

# WEBASRS – A WEB-BASED TOOL FOR MODELING AND DESIGN OF ABSTRACT UNIT-LOAD PICKING SYSTEMS

---

Jeffrey S. Smith and Sabahattin Gokhan Ozden

Department of Industrial and Systems Engineering  
Auburn University  
Auburn, AL, 36849, USA

## Abstract

This paper describes a web-based tool that supports the modeling and design of abstract unit-load picking systems. The term “abstract” implies that the model is not specific to any equipment or vendors’ products, but, instead, focuses on the generic system components such as pallets, racks, slots, forklifts, cranes, etc. that comprise typical unit-load picking systems. The objectives of the tool are to support the design of an AS/RS-based or a manual forklift-based picking system based on a set of design parameters and to be able to convert from an AS/RS design to a flat warehouse design and vice versa. The research objective is to design the formal model (the data structure and operational description) that supports the conversion from one type to the other and supports the generation of static and dynamic analysis models and the recording of the analysis results. The web implementation uses a mix of XML, HTML, JavaScript and PHP and implements two existing analysis methodologies from the literature.

## 1 Introduction

Initial design of a distribution center’s picking area involves collecting information about the storage requirements, picking throughput requirements, basic equipment constraints, and overall facility size constraints, analyzing the information/constraints, and identifying “workable” configurations that meet the constraints. Final design involves selecting one or more of the workable configurations and adding the equipment-specific details required to price and construct the facility. Gu *et al.* [1] surveyed available methodologies and tools for improving warehouse design practices and partitioned the research into five major decisions in warehouse design. Their partitioning includes: warehouse structure; sizing and dimensioning; department layout; equipment selection; and operational strategy selection. The majority of existing research focuses on analysis methods and related tools for “solving” these basic design problems and for system performance evaluation. Our focus is on a related, but different area.

Our research objective is to develop a general data model a set of related web-based tools that provides a basis for design, analysis and model *generation* in warehouse layout

design for unit-load picking areas. That is, we are developing a formal model of the system requirements, parameters, and design that is independent of the specific design methods and/or operational analysis methods used to assess the design. An instance of this formal model would provide all of the information necessary to use any design/analysis methods. This paper describes an initial version of the data model along with a set of web-based tools that demonstrate the application of the data model.

## 2 Background

A *data model* describes the data associated with a system along with the access methods for this data. There are two significant benefits of using data model. Firstly, the data model helps people with different backgrounds to communicate with each other using a common frame of reference. Secondly, the data model provides a structured representation that eliminates ambiguity [2]. In addition, since the data model provides an unambiguous description of the system, software applications can easily interact with and share the data with other applications.

We use XML [3] (extensible markup language) to store our design parameters and XSD [4] (XML schema documents) to design the data model. XML is ubiquitous in today's web and provides a formal, hierarchical structure for storing and transmitting structured data. XML is extremely flexible and is becoming increasingly used as a storage format for general applications (i.e., Microsoft Office™ started using XML file formats with its 2007 version). XSD is used to express a set of rules to which an XML document must conform in order to be considered valid [4]. We are defining set of rules in our XML Schema, so in the future if another developer wants to develop another analytical model in our tool, that analytical model must conform to our data model standards which are defined by XML schema. However, the XML schema might be also changed in the future if additional requirements are imposed. We selected XSD as our XML schema languages, because it has more features than other XML schema languages. For example, we can define specific data types in XSD (e.g., integer, string). In addition, we can also define constraints on these data types such as upper and lower bound for an integer variable.

Heragu et al. [5] present a similar online conceptualization tool to compare autonomous vehicle storage and retrieval systems (AVS/RS) and AS/RS. Although their tool focuses on a relatively narrow class of picking systems, their work identifies the set of required input and output parameters and demonstrates that simple web-based tool can help to support decision making process in warehouse design.

