MDA-EFSM Events:

Activate()
Start()
PayCredit()
PayCash()
Reject()
Cancel()
Approved()
StartPump()
Pump()
StopPump()
SelectGas(int g)
Receipt()
NoReceipt()

MDA-EFSM Actions:

StoreData	// stores price(s) for the gas from the temporary data store
PayMsg	// displays a type of payment method
StoreCash	// stores cash from the temporary data store
DisplayMenu	// display a menu with a list of selections
RejectMsg	// displays credit card not approved message
SetW(int k)	// set value for credit/cash flag
SetPrice(int g)	// set the price for the gas identified by *g* identifier
ReadyMsg	// displays the ready for pumping message
SetInitialValues	// set *G* (or *L*) to 0
PumpGasUnit	// disposes unit of gas and counts # of units disposed
GasPumpedMsg	// displays the amount of disposed gas
StopMsg	// stop pump message and receipt? msg (optionally)
PrintReceipt	// print a receipt
CancelMsg	// displays a cancellation message

Operations of the Input Processor
(GasPump-1)

```
Activate(int a) {
        if (a>0) {
              d->temp_a=a;
              m->Activate()
        }
}

Start() {
        m->Start();
}

PayCredit() {
        m->PayCredit();
}

Reject() {
        m->Reject();
}

Cancel() {
        m->Cancel();
}

Approved() {
        m->Approved();
}

PayCash(int c) {
        if (c>0) {
              d->temp_c=c;
              m->PayCash();
        }

StartPump() {
        m->SelectGas(1);
        m->StartPump();
}

PumpGallon() {
        if (d->w==1) m->Pump();
        else if (d->w==0) {
              if (d->cash<(d->G+1)*d->price) {
                    m->StopPump();
                    m->Receipt()
              }
              else m->Pump();
        }
}

StopPump() {
        m->StopPump();
        m->Receipt();
}
```

Notice:
cash: contains the value of cash deposited
price: contains the price of the selected gas
G: contains the number of gallons already pumped
w: cash/credit flag

cash , *G*, *price*, *w* are in the data store
m: is a pointer to the MDA-EFSM object
d: is a pointer to the Data Store object

Operations of the Input Processor
(GasPump-2)

```
Activate(float a, float b) {
        if ((a>0)&&(b>0)) {
              d->temp_a=a;
              d->temp_b=b;
              m->Activate()
        }
}

Start() {
        m->Start();
}

PayCredit() {
        m->PayCredit();
}

Reject() {
        m->Reject();
}

Cancel() {
        m->Cancel();
}

Approved() {
        m->Approved();
}

Super() {
        m->SelectGas(2)
}

Regular() {
        m->SelectGas(1)
}

StartPump() {
        m->StartPump();
}

PumpGallon() {
        m->Pump();

StopPump() {
        m->StopPump();
        m->Receipt();
}
```

