An Object Oriented and Axiomatic Theory of Warehouse Design

Leon F. McGinnis
Georgia Institute of Technology, leon.mcginnis@isye.gatech.edu

AN OBJECT ORIENTED AND AXIOMATIC THEORY OF WAREHOUSE DESIGN

Leon F. McGinnis
Georgia Institute of Technology

Abstract

Despite the conceptual simplicity of warehousing, the development of integrated computational tools for warehouse design has remained an elusive goal. In recent years, there has been some progress toward this goal, with a growing body of research addressing topics as diverse as the design process itself, decision support for specific design decisions, warehouse representation, integrating warehouse representation and analysis, and conceptual approaches for developing integrated warehouse design tool chains. Until now, however, there has not been a suitable warehouse design theory that would provide an integrating framework for all these disparate efforts.

This paper presents an object-oriented and axiomatic warehouse design theory. Key assumptions about the warehouse to be designed are stated as axioms, with appropriate formalisms. Using the axioms and the associated notation, a formal specification of warehouse requirements can be stated, a formal description for a warehouse design can be given, and methods can be developed for testing the warehouse design against the requirements. Moreover, the axioms provide the foundation for identifying essential warehouse design decisions, and formally stating both the criteria for evaluating those decisions and the constraints limiting those decisions.

The paper provides a conceptual and rigorous bridge between the process-oriented research on warehouse design process or workflow, and the mathematically oriented approach to warehouse design reflected in the vast literature on mathematical models and algorithms for specific warehouse design and/or operating decisions

1. Introduction

Warehouses are conceptually simple industrial facilities—goods arrive and are stored, customer orders arrive, goods are retrieved from storage and assembled to fill customer orders, and the assembled orders are packed and shipped. The most common material conversion is from one unit of handling to another, and the most complex material conversion is building mixed cases or mixed pallets. The material handling operations are picking, putting, moving, and measuring. Thus, one unfamiliar with the challenge might be forgiven for assuming that creating a warehouse design would be an easy task.

In fact, warehouse design can be quite difficult, in part because there are so few constraints on how any particular warehouse should be configured.

The literature on this problem is actually rather sparse. Most treatments of warehouse design, or facility design in general, include a step which is "generate alternatives," but don't really say ***how*** to generate these alternatives (there are exceptions, of course, which will be discussed later). The approach to warehouse design proposed in this paper clearly distinguishes two phases of design: *functional design*, which specifies *what* must be done in the warehouse—its *functions*—to fulfill its mission, and *embodiment design,* which specifies *how* the functions will be implemented—the *systems*—and what resources will be required. Creating a warehouse design requires identifying the necessary capabilities of the warehouse, establishing the essential functions the warehouse must perform, and translating these functions into an embodiment, or a configuration of a specific set of resources with specific behaviors. A single system in the warehouse may embody many functions, as will be illustrated later. The embodiment specification should be sufficiently detailed for the design to be evaluated with regard to performance, including cost, so that multiple designs can be compared and ranked. This paper proposes an object-oriented and axiomatic approach to the problem of generating alternative warehouse designs, for conventional distribution warehouses; cross-docks and factory warehouses are not explicitly addressed.

An *axiom*, according to Merriam-Webster, is "a statement accepted as true as the basis for argument or inference," (http://www.merriam-webster.com/dictionary/axiom). In following sections, a series of warehousing axioms will be proposed, and inferences drawn from those axioms. Together, the axioms and the associated inferences will be the basis for proposing:

- The semantics of warehouse requirements
- The semantics of warehouse functional design
- The semantics of warehouse embodiment design
- A general process for creating a warehouse design

No doubt, there will be some deficiencies in what is proposed. Nevertheless, it will be the first integrated and formal model of warehouse design, and moreover, it will provide a framework within which to view and evaluate the vast literature on particular warehouse operational and design decisions.

Much of the presentation in this paper will use the formalisms of OMG SysML™ (www.omgsysml.org), a formal object-oriented language for systems modeling. Two specific types of diagrams will be used extensively: structure diagrams display objects and their relationships; activity diagrams display the flow of control and objects through a network of activities. These diagrams are largely self-explanatory, except perhaps for connectors in the structure diagrams. A connector between two blocks with a white triangle at one end represents a "generalization" relationship, i.e., the object at the triangle end is the general case. A connector with a black diamond at one end represents a "part of" relationship; the object at the other end is a part of the object at the diamond end. A connector with a white diamond at one end represents a reference association.

The remainder of the paper begins with a brief review of the relevant literature. This is not a comprehensive review of warehousing or warehouse design, as there are other sources for that. Rather, section 2 focuses on what has been said in the past about the processes and tools for creating alternative warehouse designs. Section 3 discusses warehouse functional requirements. Section 4 addresses functional requirements analysis, or what must be known to support the transition from functional thinking to embodiment thinking. Section 5 proposes an approach for warehouse functional design, based on the use of activity diagrams from SysML. Section 6 explains how to use the functional design to transition to an embodiment design, and how analysis tools can be brought to bear in the process. Section 7 provides a brief example, and section 8 summarizes the contribution and identifies directions for future research and development.

2. Related Literature

In recent years, there has been some progress toward the goal of developing computer aided warehouse design methods and tools. For example, there is now a large body of research on analytic models to support specific design decisions (see, e.g., (Gu, et al, 2010) for a thorough literature review). There have been a number of investigations of the warehouse design process itself, including empirical studies (Apple, *et al*, 2011), (Bodner, *et al*, 2002), and (Goetschacklx *et al*, 2001), as well as attempts to formalize the process (McGinnis *et al*, 2006), (McGinnis, 2008), and (Sharp *et al*, 2008). The problem of modeling the warehouse itself (rather than modeling its analysis) has been explored, e.g., in (McGinnis, 2004), (Goetschalckx *et al*, 2008), and (Friemann *et al*, 2010).