In order to demonstrate the use of our data models and the associated web tools, we have implemented two examples. First, we translated the Microsoft Excel™-based ExASRS v.2.0 [6] (a tool which determines a basic configuration for a unit load AS/RS based on a set of user input). Second, we developed an implemented an algorithm for manual-pick order-picking systems based on Pohl et al. [10] to compare these two systems (unit load AS/RS and manual-pick order-picking system).

ExASRS iteratively determines whether the selected number of aisles will be sufficient to meet the throughput and storage requirements defined by user. Analytical expressions in [7], [8] and [9] are used in cycle time calculations in ExASRS. We directly used the analytical expressions that are already implemented in ExASRS and we translated them for use in our web-based tool. Pohl et al. [10] proposed analytical models for both single and dual cycle command operations in common warehouse designs. They developed analytical dual-command travel distance models for optimal paths for three different types of layouts. They concluded that what they refer to as “Layout C” is the best choice or near-best choice for a wide range of different parameters. Our tool uses analytical travel distance models for Layout C discussed in this paper (see Figure 1).

### **3 Model Assumptions**

Our data model currently consists of a two-level hierarchy – a “base model” that includes parameters common to all unit-load picking systems, an “AS/RS model” that augments the base model with the AS/RS-specific parameters, and a “Manual-pick model” that augments the base model with the parameters specific to manual picking systems. As we continue the development of the data model and extend it to accommodate additional system types, additional levels will be added to the hierarchy and additional models will be added at the existing level(s). In the following paragraphs, we give the assumptions for the three existing models.

#### **Base Model Assumptions**

The following assumptions are applied across all of the models in the hierarchy.

1. Picking unit loads;
2. Randomized storage is used; and
3. The numbers of storage and retrieval requests per hour are constant and known.

#### **AS/RS Model Assumptions**

The AS/RS model is based on ExASRS [6] and all assumptions from that work (notably [7] and [9]) are used in AS/RS model.

1. The rack is considered to be a continuous rectangular pick face with length  $L$  and height  $H$  with the I/O station located at the lower left-hand corner.
2. The S/R machine travels simultaneously in the horizontal and vertical directions.

#### **Manual-Pick Model Assumptions**

In our manual pick configurations, we assume that pallets are stacked on top of one another (possibly in racks) in parallel aisles. Workers use forklifts to pick unit-loads from pick and deposit locations and store these loads in the pallet racks and they retrieve unit-loads from the pallet racks and return them to the pick and deposit location. Since these types of warehouses are generally large, travel distances traveled by forklifts are

also large. Assumptions from Pohl et al. [10] are used in manual-pick order-picking system model.

1. No acceleration/deceleration is used;
2. A set of discrete picking aisles with continuous picking activity in each aisle and picking uniformly distributed within and among all aisles;
3. The storage and retrieval requests are assumed to be independent and processed on a first-come-first-serve basis;
4. Single and dual command order-picking;
5. All vehicles can pickup from all pick locations with equal probability;
6. One central pick and deposit location;
7. Vertical travel time to upper pallet positions is negligible; and
8. Lateral travel time within an aisle is negligible.

Figure 1 illustrates the manual picking configuration. In our web implementation, the parameters defined below are incorporated into the XML data model.

