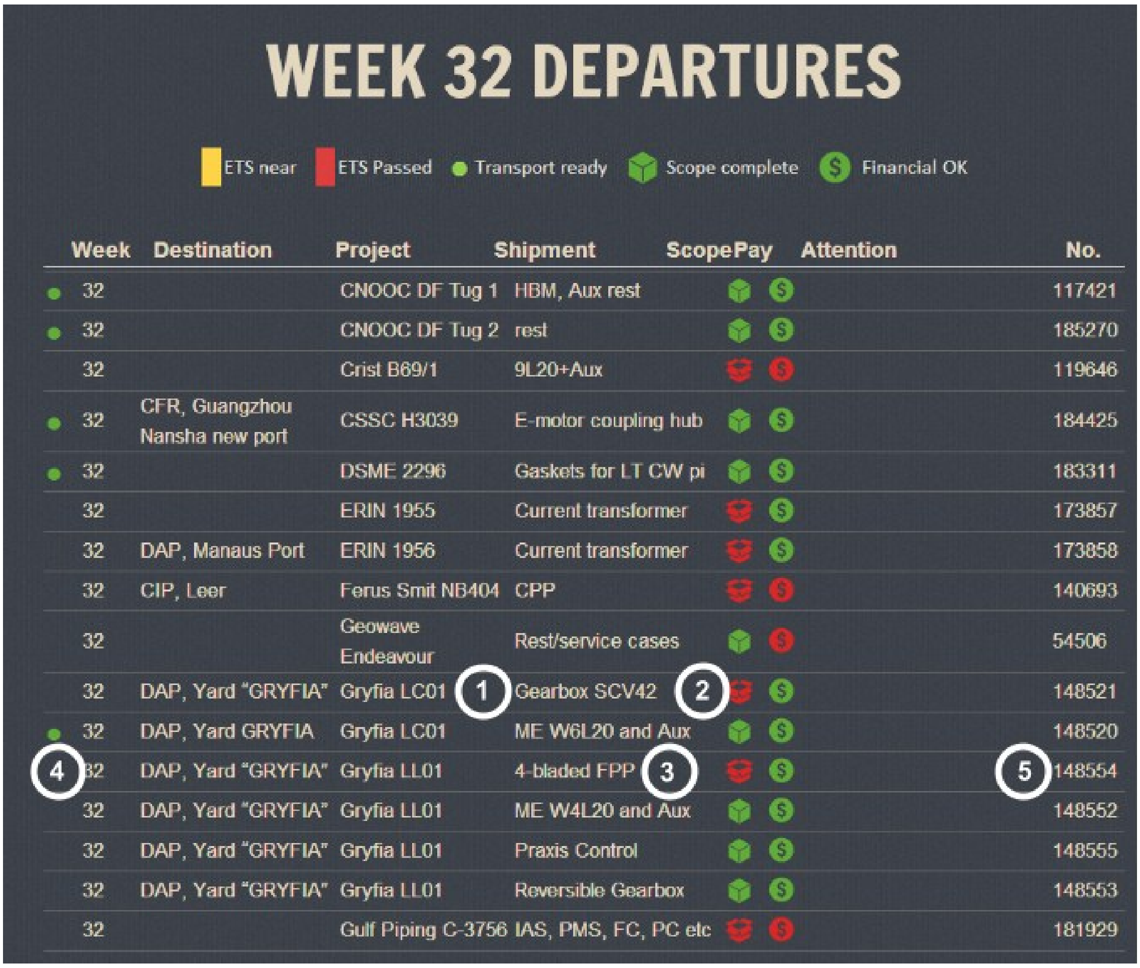
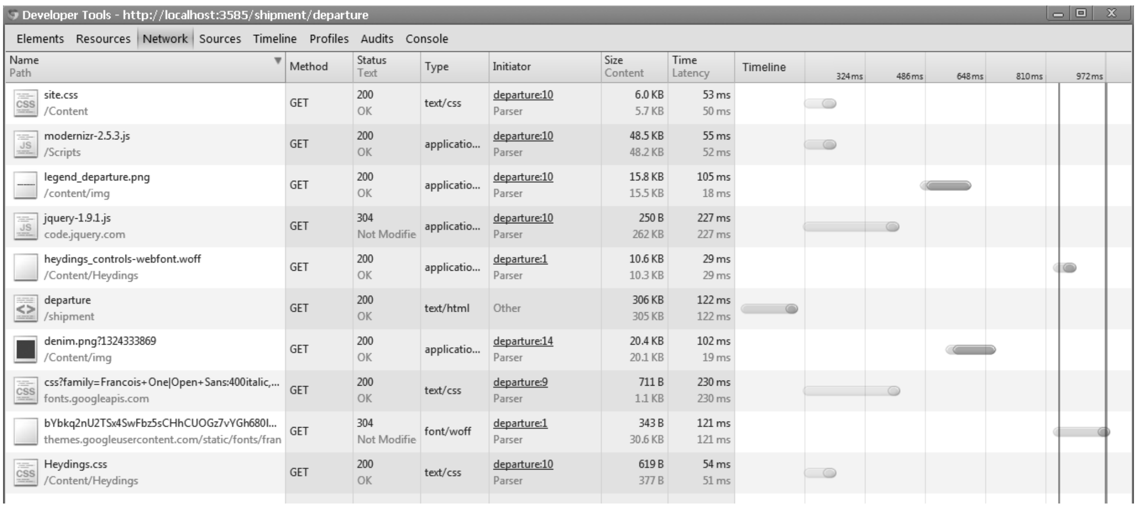


Figure 62. Reader eyes motion tracking.