A recent and fairly comprehensive review of the warehouse design literature has been published by (Baker and Cannesa, 2009). Looking at all the extant publications, they tabulated the suggested steps in the design process. They also contacted a number of firms doing warehouse design, and compiled the steps used by those firms. Finally, combining the published literature and their survey results, they proposed a synthesis of all those suggestions. While this is a valuable resource and provides some good background, it falls short of suggesting a theory of warehouse design. It identifies a number of specific tools that are or could be used in the warehouse design process, but does not provide much insight into how those tools could be integrated.

Integrating formal models and analysis has been addressed, e.g., in (Blunck *et al*, 2011) and (McGinnis and Meller, 2009). Conceptual approaches to creating integrated tool chains are described in (McGinnis, 2010). Until now, however, there has not been a suitable warehouse design theory that would provide an integrating framework for all these disparate efforts.

The distinction between the *what* of design and the *how* is also made by Suh (Suh, 1990) in his seminal work on axiomatic design. Suh associates *functional requirements* (denoted FR) with the “what” question, and *design parameters* (denoted DP) with the “how” question. In axiomatic design, the DP correspond to the physical instantiation, and lead naturally to *process variables* (denoted PV). A key goal of axiomatic design is to decompose a design problem hierarchically, and to achieve independence across the decomposition. At each level of the decomposition, the FR, DP, and PV domains are explicitly defined. The approach to warehouse design presented here resembles Suh's axiomatic design in the use of the concepts of functional requirements, functional design, and embodiment design, but departs from it in fully elaborating the functional design before attempting the embodiment design.

A very thorough treatment of engineering design theories can be found in (Tate, 1999).

3. Warehouse Functional Requirements

A warehouse is a facility that interrupts the flow of goods from the suppliers of products to the customers for those products, as illustrated in Figure 1. Warehouses serve several key commercial functions, such as protecting producers from variability in demand, and providing impedance matching between high rate production and lower rate consumption (e.g., soft drinks), or between low rate production and high rate demand (e.g., iPhones).

Figure 1 Warehouse Context

Axiom 1: A warehouse has an input flow (InFlow) of shipments from suppliers, and an output flow (OutFlow) of shipments to customers.

Both InFlow and OutFlow are examples of the general concept of "flow". Flows certainly can be observed in a warehouse, e.g., by observing what arrives to the receiving dock, what departs from the shipping dock, or any movement of goods within the warehouse. All flows share a similar underlying structure, and all flows can be considered over the long term, or in smaller time buckets, e.g., daily to understand real-time loading impacts, or quarterly, to understand seasonal variation.

Axiom 2: A warehouse *flow*, considered over any time period, consist of *orders*, which have *lines* where each line identifies a *stock keeping unit* (or SKU), a *unit of handling* (or UoH), and a *quantity*; an order also will have an associated UoH.

The relationships identified in Axiom 2 are summarized in Figure 2. Note that flows that are *internal* to the warehouse may be associated with "orders" that are generated by the warehouse management system or WMS, but flows that arrive to the receiving dock will correspond to replenishment orders from suppliers, while flows that depart from the shipping dock will correspond to customer orders.

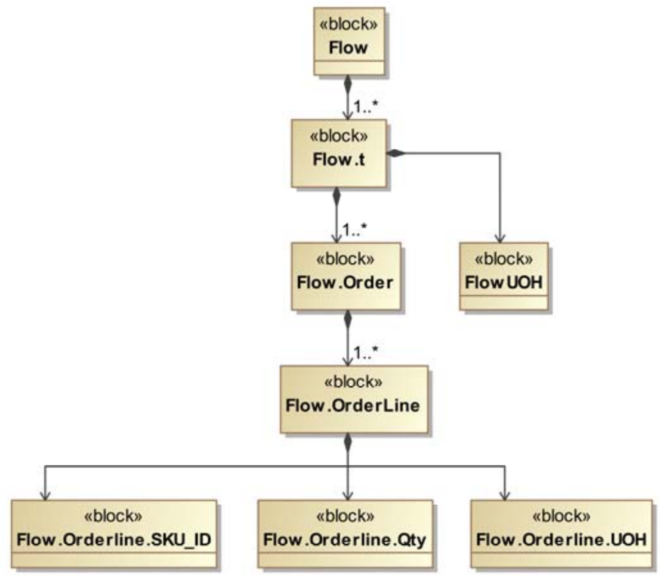

Figure 2 Warehouse Flow

Unit of Handling, or UoH, is an important concept for warehouse design, and Figure 3 illustrates the various units of handling that are possible. Note that if Figure 3 does not completely enumerate all the important units of handling, it is easily updated.

