```python
def vacuum_cleaner():
    # Taking input for room states
    goal_state = {'A': '0', 'B': '0'}    # 0 means Clean, 1 means Dirty
    cost = 0

    # Taking user input for current state
    location_input = input("Enter the location of Vacuum (A or B): ").upper()
    status_input = input("Enter the status of " + location_input + " (0 for Clean, 1 for Dirty): ")
    status_other_room = input("Enter the status of the other room (0 for Clean, 1 for Dirty): ")

    # Display initial state
    print("\nInitial State:")
    if location_input == 'A':
        print("Room A:", status_input, " | Room B:", status_other_room)
    else:
        print("Room A:", status_other_room, " | Room B:", status_input)

    # Perform actions
    if location_input == 'A':
        if status_input == '1':
            print("Room A is Dirty. Cleaning Room A...")
            goal_state['A'] = '0'
            cost += 1
        if status_other_room == '1':
            print("Moving to Room B...")
            cost += 1
            print("Room B is Dirty. Cleaning Room B...")
            goal_state['B'] = '0'
            cost += 1
    else:  # Vacuum in B
        if status_input == '1':
            print("Room B is Dirty. Cleaning Room B...")
            goal_state['B'] = '0'
            cost += 1
        if status_other_room == '1':
            print("Moving to Room A...")
            cost += 1
            print("Room A is Dirty. Cleaning Room A...")
            goal_state['A'] = '0'
            cost += 1

    # Display results
    print("\nGoal State: Room A:", goal_state['A'], "| Room B:", goal_state['B'])
    print("Performance Measure (Total Cost):", cost)

vacuum_cleaner()
```

```
Enter the location of Vacuum (A or B): A
Enter the status of A (0 for Clean, 1 for Dirty): 1
Enter the status of the other room (0 for Clean, 1 for Dirty): 1

Initial State:
Room A: 1  | Room B: 1
Room A is Dirty. Cleaning Room A...
Moving to Room B...
Room B is Dirty. Cleaning Room B...

Goal State: Room A: 0 | Room B: 0
Performance Measure (Total Cost): 3
```