Other minor changes based on user feedbacks. Some of them are 1) addition of a board of legends under the page header as a reference for new users, 2) elimination of the airplane icon to fit with multimodal purpose of the view, and 3) replacement of “W” for “Week” due to misleading interpretation (In Wärtsilä “W[number]” stands for an engine type, e.g. W32).

3.2.4. Improve page load

In the first iteration, decision to substitute graphics with `webfonts` and SVG has boosted page load due to a reduction in data exchanged. In the second iteration, there is more focus to further reduce the amount of data loaded to a page. By using element inspection tool provided by Chrome web browser, developer is able to identify a number of areas of improvement. Firstly obsolete CSS elements are eliminated. Secondly obsolete `webfonts` are removed. Lastly css files are placed in front of js files to allow parallel downloading of the two sets of files. Test results shows good data. Pages now load 25% faster; with the fastest speed measured at 238ms.



Name / Path	Method	Status / Text	Type	Initiator	Size / Content	Time / Latency
site.css /Content	GET	200 OK	text/css	departure:10 Parser	6.0 KB 5.7 KB	53 ms 50 ms
modernizr-2.5.3.js /Scripts	GET	200 OK	applicatio...	departure:10 Parser	48.5 KB 48.2 KB	55 ms 52 ms
legend_departure.png /content/img	GET	200 OK	applicatio...	departure:10 Parser	15.8 KB 15.5 KB	105 ms 18 ms
jquery-1.9.1.js code.jquery.com	GET	304 Not Modifie	applicatio...	departure:10 Parser	250 B 262 KB	227 ms 227 ms
heydings_controls-webfont.woff /Content/Heydings	GET	200 OK	applicatio...	departure:1 Parser	10.6 KB 10.3 KB	29 ms 29 ms
departure /shipment	GET	200 OK	text/html	Other	306 KB 305 KB	122 ms 122 ms
denim.png?1324333869 /Content/img	GET	200 OK	applicatio...	departure:14 Parser	20.4 KB 20.1 KB	102 ms 19 ms
css?family=Francois+One|Open+Sans:400italic,... fonts.googleapis.com	GET	200 OK	text/css	departure:9 Parser	711 B 1.1 KB	230 ms 230 ms
bYbkq2nU2TSx4SwFbz5sCHhCUOGz7vYGh680l... themes.googleusercontent.com/static/fonts/fran	GET	304 Not Modifie	font/woff	departure:1 Parser	343 B 30.6 KB	121 ms 121 ms
Heydings.css /Content/Heydings	GET	200 OK	text/css	departure:10 Parser	619 B 377 B	54 ms 51 ms

Figure 63. Network statistics before tuning.

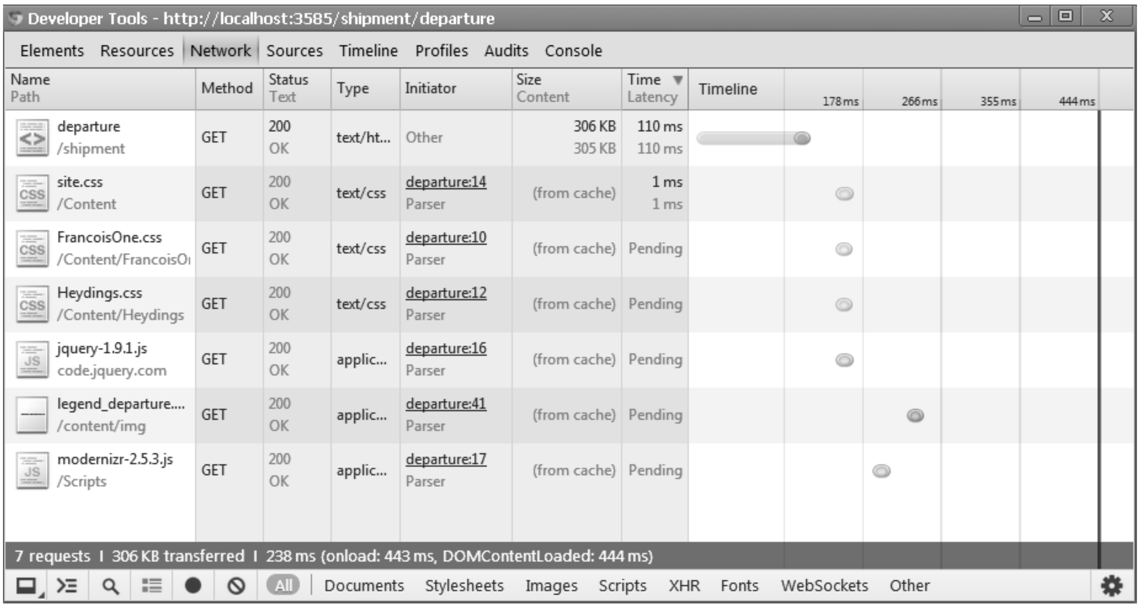


Figure 64. Network statistics after tuning.

