```
testcase validWithdrawal (inout reason_Type reason) runs on hardwareEmulator_CType system
ATM_Interface_CType {

    var boolean result;
    var operationComplete_Type erroneousWithdrawal;
    var default theDefault := activate(HWE_Default());

    // Creation of PTC
    var bankEmulator_CType BE_PTC:=bankEmulator_CType.create;

    // Mapping & Connecting ports
    map(system:niSUT, BE_PTC:niCom);
    connect(self:coHWE, BE_PTC:coBE);
    map(system:hwiSUT, self:hwiCom);

    BE_PTC.start(BE_validWithdrawal()); // start PTC
    testCaseGuard.start; // start guarding timer

    // invocation of authentication procedure
    result := authentication(validCard_Par, validPin_Par, reason);

    if (result != true) { // test case fails in authentication procedure
        setverdict(inconc); BE_PTC.stop; // stop the bank emulator test component
        stop;
    }

    // start of withdrawal procedure
    hwiCom.send(validWithdrawalOp_Template);
    reason := unknown; // return value if MTC fails in default
    alt {
        [] hwiCom.receive(operationComplete_Type:{withdrawal, true, ?})
        {
            reason := noReason;
            setverdict(pass);
            all component.done; // wait for the termination of PTC
        }
        [] hwiCom.receive(operationComplete_Type:?) -> value erroneousWithdrawal {
            setverdict(fail);
            reason := erroneousWithdrawal.reason;
        }
        [] coHWE.receive(reason_Type:?) -> value reason {
            setverdict(fail);
        }
        [] testCaseGuard.timeout {
            reason := notInTime; setverdict(fail);
        }
    }
    deactivate(theDefault);
    if (reason != noReason) { // Stopping PTC in case of a failure
        BE_PTC.stop;
        stop;
    }
}
```