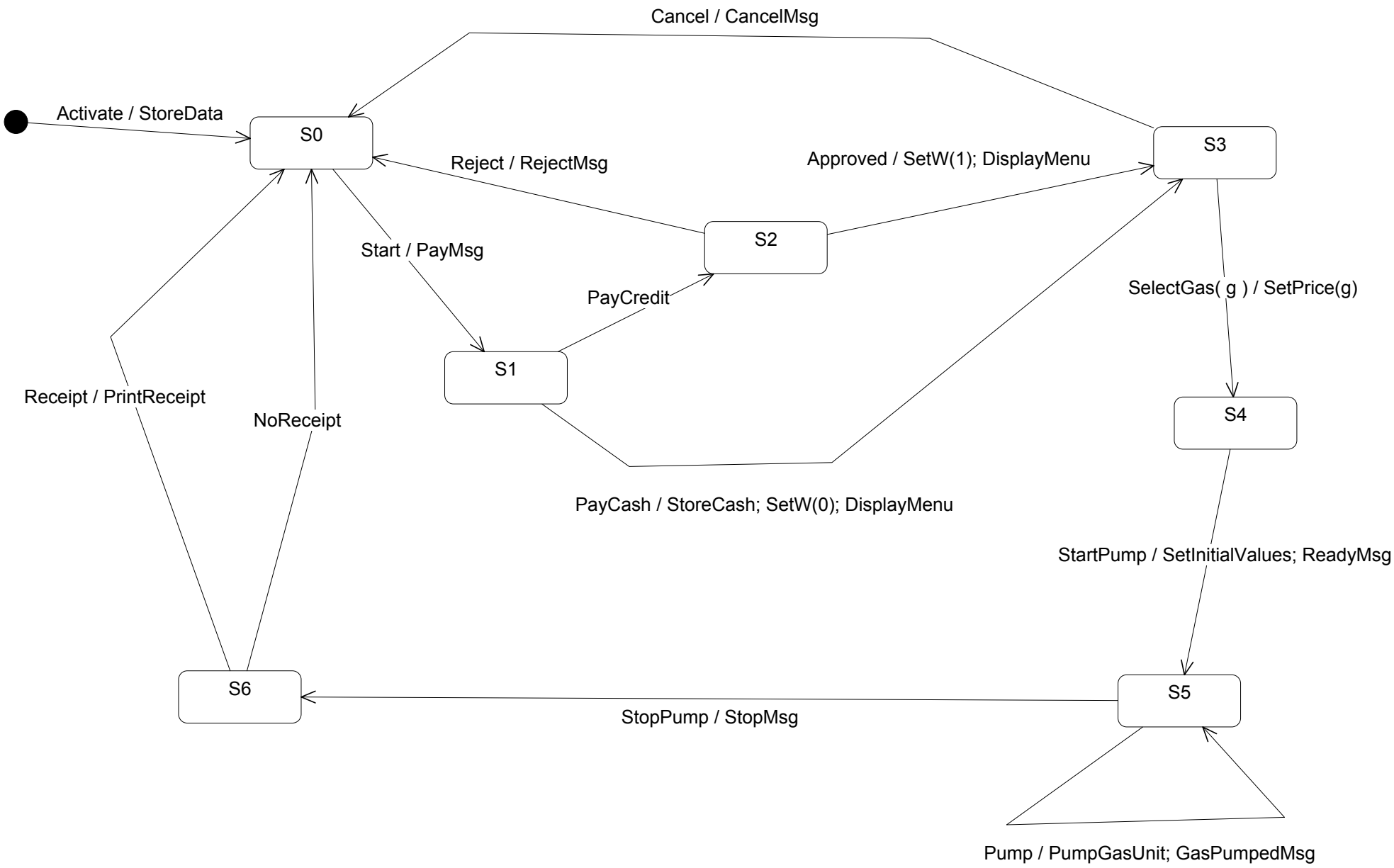
Notice:
m: is a pointer to the MDA-EFSM object
d: is a pointer to the Data Store object

Operations of the Input Processor
(GasPump-3)

```
Activate(float a, float b) {
        if ((a>0)&&(b>0)) {
              d->temp_a=a;
              d->temp_b=b;
              m->Activate()
        }
}

Start() {
        m->Start();
}

PayCash(float c) {
        if (c>0) {
              d->temp_c=c;
              m->PayCash()
        }
}

Cancel() {
        m->Cancel();
}

Premium() {
        m->SelectGas(2);
}

Regular() {
        m->SelectGas(1);
}

StartPump() {
        m->StartPump();
}

PumpLiter() {
        if (d->cash<(d->L+1)*d->price)
              m->StopPump();
        else m->Pump()
}

StopPump() {
        m->StopPump();
}

Receipt() {
        m->Receipt();
}

NoReceipt() {
        m->NoReceipt();
}
```

Notice:
cash: contains the value of cash deposited
price: contains the price of the selected gas
L: contains the number of liters already pumped

cash , *L*, *price* are in the data store
m: is a pointer to the MDA-EFSM object
d: is a pointer to the Data Store object



MDA-EFSM for Gas Pumps