3.2.5. Physical deployment

Although being accessible from desktop computer web browsers, the real-time shipment status display is designed to be display in the place that is most convenient for mobile people: in the corridors. Different corridors are examined in order to find a place that attracts the most traffic. The desired place is found and the display is planned to be mounted on the wall at the cross road of two main streets and a coffee break hub. Screen placement at this point is meant for two groups of readers: 1) those in motion and 2) those who have coffee break. A 40-inch display is purchased to fit into this location. Although not yet mounted on the wall, a separate test with the screen displaying shipment statuses yields quality as expected with desktop monitors.

Figure 65. Physical screen deployment floor plan.

Figure 66. Shipment statuses displayed on a big screen.

4. DISCUSSION

The real-time shipment status display system provides better inter-organization communication between project management and logistics control. While advanced information systems has provided a wealth of transactional data that benefit operational people work, the displays finds its unique position in providing management stakeholders better visibility to the supply chain activities without being too detailed. Data generalization mechanism and simple design have transformed raw data into a more useful presentation. In addition, the use of web technology allows accessibility from a wide range of devices and screen sizes. These characteristics result in better decision making. These benefits have brought further applications to the screen that initially were not in the scope of design.

The display if deployed in a factory shop floor will allow planners to prioritize workforce to focus on the most important orders. The deployment will increase communication between different project side, logistics control, and far-reaching networks such as in China, Brazil, and Italy. Reaching out of internal network, the display system can be used also to communicate shipment status to end customers. A screenshot of the screen was shown and has received warm reception from the customer side.

These applications have shown that the deployment of the display system in the supply chain will help organization centralize its messages to all stakeholders, helping them focus on those activities that matter to the customers. The ability to prioritize work will help smoothing bottlenecks, thus improve overall supply chain services. The display system also helps reduce unnecessary reports, giving employee more time to focus on data quality and better decision. With more awareness of the supply chain condition, the display contributes in transforming the supply chain into pulling nature.

From the system design point of view, the display system is designed to be intuitive, accessible, up-to-date, fast, and reliable. In term of system architecture, it is designed for the ease of maintenance and upgrade. The system comprises of two independent sub-systems; one is responsible for data collection and the other is for display, glued by a concrete data model. As the interface is clearly defined, a sub-system can be upgraded or replaced without affecting the other. While already functional, the system still needs further improvements to be truly real-time. More improvement needs to be done in the data collection sub system. Faster services such as enterprise service bus (ESB) to integrate data between SAP and the database should be developed. The advent of Websockets (Ubl & Kitamura 2010) has provided a new technology to eliminate periodic pooling and enable synchronous data refresh between server and client. Reduction / elimination of client-server connection latency, accompanied by a reduction of data synchronization from SAP will make the system closer to being a synchronous system.

Although designed for layout uniformity across devices and screens, the display can be more context-aware. For example, if it is displayed in a room it should only show shipments that belong to people sitting in that room. If it is displayed in a factory, only shipments that belong to the factory should be displayed. Lastly, a customer should not see another customer’s shipments.

All in all the thesis work has provided a functional system and a flexible architectural foundation that can be used, studied, and further enhanced to make extensive real-time logistics concise and delivered to the right stakeholders, thus increases company competitive advantage in logistics execution.

REFERENCES

Allen, K. Scott (2013). *Building Application with ASP.NET MVC 4* [online]. Pluralsight [cited 29 July 2013]. Available from World Wide Web: <ULR: http://pluralsight.com/training/Player?author=scott-allen&name=mvc4-building-m1-intro&mode=live&clip=0&course=mvc4-building>.

Abu, F. A. (2012). *A Business Process Modeling-Based Approach to Investigate Complex Processes: Software Development Case Study*. Business Process Management Journal 18:1, 122-137.

Ateş, Faruk, Alex Sexton, Paul Irish, Ryan Seddon, Alexander Farkas. *Moderniz Documentation* [online] [cited 28 July.2013]. Available from World Wide Web: <ULR: http://modernizr.com/docs/#installing>.

Chadwick, Jess, Todd Snyder & Hrusikesh Panda (2012). *Programming ASP.NET MVC 4*. Sebastopol: O'Reilly Media, Inc. 492p. ISBN: 978-1-449-32031-7.

Cohrs, Christian, Gisbert Loff (2003). *The SAP Gui Scripting API* [online]. *SAP AG* [cited 22 July. 2013]. Available from World Wide Web: <URL: http://www.sdn.sap.com/irj/scn/go/portal/prtroot/docs/library/uuid/80aaac18-2dfe-2a10-bbb1-ec9b3760ea4c?QuickLink=index&overridelayout=true&28385438934341>.

Cooper, Alan, Rober Reimann & Dave Cronin (2007). *About Face 3 - The Essentials of Interaction Design*. Indianapolis: Wiley Publishing, Inc. ISBN: 978-0-470-08411-3.

