

Пример построения концепции системы на технологиях интернета вещей

Приложение интернета вещей - сложная система, практически эквивалент живого организма, у которого есть своя анатомия, физиология и даже психика, причем все это существует только во взаимосвязи.

Можно сказать, что аналогом анатомии является совокупность механического конструктива элементов приложения - то, что мы применительно к стеку разработки умных взаимодействующих вещей называли инфраструктурой.

Физиология - это то, как там организованы процессы как они работают, а психика - то, что определяют логику его поведения.

Приложение проектируется как единое целое и его проектирование - это про умение создавать сложные системы, где совместно используются технологии механики, электроники, программирования работы с данными, их анализа и пр., и пр.

Перефразируя Маркса - именно этим «самый плохой разработчик отличается от самой лучшей пчелы»: он представляет, что собирается делать, до того, как начинает делать!

И представление того, что нужно для получения решения и то, как это будет работать - и есть суть проектирования.

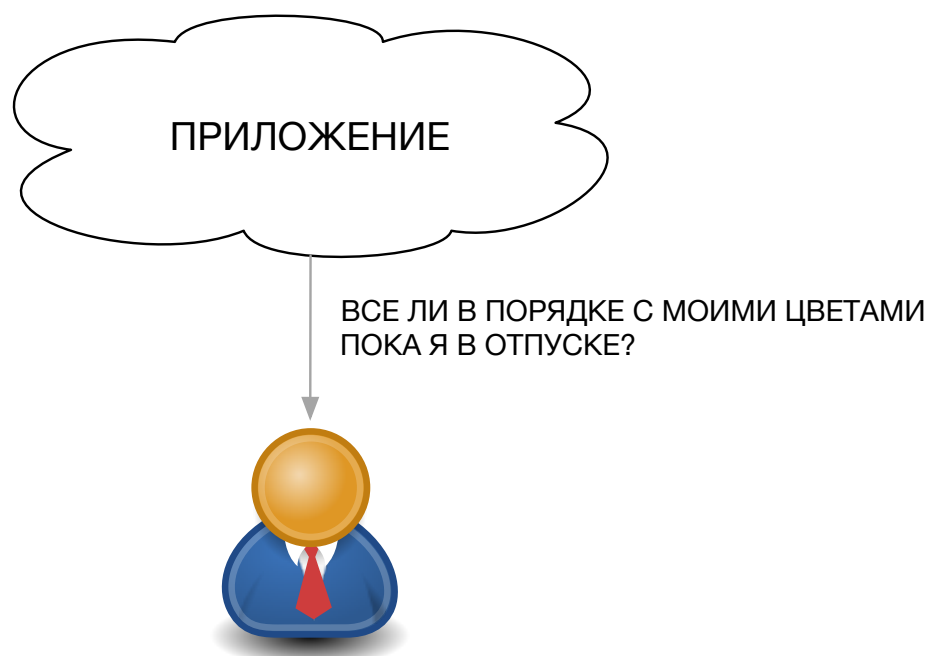
В прошлой части мы выяснили, как собрать требования пользователя, потребителя или заказчика, а теперь посмотрим, как, имея их, прийти к возможному решению.

Разберем на конкретном примере.

Пусть есть проблема: пользователь волнуется, все ли в порядке с его цветами, пока он в отпуске

Он рассмотрел разные способы, как устранить причины своего беспокойства, услышал, что есть возможность удаленного мониторинга состояния растений, и обратился к нам

Он ожидает, что мы сделаем ему
некое «приложение», которые будет
давать ему информацию о состоянии
его растений

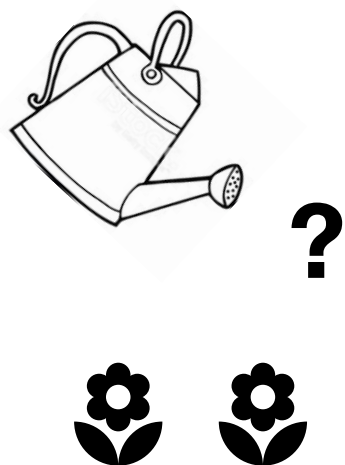


Считаем для простоты, что пользователь, соответственно, стал нашим заказчиком (хотя в общем случае это может быть, что это совершенно разные люди).

И вот мы отправили к пользователю нашего аналитика и теперь представляем требования к будущему приложению.

Мы узнаем, что суть проблемы в том, что пользователь договаривается с соседями, что они будут присматривать за его цветами и поливать их. Но он волнуется, что они могут забыть это сделать

Итак, мы сформулировали проблему:
надо узнать, были ли вовремя политы
цветы?



