



Design Document

For Insulin Pump
(Version 1.1)

Written by:
Morteza Yousef-Sanati
Hamid Mohammad-Gholizadeh

Assignment 2

Course: 703 Software Design
Professor: Dr. Tom Maibaum
March 2011

Table of Contents

I	INSULIN PUMP SPECIFICATIONS	3
	<i>What is an insulin pump?</i>	<i>3</i>
	<i>How big is an insulin pump?</i>	<i>3</i>
	<i>How does an insulin pump work?</i>	<i>3</i>
	<i>How common is an insulin pump?</i>	<i>3</i>
II	DESIGN DECISION.....	5
II.1	SYSTEM DESIGN ARCHITECTURE.....	5
II.1.1	GUI.....	5
II.1.2	Controller	6
II.1.3	Data Layer.....	6
III	OOM DIAGRAMS.....	7
III.1	MODEL LEVEL DIAGRAMS	7
III.1.1	Diagram PackageDiagram_1.....	7
III.1.2	Diagram UseCaseDiagram_1	8
III.2	PACKAGE DIAGRAMS CONTROLLER	9
III.2.1	Diagram ACTButtonStateChart.....	9
III.2.2	Diagram Controler ClassDiagram	10
III.2.3	Diagram lockPump.....	11
III.2.4	Diagram lockPumpCollaboration.....	12
III.2.5	Diagram Pump State Chart	13
III.2.6	Diagram reservoir_Sate_chart.....	14
III.2.7	Diagram resumePump	15
III.2.8	Diagram resumePumpCollaboration.....	16
III.2.9	Diagram setBasal	17
III.2.10	Diagram setBasalCollaboration.....	18
III.2.11	Diagram setBolus	19
III.2.12	Diagram setBolusCollaboration.....	20
III.2.13	Diagram setDateTime	21
III.2.14	Diagram setDateTimeCollaboration.....	22
III.2.15	Diagram showPumpStatus	23
III.2.16	Diagram showPumpStatusCollaboration.....	24
III.2.17	Diagram suspending PumpCollaboration.....	25
III.2.18	Diagram SuspendingPump.....	26
III.2.19	Diagram unlockPump	27
III.2.20	Diagram unlockPumpCollaboration.....	28
III.3	PACKAGE DIAGRAMS DATA.....	29
III.3.1	Diagram DataLayerClassDiagram.....	29
III.4	PACKAGE DIAGRAMS GUI.....	30
III.4.1	Diagram GUI Class Diagram	30
III.4.2	Diagram Pump Simulator User Interface.....	31

I Insulin Pump Specifications

What is an insulin pump?

The most recently available advance in insulin delivery is the insulin pump. In the United States, MiniMed and Disetronic market the insulin pump. An insulin pump is composed of a pump reservoir similar to that of an insulin cartridge, a battery-operated pump, and a computer chip that allows the user to control the exact amount of insulin being delivered.

How big is an insulin pump?

Currently, pumps on the market are about the size of a standard communications beeper.

How does an insulin pump work?

The pump is attached to a thin plastic tube (an infusion set) that has a soft cannula (or plastic needle) at the end through which insulin passes. This cannula is inserted under the skin, usually on the abdomen. The cannula is changed every two days. The tubing can be disconnected from the pump while showering or swimming. The pump is used for continuous insulin delivery, 24 hours a day. The amount of insulin is programmed and is administered at a constant rate (basal rate). Often, the amount of insulin needed over the course of 24 hours varies depending on factors like exercise, activity level, and sleep.

The insulin pump allows the user to program many different basal rates to allow for variation in lifestyle. In addition, the user can program the pump to deliver a bolus (large dose of insulin) during meals to cover the excess demands of carbohydrate ingestion.

How common is an insulin pump?

Over 50,000 people with diabetes worldwide are using an insulin pump. This number is growing dramatically as these devices become smaller and more user-friendly. Insulin pumps allow for tight blood sugar control and lifestyle flexibility while minimizing the effects of low blood sugar (hypoglycemia). At present, the pump is the closest device on the market to an artificial pancreas. More recently, newer models of the pump have been developed that do not require a tubing, in fact - the insulin delivery device is placed directly on the skin and any adjustments needed for insulin delivery are made through a PDA like device that must be kept within a 6 foot range of the insulin delivery device, and can be worn in a pocket, kept in a purse, or on a tabletop when working.

Probably the most exciting innovation in pump technology is the ability to use the pump in tandem with newer glucose sensing technology. Glucose sensors have improved dramatically in the last few years, and are an option for patients to gain further insight into their patterns of glucose response to tailor a more individual treatment regimen. The newest generation of sensors allows for a real time glucose value to be given to the patient. The implantable sensor communicates wirelessly with a pager-sized device that has a screen. The device is kept in proximity to the sensor to allow for transfer of data, however, it can be a few feet away and still receive transmitted information. Depending on the model, the screen displays the blood glucose reading, a thread of readings over time, and a potential rate of change in the glucose values. The sensors can be programmed to produce a "beep" if blood sugars are in a range that is selected as too high or too low. Some can provide a warning beep if the drop in blood sugar is occurring too quickly.

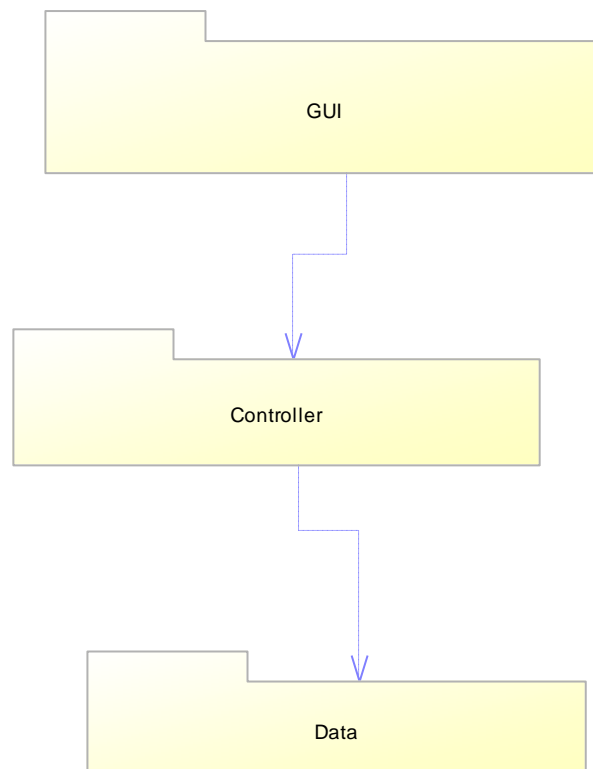
To take things one step further, there is one particular sensor that is new to the market that is designed to communicate directly with the insulin pump. While the pump does not yet respond directly to information from the sensor, it does "request" a response from the patient if there is a need for adjustments according to the patterns it is programmed to detect. The ultimate goal of this technology is to "close the loop" by continuously sensing what the body needs, and then responding by providing the appropriate dose of insulin. While this technology is a few more years in the making, the strides in this direction continue to grow.

II Design Decision

II.1 System Design Architecture

Our design is based on 3-tier architecture which is reflected by three packages: GUI, Controller and Data.

GUI includes classes related to GUI Design like, Frames, Buttons, Panels and etc. which is constructing our user interface. Controller includes core classes like classes which control the pump functionality and Data package contains a class which is responsible for storing and retrieving system data.



System Design Architecture

II.1.1 GUI

Our GUI completely reflects a real insulin pump interaction. Instead of using common GUI widgets like menus we decided to implement the menus just like the menus exist in a real insulin pump. This approach facilitates detaching the controller part of the software and embedding it in a real hardware with minimum effort.

As you will see in the Class diagram we exploited the **Command Pattern** to control the complexity of different states of the buttons of the interface. As you will see in the state chart diagrams for the ACT button, its functionality changes according to the place the user is interacting with. e.g. while the user is inside the menu the ACT button command differs

with the time the user setting the basal or bolus or even when the user want to set the date and time.

We also use **singleton pattern** for preventing some classes to have more than one instance throughout the life cycle of the system. e.g. ActBtn, UpBtn and other buttons in GUI. We should mention that we didn't indicate explicitly this pattern on these classes in class diagram. However it should be considered in implementation. We demonstrated each button as a child of a Button class setting their captions differently. It would be better to show these kinds of situations by using Object diagram and only change the value for that instances, But we avoided to use Object diagrams to avoid more complexity.

For changing the screen inside the frame, say the place we show information to the user, we will create one instance of a Panel class and will change according to each state of the pump.

II.1.2 Controller

We represented Battery and Reservoir with the corresponding classes in controller layer. These classes simulate the functionality of these hardwares like consuming the battery of decreasing the reservoir insulin.

The pump class acts as a core of a controller system. it almost controls and synchronize the activity of the pumps. There are some enumerations in control layer which represents different states of the system like "PumpState Enumeration". We used OCL constraint to specify the constraint of the system. these constraint are annotated in class diagrams elements.

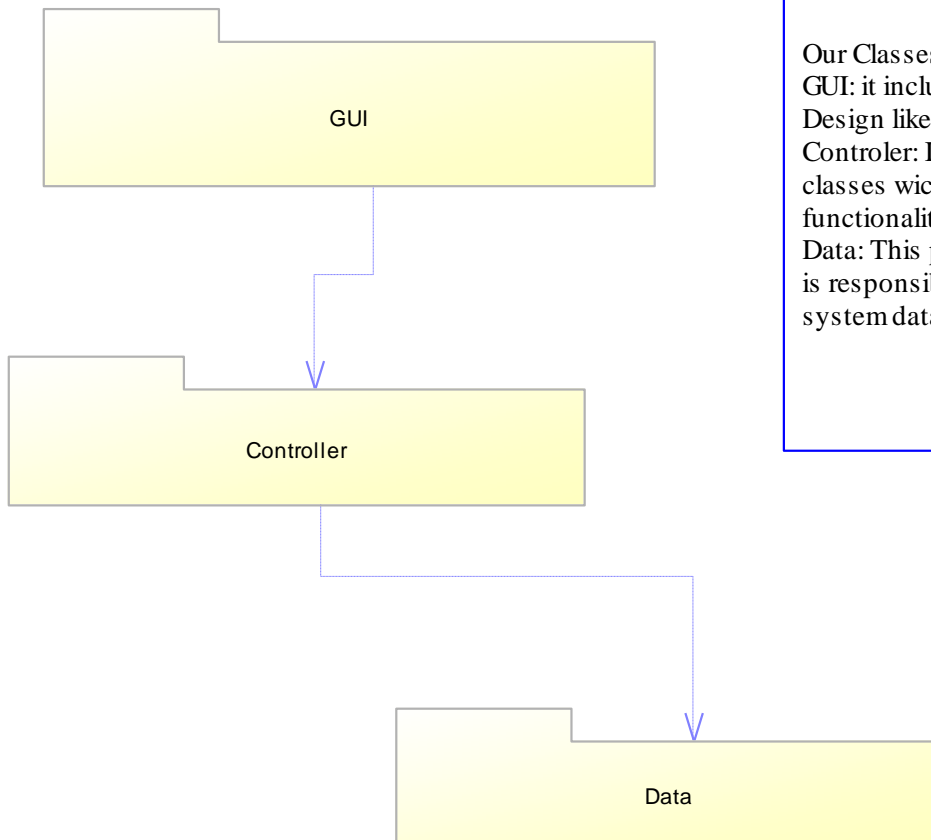
II.1.3 Data Layer

Our data layer is responsible for storing system data. Actually there is one class in this layer doing all data related storing and retrieving tasks. For example we need to store the battery level and reservoir level when the pump is turned off (program closed) and then retrieve these data again when it is turned on (program started).

