

Analysis 2

J. L. De Coi

August 4, 2006

```
Status s;
NegotiationStrategy ns;
TerminationAlgorithm ta;
Filter f;
Peer otherPeer;

void perform(FilteredPolicy fp, Notification[] na){
    Check[] c = na.check();
    // If c[i] failed, it is not put in the state
    s.add(c);
    evaluationState.add(c);
    if(ta.terminate()){
        send(TERMINATION_MESSAGE, otherPeer);
        return;
    }
    Actions[] toPerform = f.extractActions(fp);
    Actions[] unlocked;
    FilteredPolicy[] fpToSend;
    FilteredPolicy[] myFp = f.filter(toPerform);
    for each i
        if(f.isLocked(myFp[i])) fpToSend.add(myFp[i]);
        else add the corresponding action to unlocked;
    Action[] aa = ns.selectActions(unlocked);
    Notification[] naToSend = aa.perform();
    Message m = new Message(fpToSend, naToSend);
    send(m, otherPeer);
}
```