```
using System;
using System.Collections.Generic;
using System.Text;

using Microsoft.Modeling;
using Microsoft.Modeling.Xrt;

namespace PubSub.Model
{

    #region Publisher

    [TypeBinding("PubSub.Implementation.Publisher")]
    public class Publisher
    {
        internal SetContainer<Subscriber> subscribers;

        [Rule(Action = "new this.Publisher()")] //must link to real code
        public Publisher()
        {
            subscribers = new SetContainer<Subscriber>();
        }

        [Rule(Action = "this.Publish(data)")]
        public void Publish(string data)
        {
            foreach (Subscriber sub in subscribers)
                sub.messages.Add(data);
        }
    }

    #endregion

    #region Subscriber

    [TypeBinding("PubSub.Implementation.Subscriber")]
    public class Subscriber
    {
        public SequenceContainer<string> messages;

        [Rule(Action = "new this.Subscriber(pub)")] //parameter is publisher
        public Subscriber(Publisher pub)
        {
            messages = new SequenceContainer<string>();
            pub.subscribers.Add(this); //notice break of encapsulation
        }

        [Rule(Action = "this.Received(data)")]
        public void Received(string data)
        {
            Condition.IsTrue(messages.Count > 0);
            Condition.IsTrue(messages[0] == data); //receives the message that matches the data passed in
            messages.RemoveAt(0);
        }

    }
    #endregion
}
```