Design Pattern for Updating an ATG Order

This is from a post Jeremy Sears made here on the ATG\_Tech Google Group, but I thought it was nice and clear and worth reposting. All credit to Jeremy for this:

“In general, the design pattern for updating an order is as follows:

1. Acquire lock-manager write lock on profile id from the /atg/commerce/order/LocalLockManager
2. Begin Transaction
3. Synchronize on the Order object.
4. Modify Order
5. Call ((OrderImpl) pOrder).updateVersion();
6. Call OrderManager.updateOrder()
7. Release Order synchronization
8. End Transaction
9. Release lock-manager write lock on profile id from the /atg/commerce/order/LocalLockManager

If you extend atg.commerce.order.purchase.PurchaseProcessFormHander, then steps 1,2,8 & 9 are done for you in the beforeSet and afterSet method implementations. Steps 3&7 are no longer strictly necessary, but are still good practice. Steps 3-7 should be performed manually in your application code. Calling step 6 within the transaction is mandatory and the lock manager work should always wrap the transaction to prevent some rare transaction race conditions. “