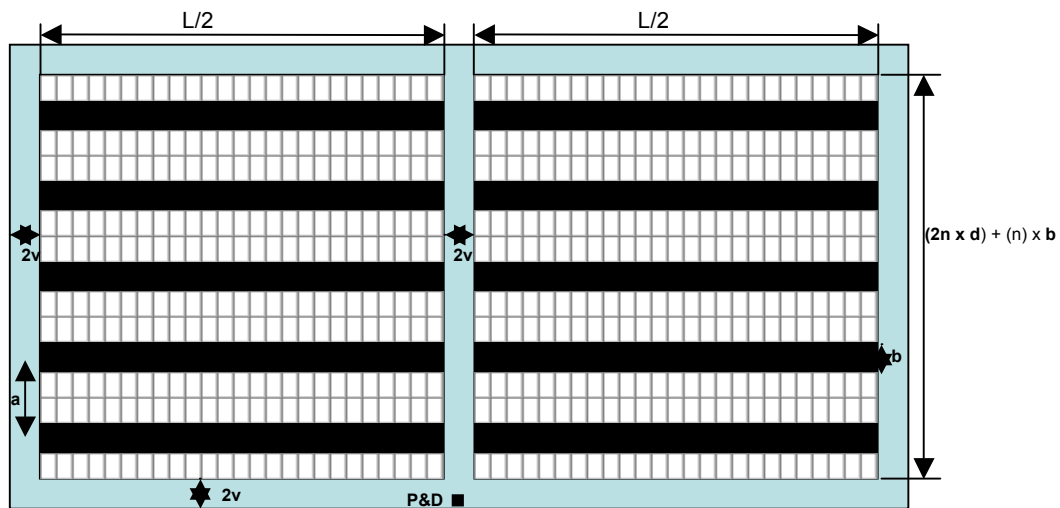


Figure 1 – Manual-Picking Order Picking System Layout

$m$ : number of picking locations in length

$l$ : length of a single location

$d$ : depth of a single location

$L$ : length of rack per aisle ( $ml$ )

$n$ : number of picking aisles (colored in black) in one part of the storage area

2*v*: Cross aisle width

*b*: Picking aisle width

*a*: Distance between picking aisles ( $b + 2d$ )

## 4 Design Parameters

The design parameters comprise the data model. As described above, the data model currently consists of a two-level hierarchy with the Base Model as the root and the AS/RS and Manual Picking models as children. The specific parameters for each model are given in the following sections.

### 4.1 Base Model Parameters

Parameter	Description
num_st_req	Number of storage requests per hour
num_ret_req	Number of retrieval requests per hour
max_st_vol	Maximum storage volume (unit loads)
st_cell_height	Storage location (cell) height including rack structure and clearances (feet)
st_cell_width	Storage location (cell) width including rack structure and clearances (feet)
st_cell_depth	Storage location (cell) depth including rack structure and clearances (feet)
aisle_width	Picking aisle width (feet)
dual_command	Percentage of dual command (DC) operations (%)
pickup_time	Time required to pickup a unit load in the rack or at the I/O points
deposit_time	Time required to deposit a unit load in the rack or at the I/O points
num_aisles	Number of aisles in the system
rack_length	Length of rack (feet)
rack_height	Height of rack (feet)
num_cols_per_rack	Number of columns per rack
num_rows_per_rack	Number of tiers (rows) per rack
st_area	Total storage area feet <sup>2</sup>
st_op_per_aisle	Number of storage openings per aisle
st_op_req_per_aisle	Number of storage openings required per aisle
avg_time_per_req	Average time per request (second)
time_av_per_req	Time available per request (second)
avg_thru	Average throughput (requests/hour)
req_thru	Required throughput (requests/hour)
TSC	Single Cycle Time (second)
TDC	Dual Cycle Time (second)

### 4.2 AS/RS Parameters

Parameter	Description
num_shuttle	Positions that are available for the AS/RS machine during one request

hor_vel	S/R Machine Horizontal Velocity (ft/min)
ver_vel	S/R Machine Vertical Velocity (ft/min)
hor_acc_dec	S/R Machine Horizontal acceleration - deceleration (ft/sec <sup>2</sup> )
ver_acc_dec	S/R Machine Vertical acceleration - deceleration (ft/sec <sup>2</sup> )
max_util	Maximum allowable utilization of the S/R machine for design purposes
rack_shape	Desired rack shape, either square-in-time or fixed height
max_height	If a fixed height rack is used, then this specifies the rack height in feet
rack_type	The type of racks used, either single-deep or double-deep
honey_combing	If double-deep rack is used, this represents the percentage of the storage locations that are not usable due to honeycombing losses
avg_time_per_req_fcfs	Average time per request on first come first served basis
avg_time_per_req_mod_fcfs	Average time per request on modified first come first served basis
avg_time_per_req_nn	Average time per request on nearest neighbor basis
avg_time_per_req_mod_nn	Average time per request on modified nearest neighbor
Th	Horizontal time (second)
Tv	Vertical time (second)
T	Maximum of horizontal time and vertical time
b	Min(Th/T, Tv/T)
TQC	Quadruple cycle time (second)
TSTC	Sextuple cycle time (second)