This paper addresses only warehouses for which the inflows arrive and the outflows depart via some sort of vehicle (van, trailer, railroad car, airplane, etc.). For such warehouses, the functional requirements which must be satisfed are to "consume" the InFlow and "produce" the OutFlow, as summarized in Axiom 3:

Axiom 3: A warehouse must have the capability and capacity to:

- Unload and store supplier orders from arriving vehicles with a specified service level

- Assemble and load customer orders onto departing vehicles with a specified service level

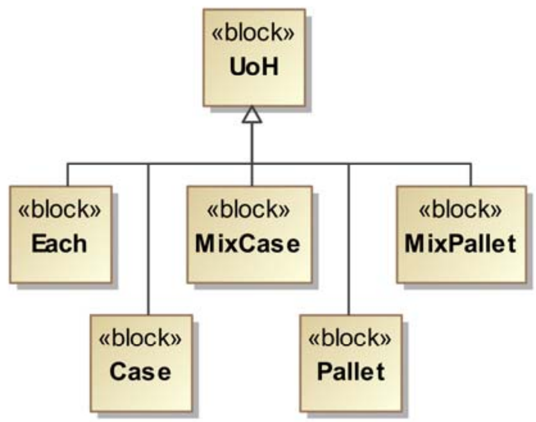

Figure 3 Units of Handling

Capability is related to the physical requirements for handling and storing goods and orders. If goods arrive on pallets, the warehouse must be able to lift and move pallets. If frozen goods arrive, the warehouse must be able to maintain their frozen state. Capacity is related to the rate at which loading and unloading operations must be performed, and the amount of storage space that must be available.

Clearly, a warehouse may have other functional requirements, e.g., conversion of the InFlow UoH to the OutFlow UoH, or moving SKUs inside the warehouse. However, those requirements are derivative, i.e., they result from the fundamental requirements for consuming the InFlow and producing the OutFlow.

For loading and unloading, the most straightforward way to specify the service level is to give a probability distribution for the time to complete in excess of some target, and the simplest version of this is a single target and a single probability, for example, "every truck arriving at the receiving dock will be unloaded within 90 minutes, with probability 0.95". For the inventory requirement, the most straightforward way to specify the service level is to give a probability that an arriving unit of handling will not find an available storage slot, e.g., "the probability that an arriving unit of handling cannot be stored will be less than 0.01". Of course, other formulations of the service level specifications are possible.

4. Functional Requirements Analysis

A key step in warehouse design is to understand, for a particular design problem, how the requirements for consuming the InFlow and producing the OutFlow, identified in Axiom 3, will drive design decision making, i.e., what capabilities and capacities must be provided. Thus, functional requirements analysis is fundamentally about characterizing the InFlow and OutFlow in useful terms. In practice and in the research literature, functional requirements analysis has not been given a formal treatment. What has been written and presented regarding warehouse requirements analysis has been largely *ad hoc*, generally oriented toward embodiment design, and using examples that highlight the idiosyncrasies of particular warehouse design scenarios. Not surprisingly, there are no general purpose tools available to support warehouse functional requirements analysis.

The intent of this discussion of functional requirements analysis is to provide some formal structure within which the issues can be identified and understood, along with a generic approach to functional requirements analysis. One goal is to identify both the needs for more formal analysis methods and tools, and the opportunities for integrating domain knowledge. It should be noted that the presentation here is explicitly geared toward supporting functional design as it will be described in the following section.

Propositions 1 and 2 help in identifying the kinds of *terms* in which InFlow and OutFlow will be characterized, but don't specify how those terms will be identified, evaluated, or used for a particular warehouse design.

Proposition 1: Functional requirements analysis identifies all attributes of the InFlow and OutFlow which are relevant for design decision making.

Proposition 2: For both InFlow and Outflow, there are three types of attributes that are relevant for design decision making:

- Physical attributes: UOH, size, weight, shape, packaging, value, etc
- Operational attributes: rate of use, rate variability, correlations, etc.
- Managerial attributes: seasonality, value, etc.

The essence of functional requirements analysis is the discovery of the particular set of attributes which are most important for expressing the requirements for a particular warehouse design problem. This task is somewhat easier for the OutFlow production requirements. The critical physical attributes of the customer orders are the UoH for the order itself, the composition of the order, i.e., a single line or multiple lines, and other physical attributes, such as weight, hazard, etc. Together, these physical attributes will dictate the required capabilities of the order assembly process itself, e.g., to build mixed cases or mixed pallets, or to lift up to 1000 kg. The critical operational attributes of the orders may be the type of shipper, e.g., so that all package handling orders are completed by a certain time, or the location of the customer, e.g., so that all orders destined for a certain geographical area can be completed together.

Once the characteristic attributes of the OutFlow have been identified, analysis of the OutFlow can be used to disaggregate the OutFlow into "order groups", i.e., orders which are sufficiently similar so that the same processes can be used for assembling and shipping all the orders in an order group. Note that this does not require knowledge of exactly how the order assembly and shipping will be done; rather, it only requires knowing that, however it is done, every order in an order group can be treated exactly the same way. This greatly simplifies the subsequent functional and embodiment design decision making, since the focus can be on a relatively small number of order group flows which are well defined in terms of physical attributes and in terms of the associated production rates. For each order group, the required "production rate" must be described.

Identifying the set of attributes to be used to characterize the InFlow consumption requirements is somewhat more difficult. Certainly, the physical attributes of the arriving SKUs will be important, such as UoH and environmental requirements. What also is important is the set of attributes which will materially impact the decision about how to store the arriving SKUs, which is, itself, a design decision, which has not yet been made, and ultimately may depend in large part on how the orders are picked, another decision that has yet to be made. The specific set of attributes that should be considered may vary by industry and/or by the scale of the warehouse operations. In general, however, it is clear that attributes like SKU cube and weight, SKU maximum and average inventory, SKU retrieval rate, SKU environmental requirements, etc., will be important. What also may be important is information such as what SKUs often appear together in customer orders.

Once the characteristic attributes of the InFlow have been identified, the SKUs can be partitioned into SKU families, i.e., groups of SKUs which are sufficiently similar that the same processes for receiving and putaway to storage can be used for all SKUs in the same SKU family. Note that this does not require knowing how the receiving, moving, putting, or storing will be done; rather, it requires only knowing that every SKU in a SKU family is sufficiently similar that all SKUs in the family can be treated exactly the same way with regard to receiving and putaway to storage. For each SKU family, the required "consumption rate" must be described.