III OOM Diagrams

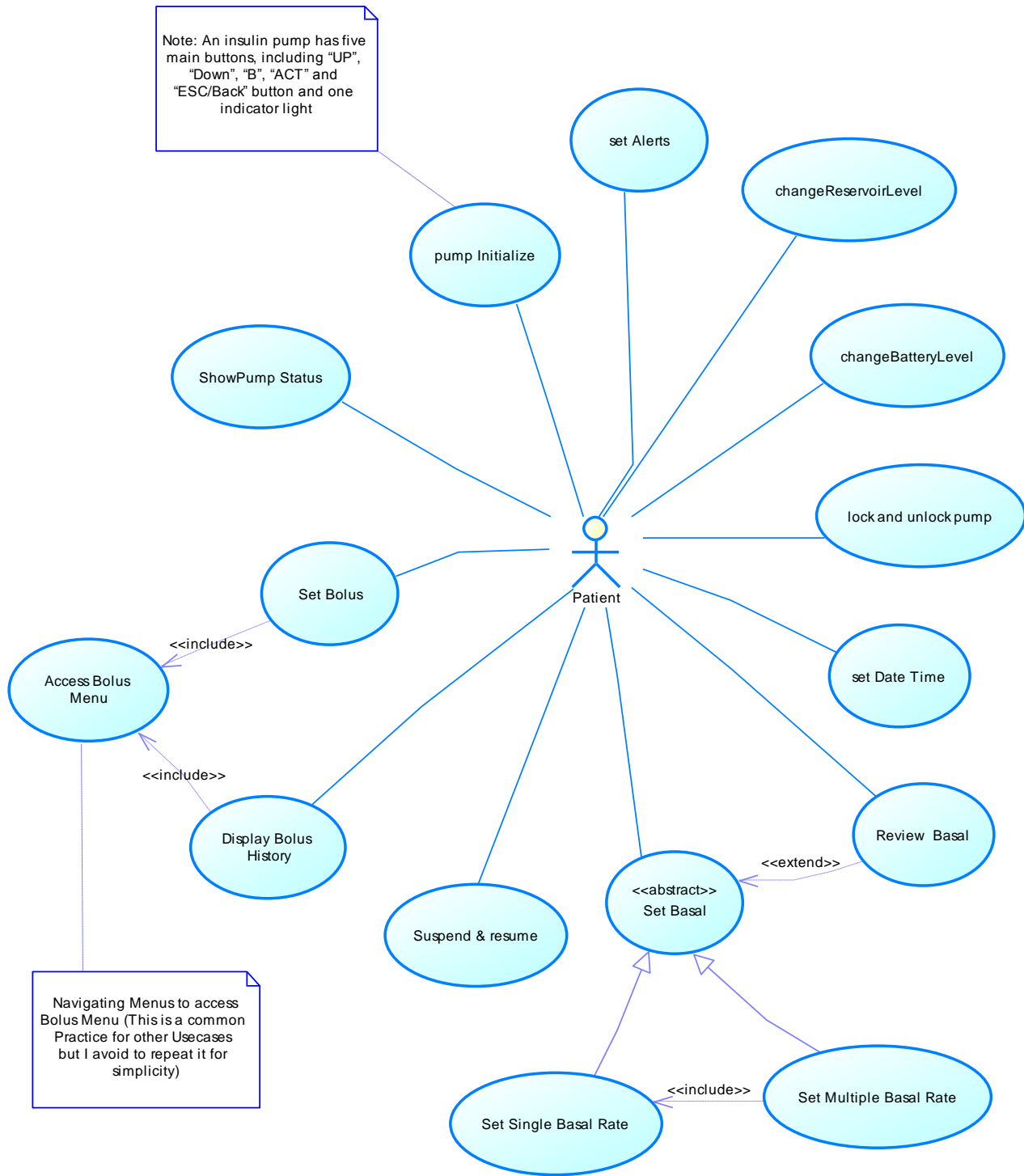
III.1 Model level diagrams

III.1.1 Diagram PackageDiagram_1



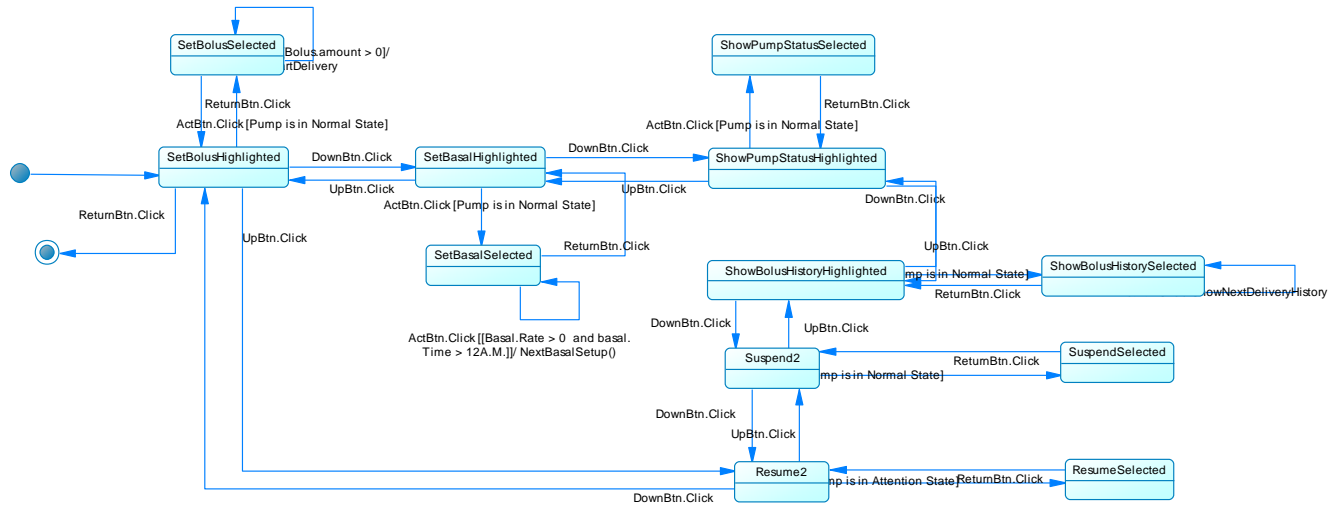
Our Classes is organized in three packages,
GUI: it includes classes related to GUI
Design like, Frames, Buttons, Panels and ...
Controller: It includes core classes like
classes wich control the pump
functionality
Data: This package contain a class which
is responsible for storing and retriving
system data