Corcoran, C. T. (1998). *ERP is Changing Manufacturing Jobs*. Infoworld 20:28, 100.

Do, Sung-Hee & Nam P. Suh (1999). *Systematic OO Programming with Axiomatic Design* [online] [cited 28 July. 2013]. Available from World Wide Web: <URL:

https://www.dropbox.com/s/si0625h63aqhel4/Do%2C%20Sung-Hee%2C%20Suh%20Nam%20P%20%281999%29%20Systematic%20OO%20programming%20with%20Axiomatic%20Design.PDF>.

Ehrs, Michael, Petri Helo, Bjarne Nordlund, Pasi Lähde, Jussi Mäkiranta, Ta Hieu (2012). *LogTrack Project Final Report*. Vaasa: University of Vaasa.

El-Haik, B. & Al-Aomar, R. (2006). *Simulation-Based Lean Six-Sigma and Design for Six-Sigma*.

El-Haik, Basem (2005). *Axiomatic Quality: Integrating Axiomatic Design with Six-Sigma, Reliability, and Quality Engineering*. New Jersey: John Wiley & Sons. 312 p. ISBN: 978-0471682738.

Farahani, Reza Zanjirani, Shabnam Rezapour & Laleh Kardar (2011). *Logistics Operations and Management Concepts and Models.* 1st Ed. London: Elsevier. 469 p. ISBN 978-0-12-385202-1.

Gallagher, Nicolas (2012). *About normalize.css* [online] [cited 28 July. 2013]. Available from World Wide Web: <ULR: http://nicolasgallagher.com/about-normalize-css/>.

GS1 (2008). *Standard International Logistics Label – STILL* [online] [cited 22.08.2013]. Available from World Wide Web: <ULR: http://www.gs1.org/docs/transportlogistics/GS1_STILL.pdf>.

GS1 (2012). Use of the Serial Shipping Container Code (SSCC) in the retail sector [online] [cited 20.08.2013]. Available from World Wide Web: <ULR: http://www.youtube.com/watch?v=R2pNIbI6kAs>.

GS1 (2013). GS1 Architecture Rapid eLearning March 2013 [online] [cited 21.08.2013]. Available from World Wide Web: <ULR: http://www.gs1.org/docs/training/architecture_elearn/index.html>.

Giachetti, Ronald E. (2010). *Design of Enterprise Systems: Theory, Architecture, and Methods*. Boca Raton: CRC Press. 447 p. ISBN 978-1-4398-1823-7.

Google (2013). *HTML5 Features Presentation* [online] [cited 28 July 2013]. Available from World Wide Web: <ULR: http://www.html5rocks.com/en/features/presentation>.

He, W. (2009). *A Solution for Integrated Track and Trace in Supply Chain Based on RFID & GPS*. Emerging Technologies & Factory Automation, 2009.ETFA 2009.IEEE Conference on , 1.

Heldman, Kim, Claudia Baca, Patti Jansen (2007). *Project Management Professional Exam Study Guide Deluxe Edition*. 2nd Ed. Indianapolis: Wiley Publishing, Inc. ISBN: 978-0-470-15251-5.

Hopp, Wallace J. (2003). *Supply Chain Science* [online]. Pohang: Pohang University of Science & Technology [cited 05 July. 2013]. Available from World Wide Web: <URL: http://proman.postech.ac.kr/lectures/MEIE676/Supply%20Chain%20Science.pdf>.

Irish, Paul (2013). *Quick guide to webfonts via @font-face* [online] [cited 28 July. 2013]. Available from World Wide Web: <ULR: http://www.html5rocks.com/en/tutorials/webfonts/quick/>.

Krug, Steve (2006). *Don't Make Me Think! A common sense approach to we usability*. 2nd Ed. Berkeley: New Riders. 216p. ISBN 0-321-34475-8

Lambert, Dougals M. & James R. Stock (1993). *Strategic Logistics Managment*. 3rd Ed. Chicago: Richard D. Irwin, Inc.. 862 p. ISBN 0-256-08838-1.

Lauterbach, Bernd, Rudiger Fritz, Jens Gottlieb, Bernd Mosbrucker, Till Dengel (2009). *Transportation Management with SAP TM* [online]. Bonn: Galileo Press [cited 11 July. 2013]. Available from World Wide Web: <URL: http://www.sap-press.de/download/dateien/1823/sappress_transportation_management_with_sap.pdf>.

Lee, Dai Gil & Nam Pyo Suh (2006). *Axiomatic Design and Fabrication of Composite Structures: Application in Robots, Machine Tools, and Automobiles*. New York: Oxford University Press. 724 p. ISBN: 0-19-517877-7.

Maciaszek, L. & Liong, B. L. (2004). *Practical Software Engineering: A Case-Study Approach*. England: Addison Wesley. 864 p. ISBN: 0321204654.

Maciaszek, Leszek , Bruce Lee Liong (2005). *Practical Software Engineering: An Interactive Case-Study Approach to Information Systems Development*. Essex: Pearson Education Limited. 864 p. ISBN 0 321 20465 4.

