



6.3.3

## State properties

# FlexBE states

---

Remember:

- A state encapsulates an atomic action in the behavior.
- ROS nodes provide **capabilities** through topics, services and ROS actions.

A FlexBE **state**:

- represents a concrete use of one or more those capabilities to perform a **specific action**.

# Properties of a FlexBE State

- Name
- State implementation

## Interface

- Parameters
- Input & Output keys
- Outcomes

**Compute pick configuration**  
*ComputeGraspState*  
Computes the joint configuration needed to grasp the part given its pose.

**Parameters**  
group: .....

**Required Autonomy Levels**  
continue: off ▼  
failed: off ▼

**Input Key Mapping**  
pose: pose .....

**Output Key Mapping**  
joint\_values: joint\_values .....  
joint\_names: joint\_names .....

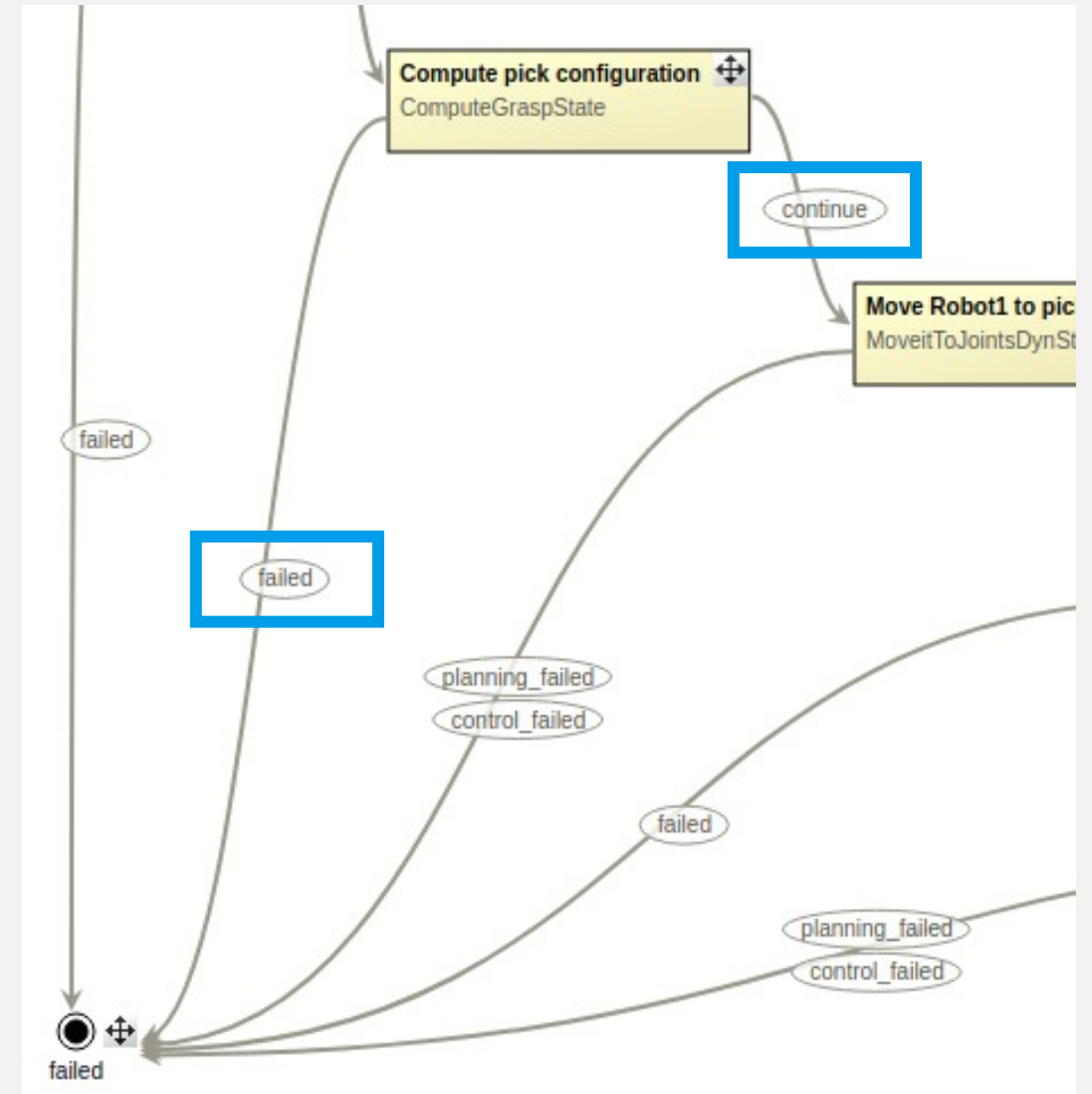
Apply Close Delete

# Properties of a FlexBE State

- Name
- State implementation

## Interface

- Parameters
- Input & Output keys
- Outcomes



# State parameters

- Configure the **static** properties of the state for your behavior.
- Use quotation marks to give them a literal value  
or
- use a behavior variable.

## Compute pick configuration

*ComputeGraspState*

Computes the joint configuration needed to grasp the part given its pose.

### Parameters

group: 'robot1'

### Required Autonomy Levels

continue: off ▼

failed: off ▼

### Input Key Mapping

pose: pose

### Output Key Mapping

joint\_values: joint\_values

joint\_names: joint\_names

Apply

Close

Delete

# State parameters

## Private Configuration

Variables enable easy configuration of constant internal values which are used multiple times. They are read-only and cannot be used in private functions.

`pick_group` = `'robot1'`



=

- Use quotation marks to give them a literal value  
or
- use a behavior variable.

## Compute pick configuration

*ComputeGraspState*

Computes the joint configuration needed to grasp the part given its pose.