### 4.3 Manual-Pick Order-Picking System Parameters

Parameter	Description
num_vehicle	Number of vehicles in the system
cross_width	Cross aisle width (feet)
veh_hor_vel	Forklift vehicle horizontal velocity (ft/min)
num_cols	Number of storage columns on each side of rack

## 5 XML Schema

As we described earlier in Section 1, an XML Schema is used to formally represent our data model. We have three sub-models, one of them is including base model parameters, one of them is including ASRS parameters and the other one is including manual-pick order-picking system parameters. These sub-models are defined as “elements” in the schema and each element is defined as a set of “parameters.” Each parameter object has name, label, value, type, description, label division id and lastly value division id. If a parameter has type 0, then it is a display only or an output parameter, otherwise it is an input parameter and it can be a textbox which only accepts integer values or it can be a combo box which has certain predefined values (such as single deep or double deep). Value stores the value of that parameter. Label division id and value division id define the places of that parameter’s label and input form in the html document. The XML schema is given below.

```

<?xml version="1.0" encoding="utf-8"?>
<xs:schema attributeFormDefault="unqualified" elementFormDefault="qualified"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="model">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="general-parameters">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="param" maxOccurs="unbounded"
type="parameters"></xs:element>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
        <xs:element name="asrs-parameters">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="param" maxOccurs="unbounded"
type="parameters"></xs:element>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
        <xs:element name="manual-parameters">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="param" maxOccurs="unbounded"
type="parameters"></xs:element>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:complexType name="parameters">
    <xs:sequence>
      <xs:element name="name" type="xs:string" />
      <xs:element name="label" type="xs:string" />
      <xs:element name="value" type="xs:unsignedByte" />
      <xs:element name="type" type="xs:unsignedByte" />
      <xs:element name="description" type="xs:string" />
      <xs:element name="labeldivid" type="xs:string" />
      <xs:element name="valuedivid" type="xs:string" />
    </xs:sequence>
  </xs:complexType>
</xs:schema>

```

## 6 Design Algorithms

### 6.1 AS/RS Design Algorithm

Our AS/RS implementation is based on ExASRS, a tool which determines the design of a unit load AS/RS [6]. The algorithm finds the minimum number of aisles and calculates the basic dimensions of the rack system. The algorithm begins with one aisle and if it does not meet the throughput requirement (number of storage/retrieval requests per hour),

it increases the number of aisles iteratively until it meets the desired throughput. Details of the algorithm and cycle time calculations are described in [6]. Our implementation is a direct translation from the VBA code to JavaScript.

## 6.2 Manual-Pick Order-Picking System Design Algorithm

Our algorithm determines the design of manual-pick order-picking system. The algorithm finds the minimum number of aisles and the basic rack dimensions. We used and slightly modified the travel distance equations that are presented in [10] to compute cycle times. Here are the broad lines of the algorithm as follows:

**Step 1:** Begin with 1 aisle

**Step 2:** Determine the number of columns per aisle with the given number of rows per rack and maximum (total) storage capacity:

$$\text{numberofcolumnsperrack} = \frac{\text{totalstoragecapacity}}{2 * (\text{numberofaisles}) * (\text{numberofrowsperrack})}$$

**Step 3:** Determine the average time per request for given warehouse design based on step 2.

// Single Command Travel Distance

$$dSC = \frac{L}{2} + 2v + an + 2v$$

// Dual Command Travel Distance

$$dDC = L \left( \frac{11n-1}{12n} \right) + v \left( \frac{2n-1}{n} \right) + a \left( \frac{4n^2-1}{3n} \right) + 2v$$

