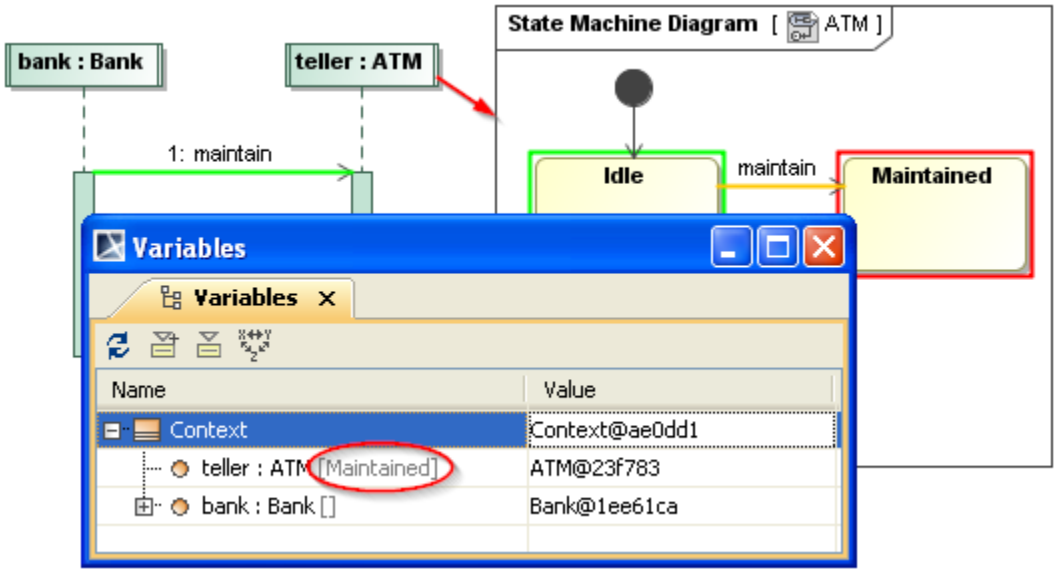
Message

On this page

According to the UML specification, a message defines a particular communication between lifelines of an Interaction. Magic Model Analyst simulates messages whose message sorts are **asynchSignal**, **synchCall**, **asynchCall**, and **reply**. You can specify a connector for the message. If the role of one connector end is the property specified by a source object of the message, Magic Model Analyst will send the message along the connector. A target object will be the object at the other end of the connector.

Asynchronous Signal Message (asynchSignal)

Magic Model Analyst simulates an asynchronous signal message by creating a signal object from a signal specified as a signature of the message. It sends the signal object to a target object asynchronously. The following figure shows that the bank sent the signal **maintain** to the teller using an asynchronous signal message. When the target object (the ATM object specifying the teller) received the signal, the State changed to **Maintained**.

