On sustainable operation of warehouse order picking systems

On sustainable operation of warehouse order picking systems

R. Andriansyah[a], **L.F.P. Etman**[a], **J.E. Rooda**[a]

[a]*Department of Mechanical Engineering, Eindhoven University of Technology, The Netherlands*
(r.andriansyah@tue.nl, l.f.p.etman@tue.nl, j.e.rooda@tue.nl)

Abstract: Sustainable development calls for an efficient utilization of natural and human resources. This issue also arises for warehouse systems, where typically extensive capital investment and labor intensive work are involved. It is therefore important to assess and continuously monitor the performance of such a system to identify possible improvements in the system configuration. We believe that a modular system architecture and an accompanying performance monitoring method serve this purpose. In this paper we advocate a system architecture with a decentralized hierarchical control structure. The architecture allows easy adjustment of system configurations and control heuristics to deal with ever-changing warehouse requirements. We also propose a performance monitoring method that only requires little shop-floor data. This method is based on the concept of effective process time and it can be used to generate key performance indicators of the warehouse. By applying the system architecture and the performance monitoring method, we believe that more efficient ways of utilizing resources and capitals can be identified to improve the sustainability of warehouse order picking systems.

Keywords: Order picking, System architecture, Decentralized control, Performance monitoring, Effective process time

1. Introduction

Warehouses serve an irreplaceable role in the supply chain operation. Some of their main functions are to better match supply with customer demands, consolidate products, reduce transportation cost, provide customer service and provide value-added processing (Bartholdi III and Hackman, 2008). Unfortunately, warehouses require large investment on labor, capitals (including land and material handling equipments), and information systems. This is particularly true for automated warehouses, which are nowadays becoming a common practice. Automated warehouses typically have higher consumption of energy and material than conventional warehouses.

These facts emphasize the relevance of sustainable design and operation of warehouses. After all, sustainable development calls for fulfilment of the needs of industry today, while protecting, sustaining, and enhancing the human and natural resources that will be needed in the future (IISD, 1992). Also, fair and efficient use of resources with respect to meeting human needs is one of the four system conditions for sustainable development according to Robert et al. (1997). Specifically for warehouses, the main parameters for sustainability are cash flow, warehouse utilization, carbon emissions and minimizations, order processing time, employee job satisfaction, and social and environmental impacts on surrounding areas (Tan et al., 2009). In this paper we argue that a flexible system architecture and an accompanying performance monitoring system are two indispensable ingredients to ensure efficient utilization of warehouse resources and thus a sustainable warehouse operation. We specifically consider warehouses with Automated Storage/Retrieval Systems referred to as the miniload-workstation (ML-WS) order picking systems.

First, we advocate a modular system architecture for ML-WS order picking systems. The architecture applies a decentralized hierarhical control structure and operational layers. These arrangements allow one to perform local adjustments easily at one part of the system without affecting the other parts of the system. As such, we believe that the system architecture has the necessary flexibility to cope with the ever-changing warehouse requirements. Furthermore, the system architecture can be easily implemented into a simulation model to evaluate what-if scenarios.

Second, we propose a performance monitoring method based on the concept of Effective Process Time (EPT). The method requires little data that can be directly obtained from the shop-floor. Our goal is to develop a practical method to monitor the real-time performance of warehouses such that any deviation from the expected performance can be detected timely. We illustrate the application of the method in one part of the warehouse namely the order picking workstation.

The remainder of this paper is organized as follows. Section 2 describes a miniload-workstation order picking system. Section 3 discusses the proposed system architecture. Section 4 presents the EPT-based performance monitoring method. Finally, Section 5 concludes the paper.

2. An ML-WS order picking system

Figure 1 shows an example of an ML-WS order picking system. This example is based on an existing warehouse. The physical structure of this system is elaborated in Section 2.1. Sections 2.2 and 2.3 describe respectively the storage and retrieval, and the item picking operation performed in such systems.