// Expected Single Command Cycle Time

$$ESC = \frac{dSC}{\text{vehiclehorizontalvelocity}}$$

// Expected Dual Command Cycle Time

$$EDC = \frac{dDC}{\text{vehiclehorizontalvelocity}}$$

// Time Scaled Single Command Cycle Time

pt: pickup time

dt: deposit time

$$tSC = ESC + pt + dt$$

// Time Scaled Dual Command Cycle Time



$$tDC = EDC + 2pt + 2dt$$

ATPR: Average Time Per Request

dual: Percentage of dual commands

$$ATPR = \frac{(tSC)(1 - dual) + \left(\frac{tDC}{2}\right)dual}{numberofvehicles}$$

**Step 4:** If the average time per request is greater than the cycle time requirement defined as a throughput by the user then increase number of aisles by 1 and go to step 2.

Otherwise stop and report the results.

## 7 Web Implementation

The web-based tool uses HTML, JavaScript, PHP and XML. HTML and JavaScript are used primarily for the user interface components. JavaScript is used to implement the algorithms and calculate the output parameters. PHP is used for XML file handling. The most important part is the XML; we used XML to define our warehouse models. Since XML is a generic way of storing and transporting data, our warehouse models can be transferred to any other tool if our XML data model is handled correctly. The address of the web application is <http://warehouselayout.org>.

Figures 2 and 3 give screen shots of the AS/RS section web application. The manual pick section of the application is very similar to the AS/RS section, with only the parameter definitions being different. When the user starts the application, s/he has an option of entering the input parameters or reading an existing XML file. Once the parameters are input, the user can use the Calculate button to run the analysis and display the resulting output parameters. The user can also run the sensitivity analysis and save the XML file using the corresponding buttons. Finally, users can upload an existing XML file using the Browse and Upload buttons. The Save/Upload capabilities allow the user to save configurations on their local computer and upload these files at a later time for further analysis.

Input Parameters		Results	
Model Name:	<input type="text" value="Model 1"/>	Number of Aisles:	3.00
Save Time:	<input type="text" value="2010-03-21"/>	Length of Rack (ft):	144.00
Number of Storage/Retrieval Shuttles:	<input type="text" value="3"/>	Height of Rack (ft):	96.00
Number of Storage requests per hr:	<input type="text" value="100"/>	Number of Columns per Rack:	36.00
Number of Retrieval requests per hr:	<input type="text" value="100"/>	Number of Rows per Rack:	24.00
Maximum Storage Volume (units):	<input type="text" value="5000"/>	Storage Area (ft <sup>2</sup> ):	5184.00
Storage Cell Height with clearances (ft):	<input type="text" value="4"/>	Storage openings per Aisle:	1728.00
Storage Cell Width with clearances (ft):	<input type="text" value="4"/>	Storage openings required per Aisle:	1666.67
S/R Machine Horizontal Velocity (ft/min):	<input type="text" value="150"/>	Avg Time per request (sec):	51.43
S/R Machine Vertical Velocity (ft/min):	<input type="text" value="100"/>	Time Available per request (sec):	52.92
S/R Machine Horizontal accel/decel (ft/sec <sup>2</sup> ):	<input type="text" value="2"/>	Avg Throughput (requests/hr):	205.78
S/R Machine Vertical accel/decel (ft/sec <sup>2</sup> ):	<input type="text" value="3"/>	Required Throughput (requests/hr):	200.00
Pickup Time (sec):	<input type="text" value="2"/>	Avg Time per request (sec) FCFS:	51.43
Deposit Time (sec):	<input type="text" value="5"/>	Avg Time per request (sec) mod. FCFS:	51.43
% Dual Command Operations:	<input type="text" value="0"/>	Avg Time per request (sec) Nearest Neighbor:	41.30
Maximum Allowable Utilization (%):	<input type="text" value="98"/>	Avg Time per request (sec) mod. Nearest Neighbor:	41.30
Rack Shape:	<input type="text" value="Square In Times"/>	Horizontal Time (sec):	57.60
Height (ft):	<input type="text" value="60"/>	Vertical Time (sec):	57.60
Rack Type:	<input type="text" value="Single Deep"/>	Max of horizontal and vertical time:	57.60
Honey Combing Loss (%):	<input type="text" value="15"/>	b:	1.00
Aisle Width (ft):	<input type="text" value="4"/>	T(SC) (sec):	84.96
Storage Cell Depth with clearances (ft):	<input type="text" value="4"/>	T(DC) (sec):	119.63
		T(QC) (sec):	188.97
		T(STC) (sec):	258.30