III.1.2 Diagram UseCaseDiagram_1

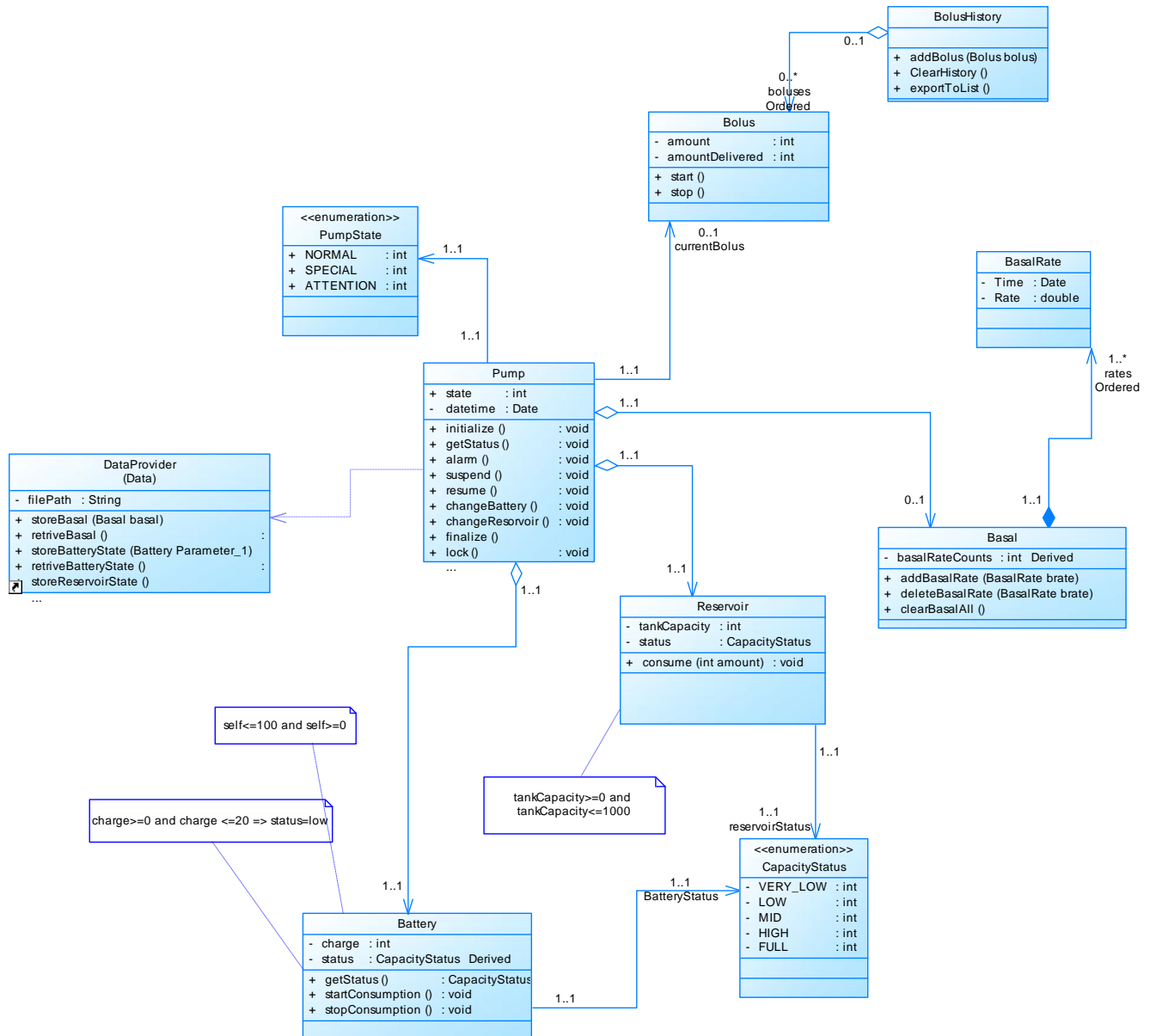


III.2 Package diagrams Controller

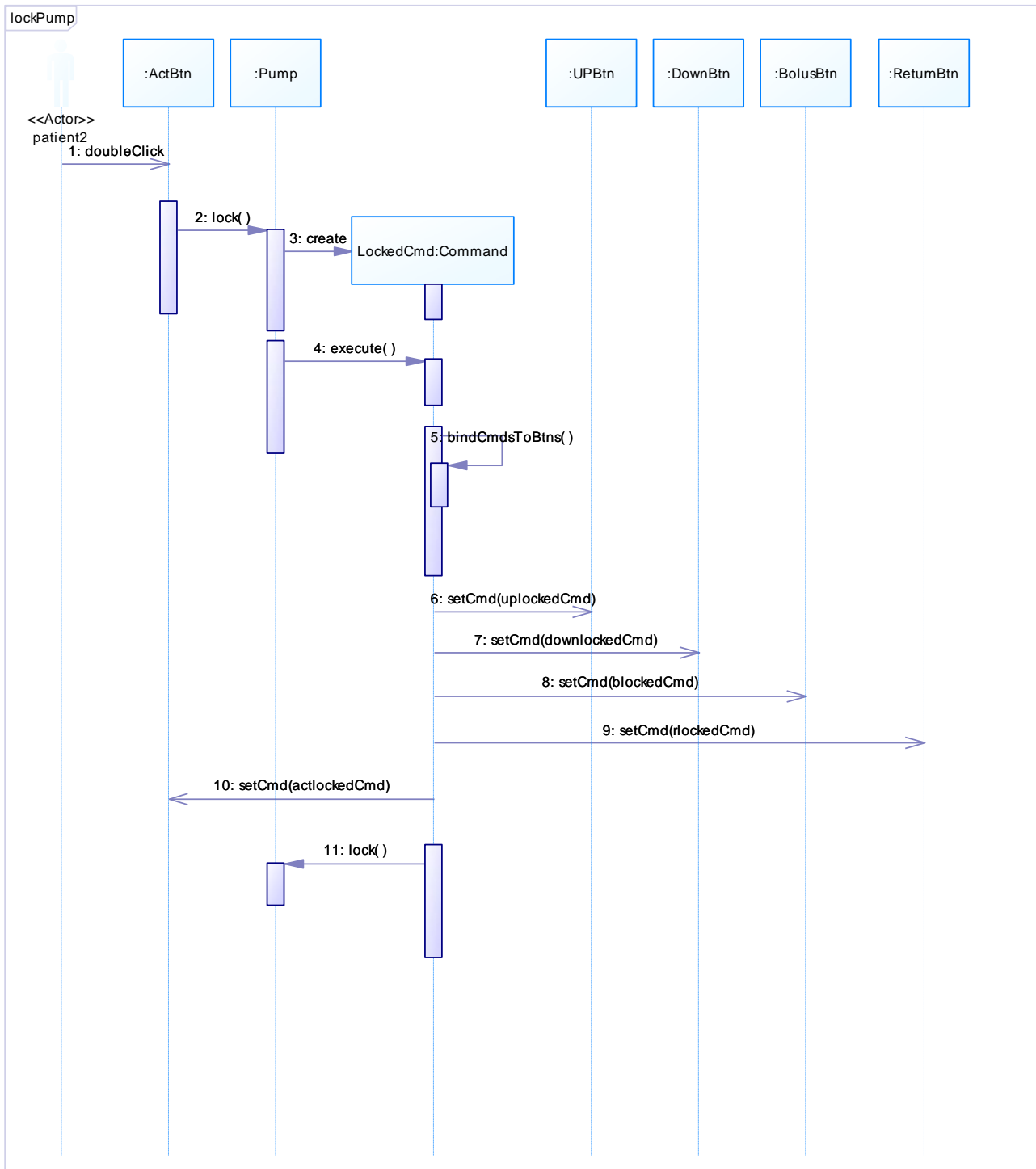
III.2.1 Diagram ACTButtonStateChart



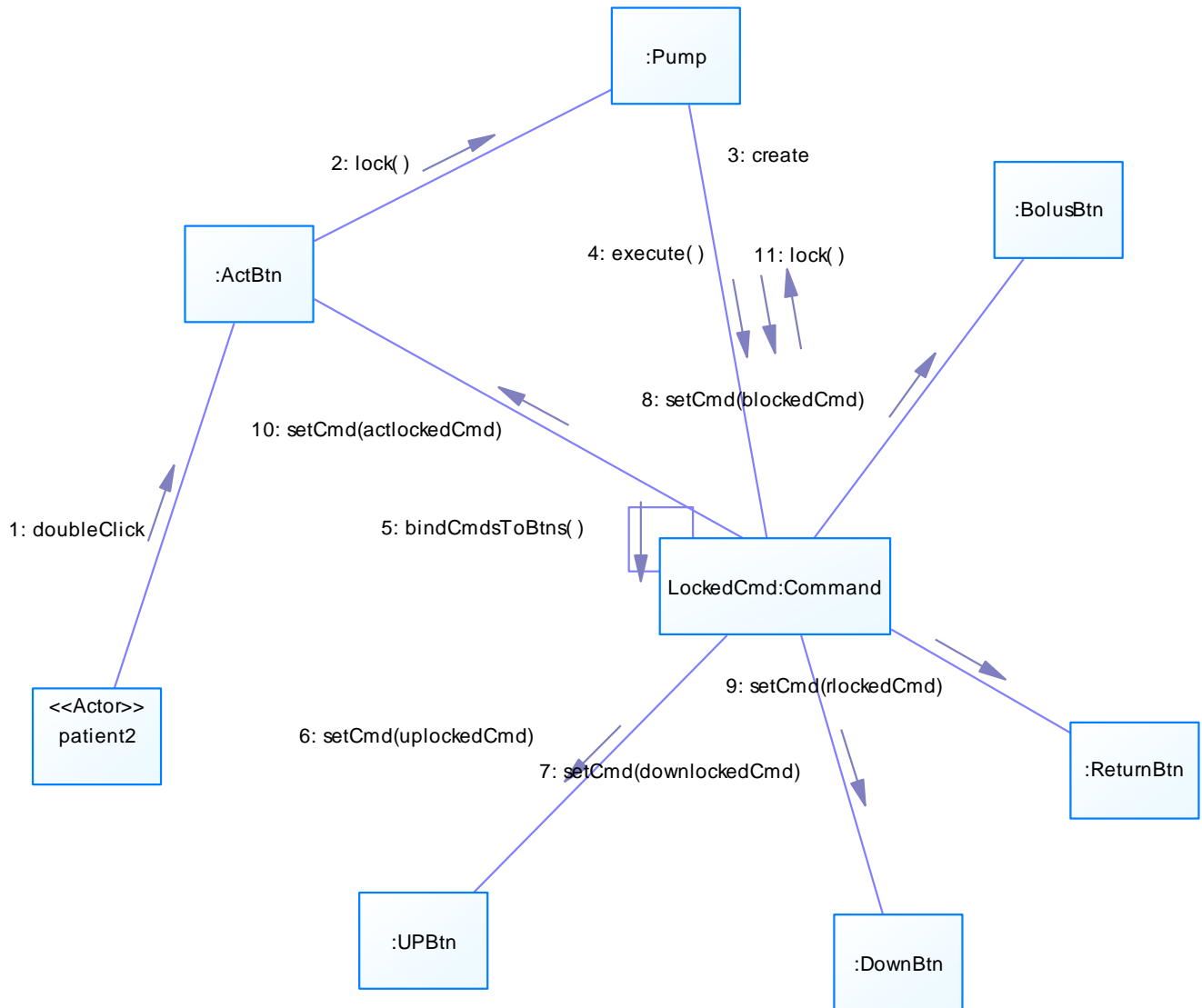
III.2.2 Diagram Controler ClassDiagram



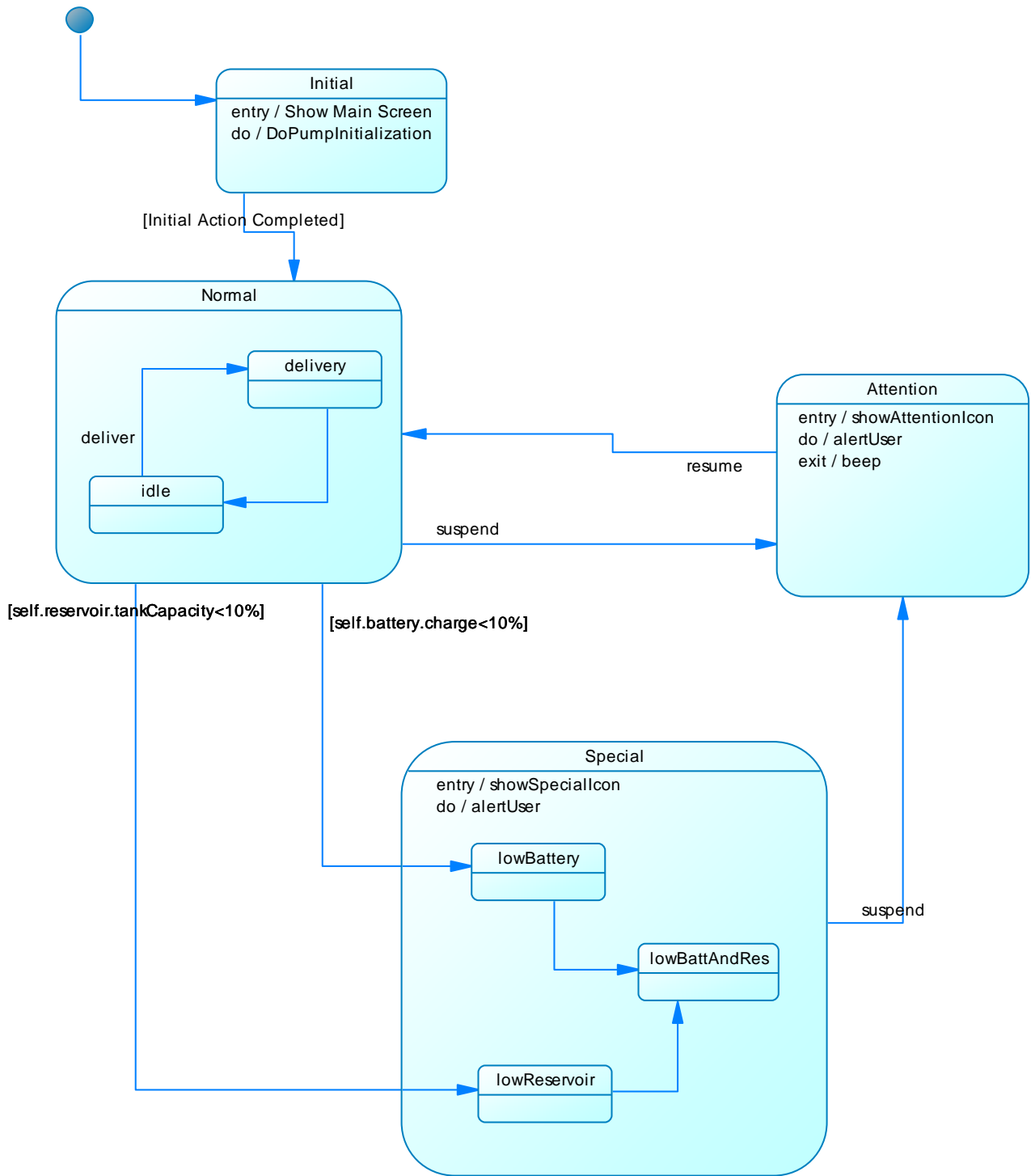
III.2.3 Diagram lockPump



