Daniel Loranger

CS255, 25-May-2025

Hamp Crafts DFD upgrades

Sources

* Supplier
* Carrier
* Customer

Sinks

* Shipper

Processes

1. – Receive Customer Order
2. – Checkout
3. – Fulfill Order
4. – Choose Supplier

Data-Flows

With the brick-and-mortar model described, some of the processes appear to be muddied and unclear, such as how and what the supplier would be providing to a Shipper (retail clerk?), and also how the carrier would be involved for a carry out order. The overall architecture appears to be a half baked version of desired online drop shipping and central warehouse / in-store sales.

* From C***ustomer Order***, unknown information is transmitted to the C***heckout*** process.
  + ***Checkout*** Process is responsible for tabulating the pricing (missing data source), and feeding the order details to the ***Carrier*** via the *Shipping Plan*.
    - ***Carrier*** – using the order details (sink) shipping logistics are passed to the ***Shipper*** (source).
  + ***Checkout*** process is responsible for passing the order process to the ***Fulfill Order*** process.
    - ***Fulfill Order*** process takes the order details (sink) and performs 3 source activities
      * ***Delivery Plan*** – The order is used to select the supplier. The usage here is unclear if this is for drop shipping, inventory replenishment (brick and mortar stock, warehouse inventory, etc.)
      * ***Shipment Schedule*** – This appears to be both a source and sync operation, details are not clear what information is passed.
      * **Order details passed to *Shipper***
  + ***Choose Supplier***process sources *contract negotiations* to the *Supplier*. Presumably this works then as a source as *Shipment Schedule* to the *Fulfill Order* and unknown data to the ***Shipper***

Additional Requirements for online store-front

* Centralized inventory (in stock/out of stock) management module
  + New data entry process (Source)
  + New data consumption process (sink)
  + New database
* Centralized pricing management module
  + New data entry process (Source)
  + New data consumption process (sink)
  + New database
* External Shipping calculation process (customer selects shipping option)
  + External API integration (source/sink)
  + Replacement process for new tools
* Web order check out separate from in-store checkout modules
  + Potentially new database
  + New handling process (source)
* User account management based on roles
  + New database to hold user accounts
  + New process, user accounts (source)
  + New process, user accounts (sink)
* Payment processing online (electronic) vs in-store (cash/electronic) modules
  + New processes for ecommerce
* Order history /details module
  + New process for order entry (source)
  + New process for order review (sink)
  + New database for order management
  + user account data (sink)
* Automated inventory replenishment module
  + inventory database (sink)
  + user account database (sink)
  + New business logics for setting rules (source)
* Order picking / prioritization module
  + Replacement process for shipping workflows (sink)
* Inventory fulfillment tracking module (supplier orders pending delivery type stuff)
  + Replacement process for procurement orders (source)
  + Replacement process for procurement receipts (sink)

-- Integrations

Upon reviewing the existing work flows that are presented by the DFD, it is clear that processes are sub-optimal at best and are heavily dependent on human management to ensure inventory levels are tracked, resupply orders are processed, and shipments are managed. While the guidance is to integrate into the existing models, it would be my recommendation to actually overhaul the work-flows with a modern approach that will ultimately serve the customer better.

While integrating into the current workflows (patching at best) will work for a single centralized storefront, it will not scale to multiple store-fronts which are comingled with online retail web-stores. In the existing workflows, there is too much room for human error and inventory shortages will undoubtedly happen. This will result in back-orders, and even cancelled orders due to delayed shipments. Customer shipping details will be missed, or entered with errors, leading to customer frustrations, and general dis-satisfaction, negatively impacting the business as a whole.

*Employees are assigned to check the inventory system to ensure all orders are processed in a timely manner. Communication regarding inventory shortages or delays in delivery is a manual process for Hamp Crafts if a potential shortage occurs. Otherwise, the inventory specialist and order fulfillment team input information into the local database regarding order statuses and tracking of shipments.*

If the customer does want to continue with the existing workflows, the web order system could be added as a layer above the existing cashier system, where the agent will manually transfer the web order into the existing work-flows, allowing existing processes to continue working undisturbed.

An order history module could be inserted inside the shipper process to update the customer order information, providing electronic tracking/etc. with minimal impact to the shipping agent, while improving customer order tracking. Currently this information is manually entered into a local database anyways, so this would just be doing the same task in a dedicated tool/subprocess.

Implementing an inventory management process would have a more detailed data workflow, as it would be sinking order quantities from the web cart and store-front sales, while also sourcing the inventory levels to the web cart system so web customers can monitor the inventory available for purchase. This would also have interactions with the choose supplier & supplier entities and it would allow the buyers and vendors to better track the business needs, so they could also act a data sink to the inventory levels information managed by this new process.

A centralized price management process would need to be implemented as the store-front typically uses labels for discounted products, bulk purchases, etc. These labels would not translate to online sales portals, leading to pricing mismatch and confusion. An example would be an online sale price being lower than a retail storefront price, which causes a customer to challenge the in-store pricing causing confusion, congestion, and overall dissatisfaction by the customers and also the sales agent. This tool would be implemented as a sink to the procurement team who can update pricing based on market conditions, availability, etc. This data would then act as a source to the customer order workflows to show the real-time pricing during the shopping experience.

--Counter Recommendation

While potentially painful during initial roll-out due to retraining staff, it is strongly advised to instead move towards a modern point of sale model. In a point-of-sale workflow, inventory is managed at a central cloud-based database, which is automatically updated in real-time as goods are sold and brought into stock (deliveries). Customers and sales staff can instantly check inventory levels of all items at the company level ensuring only available goods are purchased (or backordered if desired).

Orders will be capable of being processed in a variety of ways, such as cash/electronic payments at a retail store-front (skipping shipments as irrelevant), or utilizing a web cart system for website purchases. Shipping carrier integrations are also well established allowing both the customer and the shipping agent to select the desired shipping methods at the time of checkout (customer), and also the shipping agent (carrier outages worked around, etc.)

Once orders are captured, shippable orders can be prioritized for next day deliveries first, followed by all others in a FIFO priority. This will automatically handle the picking order ensuring all orders are processed in a timely fashion, while also keeping the customer updated about shipment status, etc.

Bank account funding under this point-of-sale model is a well-established online business model that can utilize existing encrypted API’s from the local bank, or national services such as VISA/PayPal, etc. which ensure both security and protections to both the vendor and the customer.

Utilizing a centralized inventory management system will also allow auto-replenishment of products as desired, or alerting buyers to manually trigger replenishment orders as desired. Trending automation can also be implemented to help alleviate the manual tracking by alerting of trending up and trending down product sales.