The selection of the specific attributes to be used in the functional requirements analysis for a particular warehouse design is itself a design decision. As such, it should not be expected that there is a rule or an algorithm that will make the selection. Statistical analysis of prior supplier and customer orders can be helpful, but only to the extent that it either confirms or refutes the assumption that some particular attribute is important. The prospective warehouse designer has two possible strategies:

- Try many different attribute selections, complete the functional and embodiment design for each, and choose the one that leads to the best embodiment design, or
- Rely on experience and judgment to identify an attribute selection.

The first strategy is unlikely to be practical, due to the time and cost of detailed functional and embodiment design. The second strategy might seem to be impossible (one can't design unless one already has experience in designing), but actually has considerable promise. In (Apple, Meller and White, 2010), a number of tables and charts are presented which capture or have the potential to capture domain specific knowledge which could be of use in selecting the attributes to characterize the InFlow and OutFlow. In fact, a promising direction for further development is the design of data structures for capturing the kinds of domain knowledge that would helpfully inform attribute selection in functional requirements analysis.

The essential result of analyzing the InFlow consumption and OutFlow production requirements is the identification of the SKU families and order groups, along with the specification of the rates of each. The rate specification can be simply an average value, an estimate of mean and variance, or even a complete distribution. The more complete the specification, i.e., the more information it contains, the more thorough the evaluation can be for the performance of any proposed warehouse design.

It would be difficult to overstate how important functional requirements analysis is to the final design of the warehouse. A poor choice of SKU families or order groups can lead to an initial functional design that simply cannot be embodied in an effective form. In fact, it can be argued that selecting the appropriate SKU families and order groups is the fundamental and most creative aspect of warehouse design. Thus, it is likely the hardest to capture, document, and teach.

5. Functional Design

Essentially, what a warehouse does is to transform its InFlow, i.e., the stream of arriving supplier orders, into its OutFlow, or stream of departing customer orders. Specifying this transformation process is, in fact, the essence of the warehouse design problem. The usual approach to warehouse design is to try to specify the embodiment of this process, i.e., what technologies and practices will be used to pick and assemble customer orders. The approach proposed here is quite different and begins by trying to understand the transformation process at a functional level. In this context, the functions of a warehouse are abstract processes which make abstract changes to an abstract flow. For example, conceptually, the "receiving function" is an abstract process that transforms the InFlow over some time period into a set of units of handling ready to be put away into some storage function. The "receiving function" concept can be understood without specifying the details of the InFlow (the specific orders, SKUs, etc.) or the details of the putaway (conveyor, lift truck, etc.) or the details of the storage (block stacking, pallet rack, AS/RS, etc.). In a similar way, a set of warehouse functions can be identified that are understood in the warehousing domain without detailed definition, which leads to Axiom 4:

Axiom 4: To transform InFlow to OutFlow, a warehouse must be capable of the (abstract) functions of receiving, putting, storing, retrieving, moving, sorting, assembling, and shipping.

As generally understood, the receiving function breaks down incoming supplier orders (or replenishment orders) for putaway (via a move function) into the storage function. When customer orders are to be assembled, a retrieval function removes goods from storage, an assembly function creates the complete customer order, a movement function transports goods and orders as necessary, and the shipping function makes the orders ready to leave the warehouse for transport to customers. Note that the descriptions here are in terms of functions, not in terms of the embodiment of those functions, and functions can be identified without necessarily specifying the order in which they are performed. Figure 4 summarizes the essential warehouse functions.

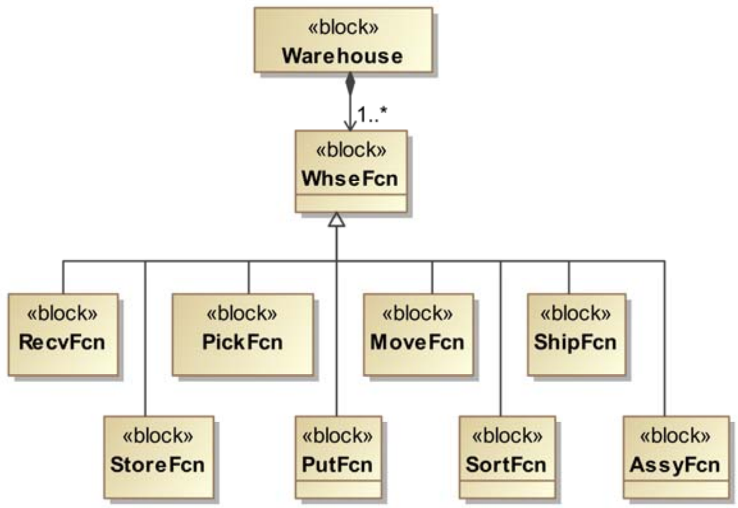

Figure 4 Warehouse Functions

It is important to emphasize that "functions" are abstract concepts, which ultimately will be instantiated in physical systems. Functional thinking is about "what must happen" rather than "how it happens".

The result of functional requirements analysis is a set of SKU families (resp. order groups), such that all SKUs (resp. orders) in a particular SKU family (resp. order group) can be treated in exactly the same way for receiving, putaway, and (initial) storage (resp. order assembly and shipping). Axioms 5 and 6 translate this into an operational approach to functional design.

Axiom 5: Each SKU family in the InFlow can be associated with a unique storage function.

Axiom 6: Each customer order group in the OutFlow can be associated with a unique assembly function.