III.2.4 Diagram lockPumpCollaboration

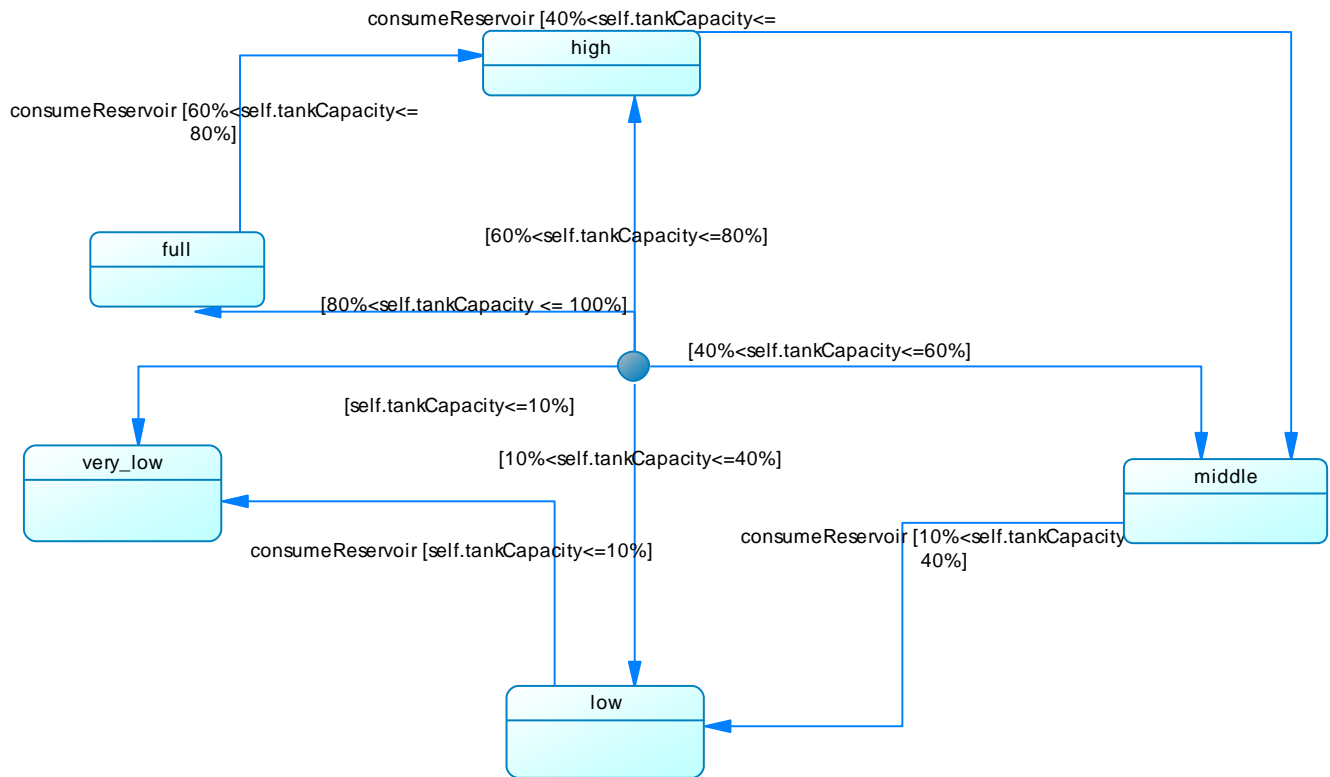


III.2.5 Diagram Pump State Chart

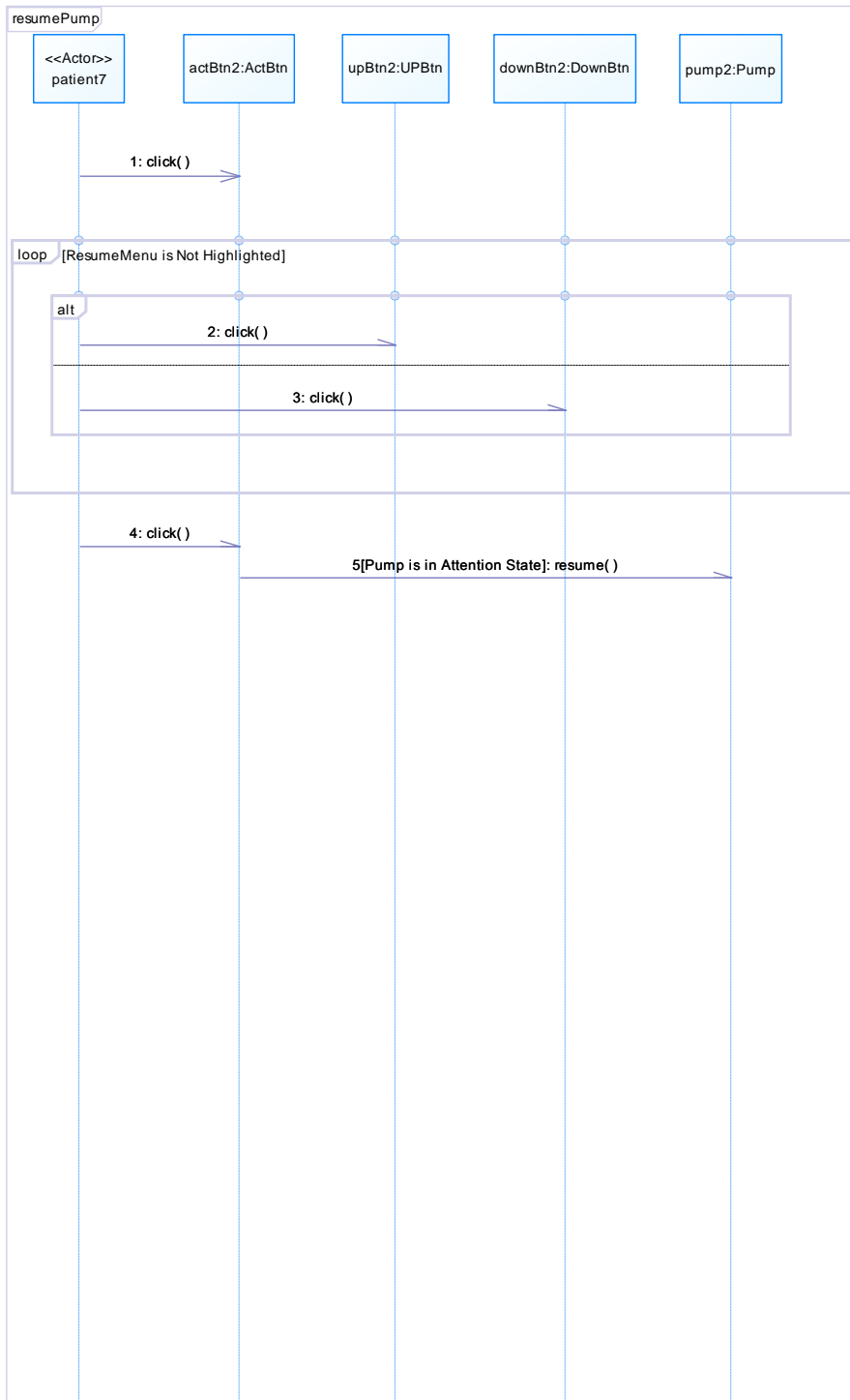


III.2.6 Diagram reservoir_Sate_chart

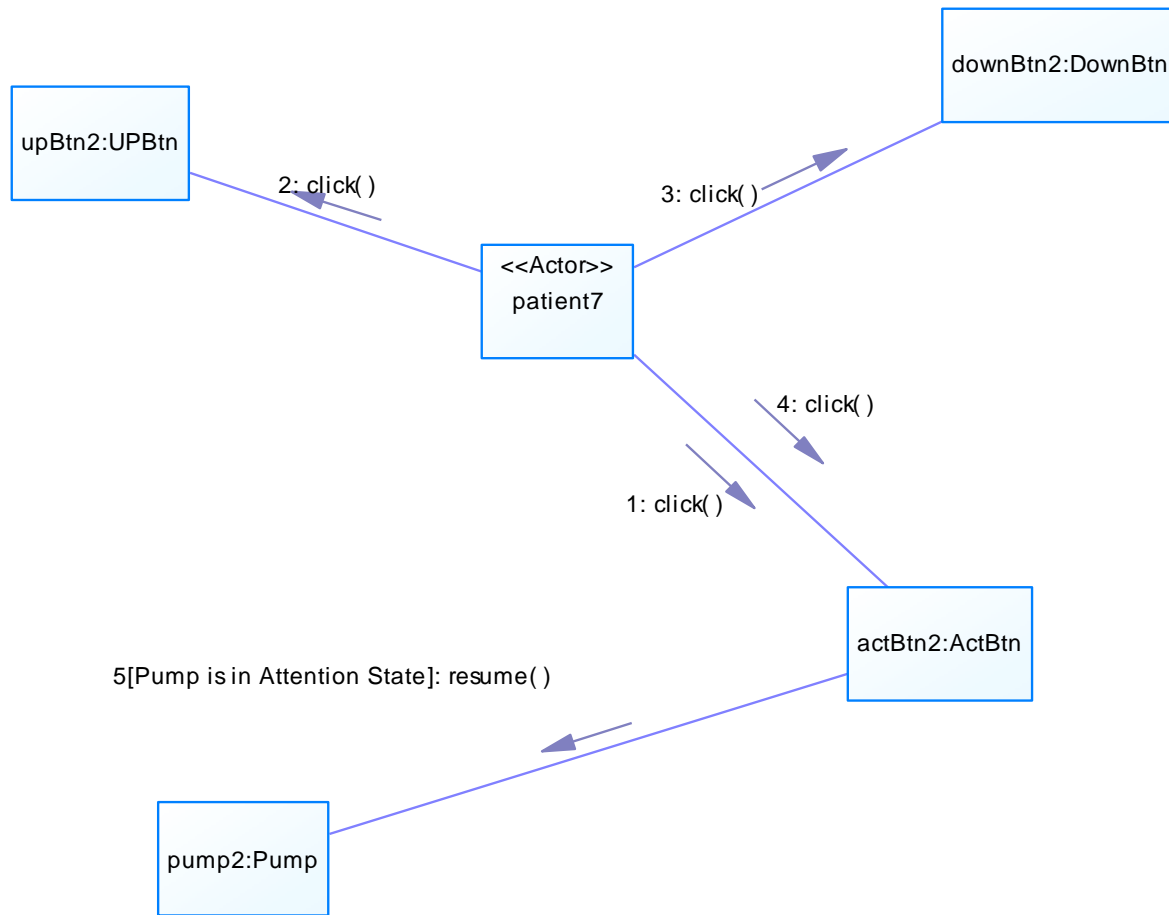
Reservoir of pump must be changed in order to get more capacity then there are not reverse transition from lower capacities to higher ones.