Manian, Divya (2009). *The Truth about Doctypes* [online] [cited 28 July.2013]. Available from World Wide Web: <ULR: http://nimbupani.com/the-truth-about-doctypes.html>.

Marcotte, Ethan (2009). *Fluid Grids* [online] [cited 28 July 2013]. Available from World Wide Web: <ULR: http://alistapart.com/article/fluidgrids>.

Marcotte, Ethan (2010). *Responsive Web Design* [online] [cited 28 July. 2013]. Available from World Wide Web: <ULR: http://alistapart.com/article/responsive-web-design>.

Nguyen, Duy (2010). *Touch Screen Operated Data Warehouse Application* [online]. Vaasa: Vaasa University of Applied Sciences [cited 07.09.2013]. Available from World Wide Web: <URL: http://publications.theseus.fi/bitstream/handle/10024/23936/Nguyen_Duy.pdf>.

Pimmler, Thomas U. & Steven D. Eppinger (1994) [online]. *Integration Analysis of Product Decompositions* [cited 27 July. 2013]. Cambridge: Massachusetts Institute of Technology. Available from World Wide Web: <URL: http://web.mit.edu/eppinger/www/pdf/Pimmler_DTM1994.pdf>.

Royce, Winston W. (1970). *Managing the Development of Large Software Systems* [online] [cited 28 July. 2013]. Available from World Wide Web: <URL: http://leadinganswers.typepad.com/leading_answers/files/original_waterfall_paper_winston_royce.pdf>.

S.-J. Kim (1991). *Design of Software System Based on Axiomatic Design*. CIRP Annals - Manufacturing Technology 40:1, 165-170.

Sage, Andrew P., James E. Armstrong, Jr. (2000). Introduction to Systems Engineering. New York: John Wiley & Sons. 568 p. ISBN 978-0471027669.

Shamsuzzoha, A. (2011*). Logistics Tracking: An Implementation Issue for Delivery Network*. Technology Management in the Energy Smart World (PICMET), 2011 Proceedings of PICMET '11.

Shamsuzzoha, Ahm & Petri Helo (2011). *Real-time Tracking and Tracing System: Potentials for the Logistics Network*. Kuala Lumpur: Proceedings of the 2011 International Conference on Industrial Engineering and Operations Management [cited 09 July. 2013]. Available from World Wide Web: <URL: http://www.iieom.org/ieom2011/pdfs/IEOM038.pdf>.

Sandhu, M. (2006) 'Project logistics with the dependency structure matrix approach – an analysis of a power plant delivery', Int. J. Logistics Systems and Management, Vol. 2, No. 4, pp.387–403.

Sikos, Leslie F. (20011). *Web Standards - Mastering HTML5, CSS3, and XML*. New Yord: Apress. 412p. ISBN: 978-1-4302-4042-6.

Sommerville, Ian (2007). *Software Engineering*. 8th Ed. Essex: Pearson Education Limited. 864 p. ISBN 978-0321313799.

Suh, Nam P. (1997). *Design of Systems* [online] [cited 21 July. 2013]. Available from World Wide Web: <ULR: https://www.dropbox.com/s/1rk3pnza35vvgac/Suh%20Nam%20P%20%281997%29.%20Design%20of%20Systems.PDF>.

Symonds, J. A. (2008). *Using RFID to Track and Trace High Value Products: The Case of City Healthcare*. Journal of Cases on Information Technology 10:1, 1-13.

Ubl, Malte & Eiji Kitamura (2010). *Introducing WebSockets: Bringing Sockets to the Web* [online]. Google [cited 03 August 2013]. Available from World Wide Web: <ULR:http://www.html5rocks.com/en/tutorials/websockets/basics/>.

W3C (2011). *Scalable Vector Graphics (SVG) 1.1 (Second Edition)* [online] [cited 28 July 2013]. Available from World Wide Web: <ULR: http://www.html5rocks.com/en/features/presentation>.

W3C (2013). *CSS Values and Units Module Level 3* [online] [cited 28 July 2013]. Available from World Wide Web: <ULR: http://www.w3.org/TR/css3-values/#rem-unit>.

Williams, Glynn C. (2008). *Implementing SAP ERP Sales & Distribution*. 530p. New York: McGraw-Hill. ISBN: 0-07-164358-3.