### Parameters

group: `'robot1'`

### Required Autonomy Levels

continue: `off` ▼

failed: `off` ▼

### Input Key Mapping

pose: `pose`

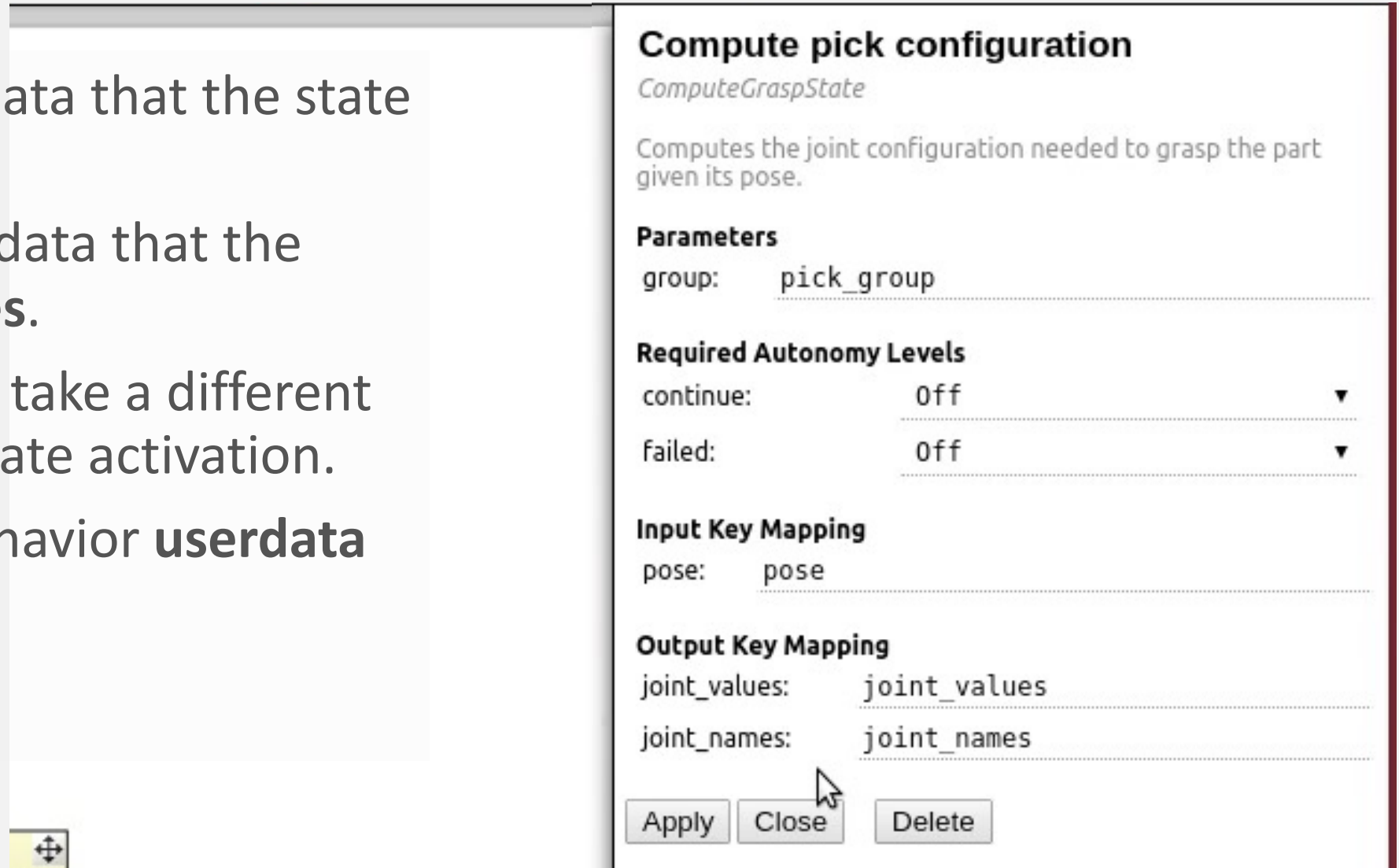
### Output Key Mapping

joint\_values: `joint_values`

joint\_names: `joint_names`

# Input and Output Key Mappings

- **Inputs** keys: data that the state **requires**.
- **Output** keys: data that the state **produces**.
- **Dynamic**: can take a different value every state activation.
- Values are behavior **userdata** variables.



The screenshot shows a dialog box titled "Compute pick configuration" with the subtitle "ComputeGraspState". The description states: "Computes the joint configuration needed to grasp the part given its pose." The dialog includes several sections: "Parameters" with a "group:" field set to "pick\_group"; "Required Autonomy Levels" with "continue:" and "failed:" dropdowns both set to "off"; "Input Key Mapping" with a "pose:" field set to "pose"; and "Output Key Mapping" with "joint\_values:" set to "joint\_values" and "joint\_names:" set to "joint\_names". At the bottom are "Apply", "Close", and "Delete" buttons. A mouse cursor is pointing at the "Close" button. A small crosshair icon is visible in the bottom left corner of the dialog area.

**Compute pick configuration**  
*ComputeGraspState*

Computes the joint configuration needed to grasp the part given its pose.

**Parameters**  
group:

**Required Autonomy Levels**  
continue:  ▼  
failed:  ▼

**Input Key Mapping**  
pose:

**Output Key Mapping**  
joint\_values:   
joint\_names:

# Input and Output Key Mappings

## State Machine Userdata

The userdata of a state machine can be used to pass any data from one state to another. Userdata values may be changed by states during runtime. Make sure you define default values for all userdata keys.

part\_pose = []  
pick\_configuration = home1  
 =

value every state activation.

- Values are behavior **userdata** variables.

## Compute pick configuration

*ComputeGraspState*

Computes the joint configuration needed to grasp the part given its pose.