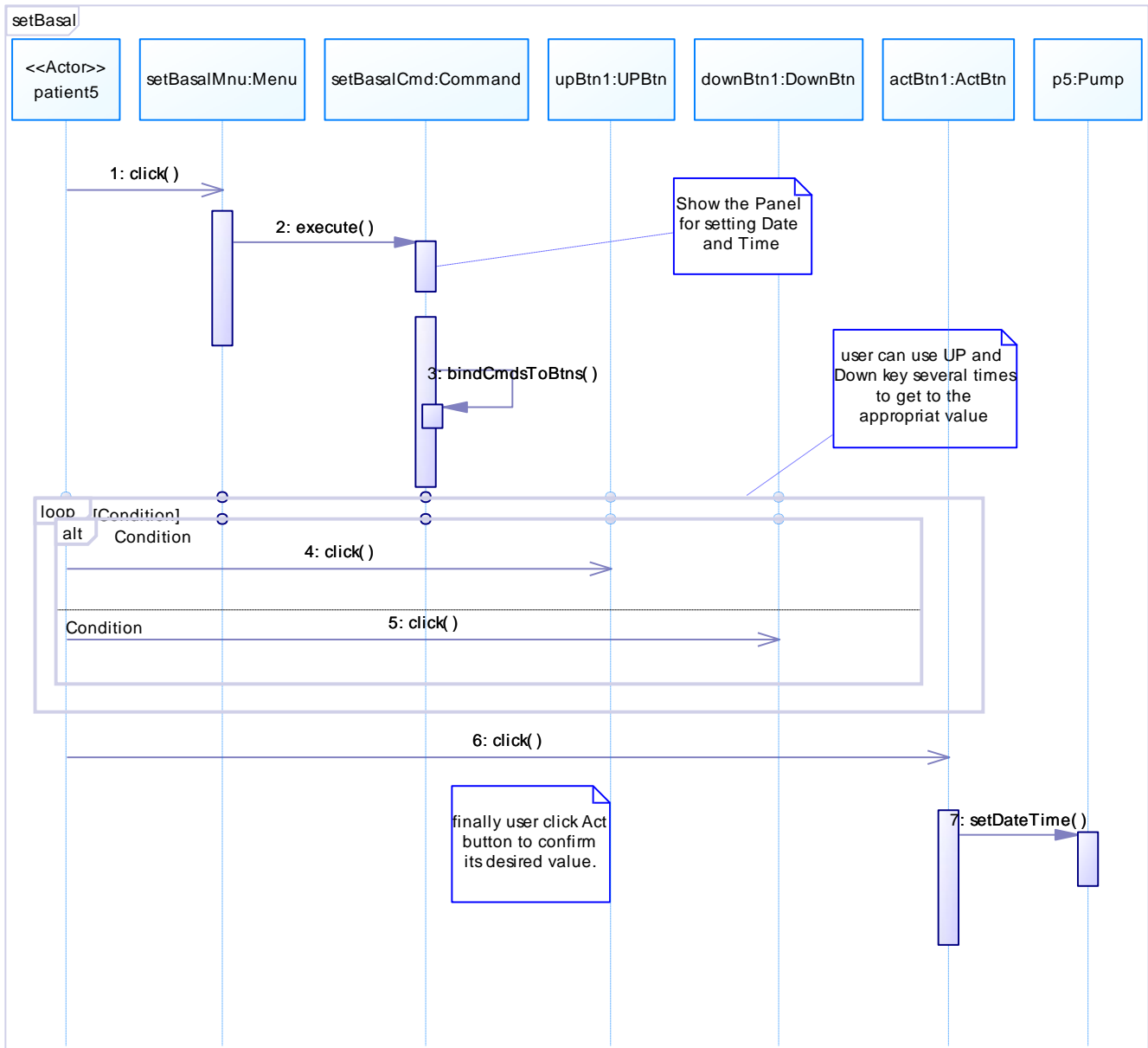
III.2.7 Diagram resumePump



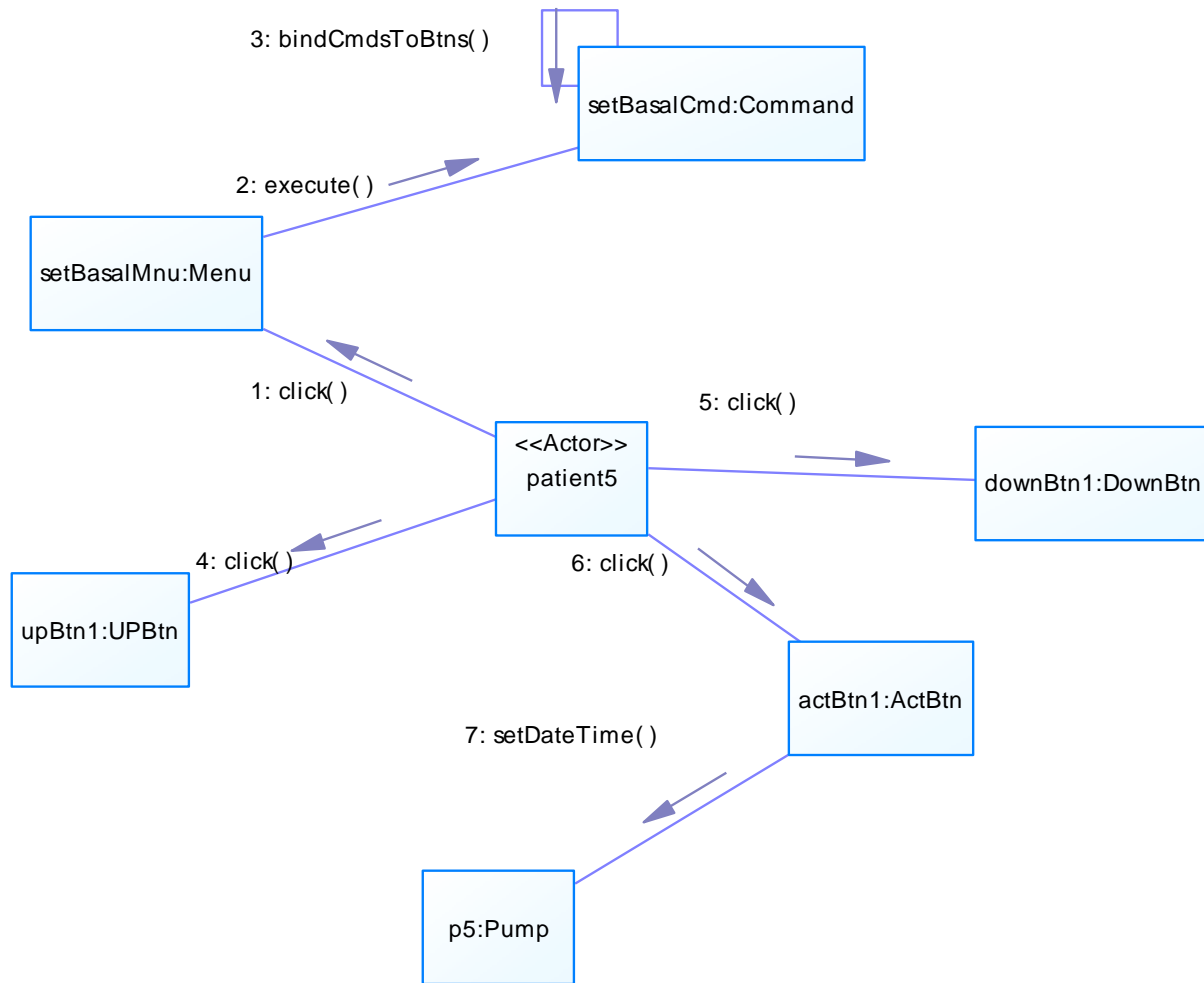
III.2.8 Diagram resumePumpCollaboration



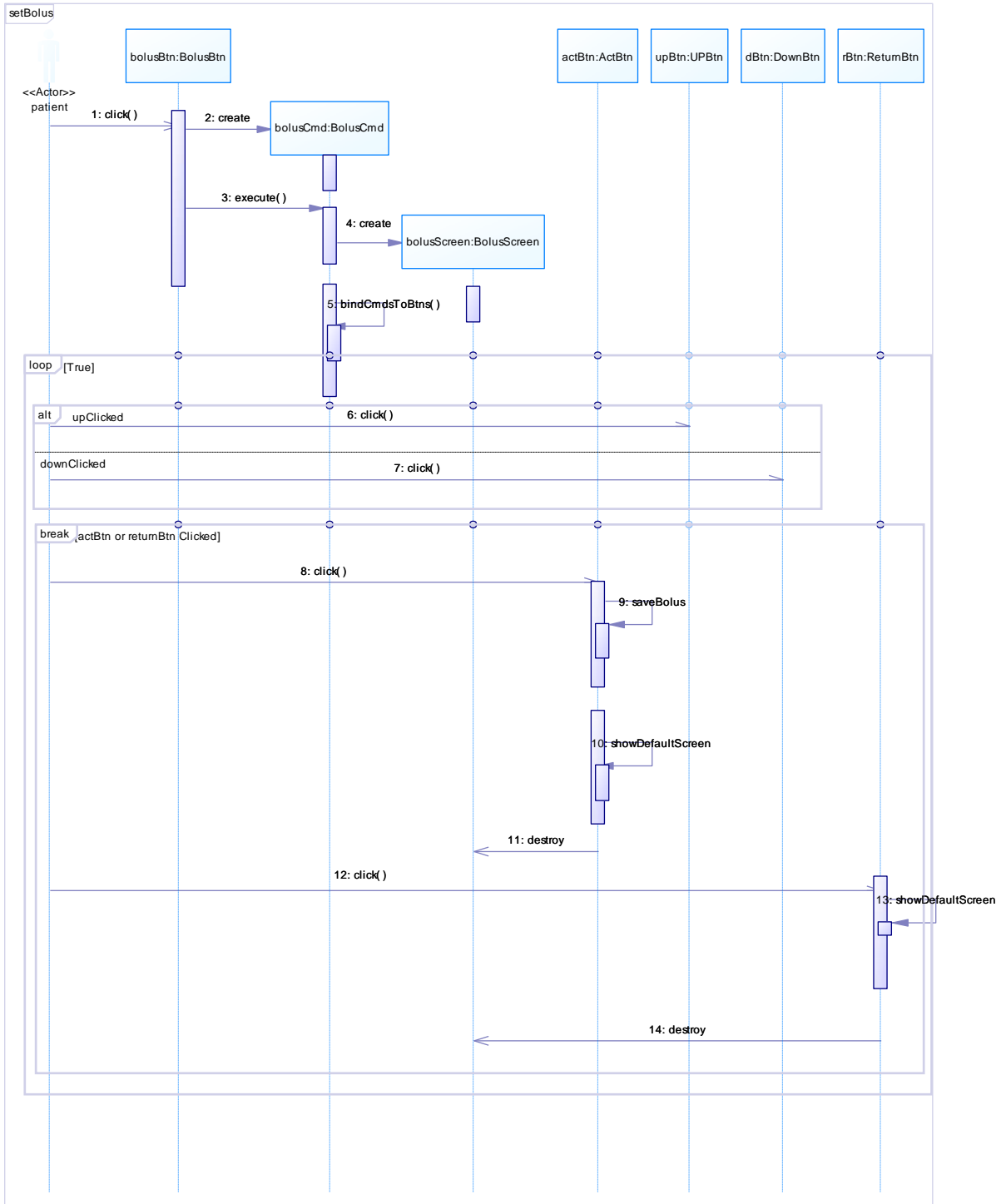
III.2.9 Diagram setBasal



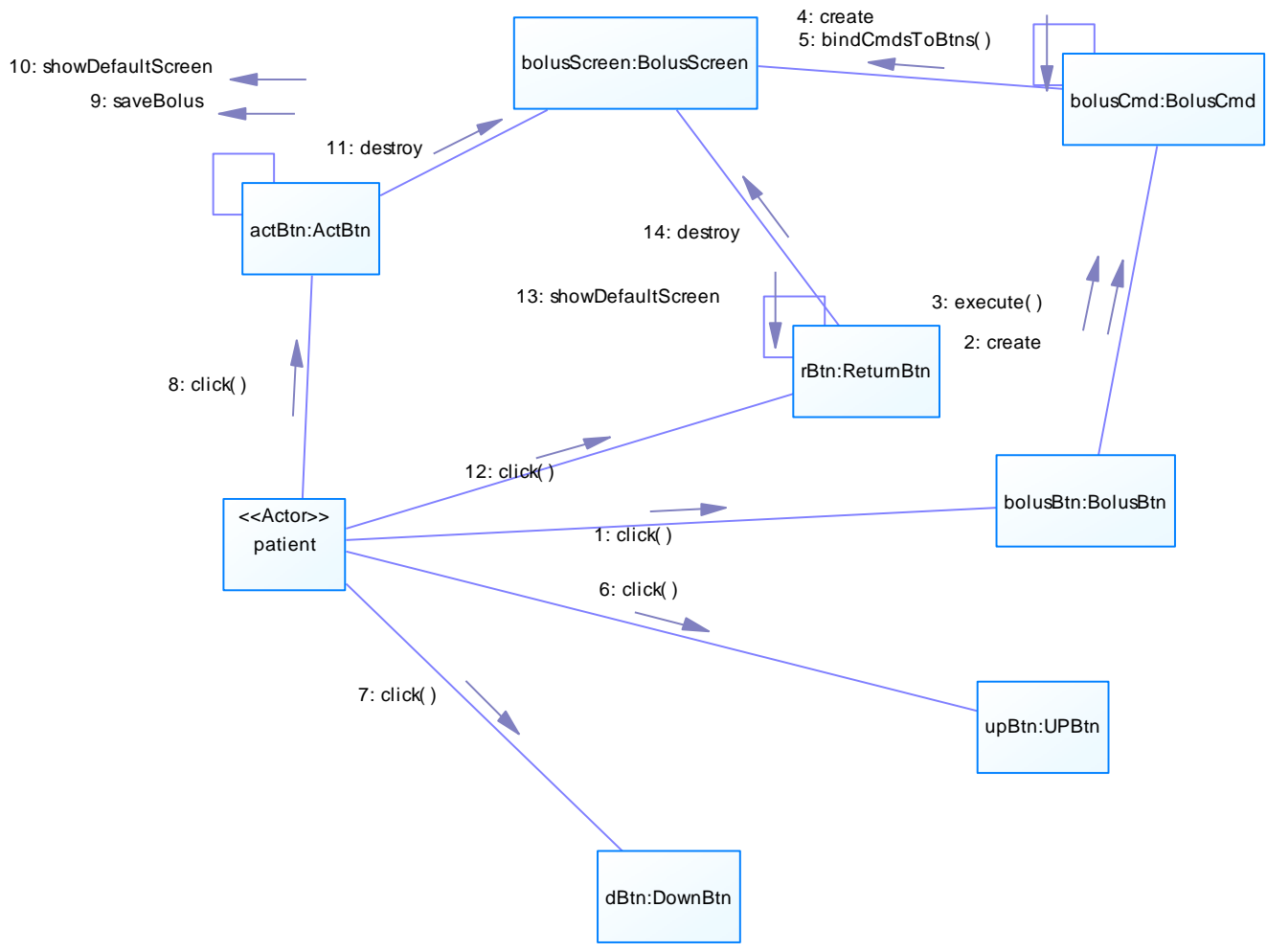
III.2.10 Diagram setBasalCollaboration