Figure 5 displays the relationships which are implied by Axioms 5 and 6. As a result of Axioms 5 and 6, we can propose the following:

Proposition 3: Functional design is initiated with the following:

1. a single receiving function,
2. a single shipping function,

3. for each SKU family, a move function (from the receive function), a put function, and a store function, and
4. for each order group, an assembly function, and a move function (to the ship function)

One implication of Proposition 3 is that the functional design process will define, for each SKU family, corresponding move, put, and storage functions. Similarly, for each order group, functional design will define corresponding assembly and move functions. Each move so defined has an associated *flow*, either a flow defined by a SKU family, or one defined by an order group. These flows conform to the structure illustrated in Figure 2.

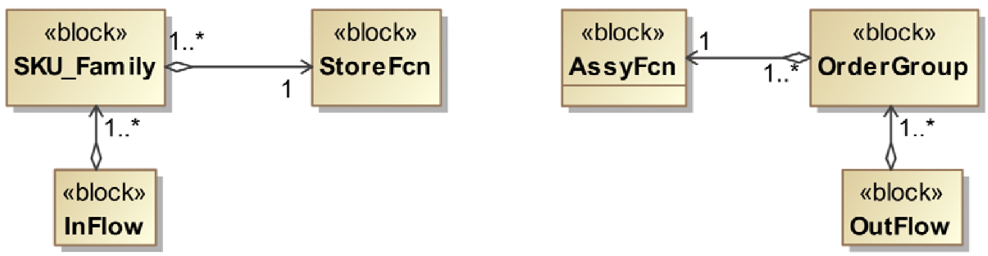

Figure 5 SKU Family and Order Group

What remains is to specify those functions necessary to remove products from their respective storage functions and move them to the order assembly functions to produce the orders in the OutFlow. Note that a particular SKU family may contain SKUs that are required by more than one order group, and, similarly, a particular order group may require SKUs from more than one SKU family. Clearly, whenever a SKU in a SKU family is required by an order group (for its assembly function) there must be, in some form, a retrieval function (from the storage function) and a move function (to the assembly function). As before, the move function corresponds to a *flow*, which is defined by the intersection of the corresponding SKU family and order group.

It is important to note that there may be sub-assembly functions, and that move functions may be applied to a single SKU, to a batch of the same SKU, or to a batch of different SKUs. For example, if items are order picked to a carton from bin shelving, and the carton is transported to a mixed pallet build station, the “order pick to carton” constitutes a sub-assembly function. These kinds of variations are easily accommodated, as are reserve storage with forward picking systems.

The next design phase will be embodiment design, in which the functional specification will be converted into a system specification with particular resources identified for each system. While it may be troubling to contemplate, there is not a crisp boundary between functional design and embodiment design, because there are implied limitations associated with a particular functional design that will become obvious when embodiment is considered, and lead to revisions to the functional design. For example, a functional design containing the design fragment (StoreFcn, RetrieveFcn, MoveFcn, AssyFcn) implies that what is retrieved and moved is the SKU in the amount needed for either a single customer order or a set of customer orders being simultaneously assembled. When considering the embodiment, it may become clear that this move function cannot be embodied efficiently, due, say, to the distances involved and frequencies of move. The result may be a revision to the functional design, for example, in which a number of SKUs are retrieved and assembled for movement to the order assembly station where they are combined with other SKUs from other storage functions.

The consequence of this observation is that design will necessarily iterate between functional and embodiment design. In contemporary practice, functional design is implicit and this iteration is only apparent in the sequence of embodiment designs that are proposed.

There is a relatively small set of warehouse function types, as illustrated in Figure 4. The number of storage functions in a warehouse design is the same as the number of SKU families, and the (initial) number of assembly functions is the same as the number of order groups. The number of functions required to bridge from storage to order assembly is probably bounded in practice by:

$$2 \times (\text{number of SKU families}) \times (\text{number of order groups})$$

This is probably a very loose bound.

Some functions are associated with a single flow, e.g., a move function, while some are associated with multiple flows, e.g., a receiving function has one InFlow and flows out that correspond to each of the SKU families. The flows associated with a function will carry information that dictates both the capability and capacity of the system that implements the function.

6. Embodiment Design

Functional design specifies what must happen to translate the InFlow to the OutFlow, and embodiment design specifies how the functional design will be accomplished. Embodiment specifies the set of physical systems in the warehouse, what resources are assigned to each system, and where are the interfaces between systems. Every function in the functional design must either be incorporated into some system, or shared between two or more systems. Thus, one way to approach embodiment design is as a problem of partitioning the set of functions into a set of systems.

Given the number of basic warehouse functions that will appear in the functional design for any interesting warehouse, there may be a very large number of ways to partition. However, there are some constraints that may simplify the problem. When some set of functions is designated to a particular system, the capabilities of that system must be compatible with the definition of the associated functions. For example, the UoH for each function in the set must be accommodated by the system specification. Similarly for any environment conditions, weight, hazardousness, etc.

Embodiment is, especially so, a phase of design where knowledge of warehouses and of previous designs can be quite valuable. In particular, the kind of domain knowledge that is illustrated in (Apple, Meller and White, 2010) can be extremely useful in embodiment design.

When a particular (embodiment) system type is chosen, and functions are assigned to this system, the system must be sized and configured, so that it has both the capabilities and the capacity that can be inferred from the set of functions assigned to it. This is the stage of design where many of the analysis models from the literature can be brought to bear. For example, if a number of storage functions are assigned to a single deep pallet rack system, the integration of their attributes can be used to determine the number of storage locations required, and perhaps the number of aisles and aisle lengths that will minimize expected pallet retrieval times.