### Parameters

group: pick\_group

### Required Autonomy Levels

continue: off ▼

failed: off ▼

### Input Key Mapping

pose: pose

### Output Key Mapping

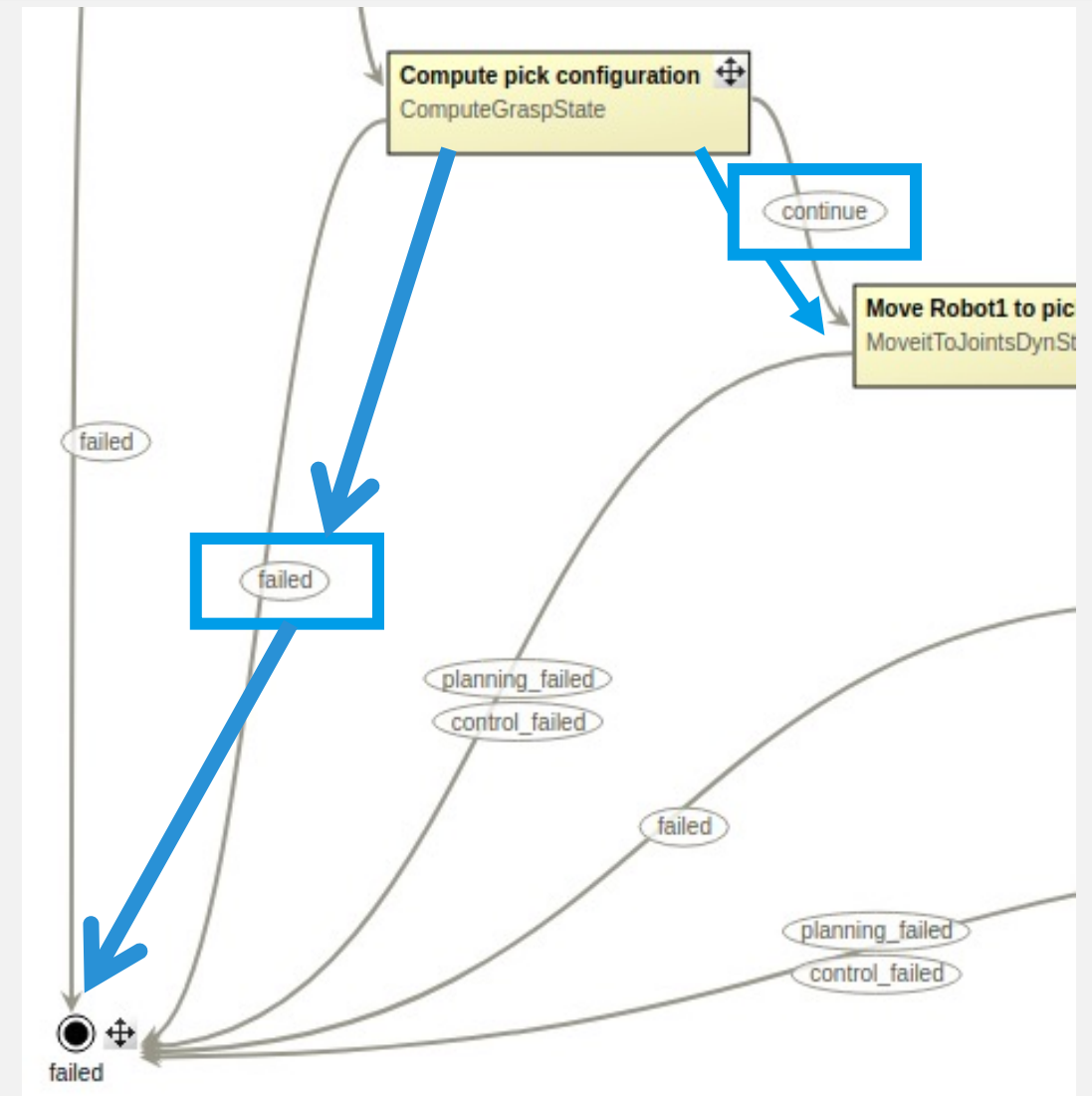
joint\_values: joint\_values

joint\_names: joint\_names



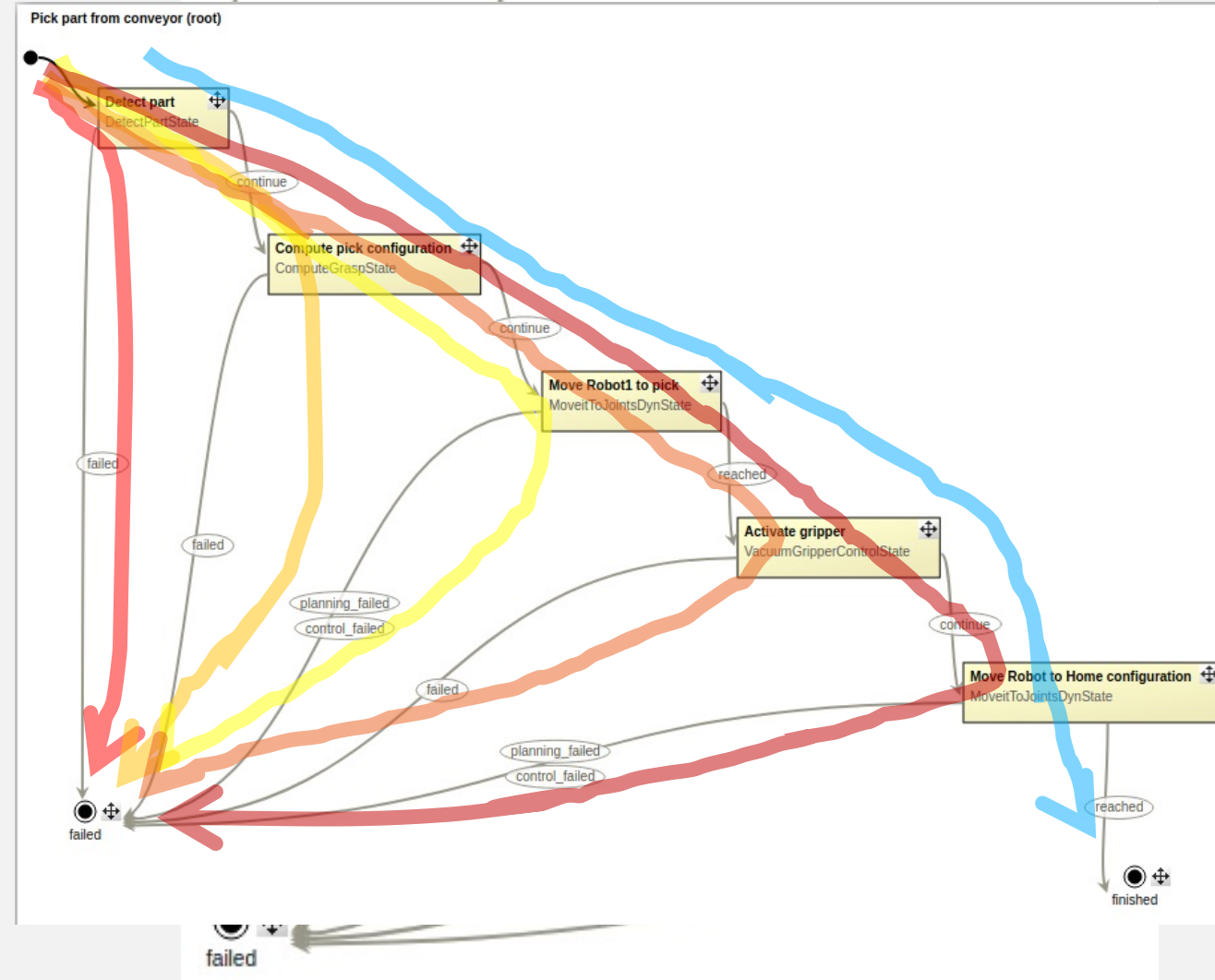
# State Outcomes

- Define possible results of execution.
- Possible transitions from the state.
- Shape the possible execution flows in the state machine.



# State Outcomes

- Define possible results of execution.
- Possible transitions from the state.
- Shape the possible execution flows in the state machine.



# Properties of a FlexBE State

- Name
- State implementation
- Parameters
- Input & Output key Mappings
- Outcomes

## Compute pick configuration

*ComputeGraspState*

Computes the joint configuration needed to grasp the part given its pose.

### Parameters

group: pick\_configuration

### Required Autonomy Levels

continue: off ▼

failed: off ▼

### Input Key Mapping

pose: part\_pose

### Output Key Mapping

joint\_values: pick\_configuration

joint\_names: joint\_names

Apply

Close

Delete