

3.5 Evolution

3.5.1 The main AquaLush irrigation software components must be reusable in later products.

3.5.2 AquaLush must eventually control a variety of irrigation valves.

3.5.3 AquaLush must accommodate implementation of traditional timer-controlled irrigation.

4. Data Requirements

4.1 AquaLush must record the following data in a persistent store:

- Type of each irrigation valve;
 - Location of each irrigation valve;
 - Flow rate of each irrigation valve;
 - Status of each irrigation valve;
 - Location of each moisture sensor;
 - Status of each moisture sensor;
 - Association between valves and sensors;
 - Current mode of operation;
 - Irrigation times;
 - Critical moisture levels; and
 - Water allocation.
-

B.6 AquaLush Use Case Model

Actors

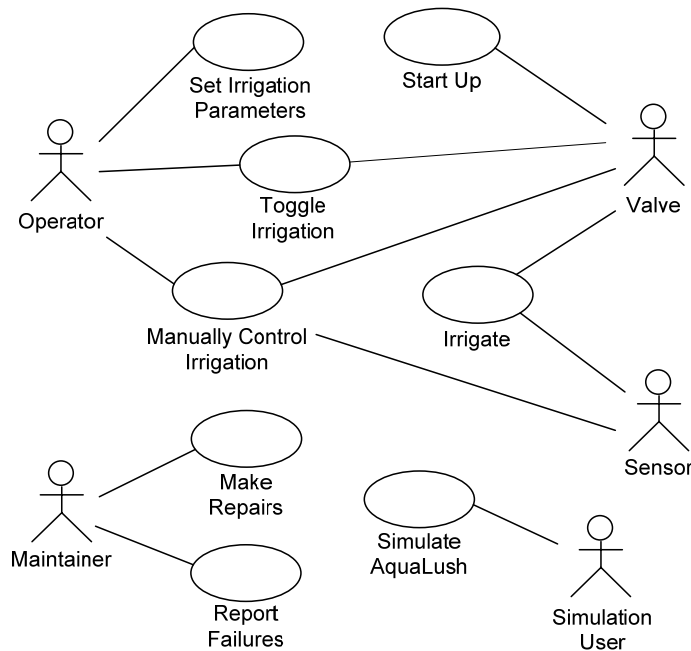
Sensor—A moisture sensor device.

Valve—An irrigation valve device.

Operator—A user operating AquaLush on a day-to-day basis.

Maintainer—A user performing maintenance tasks on AquaLush.

Simulation User—A person using the Web-based AquaLush simulation.

Use Case Diagram**Figure B-6-1 AquaLush Use Case Diagram****Use Case Descriptions****Use Case 1: Toggle Irrigation****Actors:** Operator, Valve**Stakeholders and Needs:**

Operators—To control irrigation times, to continue operating as normally as possible in the face of Valve and Sensor failures.

Managers, Marketers—To achieve 5% market share.

Maintainers—To detect and record Valve failures, to recover from power failures without Maintainer intervention.

Preconditions: None.

Postconditions: AquaLush is set to irrigate either automatically or manually, and this setting is recorded in a persistent store.

All Valve failures are recorded.

Persistent store failures are reported.

Trigger: Operator sets the mode.**Basic Flow:**

1. Operator sets the mode from manual to automatic.
2. AquaLush turns off any Valves that may be on.
3. AquaLush records the mode as automatic, confirms the change, and waits for the next irrigation time.

Extensions:

- 1a Operator sets the mode from automatic to manual:
 - 1a1. AquaLush closes all Valves.
 - 1a2. AquaLush records the mode as manual and confirms the change, and the use case ends.
- 1b Operator sets the mode to its current setting:
 - 1b1. AquaLush confirms the current mode and the use case ends.
- 2a A Valve fails:
 - 2a1. AquaLush tries twice more to close the Valve, and if it succeeds the use case continues.
 - 2a2. AquaLush alerts the Operator of the failure and records that this Valve failed in its persistent store; then the use case continues.
- 1a2a, 3a, 2a2a AquaLush cannot write to its persistent store:
 - 3a1. AquaLush alerts the Operator of the failure and the use case continues.

Use Case 2: Set Irrigation Parameters

Actors: Operator

Stakeholders and Needs:

Operators, Purchasers—To schedule irrigation for certain times, to set the moisture levels that control irrigation, to set the maximum amount of water used in irrigation.

Preconditions: None.

Postconditions: AquaLush automatic irrigation parameters are set, and these settings are recorded in a persistent store.

All Valve failures are recorded.

Persistent store failures are reported.

Trigger: Operator sets automatic irrigation parameters.

Basic Flow:

- 1. Operator sets the irrigation time, water allocation, or critical moisture levels.
- 2. AquaLush validates the new setting(s).
- 3. AquaLush records and confirms the new setting(s).