2.1. Physical structure

Three areas can be distinguished in the miniload-workstation order picking system, namely *miniloads, workstations,* and *conveyors.* Miniloads provide temporary storage spaces for product totes. At the workstation, items are picked from product totes and put into order totes. Conveyors connect the miniload area to the workstation area, and the other way around, for transporting the product totes.

Miniloads are automated storage racks equipped with cranes to serve two functions, namely the storage and retrieval of product totes. Each miniload consists of two single-deep racks with a single crane in the middle to access product totes. The cranes move horizontally along the aisle between the racks, while the holder of product totes move vertically to store or retrieve the totes.

An order picking workstation has a number of input and output buffers. A picker is available at the workstation to serve customer orders. The picker picks items from the product totes and put them into the corresponding order tote. If all items for an order have been picked, then the order is said to be finished and the next order can be processed.

The central conveyor loop transports product totes from the miniload area to the workstation area, and the other way around. As there is only a limited number of positions on the conveyor, only product totes that have successfully reserved a position are allowed to enter the conveyor.

2.2. Storage and retrieval

Storage happens when a product tote needs to be kept temporarily in the miniload until it is required to fulfill an order. Two types of product tote exist, namely *replenishment* and *returning* product totes. A replenishment product tote is a new tote that is full with items. A returning product tote is a tote that has just finished being picked at the workstation but still contains some items left.

Retrievals take place at the miniload and start when the next order to be completed has been chosen from a list of all available orders. The chosen order is further divided into jobs, which specify the SKU types and the required number of items to be picked. These jobs are then assigned to the five miniloads. When a miniload is assigned with a retrieval job, it reserves a number of product totes until the required quantity of items is covered by the items in the reserved tote(s). Once a product tote is reserved for a job, items in that tote can only be used to fulfill that particular job and may not be used for other jobs. The reserved totes are retrieved by the miniload cranes and put on the output buffer of the miniload. The

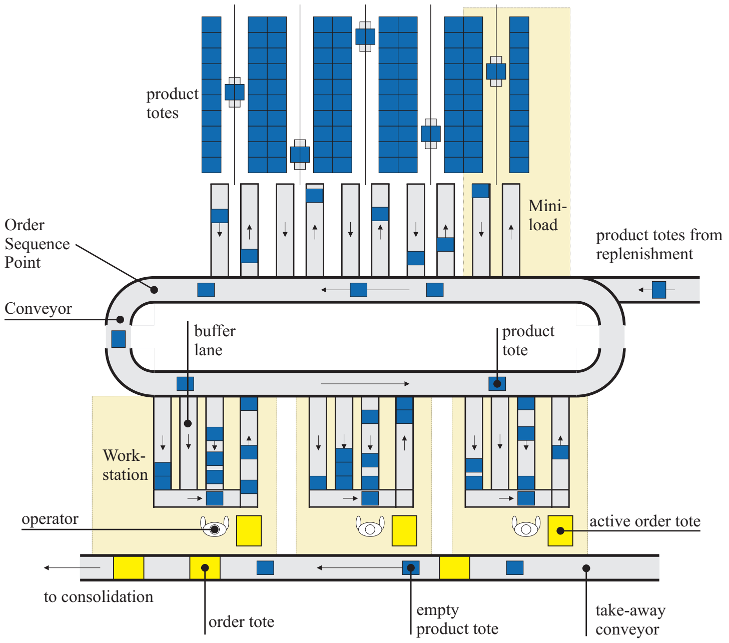


Figure 1: Miniload-workstation order picking system.

totes wait until they get access to the central conveyor loop to be sent to one of the workstations.

2.3. Item picking

Once a product tote has reached its destination workstation, an operator picks the required amount of items and put the item(s) into an order tote. When all items required for an order are already picked, the order tote is moved to the take-away conveyor.

Following item picking, the operator checks whether the product tote has become empty. If this is the case, the empty product tote is put on the take-away conveyor along with the finished order totes to be sent to a consolidation area. Alternatively, if the product tote still contains any items left, the tote is put on the central conveyor loop to be stored again in one of the miniloads.