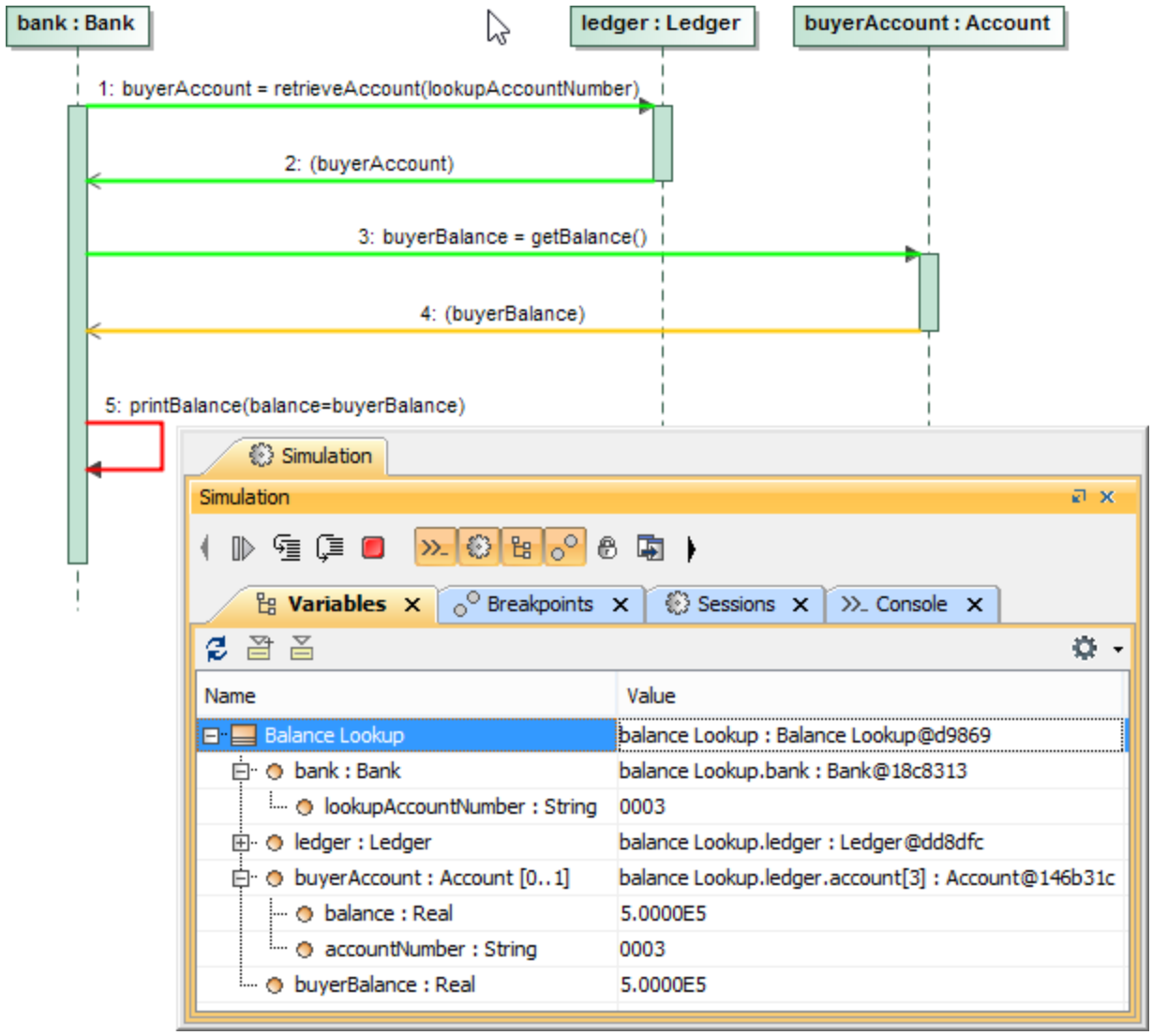


Simulation console showing the simulation of an asynchronous signal message.

Synchronous Call Message (synchCall)

When executing a synchronous call message, Magic Model Analyst simulates a Behavior (the method of called operation of a target object) from beginning to end before it can carry out the next message. The simulation method is similar to that of the call operation Action with "isSynchronous = true".

You can substitute values for input parameters of the called operation by specifying argument values on a synchronous call message.



Simulation console showing the execution of an asynchronous call message.

The preceding figure illustrates the synchronous call messages simulation. A synchronous call message **1** shows the bank object calling the operation **retrieveAccount(accountNumber : String)** of the ledger object. The substitution value for **accountNumber** is the value of **lookupAccountNumber**, which is 0003 in this example. **lookupAccountNumber** is the property of the bank.

The simulation looked for the account object with that particular **accountNumber** and returned it to the bank object with a reply message **2** (2 is a reply message of 1). The returned account object would be specified as the value of the **buyerAccount** represented by the lifeline **buyerAccount:Account**. The bank object would then call the operation **getBalance()** of the account object with a synchronous call message **3** that caused the balance value to reply to the bank with a reply message **4** (4 is a reply message of 3). Finally, the bank called itself to print the balance value with the operation **printBalance(balance:Real)**.

Asynchronous Call Message (asynchCall)

The simulation of an asynchronous call message is similar to that of a synchronous call message. In this simulation, Magic Model Analyst simulates a Behavior (a method of a called operation of a target object) on a new thread. Magic Model Analyst will immediately proceed to the next message once the simulation of the Behavior starts. It will not wait until the simulation of the Behavior completes. This type of simulation is similar to that of a call operation Action with "isSynchronous = false".

Reply Message (reply)

A reply message is a message sent as a reply to a call message (synchronous or asynchronous). If there is an argument value specified in the reply message, Simulation will evaluate the argument value.

> **Note**
> You can stop the execution at breakpoints by setting the **Constraint Failure As Breakpoint** option to **true** in the «SimulationConfig».

If the argument is not the name of a property, Simulation will verify (compare) the returned value from the called operation with the argument value.

If the result of evaluating the argument is the name of a property owned by a target object, a context object, or a source object, the value returned from the operation will be defined as the value of that property. The reply message **2** in the preceding figure has an opaque expression whose body is **buyerAccount** as an argument value.

buyerAccount is the property of the **Balance Lookup** Class which is the context of a sequence diagram. Therefore, the returned account object from the operation of the ledger object **retrieveAccount(accountNumber : String)** will be set as a value of the property **buyerAccount** when the reply message 2 is simulated.