Extensions:

- 2a A setting is invalid:
 - 2a1. AquaLush notifies the Operator of the invalid setting.
 - 2a2. Operator adjusts the setting.
 - 2a3 AquaLush checks the setting and either resumes the basic flow or returns to step 2a1.
- 3a AquaLush cannot write to its persistent store:
 - 3a1. AquaLush alerts the Operator of the failure and the use case continues.
- 3b Irrigation is in progress when the irrigation time is set:
 - 3b1. AquaLush completes the current irrigation cycle, and the new irrigation time takes effect upon completion of the cycle.
- 3c Irrigation is in progress when the water allocation is set:
 - 3c1. AquaLush recomputes the allocations for each irrigation zone, and they take effect immediately.
- 3d Irrigation is in progress when the critical moisture levels are set:
 - 3d1. AquaLush uses the new critical moisture levels to control irrigation immediately.

Use Case 3: Manually Control Irrigation**Actors:** Operator, Valve, Sensor**Stakeholders and Needs:**

Operators—To schedule irrigation for certain times, to continue operating as normally as possible in the face of Valve and Sensor failures, to irrigate the site.

Maintainers—To detect and record Valve failures, to recover from power failures without Maintainer intervention.

Preconditions: AquaLush is in manual mode.**Postconditions:** All Valve and Sensor failures are recorded.

Persistent store failures are reported.

Trigger: Operator selects a non-empty set of closed Valves and directs that they be opened.**Basic Flow:**

1. Operator selects a non-empty set of closed Valves and directs that they be opened.
2. AquaLush opens each Valve in the set and displays to the Operator for each open Valve: its location, how long it has been open, how much water it has used, and the moisture level of its associated Sensor. AquaLush also shows the total water used since the start of the use case.
The following Operator action and AquaLush responses may be done in any order, and they may be done repeatedly.
3. Operator selects a set of open Valves and directs that they be closed.
4. AquaLush closes the indicated Valves and removes them from the Valve status display.
5. Operator selects a set of closed Valves and directs that they be opened.
6. AquaLush opens the selected Valves and adds them to the Valve status display.
The following steps are taken to end the use case.
7. The Operator indicates that he or she is finished.
8. AquaLush acknowledges that manual irrigation is finished and closes all Valves.

Extensions:

2a,4a,6a,8a A Valve fails:

2a1. AquaLush tries twice more to manipulate the Valve, and if it succeeds, the use case continues.

2a2. AquaLush alerts the Operator of the failure and records that this Valve failed in its persistent store; then the use case continues.

2b A Sensor fails:

2b1. AquaLush tries twice more to read the Sensor, and if it succeeds, the use case continues.

2b2. AquaLush alerts the Operator of the failure and records that this Sensor failed in its persistent store; then the use case continues.

2*2a AquaLush cannot write to its persistent store:

2a2a1. AquaLush alerts the Operator of the failure, and the use case continues.

Use Case 4: Make Repairs**Actors:** Maintainer**Stakeholders and Needs:**

Maintainers—To fix problems easily.

Preconditions: None.**Postconditions:** The clock is reset or AquaLush is notified of repaired Valves and Sensors.

The persistent store is updated when hardware is repaired.

Persistent store failures are reported.

Trigger: Maintainer begins a repair session.

Basic Flow:

1. Maintainer begins a repair session.
2. AquaLush displays the set of failed Valves and Sensors.
The following Maintainer actions and AquaLush responses may be done (as pairs) in any order, and they may be done repeatedly.
3. Maintainer sets the current time.
4. AquaLush resets its clock and confirms it to the Maintainer. The current time is used to control automatic irrigation immediately.
5. Maintainer indicates which of the failed Valves and Sensors are now repaired.
6. AquaLush adjusts its persistent store to record the repairs. AquaLush begins using the repaired Sensors and Valves for irrigation immediately.
7. The use case ends when the Maintainer indicates that repairs are complete.

Extensions:

- 6a AquaLush cannot write to its persistent store:
 - 6a1. AquaLush alerts the Maintainer of the failure and the use case continues.

Use Case 5: Report Failures

Actors: Maintainer

Stakeholders and Needs:

Maintainers—To fix problems easily.

Preconditions: None.

Postconditions: Failed Valves and Sensors are reported.

Trigger: Maintainer requests a failure report.

Basic Flow:

1. Maintainer requests a failure report.
2. AquaLush displays the failed Valves and Sensors and their locations, or indicates that there are no failures.
3. Maintainer ends the use case.

Extensions:

None.

Use Case 6: Start Up

Actors: Valve

Stakeholders and Needs:

Maintainers—To recover from power failures without Maintainer intervention.

Operators—To continue operating as normally as possible in the face of Valve and Sensor failures.

Preconditions: AquaLush was powered down.

Postconditions: If persistent storage is intact and readable, then AquaLush is restored to its state prior to shutdown, except that all Valves are closed.

All Valve failures are recorded or persistent store failure is reported.