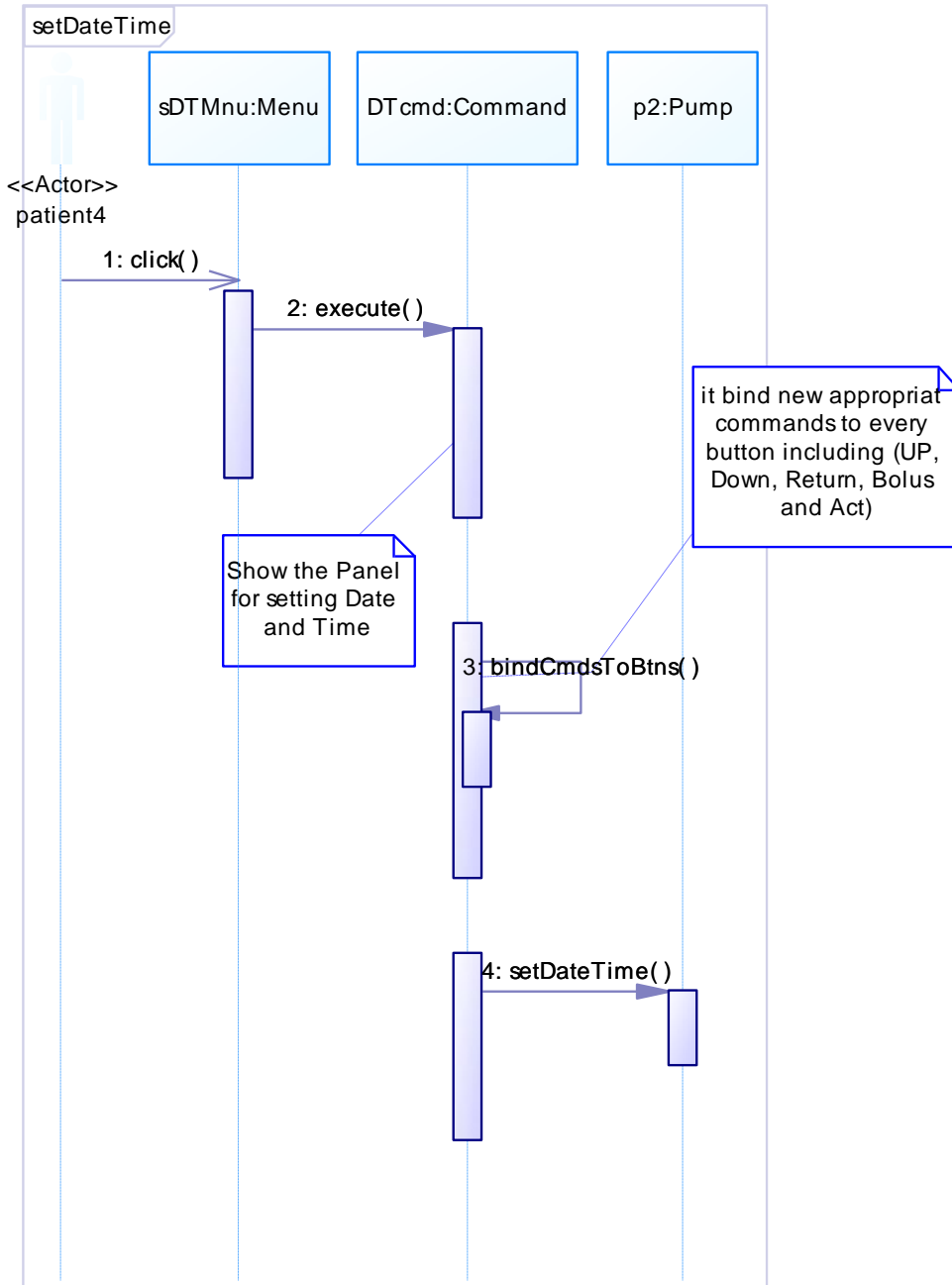
III.2.11 Diagram setBolus



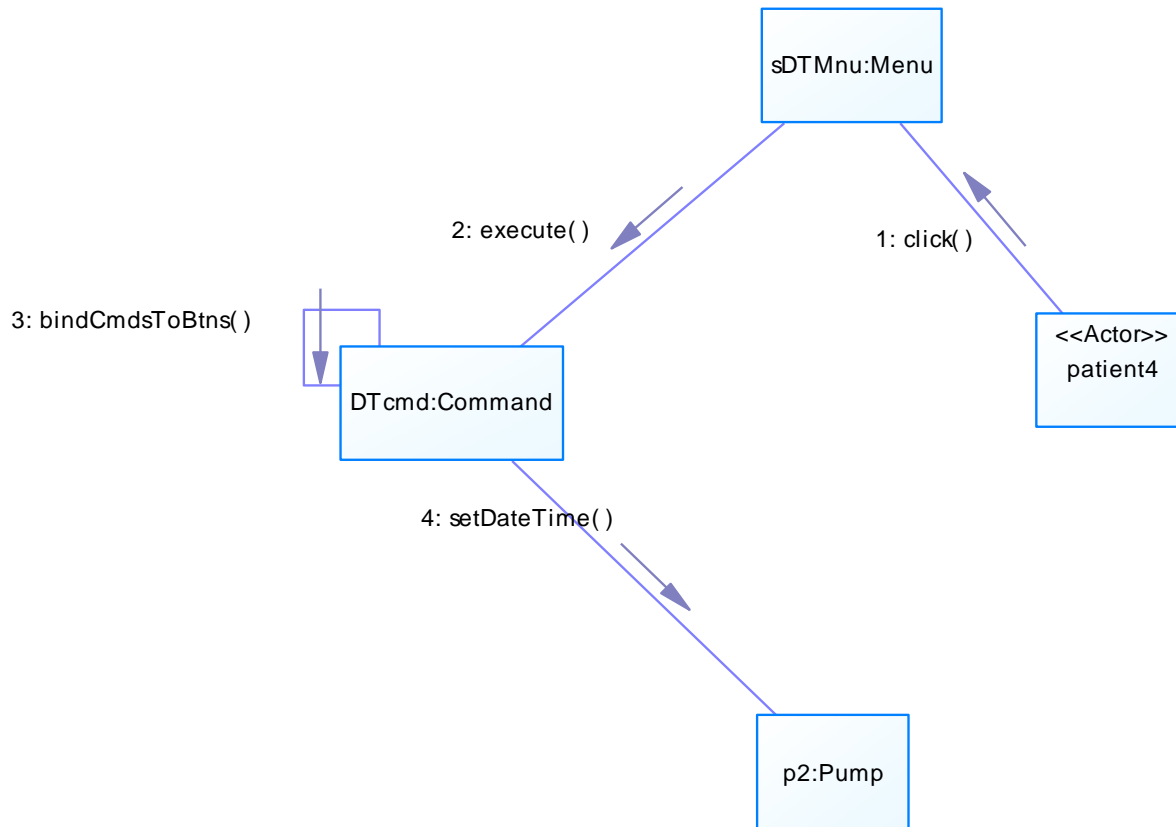
III.2.12 Diagram setBolusCollaboration



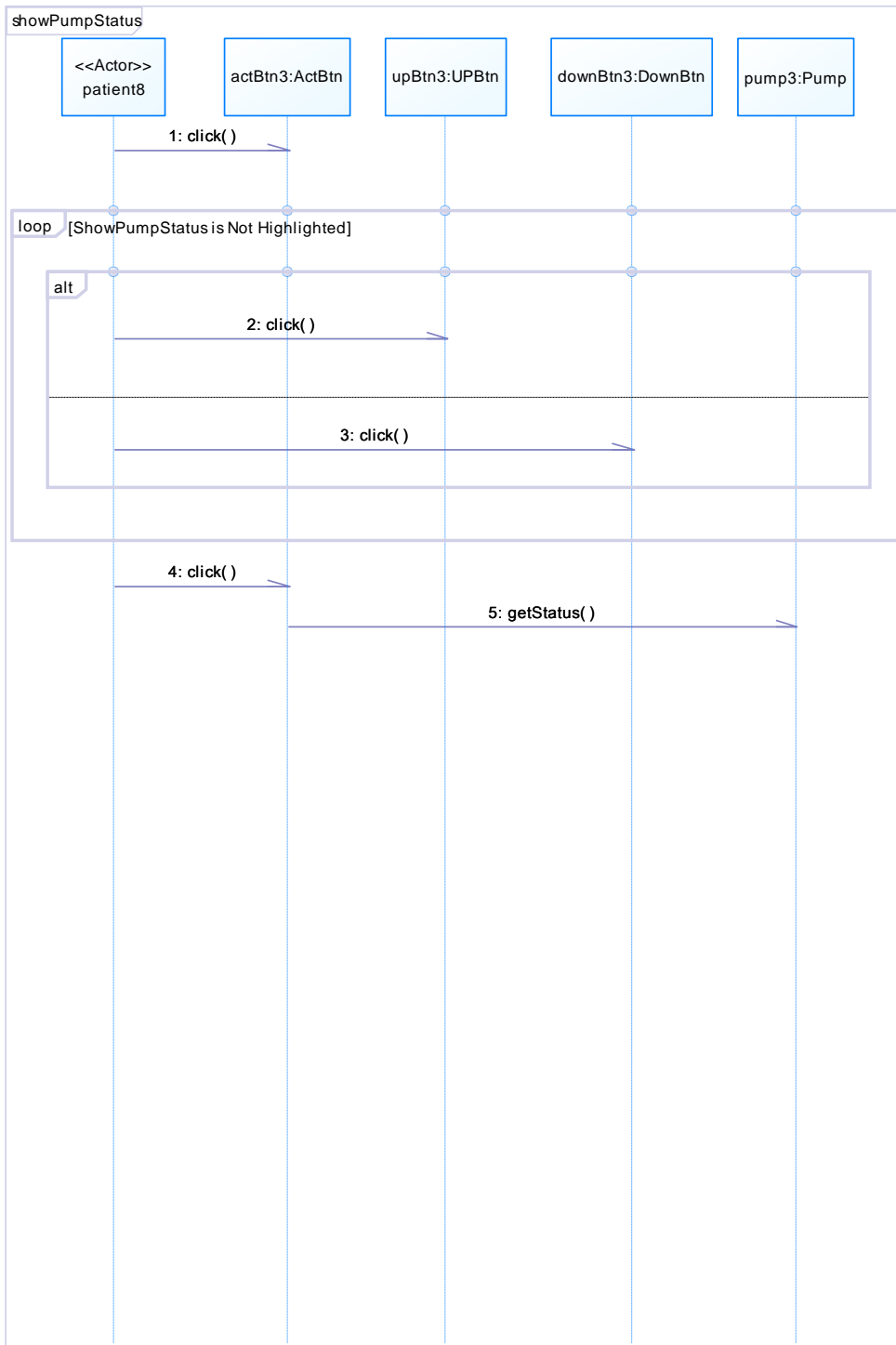
III.2.13 Diagram setDateTime



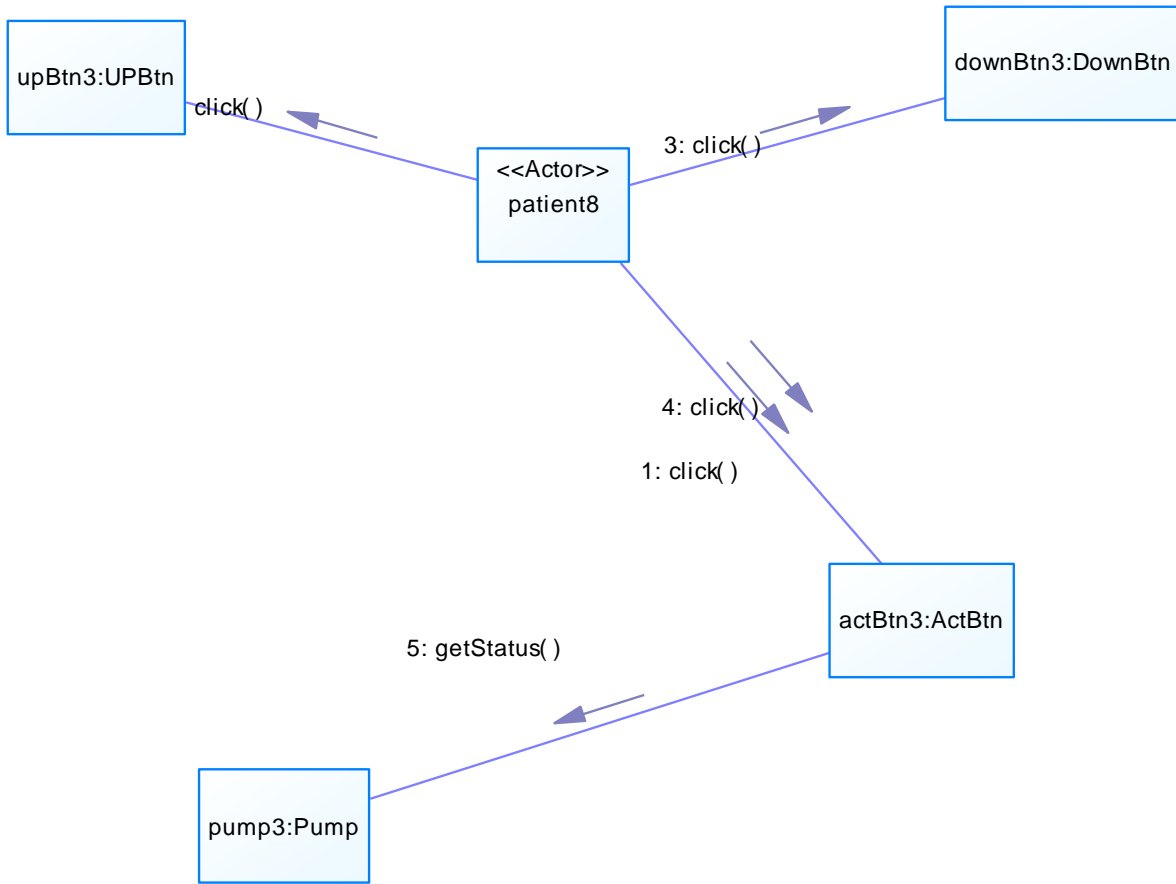