3. System architecture for ML-WS systems

We propose a system architecture as shown in Figure 2 to be used for ML-WS systems. This architecture has been developed in such a way that modularity is supported. The key features of the architecture is a decentralized hierarchical control structure with a clear distinction between operational layers. With these features, changes pertaining to system configurations and control heuristics can be made locally with as little influence as possible on the other parts of the system. The architecture is also easy to comprehend intuitively.

3.1. Areas and layers

In the system architecture we define areas and operational layers. We distinguish three areas and four layers, as shown in Figure 2. Here, circles represent processes and arrows represent communication between (two) processes. Similar to the physical structure of the system, the three areas are miniload, workstation, and conveyor area, respectively. The four operational layers are *order layer, global control layer, local control layer,* and *material flow layer* (see Figure 2).

The *order* layer contains all operations that are related to the administration of demand and supply. These operations include the creation of new customer orders by order generator `GO` and the placement of inventory replenishment orders by replenishment planner `PR`. The customer orders are delivered to the miniload area by miniload planner `PM`.

The *control* layer contains processes that record all relevant information used for decision-making in each area within the system. This layer is further divided into *global control* and *local control* layers. The difference between the two layers is the scope of information that is accessible in each layer.

The *global controller* itself holds information over *all* subsystems beneath its supervision. That is, global miniload controller `GM` possesses (abstracted) information about all five miniload subsystems `MLS`. Similarly, global workstation controller `GW` has access to (abstracted) information of the three workstation subsystems `WS`.

The *local controller* contains information pertaining to the specific subsystem in its scope. Local miniload controller `LM`, for example, has access to information *only* from physical miniload `ML` under its supervision. As such, a local controller is not aware of the presence of other local controllers in the system. The same holds for local workstation controllers `LW`.

The *material flow* layer represents the physical material (product totes) movement. Processes that belong to this layer include the input and output (I/O) buffers `BI` and `BO`, I/O conveyor `TI` and `TO`, and the

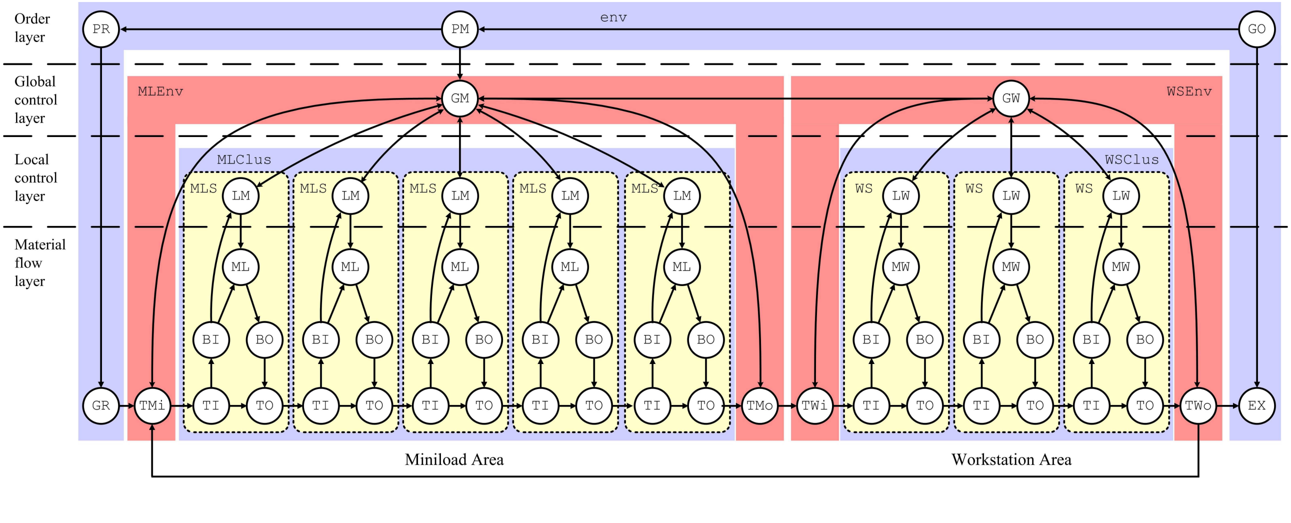


Figure 2: ML-WS System architecture.

physical miniload and workstation ML and MW, respectively. Note that the I/O buffers and I/O conveyor are present both in the miniload and workstation areas.