Trigger: The system is powered up and AquaLush is restarted.

Basic Flow:

1. AquaLush reads its persistent store and establishes its configuration.
2. AquaLush reads its persistent store and restores all its parameters.
3. AquaLush reads its persistent store and closes all working Valves. If AquaLush is in automatic mode, it waits for the irrigation time.

Extensions:

*a AquaLush cannot read its persistent store:

*a1. AquaLush displays a failure message and suspends operation.

3a A Valve fails:

3a1. AquaLush tries twice more to close the Valve, and if it succeeds the use case continues.

3a2. AquaLush records that this Valve failed in its persistent store.

3a3. AquaLush goes on with its processing.

3a2a AquaLush cannot write to its persistent store: The use case continues.

Use Case 7: Irrigate

Actors: Valve, Sensor

Stakeholders and Needs:

Management, Developers, Marketers, Operators, Purchasers—To irrigate only until the desired moisture level is achieved.

Operators, Purchasers—To monitor the amount of water used in irrigation and to stop irrigation when the maximum allocated amount of water is reached.

Maintainers—To detect Valve and Sensor failures, to recover from power failures without Maintainer intervention.

Operators—To continue operating as normally as possible in the face of Valve and Sensor failures.

Preconditions: AquaLush is in automatic irrigation mode.

Postconditions: All irrigation zones have been irrigated until either their Sensors indicate that the critical moisture level has been reached or the water allocated for the zone is used up.

All Valve and Sensor failures are recorded.

All persistent store failures are reported.

Trigger: The current time is the irrigation time.

Basic Flow:

1. AquaLush reads each (working) Sensor. If a Sensor reports moisture below its zone's critical moisture level, then AquaLush places that zone on an active zone list.
2. AquaLush counts the (working) Valves in the active zones and divides the water allocation by this count to get a Valve allocation. Each zone is assigned a zone allocation, which is the Valve allocation multiplied by the number of Valves in the zone.
For each active zone:
3. AquaLush opens all the (working) Valves in the zone, and marks the time when it does so.
4. Every minute, AquaLush reads the zone Sensor and also computes the zone water usage by multiplying the total flow rate for all Valves in the zone by the time that the Valves have been open.
5. If the zone's critical moisture level is reached, or the zone allocation is reached, then AquaLush closes all the Valves in the zone.

The use case ends when all zones in the active zone list have been irrigated.

Extensions:**1a A Sensor fails:**

- 1a1. AquaLush tries twice more to read the Sensor, and if it succeeds goes on as before.
- 1a2. AquaLush records that this Sensor failed in its persistent store.
- 1a3. AquaLush continues with the next Sensor.

3a,4a2a,5a A Valve fails:

- 3a1. AquaLush tries twice more to manipulate the Valve, and if it succeeds the use case continues.
- 3a2. AquaLush records that this Valve failed in its persistent store.
- 3a3. AquaLush goes on with its processing.

4a A Sensor fails:

- 4a1. AquaLush tries twice more to read the Sensor, and if it succeeds the use case continues.
- 4a2. AquaLush closes all Valves in the zone.
- 4a3. AquaLush records that this Sensor failed in its persistent store.
- 4a4. AquaLush continues with step 5 as if the Sensor had reached the zone's critical moisture level.

5b The zone allocation is not exhausted:

- 5b1. AquaLush adds the remainder of the zone allocation to the zone allocations of the unirrigated active zones and divides this sum by the number of Valves in the unirrigated active zones to compute a new Valve allocation.
- 5b2. AquaLush recomputes the zone allocations for each active unirrigated zone by multiplying the new Valve allocation by the number of Valves in the zone.

1a2a,3a2a,4a3a AquaLush cannot write to its persistent store: AquaLush alerts the Operator of the failure and the use case continues.

Use Case 8: Simulate AquaLush

Actors: Simulation User

Stakeholders and Needs:

Purchasers, Management, Developers, Marketers—To have a Web-based simulation providing an accurate representation of the actual product and its use.

Preconditions: None.

Postconditions: Simulators have interacted with an accurate simulation.

Trigger: The simulation Web page is loaded.

Basic Flow:

Any of the following Simulator actions may be done in any order, and they may be done repeatedly.

1. Simulator uses the simulated control panel to Toggle Irrigation, Set Irrigation Parameters, Manually Control Irrigation, Make Repairs, or Report Failures.
2. AquaLush Simulator displays the simulated effects on the representation of a site with irrigation zones containing Sensors and Valves. AquaLush Simulator also displays the simulated time, simulation speed, Sensor readings, Valve flows, water evaporation rate, and failed Valves and Sensors.
3. AquaLush Simulator speeds up or slows down the simulation, sets the simulated time, sets the simulated water evaporation rate, or makes Valves or Sensors fail or be repaired.
4. AquaLush Simulator alters the simulation in response to the new parameter settings.

The use case ends when the simulation Web page is exited.