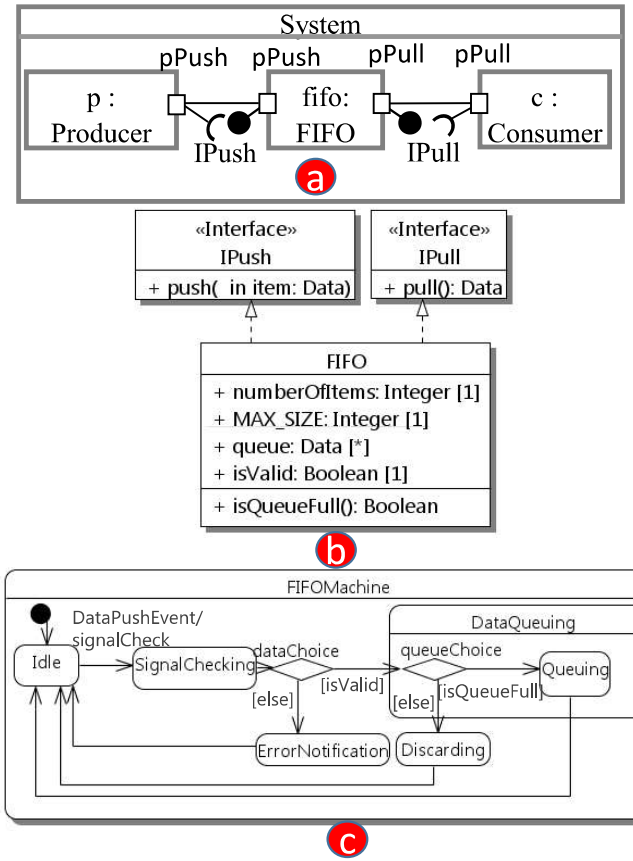


```
1. class System {
2. public:
3.   Producer p;
4.   Consumer c;
5.   FIFO fifo;
6.   void configuration(){
7.     bindPorts(p.pPush, fifo.pPush);
8.     bindPorts(c.pPull, fifo.pPull);
9.   }
10.}
11.class IPull {
12.public:
13.  virtual Data* pull() = 0;
14.}
15.class IPush {
16.public:
17.  virtual void push(Data& data) = 0;
18.}
19.class Producer {
20.public:
21.  RequiredPort<IPush> pPush;
22.};
23.class Consumer {
24.public:
25.  RequiredPort<IPull> pPull;
26.};
27.class FIFO : public IPush, IPull {
28.public:
29.  ProvidedPort<IPush> pPush;
30.  ProvidedPort<IPull> pPull;
31.  Data* pull(){//fine-grained code}
32.  void push(Data& data){//..}
33.  //attributes + methods…
```

```
34.Statemachine FIFOMachine {
35.  InitialState Idle;
36.  State SignalChecking {
37.    StateEntry entryCheck;
38.    StateExit exitCheck;
39.  };
40.  State ErrorNotification {
41.    StateEntry entryError;
42.  };
43.  State DataQueuing {
44.    StateEntry entryQueue;
45.    PseudoChoice queueChoice;
46.    State Queuing;
47.  };
48.  State Discarding;
49.  PseudoChoice dataChoice;
50.  CallEvent(DataPushEvent,push(Data&));
51.  TransitionTable {
52.    ExT(Idle,SignalChecking,
53.        DataPushEvent,NULL,signalCheck);
54.    ExT(SignalChecking,dataChoice,
55.                  NULL,NULL,NULL);
56.    ExT(dataChoice,queueChoice,
57.                  NULL,Valid,NULL);
58.  }
59.};
60.void entryCheck(){//fine-grained code}
61.void exitCheck(){//fine-grained code}
62.void entryError(){//fine-grained code}
63.void signalCheck(Data& item) {
64.//trans effect from Idle to SignalChecking
65.}
66.}
```

d