Processes TMi, TMo, TWi, TWo, TI and TO in the architecture altogether form the conveyor area. Note that the conveyor area is treated differently from the other two areas. Controllers for the conveyor area are integrated in the controller for the miniload and workstation areas. The conveyor requires information about the destination miniload or workstation for the totes. This information is assumed to be contained in the totes themselves. As such, there is no need to specify a separate controller for the conveyor.

3.2. Decentralized control structure

According to Sandell et al. (1978), the presupposition of centrality fails to hold in large systems due to either lack of centralized information or lack of centralized capability. An appealing alternative is to utilize a decentralized control instead. As outlined by Anderson and Bartholdi III (2000) from industrial case studies, advantages of utilizing a decentralized control include cheap processing units, local information in the processing units, cheap data collection, simple data processing algorithms, quick data processing, robustness to system failures, and real time operation of the processing units. In the system architecture we advocate, a decentralized control is implemented in a two-layer structure. Each controller is responsible for making decisions within its own scope based on the communication with the surrounding processes.

Different types of decisions are made in each layer of the decentralized control structure. For instance, global miniload controller GM makes a decision about which order will be completed next. To make such a decision, GM maintains a list of available orders, a list of available SKUs from all five LMs, and a list of available workstations from GW. GM also decides which of the five miniloads is assigned with jobs from the new order. This decision is made based on information about which miniload contains the oldest tote of the SKU required by the jobs, which is provided by all five LMs. The jobs are then assigned to that miniload. Finally, GM decides in which miniload a returning tote is stored.

For one particular miniload, local controller LM decides what storage or retrieval action is taken by the physical miniload. It also decides when the action is executed. The decision is based on real-time data about the number of totes that needs to be stored and retrieved (contained in BI and LM, respectively).

Similarly in the workstation area, global workstation controller GW decides whether a new order is allowed to enter the workstations. This decision is based on information about the total number of orders currently active in the workstations. Local workstation controller LW determines to which of the three buffer lanes an arriving tote will be put.

We argue that information can be utilized efficiently in such decentralized, autonomous control. Only relevant information for decision making is communicated between processes. Communication events happen exactly at the decision moments with as few communications as possible. Also, changes in one part of the system can be made locally with as little influence as possible to the surrounding parts.

3.3. Subsystems

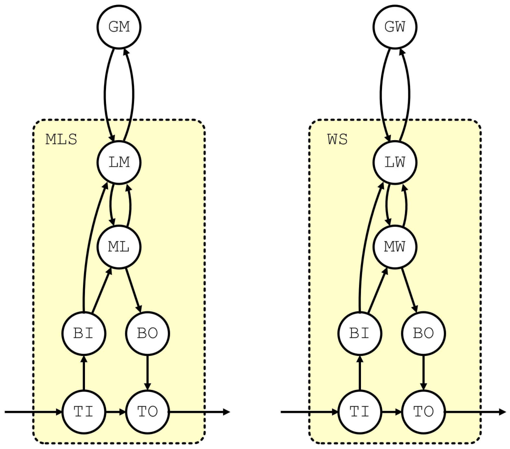
Increased modularity is also gained from creating miniload and workstation subsystems. MLS and WS consist of a number of processes that together represent respectively one miniload and one workstation (see Figure 3).

The modularity of the system architecture as reflected by the subsystems provides scalability. The number of subsystems such as MLS and WS can be easily increased or decreased, and the respective global controllers GM and GW easily adjusted.