7. Example

One of the few illustrations of a process for generating a warehouse design is given in (Apple, Meller, and White, 2010), and their example will be the basis for demonstrating the approach being proposed in this paper. The scenario they posed involves servicing orders for automobile repair parts in which the basic UoH is a case—the implication being that very rarely is a full pallet required for an order line. In the following, the parameters for the case have been rounded and some values inferred from the given data.

Functional Requirements Analysis

There are two categories of orders, from "jobbers" and from distribution centers (DCs), in the OutFlow, but the only significant distinction between them is that jobber orders have many fewer lines, on average. Approximately 25 orders are shipped each day, for a total of about 100 pallets and about 4000 cases. A large fraction of the orders will contain multiple pallets. The functional requirement for assembly is to be able to retrieve cases and build mixed pallets, and to stage multiple pallets for a single customer order.

The InFlow consists of full or partial pallets representing approximately 8000 SKUs, and statistical analysis of historical data reveals that about 6000 SKUs have less than a full pallet of inventory, 25% have only a single pallet, and only 3% have more than 5 pallets. In terms of turn-over, fewer than 1% of the SKUs accounted for 23% of the throughput on a unit basis and 34% on a cube basis. Based on storage UoH and turnover, it was determined that there were four families of SKUs:

- Less than pallet inventory: ~6000 SKUs
- Pallet inventory-super fast mover: ~100 SKUs
- Pallet inventory-fast mover: 1000 SKUs
- Pallet inventory-slow mover: ~1000 SKUs

Further analysis of the InFlow is summarized in Table 1, where "pick density" is the expected number of picks per day for a SKU in the corresponding SKU family, and “pallets per day” is the pallet cube equivalent of the picks from the corresponding SKU family. The “approximate space” is a very crude calculation intended only to determine the approximate magnitude of floor space required.

Table 1 Requirements Analysis

Family	*Number*	*% Throughput*	*Pick Density*	*Pallet per Day*	*Slots*	*Approx Space*
Case	6000	20	0.133	16	6000	20000
PalletSlow	1000	20	0.8	16	1000	10000
PalletFast	1000	35	1.4	28	2000	20000
PalletSuperFast	100	25	10	20	1000	1000

Functional Design

Functions already identified are the receiving and shipping functions, the four storage functions associated with SKU families, and the mixed pallet assembly function associated with the single order group. Clearly, functions will be needed to represent goods being moved from receiving to storage, put in storage, retrieved from storage, moved to assembly, and moved from assembly to shipping. Figure 6 displays a trivial way to represent these functions, implying that the goods required for each order are retrieved from storage and moved directly to order assembly. However, this functional design may not be one that can be embodied effectively. In thinking ahead to embodiment design, the following observations are pertinent:

- Picking directly to pallet will be problematic, because of the very low pick density for two of the four SKU families, and because so many of the orders will require multiple mixed pallets
- Zone picking to pallet and then consolidating at assembly will not be efficient, again due to the pick densities, the travel to assembly, and the consolidation
- Batch picking seems attractive, especially from the Case SKU family, since it partially compensates for the low pick density.
- Batch picking will require some kind of accumulation/sortation prior to assembly
- The SKUs required from the PalletSuperFast family could be picked directly to pallet and used as the “seed” for building a customer’s mixed pallet(s)

Based on these considerations, an alternative functional design is displayed in Figure 7.

Embodiment Design

In embodiment design, the identified functions are assigned to specific systems for implementation, and specific resources are allocated to those systems. In the RayLoc case, one way to group functions is indicated in Figure 8. This grouping is a “design decision” in the sense that there is not necessarily a specific rule or algorithm that will dictate what the grouping should be. Rather, the grouping is based on looking at the complete set of capability and capacity requirements, on general knowledge about how those requirements will interact, and on intuition about what constitutes a “good”

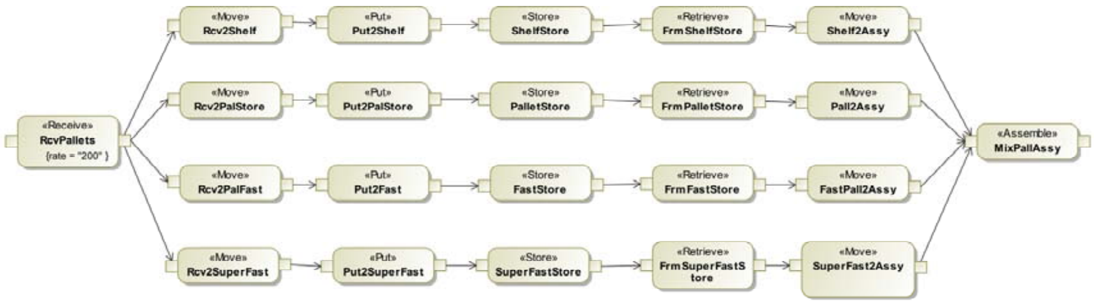

Figure 6 Initial Functional Design

warehouse design. The grouping decision could also benefit from the kind of knowledge identified in (Apple, Meller, and White, 2010) as “matrix solutions guides”.

Figures 8 and 9 go here

The less-than-pallet inventory store function is embodied as a hand-stacked shelving storage system (system 3 in Figure 8). The super-fast pallet storage function is embodied as pallet flow lanes (system 5). The remaining two pallet-based SKU family storage functions are combined in a pallet rack system (system 4). Goods are moved from receiving to the shelf storage using powered pallet jacks (system 1), and to the two pallet storage systems using lift trucks (system 2). Goods are batch picked from shelf storage using powered pallet jacks (system 6) and then loaded onto a conveyer system (system 9) for movement to the order assembly area. Goods are batch picked from the pallet storage (system 4) using lift trucks (system 8) and then loaded onto a conveyor (system 9) for delivery to the order assembly area. At the end of the case conveyor (system 9) there is a sortation system (system 10) that directs cases to the correct assembly station. The assembly stations constitute system 11.

