

EAI via MDA
SPA Architecture Day 22 Jan 2005

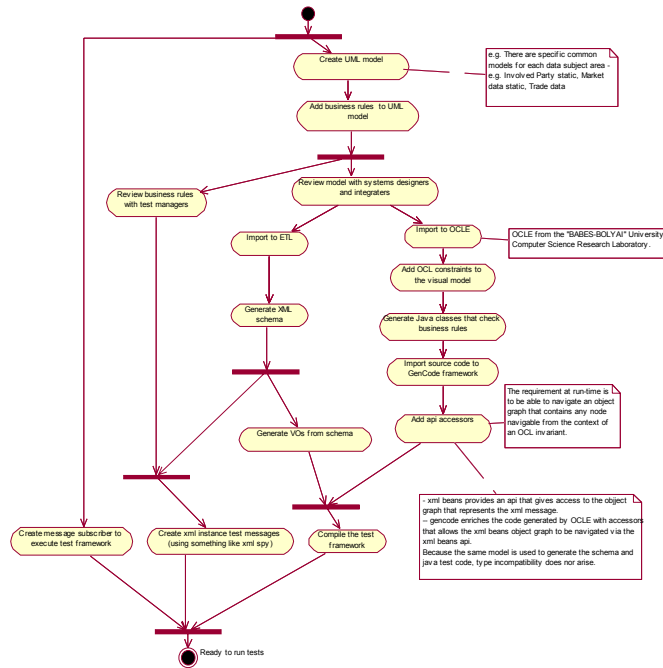
rob.james@hsbcgroup.com

ONSCREEN

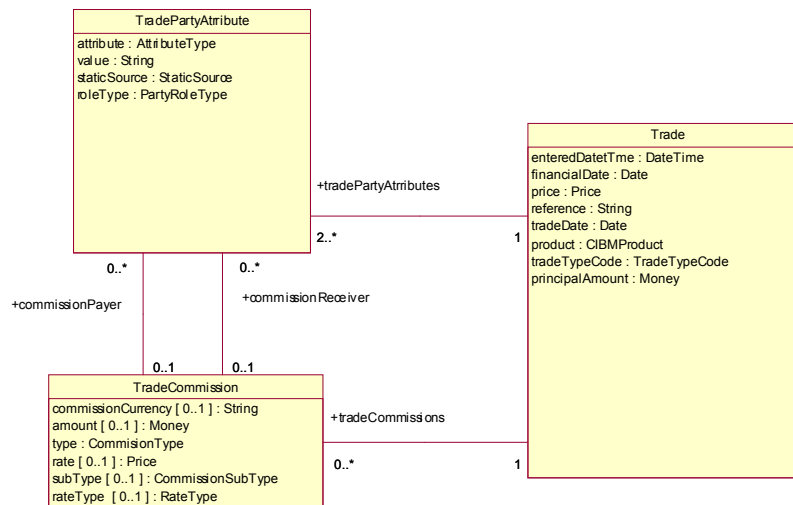
Context

- **Corporate Investment Banking and Markets (CIBM)**
 - Hundreds of 3rd party and bespoke applications
 - Dozens that deal with specific functions relating to the life-cycle of trades (and need to share common business semantics for particular data sets)
- **We estimate that 70% of the IT spend is on integration and testing**
- **So we would like to automate / generate as much of this process as possible**
- **We have noticed that MDA technologies seem (at present) better able to support this part of the development life-cycle than others**