III.2.14 Diagram setDateTimeCollaboration



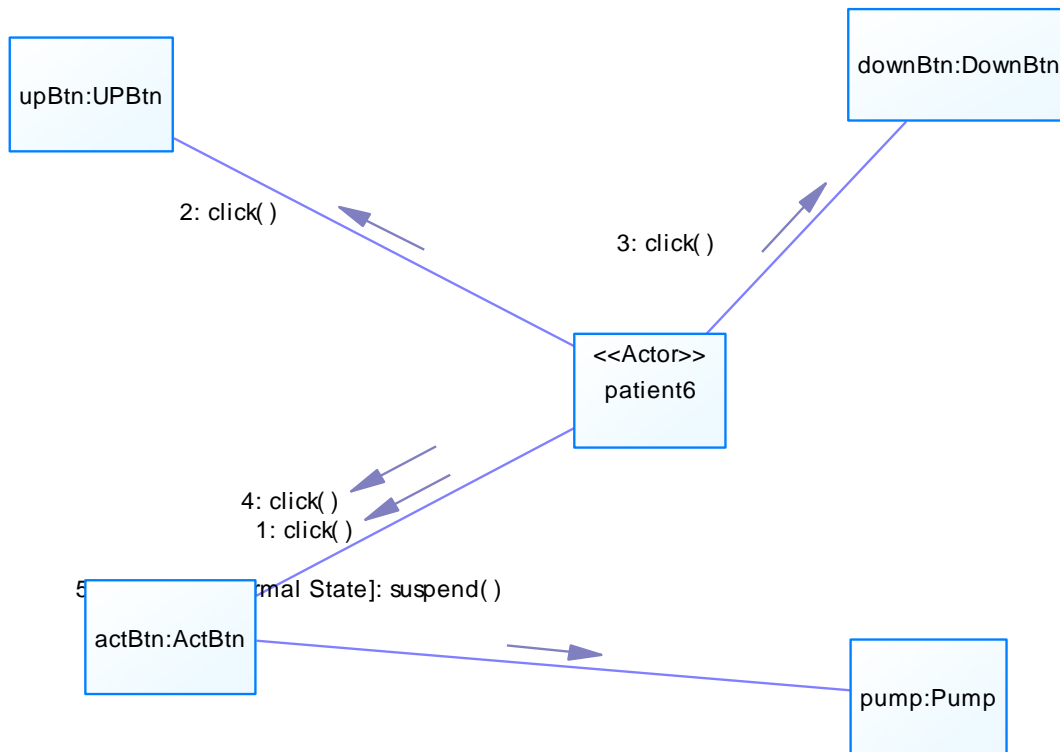
III.2.15 Diagram showPumpStatus



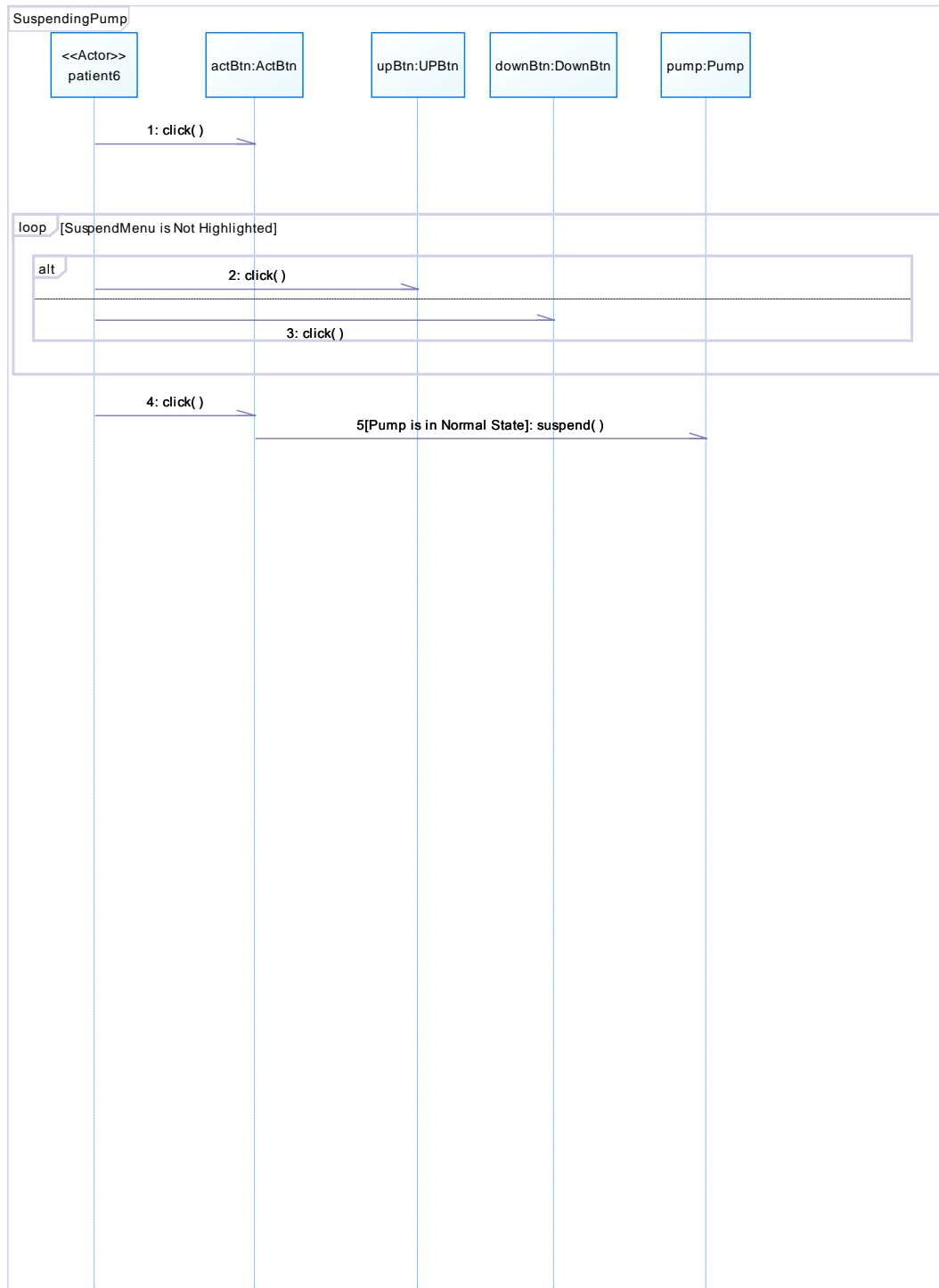
III.2.16 Diagram showPumpStatusCollaboration



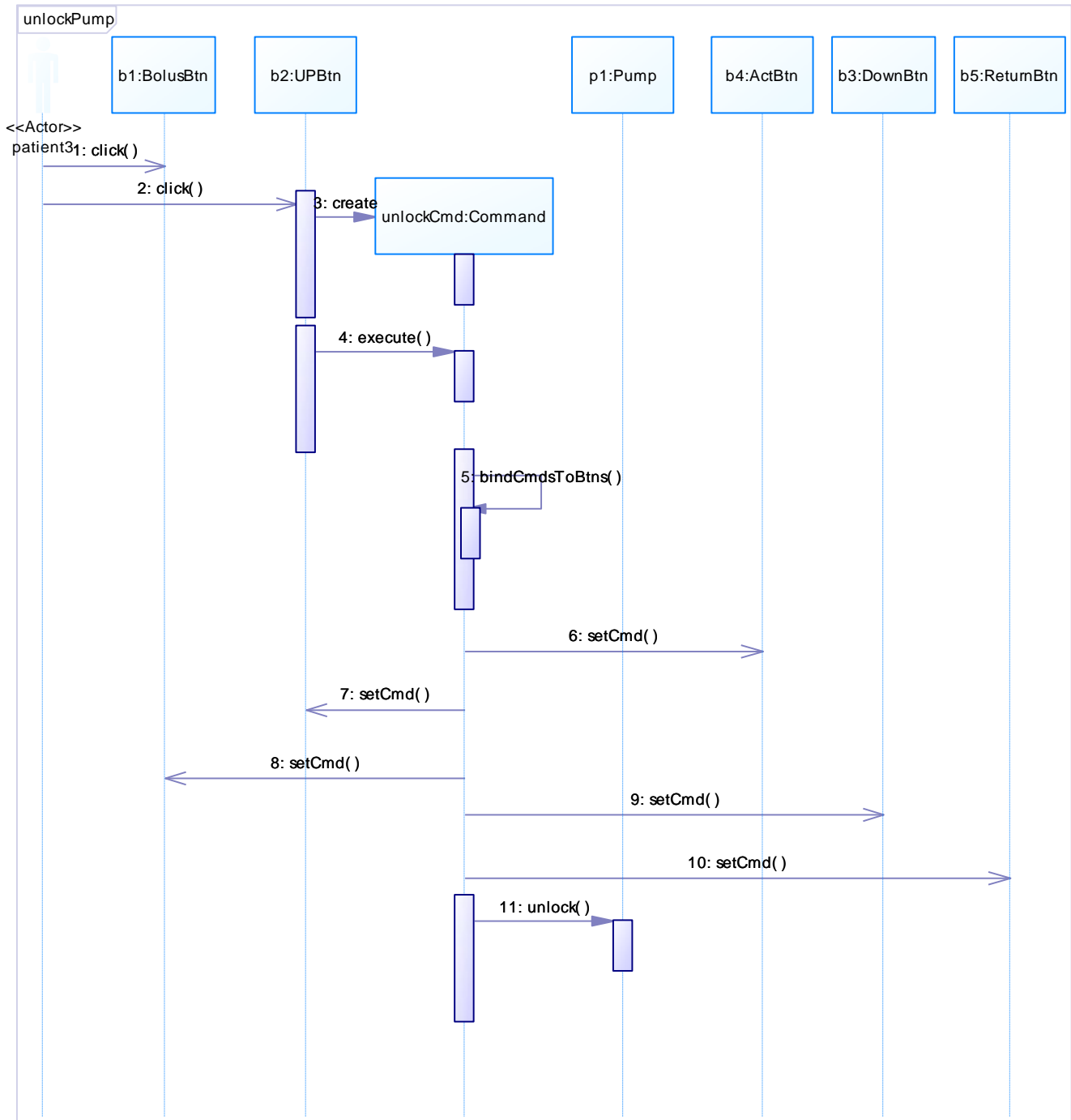
III.2.17 Diagram suspending PumpCollaboration