File:

Figure 2 - Web User Interface

Minimum Storage Volume	<input type="text"/>
Maximum Storage Volume	<input type="text"/>
Minimum Number of Aisles	<input type="text"/>
Maximum Number of Aisles	<input type="text"/>
Minimum Throughput (#req/hr)	<input type="text"/>
Maximum Throughput (#req/hr)	<input type="text"/>
Minimum %Dual Command Ops	<input type="text"/>
Maximum % Dual Command Ops	<input type="text"/>
Cost of a single SR machine	<input type="text"/>
Cost of a storage cell location	<input type="text"/>

Figure 3 - Sensitivity Analysis

## 8 Conclusions and Future Work

A web-based tool that supports the modeling and design of abstract unit-load picking systems has been described in this paper. The critical component of the tool is a formal data model that describes the system parameters in an unambiguous language (XML) and independently of the analysis tools used during the picking system design process. Two existing analysis methodologies – one for AS/RS configurations and the other for manual picking systems – were used to demonstrate the data model and the tool.

Moving forward, we are concentrating in three areas: Expanding and refining our data model to incorporate additional parameters required for other existing analysis models; Enhancing our web system to improve the “user experience”; and adding behavioral components to the data model. With the behavioral and structural components, we will be able to generate additional types of analysis models (e.g., simulation models). Our current work in this area uses SysML to describe the behavioral aspects of the system.

## 9 References

- [1] Gu, J., M. Goetschalckx, L. F. McGinnis, “Research on warehouse design and performance evaluation: A comprehensive review”, *European Journal of Operational Research* 203 (3), 539-549, 2010
- [2] Hoberman, S., “Data Modeling Made Simple 2nd Edition”, Technics Publications, LLC, New Jersey, 2009
- [3] Extensible Markup Language (XML), <http://www.w3.org/XML/>
- [4] W3C XML Schema, <http://www.w3.org/XML/Schema.html#dev>
- [5] Heragu, S.S., X. Cai, A. Krishnamurthy and C.J. Malmborg, “Web Interface for the Conceptualization of AVS/RS and AS/RS”, *Progress in Material Handling Research: 2009*, K. Ellis, R. Meller, M. Ogle, B. Peters, G. Taylor, and J. Usher, Eds, Material Handling Institute, 2008, pp. 251-292.
- [6] DePuy, G. W., “Multiple Shuttle ASRS with Acceleration/Deceleration Considerations”, 2007, <http://www.mhia.org/learning/resources/freedownloads/6273/exasrs-v2-0>
- [7] Bozer, Y.A., J.A. White, “Travel-time models for automated storage/retrieval systems” *IIE Transactions* 16 (4), 329–338, 1984
- [8] Chang, D.T., U.P. Wen, J.T. Lin, “The impact of acceleration/deceleration on travel-time models for automated storage/retrieval systems”, *IIE Transactions* 27 (1), 108–111, 1995
- [9] Meller, R.D., A. Mungwattana, “Multi-shuttle automated storage/retrieval Systems” *IIE Transactions* 29 (10), 925–938, 1997
- [10] Pohl, L. M., R. D. Meller, K. R. Gue, “An analysis of dual-command operations in common warehouse designs”, *Transportation Research Part E: Logistics and Transportation Review* 45 (3), 367-379, 2009