At this point, because each *flow* in the functional design is completely defined, there is enough information to permit the optimization of each of the systems in the embodiment, i.e., to choose the specific technology (e.g, type of lift truck used in system 2), and to determine the configuration (e.g., the aisle configuration for shelf storage or the numbers of pallet jacks and lift trucks needed). Any appropriate optimization can be done, such as turnover based slotting, order batching, etc. Note also, that it would be quite reasonable to combine systems 2, 7 and 8 into one integrated lift truck fleet that was dispatched as needed.

8. Conclusions and Future Work

Four fundamental concepts for warehouse design have been identified and illustrated: (1) flow (a stream of orders with attributes); (2) functional requirements (the SKU families and order groups and their attributes); (3) functional design (specifying what must be done to convert SKU family flows into order group flows; and (4) embodiment design (specifying how each function will be implemented).

Three distinct types of warehouse design decisions have been identified: (1) the selection of SKU families and order groups, using empirical knowledge gleaned from descriptions of the SKUs and customer orders and domain knowledge of warehousing industries and systems; (2) the specification of the function flow network, defining what must be done to translate the SKU InFlow to the customer order OutFlow; and (3) the allocation of functions to systems and the configuration of those systems.

There are essentially two opportunities for modeling to contribute to the warehouse design process: (1) models can be used to *evaluate a decision,* i.e., to predict what the impact on warehouse performance will be as a result of the decision; and (2) to *suggest a decision* in situations where the alternatives or the space of alternatives and a definitive criterion for selection can be defined, e.g., optimizing the configuration of a pallet rack storage system. Evaluation models are appropriate across all three kinds of decisions, and there are numerous examples. Normative models in the extant literature are largely confined to support of embodiment decisions. One very interesting direction for future research is to explore the use of normative models in developing the function flow network and in requirements analysis.

This paper demonstrates that warehouse design can be given a formal structure. Such formalism is a prerequisite to the development of computer aided warehouse design tools. What remains to be accomplished is the realization of such a tool.

9. Acknowledgements

The ideas presented here have had a very long gestation, and whatever clarity they may exhibit is due to the generosity of many people. My colleagues at Georgia Tech, Gunter Sharp and Marc Goetschalckx, spent many hours in informal seminars with me and graduate students in the Keck Virtual Factory Lab, including Wen-chih Chen, Andy Johnson, and Jinxiang Gu. Jim Apple joined that discussion on occasion and on many other occasions has provided enlightenment. Russ Meller has challenged, encouraged, and occasionally abetted these efforts. Detlef Spee and Michael Schmidt from the Fraunhofer IML, and students they have sponsored from TU Dortmund (Bastian Himmelroeder, Manuel Klennert, Felix Friemann, Henning Blunck, Daniel Vogdt, Stefen Schieweck, Tim Skrotzki, and Martin Thormann) have been wonderful collaborators in exploring SysML for articulating both an object-oriented specification of the warehousing domain and a practical warehouse design workflow. None of these people bear any responsibility for egregious conceptual errors I may have made.

10. References

Apple, J. M., R. D. Meller, and J. A. White (2010), “Empirically-Based Warehouse Design: Can Academics Accept Such an Approach?” in Progress in Material Handling Research: 2010, K. Ellis, K. Gue, R. de Koster, R. Meller, B. Montreuil, and M. Ogle (eds), Material Handling Industries of America.

Baker, Peter, and Marco Canessa, “Warehouse design: A structured approach”, European Journal of Operational Research, Volume 193, Issue 2, 1 March 2009, Pages 425-436.

Blunck, H., D. Vogdt, L. McGinnis (2011), “Formal Design Models for Order Picking Systems,” IERC, Reno, NV, May, 2011.

Bodner, D. A., T. Govindaraj, K. N. Karathur, N. F. Zerangue and L. F. McGinnis,(2002), "A Process Model and Support Tools for Warehouse Design," in Proceedings of the 2002 NSF Design, Service and Manufacturing Grantees and Research Conference, 2002.

Friemann, F., M. Klennert, and L. McGinnis (2010), “Model Based Systems Engineering in Warehouse Analysis and Design,” IERC, Cancun, Mexico, June 7, 2010.

Goetschalckx, M., T. Govindaraj, D. A. Bodner, L. F. McGinnis, G. P. Sharp and K. Huang (2001), "A Review and Development of a Warehousing Design Methodology, Normative Model, and Solution Algorithms," in Proceedings of the 2001 Industrial Engineering Research Conference, Atlanta, GA: Institute of Industrial Engineers.

Goetschalckx, Marc, Leon McGinnis, and Gunter Sharp (2008), “Modeling Foundations for Formal Warehouse Design,” in Progress in Material Handling Research: 2008, K. Ellis, R. Meller, M. Ogle, B. Peters, G. Taylor, and J. Usher (eds), Material Handling Industries of America.

Gu, Jinxiang, Marc Goetschalckx, and Leon McGinnis, (2010), “Research on warehouse design and performance evaluation: a comprehensive review,” European Journal of Operational Research 203, 539-549.

Karathur, K. N., T. Govindaraj, D. A. Bodner and L. F. McGinnis, (2002), "Design of a Computational Tool for Warehouse (Re)Design and Analysis," in Proceedings of the Industrial Engineering Research Conference.

McGinnis, L. F., (2008), “Facility Design Workflow Management,” IERC, Vancouver, Canada, May 21, 2008.