3.4. Alternative system architecture

An implicit assumption has been made for the proposed system architecture in Figure 2, namely that there is a single route on the conveyor which stops at every miniload and workstation. Totes routed to the furthest miniload/workstation always travel through the other miniloads/workstations. This assumption is valid for the specific example of ML-WS system given in Figure 1.

There may also be other configurations where totes heading to one workstation may (partly) have a different route than totes heading to other worksta-



(a) ML subsystem. (b) WS subsystem.

Figure 3: Miniload and workstation subsystems.

tions. In the architecture of Figure 2, the routing and traveling of totes from the miniloads to the workstations and vice versa are contained in the conveyor area. Depending on the case at hand, the architecture of the material flow through the conveyor may need to be adjusted.

Figure 4 shows an example of an alternative system architecture for the conveyor area. This figure can be seen as an excerpt of Figure 2 for processes between TWi and TWo in the workstation area. We now account for totes having specific routings depending on the destination workstation. Similar modifications may arise for the miniload area if there are also different routes to and from various miniloads. Note that the control structure remains exactly the same.

3.5. Simulation study

The proposed system architecture can be directly implemented using a discrete-event simulation language. Owing to the modularity of the system architecture, we can exploit one of the main strength of simulation namely to evaluate what-if scenarios by combining numerous design aspects in combination with control policies (Roodbergen and Vis, 2009).

In Andriansyah et al. (2009a), we have implemented the proposed system architecture using a process-algebra based simulation language χ (Chi) 1.0 (Hofkamp and Rooda, 2008). The χ language is highly suitable to model parallel systems with concurrent processes. We also showed that the system architecture can be easily adjusted to incorporate different system configurations and control heuristics. Specifically, we altered the number of miniloads in the system and used two different control heuristics implemented locally at LM. The system throughput resulting from these changes are shown in Figures 5 and 6.

4. Performance monitoring method

We believe that a performance monitoring system should complement the proposed system architecture to allow for a sustainable warehouse operation. The main purpose of a performance monitoring system is to keep track of the system performance such that deviations from the expected performance can be quickly detected. We conjecture that the required performance monitoring system should act on the various servers that make up the entire warehouse network. That is, we propose to measure the EPT (Effective Process Time) distribution of the miniload and workstation in the queueing network. Here the EPT represents the aggregated process time, which includes besides the raw process time also the non-preemptive and preemptive outages such as setup, operator unavailability, breakdown, and other kinds of disturbances. In the EPT framework, we measure EPT distributions without characterizing the contributing outages. This means that the EPTs can be measured on a day to day basis, similar as the throughput and flow time of the system. A second advantage of the EPT concept is that the measured EPT mean and variance can be feed into a simplified simulation or analytical queueing network model. This enables simulation-based optimization of the

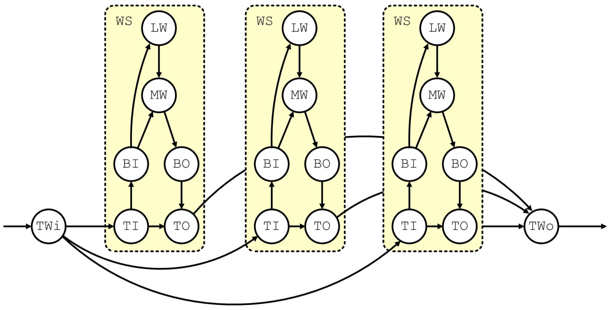


Figure 4: Alternative system architecture: conveyor area.

