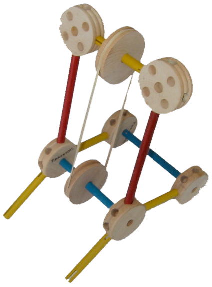

Pat’s Checkbook Discussion

February 28, 2014

This scenario describes Pat’s checkbook problem for which we will model a “checkbook assistant.” *L. J. Waguespack, Ph.D.*

Object Modeling Pat’s Checkbook

Pat’s checkbook problem is modeled on the popular personal accounting tool, Quicken. You might want to look over Quicken’s capabilities: http://quicken.intuit.com/.

Pat’s checkbook problem requires that you extract the “business rules” based on the scenario presented by Pat. A convenient simplification of the problem is to assume that all financial activity is channeled through “bank accounts.” In that manner you won’t need to be concerned by cash or credit activities outside of paying credit card bills. Another simplification is realizing that you’re modeling this problem from Pat’s perspective. You’re not modeling the bank; although Pat may need some service of the bank (e.g. whether a payment or deposit that Pat had noted has in fact been processed (aka “cleared”) by the bank as well; just like you would do by calling the bank and describing a transaction for the bank to verify as cleared.

The real challenge in the checkbook problem arises from the seemingly simple remark about “using the checkbook information in preparing an end of year tax return.” There are sample copies of the 1040 tax form and instructions on the course website in case you haven’t seen one before. (You won’t need any detailed tax knowledge, but the structure of the form and the required information will help you understand the modeling task.) To handle taxes you need to keep track of individual payments or deposits that may be “tax related.” If every individual item were remembered as a single transaction in Pat’s “checkbook” then the accounting might be more straightforward. But, some payments (such as paying a monthly credit card bill) may have multiple underlying purposes: some for food, some for gas, some for charity. Only some would have tax implications (e.g. charity related payments - a deductible expense). Interestingly enough you might notice that on the 1040 tax form there’s actually a line set aside for noting the expenses that you determine to be “charity.” And in fact any and all “tax-related” information (deposits or payments) have respective lines on the tax form and instructions for how those values are combined to determine tax due or refund due.

Your challenge is that the bank doesn’t know why a payment is made! So, although a bank statement (if you need one) may have a complete list of all the deposits and payments, it doesn’t have a notation for the “purpose” of any one of those. That’s why in the scenario it says, “Pat keeps notes on expenses.” Your model must account for these “purposes” indicating how they would be organized and remembered along with the specific payment that proves the expense existed. (By the way, purposes can relate to deposits as well because some deposits are tax-related like salary, but others may not be like refunds on non-tax-deductible items.)

Remember that we don’t model recalculable results in UML. But we do model services in classes that are “smart enough” and have “resources sufficient” to produce the results / information we would expect as a report (or tax return).

The final challenge is account reconciliation. Simply it means “Does the information Pat has match up with the information the bank has regarding the payments and deposits in any particular account?” If you have ever “balanced” your checkbook you’ve probably used the monthly bank statement. But, notice that a bank statement is a report and we don’t model those. We do model a class (or collection of classes) whose objects would be capable of providing the information found in a bank statement. You should also remember that we’re looking for “business rules” rather than “the way it’s usually done” when we business model. So, just because you may have used a monthly bank statement to balance your checkbook in the past, is a monthly statement really necessary. Can’t you find out if a single payment or deposit has cleared the bank in some other way. The “monthly bank statement” is really an “accident of implementation.” Historically, banks weren’t able to get the information gathered, collated and distributed more frequently than monthly, hence - monthly statement. But that wasn’t because of the business rules, but rather a technological “accident.” Since we’re business modeling, we don’t have to worry about “implementation,” only the business rules. So, think “out of the box!” which is the first step in “process reengineering.”

This is a fair summary of all the “hints” I’ll offer across the learning teams in class. Good Luck!!