McGinnis, L., (2010), “The Future of Modeling in Material Handling Systems,” ?” in Progress in Material Handling Research: 2010, K. Ellis, K. Gue, R. de Koster, R. Meller, B. Montreuil, and M. Ogle (eds), Material Handling Industries of America.

McGinnis, Leon F., (2004), “Developing a Reference Model for Warehouse Specifications,” 2004 IIE Research Conference, Houston, TX.

McGinnis, Leon F., Marc Goetschalckx, and Gunter Sharp, (2006), “A Comprehensive Model of Traditional Warehouse Design,” in Progress in Material Handling Research: 2006, R. Meller, M. Ogle, B. Peters, G. Taylor, and J. Usher (eds), Material Handling Industries of America.

McGinnis, Leon, and Russell Meller, (2009), “Operations Research in Warehouse Design: Is There a Role for Synthesis (vs. Analysis) in INFORMS?”, INFORMS Annual Meeting, San Diego, CA, October 13, 2009.

Sharp, Gunter, Marc Goetschalckx, and Leon McGinnis, (2008) “A systematic warehouse design workflow: focus on critical decision,” in Progress in Material Handling Research: 2008, K. Ellis, R. Meller, M. Ogle, B. Peters, G. Taylor, and J. Usher (eds), Material Handling Industries of America.

Su, N. P. (1990), The Principles of Design, Oxford University Press.

Tate, Derick (1999), A Roadmap for Decomposition: Activities, Theories, and Tools for System Design, Department of Mechanical Engineering, Massachusetts Institute of Technology.

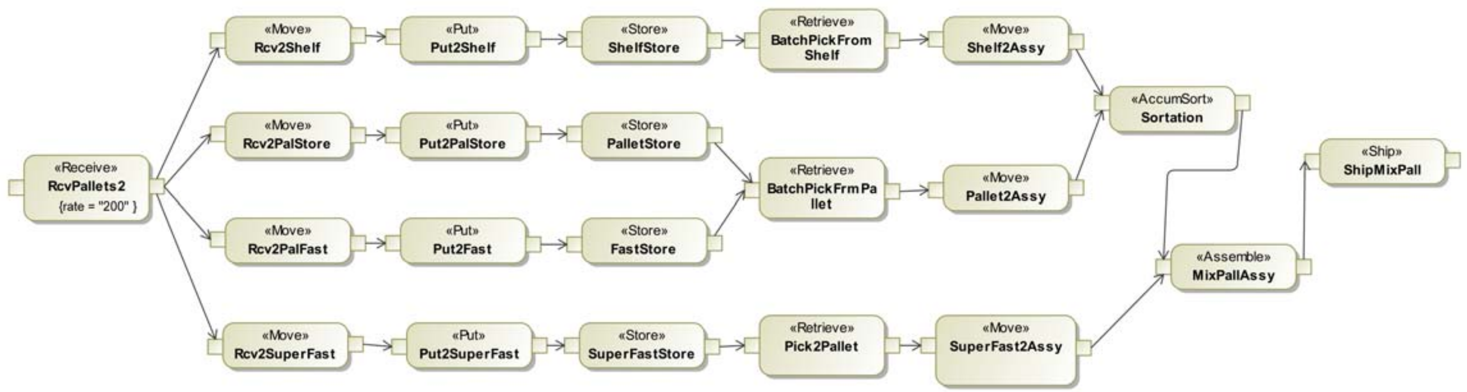

Figure 7 Revised Functional Design

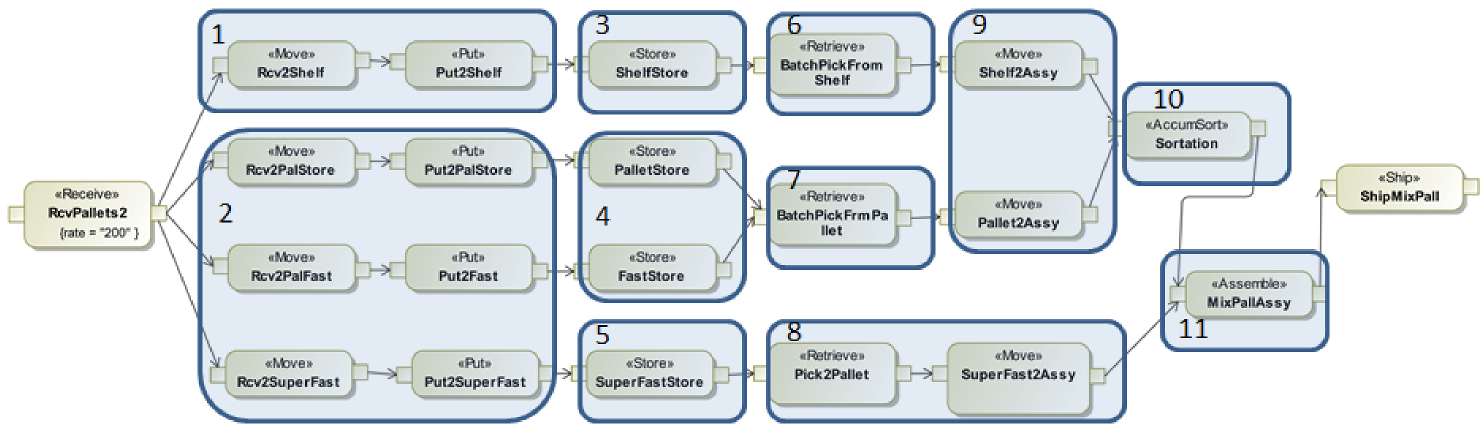

Figure 8 Function Allocation to Systems