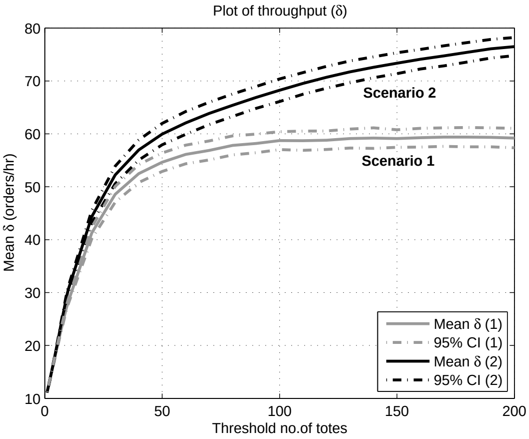


Figure 5: System throughput from two different control heuristics (Andriansyah et al., 2009a).

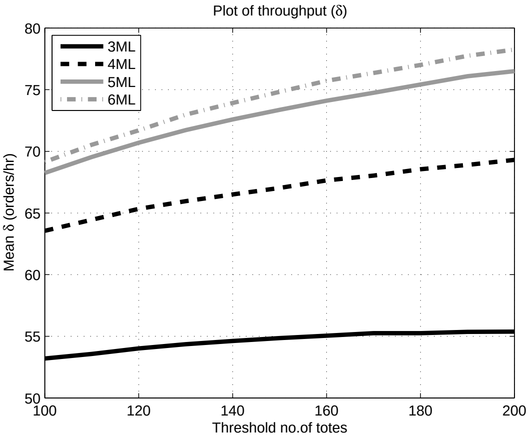


Figure 6: System throughput from four different system configurations (Andriansyah et al., 2009a).

warehouse system. For these reasons, we develop an EPT-based performance monitoring method.

To illustrate the EPT-based performance monitoring method, we focus on one order picking workstation. Figure 7 depicts an order picking workstation of the ML-WS system shown in Figure 1. Product totes retrieved from the miniloads arrive at the buffer conveyors. These totes are required to fulfill orders. A picker is present at the workstation to pick items from the product totes and put them in an order tote. The picker can only work at one order at a time. The order that is being processed is referred to as the current order. If all items for the current order have been picked, the order is said to be finished and the picker moves the finished order tote to a take-away conveyor that brings the order tote to a consolidation area. If a product tote is not yet empty after item picking, the tote will be returned to a miniload using a return conveyor.

An order picking workstation is characterized by several process time components (see Figure 8). At the core of the process is the time required for picking items, which is referred to as the raw pick time. Next to the raw pick time, pickers may require some setup time (change-over time) between processing of orders. Conveyor systems may break down, causing unavoidable delays. Picker availability is also an issue since it is likely that a picker is sometimes not present at the workstation. Quantifying each of these process time components is difficult. Therefore we aggregate them into a single EPT distribution. The idea is then to reconstruct the EPT distribution directly from tote arrival and departure times registered at the operating order picking workstation under consideration, with the obvious advantage that one does not need to quantify each component contributing to the process time. An EPT realization is calculated for each departing tote, which equals the total amount of time a tote *claims* capacity even if the tote is not yet in physical process. It represents the aggregation of all components that contribute to the processing time.

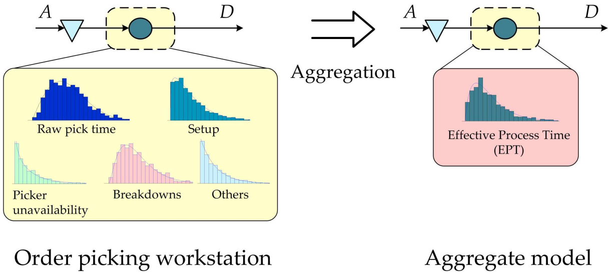


Figure 8: Aggregation method (Andriansyah et al., 2009b).

Figure 9 shows an example of arrivals and departures of six totes at an order picking workstation. Totes 1, 2, and 3 belong to order p, Tote 4 belongs to order q, and Totes 5 and 6 belong to order r. An arrival A_i occurs at the moment a product tote i enters the buffer conveyor of the order picking workstation. A departure D_i occurs when item picking has been finished and the respective product tote i is moved to the return conveyor or to the take-away conveyor.