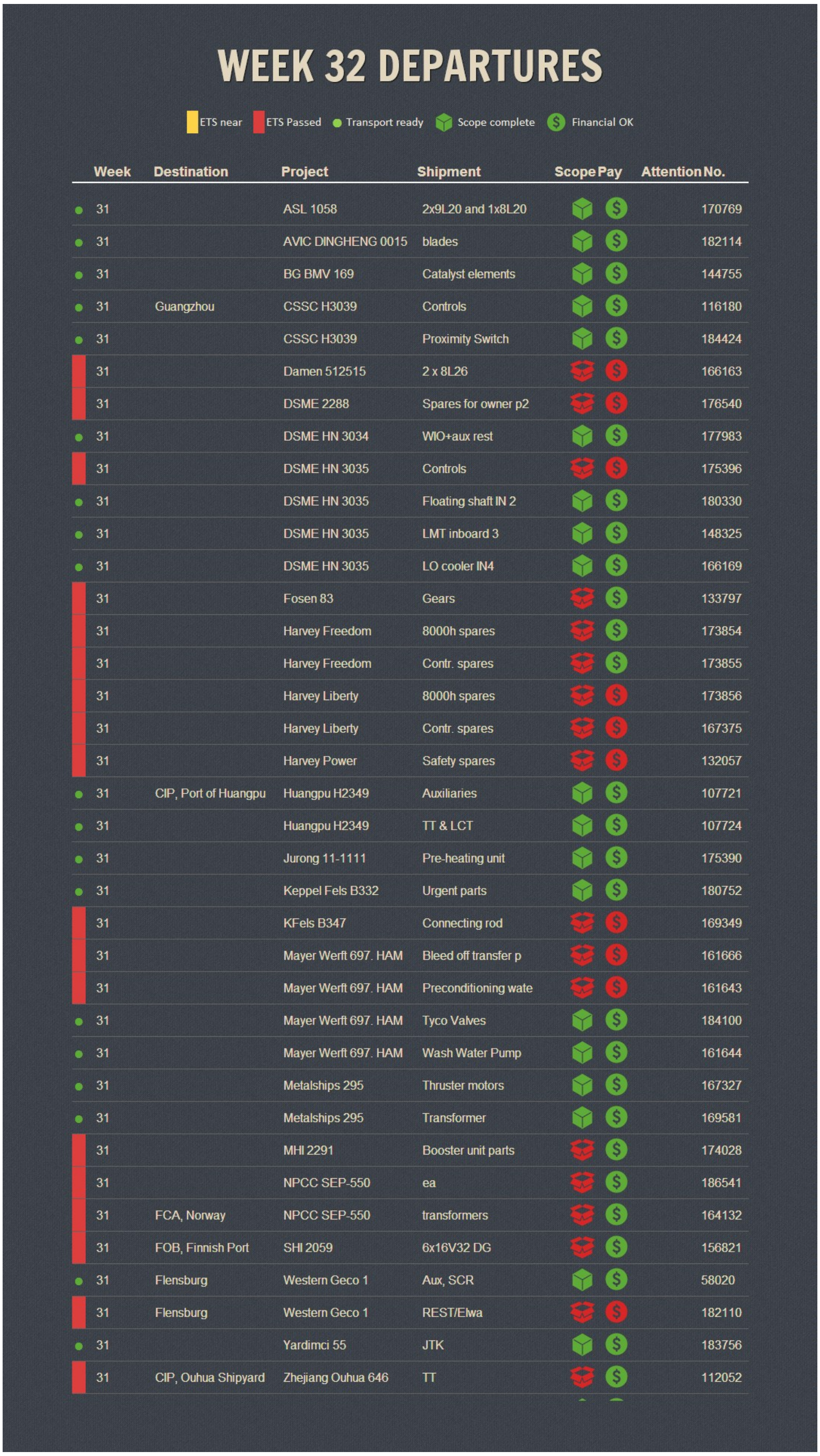
APPENDIX 1. Full Design Matrix for FR1 and FR2

	DP0: Design Parameters (How)	DP1: Database design technique	DP1.1: Entity-relationship technique	DP1.2: Within two weeks back and fort from this week	DP1.3: 1 minute	DP1.4: 6:00 to 21:00	DP2: Scheduled Data gathering technology	DP2.1: Database management system and storage location	DP2.1.1: Microsoft SQL Server DBMS	DP2.1.2: Existing test server	DP2.2: MS Excel macro to read data	DP2.2.1: SAP data connection mechanism	DP2.2.1.1: Class CSAPConnection	DP2.2.1.2: Class CSAPSessionHandler	DP2.2.2: SAP data retrieval mechanism	DP2.2.2.1: function getVT11List()	DP2.2.2.2: function getZ_ShipDest1List()	DP2.3: MS Excel macro to sync data	DP2.3.1: function getlShipments()	DP2.3.2: function getAndMergeDShipments()	DP2.3.3: functinon InsertAllShipmentsToDatabase()
FR0: Functional Requirements (What)	X																				
FR1: Provide good data		X					O	O	O	O	O	O	O	O	O	O	O	O	O	O	O
FR1.1: Concise set of models			X	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O
FR1.2: Limited data			O	X	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O
FR1.3: Frequency of update			X	X	X	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O
FR1.4: Time of update			O	X	X	X	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O
FR2: Provide efficient scheduled data I/O		X	O	O	O	O	X														
FR2.1: Quick data access		O	O	O	O	O		X			O	O	O	O	O	O	O	O	O	O	O
FR2.1.1: Database management system		O	O	O	O	O			X	O	O	O	O	O	O	O	O	O	O	O	O
FR2.1.2: Database storage location		O	O	O	O	O			X	X	O	O	O	O	O	O	O	O	O	O	O
FR2.2: Read SAP data		O	O	O	O	O		O	O	O	X							O	O	O	O
FR2.2.1: Manage SAP connection		O	O	O	O	O		O	O	O		X			O	O	O	O	O	O	O
FR2.2.1.1: Get SAP GuiConnection		O	O	O	O	O		O	O	O			X	O	O	O	O	O	O	O	O
FR2.2.1.2: Get SAP GuiSession		O	O	O	O	O		O	O	O			X	X	O	O	O	O	O	O	O
FR2.2.2: Save data from SAP to local storage		O	O	O	O	O		O	O	O		X	X	X	X			O	O	O	O
FR2.2.2.1: From transaction VT11		O	O	X	O	O		O	O	O		O	O	O		X	O	O	O	O	O
FR2.2.2.2: From transaction Z_SHIPDEST1		O	O	X	O	O		O	O	O		O	O	O		X	X	O	O	O	O
FR2.3: Sync data to database		O	O	O	O	O		X	O	O	X	O	O	O	O	O	O	X			
FR2.3.1: Get shipments collection from local storage		O	O	O	O	O		O	O	O	O	O	O	O	O	O	O		X	O	O
FR2.3.2: Get and merge shipments collection from database		O	O	O	O	O		O	O	O	O	O	O	O	O	O	O		X	X	O
FR2.3.3: Write merged shipments to database		O	O	O	O	O		O	O	O	O	O	O	O	O	O	O		O	X	X
FR2.4: Scheduled data sync		O	O	O	O	O		O	O	O	X	O	O	O	O	O	O	X	O	O	O