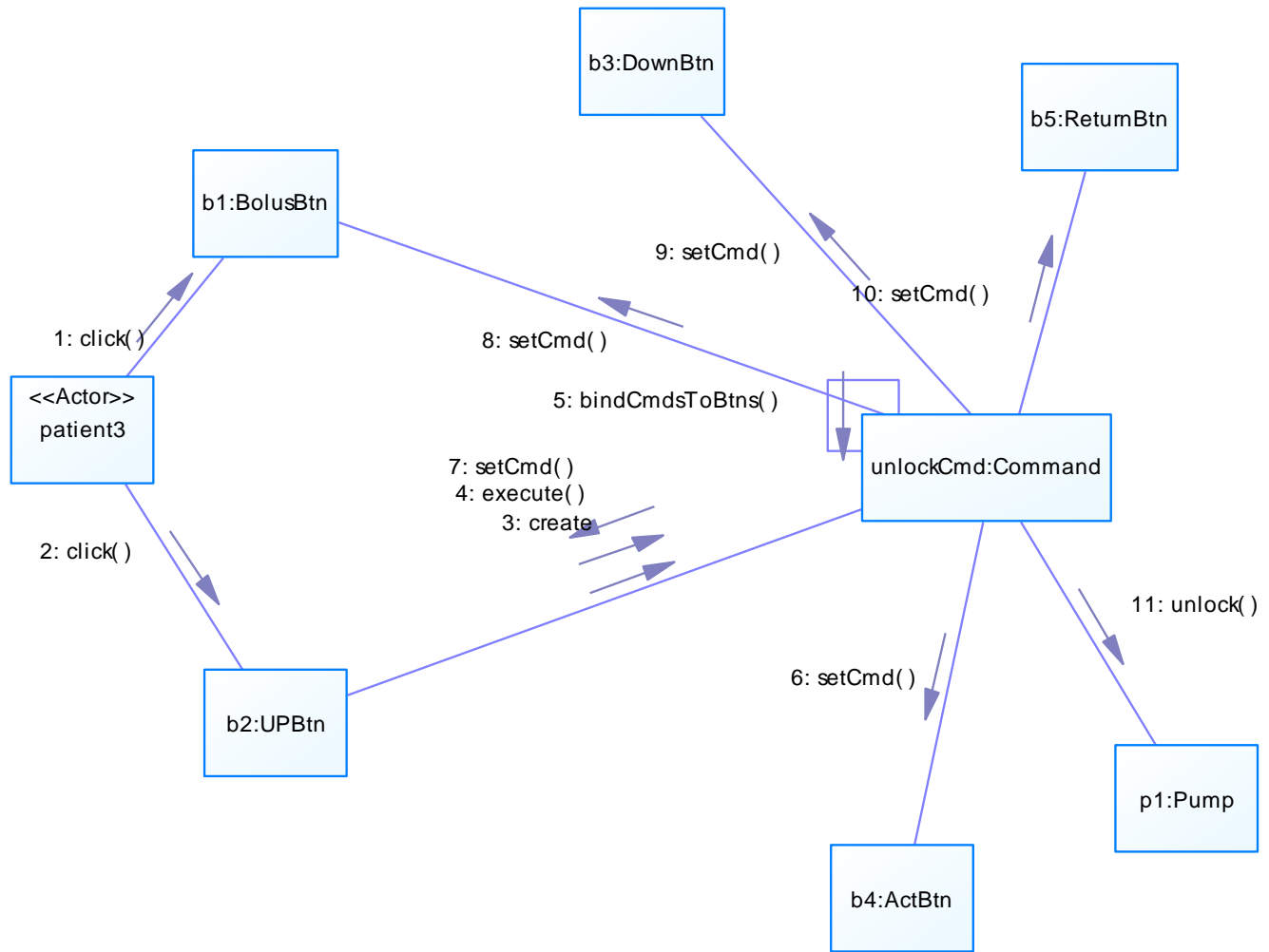
III.2.18 Diagram SuspendingPump



III.2.19 Diagram unlockPump

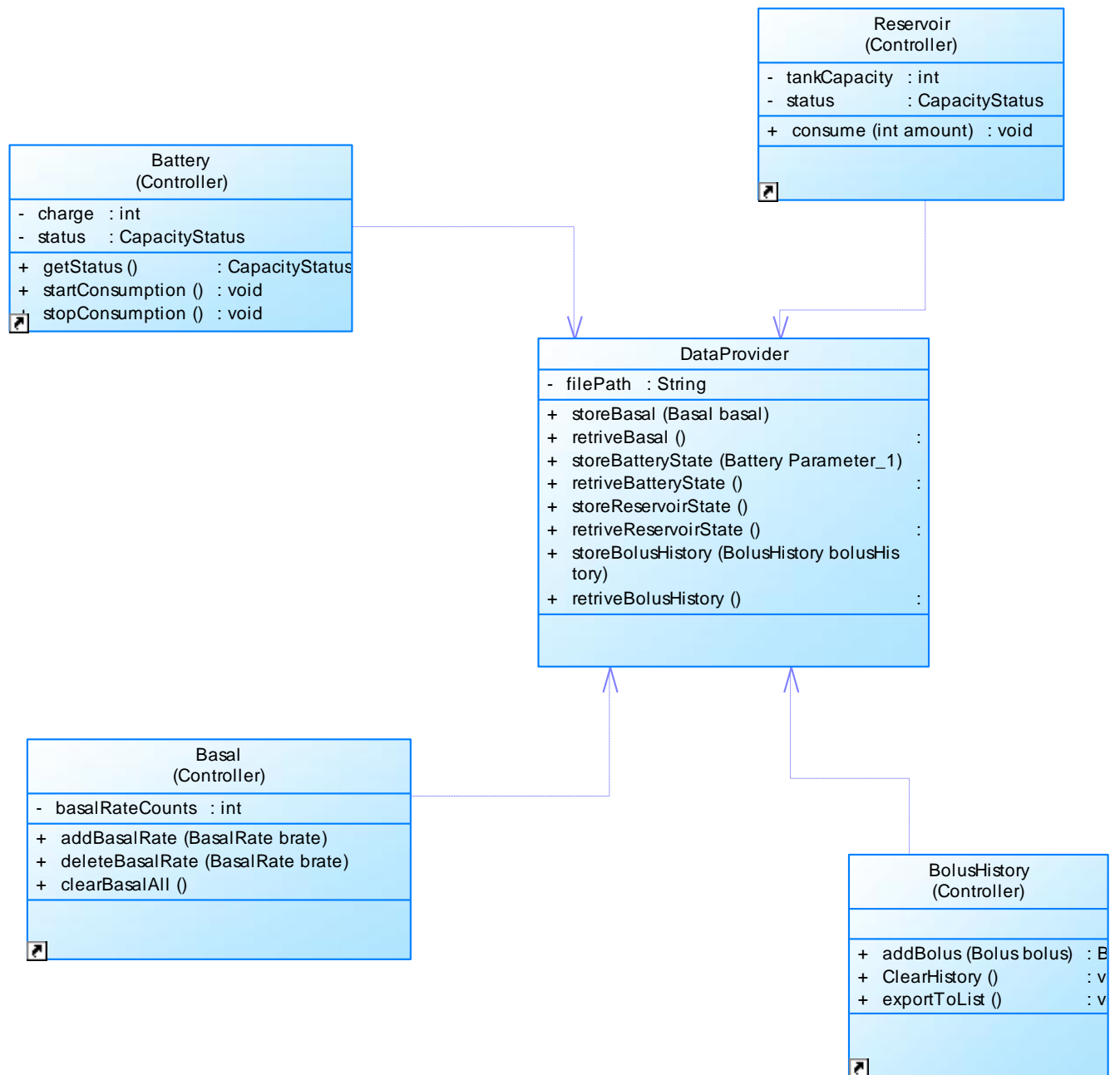


III.2.20 Diagram unlockPumpCollaboration



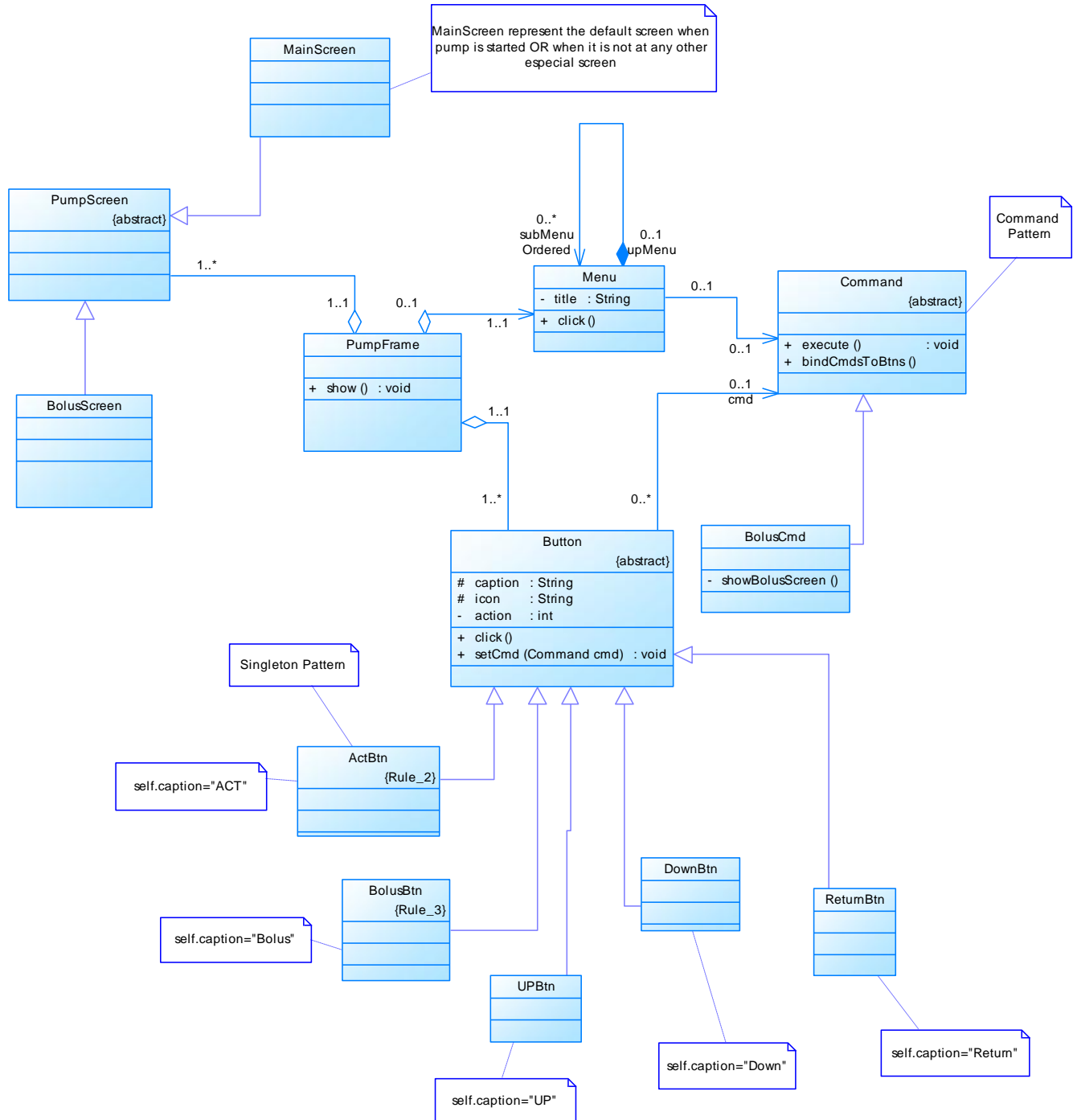
III.3 Package diagrams Data

III.3.1 Diagram DataLayerClassDiagram



III.4 Package diagrams GUI

III.4.1 Diagram GUI Class Diagram



III.4.2 Diagram Pump Simulator User Interface