Process overview

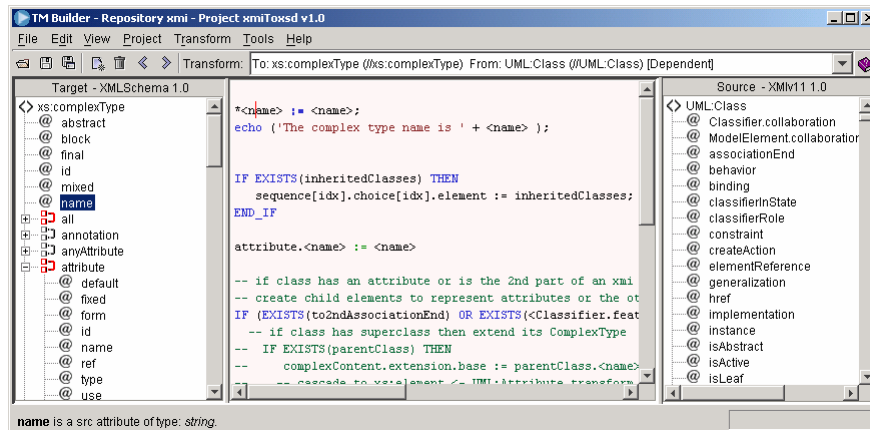


Create UML (platform independent) model



Meta model mapping

ONSCREEN



Add business rules

ONSCREEN

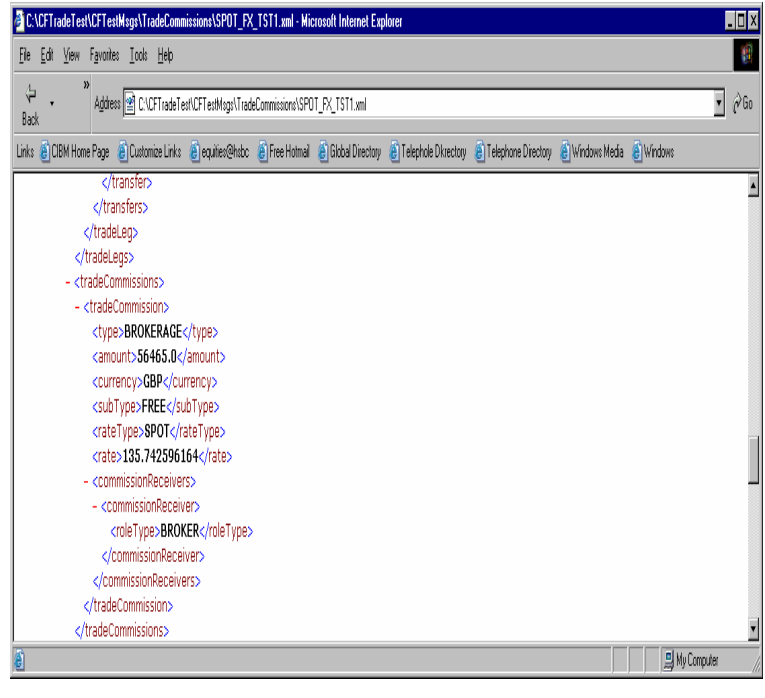
-- Constraints on TradeCommission

-- context TradeCommission
-- If commission type is BROKERAGE then the commission receiver's role type must be BROKER

-- context TradeCommission
-- If a trade commission's commission type is BROKERAGE then either
-- it has no commission subtype or the commission subtype is FREE

**At this point the model is reviewed by test managers
and system designers**

Create test messages



Add OCL

-- Constraints on TradeCommission

-- If commission type is BROKERAGE then the commission receiver's role type must be BROKER

```
context TradeCommission inv validCommissionReceiverRoleForBrokerage:  
self.type = CommissionType::BROKERAGE implies  
self.commissionReceiver -> notEmpty() and  
self.commissionReceiver.roleType = PartyRoleType::BROKER
```

-- If a trade commission type is BROKERAGE then either
-- It has no commission subtype or the commission subtype is FREE

```
context TradeCommission inv validCommissionSubtypeForBrokerage:  
self.type = CommissionType::BROKERAGE implies  
self.subType -> isEmpty() or  
(self.subType -> notEmpty() and self.subType = CommissionSubType::FREE)
```

Generate java code using OCLE

```

public void check_TradeCommission_validCommissionReceiverRoleForBrokerage() {
    CommissionType commissionTypeType = TradeCommission.this.type;
    boolean bEquals = commissionTypeType.equals(CommissionType.BROKERAGE);
    Set setCommissionReceiver = TradeCommission.this.getCommissionReceiver();
    boolean bNotEmpty = CollectionUtilities.isNotEmpty(setCommissionReceiver);
    Set setCommissionReceiver0 = TradeCommission.this.getCommissionReceiver();
    //evaluate 'collect(roleType)':
    List bagCollect = CollectionUtilities.newBag();
    final Iterator iter = setCommissionReceiver0.iterator();
    while (iter.hasNext()) {
        final TradePartyAttribute decl = (TradePartyAttribute)iter.next();
        PartyRoleType partyRoleTypeRoleType = decl.roleType;

        bagCollect.add(partyRoleTypeRoleType);
    }
    bagCollect = CollectionUtilities.flatten(bagCollect);

    boolean bEquals0 = bagCollect.equals(PartyRoleType.BROKER);
    boolean bAnd = bNotEmpty && bEquals0;
    boolean bImplies = !bEquals || bAnd;
    if (!bImplies) {
        System.err.println("invariant 'validCommissionReceiverRoleForBrokerage'
                           failed for object "+TradeCommission.this);
    }
}