In Andriansyah et al. (2009b) we propose to calculate the EPT realizations using the following sample path equation:

$$\mathrm{EPT}_i = D_i - \max\{A_i, D_{i-1}\} \tag{1}$$

here D_i denotes the time epoch of i^{th} departing tote. A_i denotes the arrival epoch of the corresponding i^{th} departing tote. The bottom part of Figure 9 illustrates how EPT realizations are obtained using Equation (1). When EPT realizations from all departing totes have been obtained, an EPT distribution can be constructed. The EPT distribution represents the actual pick rate at an order picking workstation.

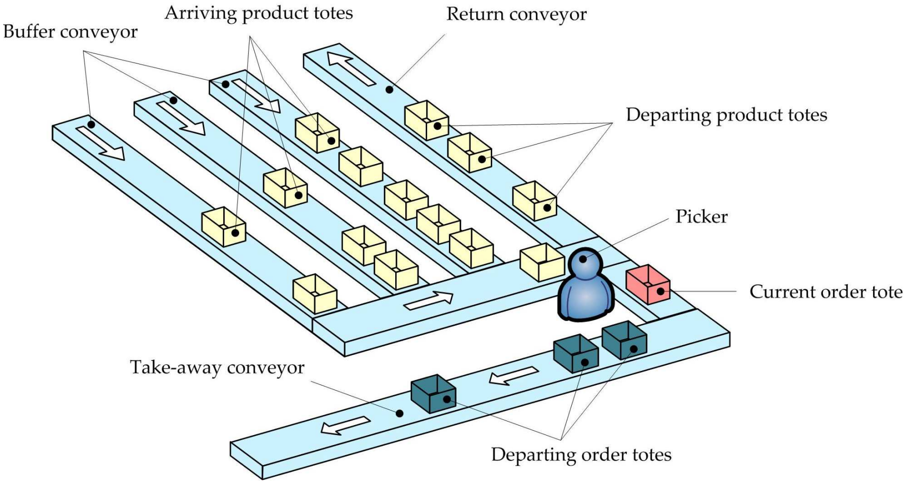


Figure 7: Order picking workstation (Andriansyah et al., 2009b).

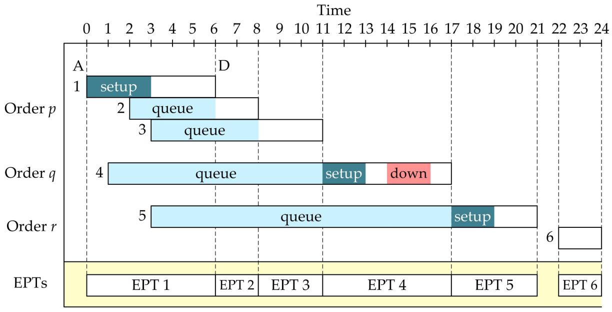


Figure 9: Gantt chart example (Andriansyah et al., 2009b).

Using the system thinking approach, Tan et al. (2009) argued that the actual pick rate reflects the staff's level of job satisfaction and productivity, which is also relevant to warehouse sustainability. The actual pick rate can be compared to the expected pick rate for performance analysis purposes. Furthermore, if this method is implemented in real-time at an order picking workstation, then any exceptionally large EPT realizations can be immediately detected. This will make it possible to identify the root cause of such large EPT realizations.

An application of the EPT method to the order picking workstation depicted in Figure 7 has been done in Andriansyah et al. (2009b). In our study, we used the method to accurately predict the mean and variability of tote and order flow times at different utilization levels. As such, the EPT realizations, the EPT distribution, and the flow time predictions resulting from the proposed method can be used identify possible improvements for order picking activities. In de Koning (2008) EPT-based modeling for the performance analysis of miniload systems is considered.

5. Conclusion and discussion

Efficient resource utilization is a prerequisite for sustainable warehouse order picking. This is mainly because warehouse order picking systems are labor and capital intensive. They require large investments on human and natural resources. In this regard, our study considers warehouse order picking sustainability from the perspective of efficient utilization of resources throughout the warehouse lifetime.