APPENDIX 2. Full Design Matrix for FR3 and FR4

	DP0:	DP1: Database design technique	DP2: Scheduled data gathering technology	DP3: Layout design techniques	DP3.1: Mobile person in need for spot-on and fast layout	DP3.1.1: Airport screen inspired layout	DP3.1.2: Design for scanning, not reading	DP3.2: Transient posture	DP3.2.1: Less data, no logistics jargon	DP3.2.2: Clear text, clear graphics	DP3.2.3: No interpretation, no guessing	DP4: Web technology	DP4.1: Client-side adaptive screen layout	DP4.1.1: HTML5, CSS3, and JavaScript	DP4.1.1.1: <!DOCTYPE html>	DP4.1.1.2: normalize.css	DP4.1.1.3: Modernizr.js	DP4.1.1.4: JQuery	DP4.1.2: CSS3	DP4.1.2.1: rem unit	DP4.1.2.2: @media	DP4.1.2.3: webfont and SVG	DP4.2: ASP.NET	DP4.2.1: MVC4	DP4.2.2: Free of charge	DP4.2.2.1: MS Visual Web Developer 2010 Express	DP4.2.2.2: Google Chrome Developer Tools
FR0: Functional Requirements (What)	X																										
FR1: Provide good data		X	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O
FR2: Provide efficient scheduled data I/O		O	X	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O
FR3: Provide intuitive layout		O	O	X								O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O
FR3.1: Know user persona		O	O		X			O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O
FR3.1.1: Speak their language		O	O			X	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O
FR3.1.2: Quick		O	O			O	X	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O
FR3.2: Know application posture		O	O		X	O	O	X				O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O
FR3.2.1: Simple		O	O		O	O	O		X	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O
FR3.2.2: Clear		O	O		O	O	O		O	X	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O
FR3.2.3: To-the-point		O	O		O	O	O		X	X	X	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O
FR4: Provide high accessibility		X	X	X	O	O	O	O	O	O	O	X															
FR4.1: Support multiple reading devices		O	O	O	O	O	O	O	O	O	O		X										O	O	O	O	O
FR4.1.1: Cross-browser compatible		O	O	O	O	O	O	O	O	O	O			X					O	O	O	O	O	O	O	O	O
FR4.1.1.1: Avoid browser's quirks mode		O	O	O	O	O	O	O	O	O	O				X	O	O	O	O	O	O	O	O	O	O	O	O
FR4.1.1.2: Consistent elements rendering		O	O	O	O	O	O	O	O	O	O				O	X	O	O	O	O	O	O	O	O	O	O	O
FR4.1.1.3: Browser feature dection		O	O	O	O	O	O	O	O	O	O				O	O	X	O	O	O	O	O	O	O	O	O	O
FR4.1.1.4: Consistent JavaScript behavior		O	O	O	O	O	O	O	O	O	O				O	O	O	X	O	O	O	O	O	O	O	O	O
FR4.1.2: Cross-device compatible		O	O	O	O	O	O	O	O	O	O			O	O	O	O	O	X				O	O	O	O	O
FR4.1.2.1: Relative unit of measurement		O	O	O	O	O	O	O	O	O	O			O	O	O	O	O		X	O	O	O	O	O	O	O
FR4.1.2.2: Media queries		O	O	O	O	O	O	O	O	O	O			O	O	O	O	O		X	X	O	O	O	O	O	O
FR4.1.2.3: Undistorted graphics		O	O	O	O	O	O	O	O	O	O			O	O	O	O	O		O	O	X	O	O	O	O	O
FR4.2: Capable web platform		O	O	O	O	O	O	O	O	O	O		O	O	O	O	O	O	O	O	O	O	X				
FR4.2.1: Provide clear separation of development concerns		O	O	O	O	O	O	O	O	O	O		O	O	O	O	O	O	O	O	O	O		X	O	O	O
FR4.2.2: Good SDK		O	O	O	O	O	O	O	O	O	O		O	O	O	O	O	O	O	O	O	O		X	X		
FR4.2.2.1: For development		O	O	O	O	O	O	O	O	O	O		O	O	O	O	O	O	O	O	O	O		O		X	O
FR4.2.2.2: For performance optimization		O	O	O	O	O	O	O	O	O	O		O	O	O	O	O	O	O	O	O	O		O		O	X