```

The XML Beans api

- XMLBeans provides three major APIs:
 - XmlObject The java classes that are generated from an XML Schema are all derived from XmlObject. These provide strongly typed getters and setters for each of the elements within the defined XML. Complex types are in turn XmlObjects. For example getCustomer might return a CustomerType (which is an XmlObject). Simple types turn into simple getters and setters with the correct java type. For example getName might return a String.
 - XmlCursor From any XmlObject you can get an XmlCursor. This provides efficient, low level access to the XML Infoset. A cursor represents a position in the XML instance. You can move the cursor around the XML instance at any level of granularity you need from individual characters to Tokens.
 - SchemaType XMLBeans provides a full XML Schema object model that you can use to reflect on the underlying schema meta information. For example, you might want to generate a sample XML instance for an XML schema or perhaps find the enumerations for an element so that you can display them.

Enrich generated code

- OCLE code is generated in terms of the OCL context.
- The tool expects the users to populate an object graph with (an instance of) the context as the root.
- In order to use the checking code in a run-time environment we need to enrich this code so that it can be invoked in terms of the object graph produced by xml beans (or some other api).

Find accessor methods in code generated by xml beans using reflection

```

public void findGetters(buildImportStatements myPkg) {
    Iterator i = constraints.iterator();
    boolean pass=true;
    String keep=new String();
    String myType= new String();
    int lc=0;
    while (i.hasNext())
    {...
    // now go find using the symbol the method required for the getter
    getName=getName.toLowerCase();

    //System.out.println("Value is " + getName);
    try {
        Class c = Class.forName(myPkg.getPkgName()+"."+myType);
        Method m[] = c.getDeclaredMethods();

        for (int idx = 0; idx < m.length; idx++) {

            //m[idx].getName().toLowerCase().equals(getName);
            // System.out.println("Value is " + m[idx].getName().toLowerCase() + "" + getName);

            if (m[idx].getName().toLowerCase().indexOf("xget") < 0 &&
                m[idx].getName().toLowerCase().indexOf("xset") < 0 &&
                m[idx].getName().toLowerCase().indexOf("set") < 0 ) {

                // System.out.println("Value is " + m[idx].getName().toLowerCase() + "" + getName);

                if ( m[idx].getName().toLowerCase().indexOf(getName) > 0 ) {
                    //System.out.println("OK THE CALL I WILL MAKE IS " + m[idx].getName());
                    keep=keep.substring(0, keep.indexOf(",");
                    keep=keep.concat(", " + m[idx].getName() + "());");

                constraints.set(lc-1, keep);

```

Framework code

```
public void check_TradeCommission_validCommissionReceiverRoleForBrokerage() {
    boolean bImplies = false;
    TradeCommission[] tc = {};
    if ( getTrade().getTradeCommissions() == null )
        return;
    tc = getTrade().getTradeCommissions().getTradeCommissionArray();
    boolean bEquals=false;
    for ( int idx = 0; idx < tc.length; idx ++ ) {
        bEquals = tc[idx].getType().equals(CommissionType.BROKERAGE);
        if ( !bEquals )
            continue;
        TradeCommission.CommissionReceivers tcr =
            tc[idx].getCommissionReceivers();
        if ( tcr == null ) {
            continue;
        }
        TradePartyAttribute[] tpa = tcr.getCommissionReceiverArray();
        if ( tpa == null ) {
            continue;
        }
        for ( int i = 0; i < tpa.length; i++ ) {
            if ( !tpa[i].getRoleType().equals(PartyRoleType.BROKER) )
                continue;
            bImplies=true;
        }
    }
    if ( !bImplies && bEquals ) {
        exceptions.add("invariant
'check_TradeCommission_validCommissionReceiverRoleForBrokerage' failed for object
"+TradeCommissionChecker.this);
    }
}
```

ONSCREEN

Using the framework

- **Compile framework code**
- **Create subscriber**
- **Store faulty messages and generate xml error output**

ONSCREEN

**Only one step
not checked by
business**