Warehouses are nowadays facing ever-changing requirements. The rise of internet orders, for example, has caused warehouses to serve not only bulk orders from e.g., supermarkets and retailers, but also small orders from individuals via the internet. Another example is that warehouses have to deal with increasingly tight delivery due dates to ensure customer satisfaction. In order to cope with these challenges, warehouses need flexibility so that any necessary adjustments to the warehouse system can be made quickly and easily. Moreover, warehouses should be able to continuously monitor their performance and to readjust configurations and operational settings accordingly. We believe that a flexible system architecture and a performance monitoring method would serve these purposes.

We advocate a system architecture that contains a decentralized hierarchical control structure with operational layers. This architecture sustains modularity such that different system configurations, design parameters, and control heuristics can be easily incorporated. Secondly we propose a performance monitoring system based on the effective process time. The method requires arrival and departure data of totes as input, for example at the workstation and at the miniload. For each of these servers in the warehouse network, EPT distributions are obtained. The EPT mean and variance can be used as performance indicators and also as input in a simulation model or analytical queueing model of the warehouse system. With this performance monitoring system we believe deviations from the expected performance can be detected early and subsequent corrective actions can be performed quickly. All of these ensure efficient utilization of resources, which eventually leads to improvement of warehouse sustainability.

Acknowledgement

This work has been carried out as part of the FALCON project under the responsibility of the Embedded Systems Institute with Vanderlande Industries as the industrial partner. This project is partially supported by the Netherlands Ministry of Economic Affairs under the Embedded Systems Institute (BSIK03021) program. We would like to thank Willem de Koning, Richard Jordan, Bruno van Wijngaarden, Roelof Hamberg, and Jacques Verriet for their contributions in developing the system architecture.

References

Anderson, C., Bartholdi III, J. J., 2000. Centralized versus decentralized control in manufacturing: lessons from social insects. In: Proceedings Complexity and Complex Systems in Industry. University of Warwick, pp. 92–105.

Andriansyah, R., de Koning, W. W. H., Jordan, R. M. E., Etman, L. F. P., Rooda, J. E., 2009a. A process algebra based simulation model of a miniload workstation order picking system. Submitted.

Andriansyah, R., Etman, L. F. P., Rooda, J. E., 2009b. Simulation model of a single-server order picking workstation using aggregate process times. SIMUL 2009 accepted.

Bartholdi III, J. J., Hackman, S. T., 2008. Warehouse and Distribution Science. The Supply Chain and Logistics Institute, School of Industrial and Systems Engineering, Georgia Institute of Technology, release 0.89 Edition.

de Koning, W. W. H., 2008. Modeling a storage and retrieval system: architecture and model aggregations. Master's thesis, Eindhoven University of Technology.

Hofkamp, A. T., Rooda, J. E., 2008. Chi 1.0 reference manual. Systems Engineering Report 2008-04, Eindhoven University of Technology, Eindhoven.
URL http://se.wtb.tue.nl/sewiki/chi

IISD, 1992. Business strategy for sustainable development: leadership and accountabililty or the for the 90s. The International Institute for Sustainable Development, Deloitte & Touche and the World Business Council for Sustainable Development.

Robert, K. H., Daly, H., Hawken, P., Holmberg, J., 1997. A compass for sustainable development. International Journal of Sustainable Development and World Ecology 4, 79–92.

Roodbergen, K. J., Vis, I. F. A., 2009. A survey of literature on automated storage and retrieval systems. European Journal of Operational Research 194 (2), 343–362.

Sandell, J. N., P, P. V., Althans, M., Safonov, M., 1978. Survey of decentralized control methods for large scale systems. IEEE Transactions on Automatic Control 23, 108–128.

Tan, K. S., Ahmed, M. D., Sundaram, D., 2009. Sustainable warehouse management. Proceedings of EOMAS 2009 458.