APPENDIX 3. Full Screen Departure View



WEEK 32 DEPARTURES

ETS near | ETS Passed | Transport ready | Scope complete | Financial OK

Week	Destination	Project	Shipment	Scope	Pay	Attention	No.
31		ASL 1058	2x9L20 and 1x8L20				170769
31		AVIC DINGHENG 0015	blades				182114
31		BG BMV 169	Catalyst elements				144755
31	Guangzhou	CSSC H3039	Controls				116180
31		CSSC H3039	Proximity Switch				184424
31		Damen 512515	2 x 8L26				166163
31		DSME 2288	Spares for owner p2				176540
31		DSME HN 3034	WIO+aux rest				177983
31		DSME HN 3035	Controls				175396
31		DSME HN 3035	Floating shaft IN 2				180330
31		DSME HN 3035	LMT inboard 3				148325
31		DSME HN 3035	LO cooler IN4				166169
31		Fosen 83	Gears				133797
31		Harvey Freedom	8000h spares				173854
31		Harvey Freedom	Contr. spares				173855
31		Harvey Liberty	8000h spares				173856
31		Harvey Liberty	Contr. spares				167375
31		Harvey Power	Safety spares				132057
31	CIP, Port of Huangpu	Huangpu H2349	Auxiliaries				107721
31		Huangpu H2349	TT & LCT				107724
31		Jurong 11-1111	Pre-heating unit				175390
31		Keppel Fels B332	Urgent parts				180752
31		KFels B347	Connecting rod				169349
31		Mayer Werft 697. HAM	Bleed off transfer p				161666
31		Mayer Werft 697. HAM	Preconditioning wate				161643
31		Mayer Werft 697. HAM	Tyco Valves				184100
31		Mayer Werft 697. HAM	Wash Water Pump				161644
31		Metalships 295	Thruster motors				167327
31		Metalships 295	Transformer				169581
31		MHI 2291	Booster unit parts				174028
31		NPCC SEP-550	ea				186541
31	FCA, Norway	NPCC SEP-550	transformers				164132
31	FOB, Finnish Port	SHI 2059	6x16V32 DG				156821
31	Flensburg	Western Geco 1	Aux, SCR				58020
31	Flensburg	Western Geco 1	REST/Elwa				182110
31		Yardimci 55	JTK				183756
31	CIP, Ouhua Shipyard	Zhejiang Ouhua 646	TT				112052

APPENDIX 4. Full Screen Arrival View

WEEK 32 ARRIVALS

ETA Passed Arrived

Week	Destination	Project	Shipment	Attention	No.
29		Huangpu 8000kW-16	CPP From CN	Arrived w 29	58655
31		BP Quad 204 at HHI	Generator pump	Arrived w 31	183104
31		Celiktrans 44	Rest PO4502454396	Arrived w 31	177985
31		Concordia 674	coupling	Arrived w 31	183470
31		CSBC Navy H1025	rest parts	Arrived w 31	150955
31		DSME 2296	hex nut washer	Arrived w 31	184110
31	DAP, Kherson Shipyard	Kherson 8003	Electronic board tel	Arrived w 31	180756
31	Geoje Si, Gyeongsangnam Do	SHI CARDIFF 1979	masino	Arrived w 31	183758
31		YX3195 - 90m Coastal Oil Tanker	JTK	Departed w 31	182118
32		Armon G002	vulkan	Arrived w 32	183105
32		DSME 2296	Gaskets for LT CW pi	Arrived w 32	183311
32		DSME 3033	Valve assembly	Departed w 31	171469
32	CIP, Leer	Ferus Smit NB404	9L32	Arrived w 32	140691
32	DAP, Yard "GRYFIA"	Gryfia LC01	CPP 4D505	Arrived w 32	148522
32		SHI 1941	Aux air	Departed w 31	181881
32		TERSAN 1019	WGLS	Arrived w 32	177536
32		TERSAN 1020	Circulation pump	Arrived w 32	184101
32	CIP, Port of Zhenjiang	Zhenjiang VZJ 6258-1201	Stern. seals & tanks	Arrived w 32	177679
37		DSME HN 3035	LMT inboard 1	Departed w 32	108495
37	FOB, Hamburg	Hyundai MIPO H8090	Hydraulic Equipment	Departed w 32	124938
37		Zhenjiang VZJ 6258-1201	6L32 genset,aux	Departed w 32	107740
37		Zhenjiang VZJ 6258-1201	El&aut	Departed w 32	166167
38	ULSAN	BP Quad 204 at HHI	Thrust. scope small	Departed w 31	181506
38	FCA, Mäntyluoto	HHI 2606	auxiliaries	Departed w 31	169939
38		SHI 1941	AUX	Departed w 31	177299
38	FCA, Mäntyluoto	SHI 2049	auxiliaries	Departed w 31	170761