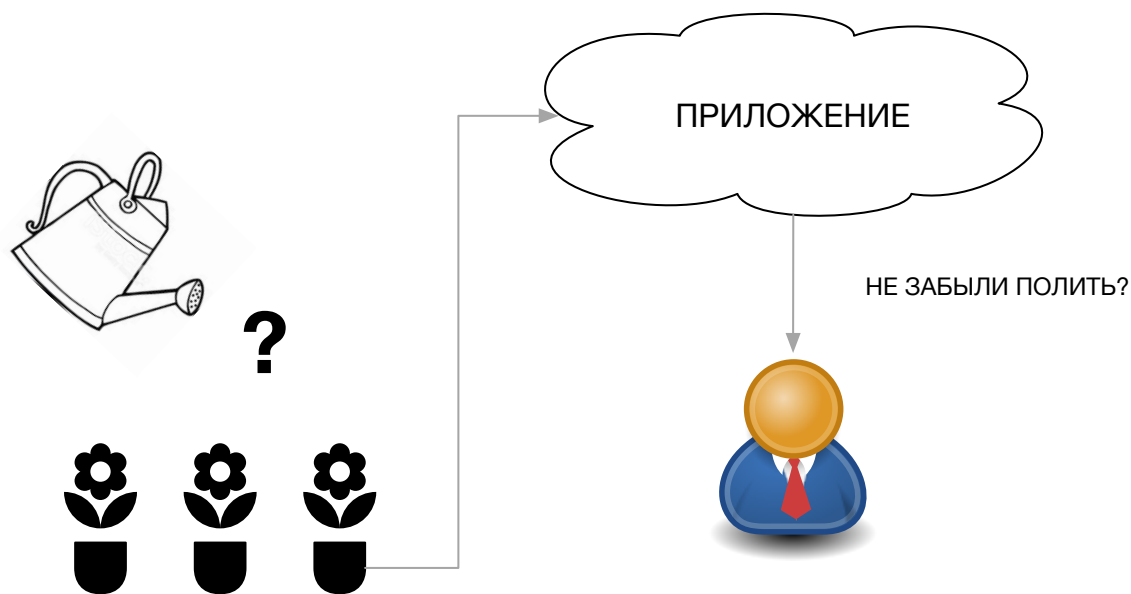
Если проблема идентифицирована,
на нужно выбрать способ ее решения
- Как мы можем проверить.
поливались цветы или нет?

Например, мы можем отслеживать,
заходили ли соседи в квартиру,
поставив соответствующий датчик а
входную дверь.

Или мы можем отсиживать,
пользовались ли лейкой, поставив на
нее RFID-метку

Но присутствие в квартире или
пользование лейкой не означает, что
цветы политы должным образом.

Поэтому мы продолжаем рассматривать разные способы и останавливается на том, что решаем просто измерять важность почвы в горшках.



Теперь нам надо понять, какие средства для этого нам понадобятся. Но это можно понять, если рассмотреть «деятельность», которую должна осуществлять система, и то, что она должна для этого «уметь»

Начнем с деятельности: система должна определять влажность почвы в горшках и сообщать об ее уровне пользователю. Если он низкий - она также должна выдавать ему специальное предупреждение.

Что система должна «уметь», чтобы это делать?

Если мы расписали деятельность, то необходимые умения, «компетенции» должны быть достаточно очевидны:

1. Система должна уметь измерить влажность почвы
2. Система должна уметь сообщить пользователю об уровне важности
3. Система должна уметь сообщить, если почва сухая

Начинаем искать средства для данных умений.

Чтобы измерить влажность почвы - берем датчик влажности.

Сразу же пройдем по стеку разработки:

- нам будет нужен какой-то конструктив для его установки, или возможно он будет интегрирован в горшок, или как-то еще - то, что мы называли «инфраструктурой»
- собственно, конкретный датчик
- «коннективность» - то, что будет обеспечивать передачу данных в сеть

- программа, которая будет выполнять необходимую обработку данных - «аналитика»

- и, наконец, понять, как они будут представлены

Со средством изменения влажности разобрались.

С помощью какого средства система может сообщить пользователю уровень влажности?

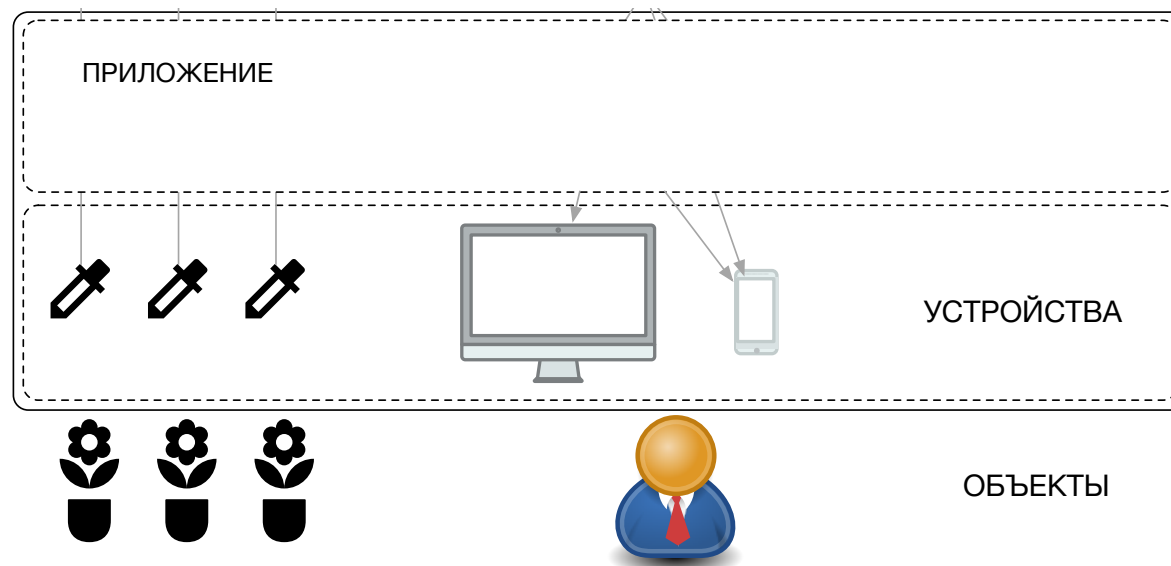
Пожелание пользователя мобильный телефон, а также возможность посмотреть эти данные компьютера. Прикинули - да, это технически осуществимо и примерно понятно как сделать.

Как выдать предупреждение о том, что почва сухая?

Так же начинаем рассматривать способы: можно, чтобы появилось сообщение, можно - какой-то звуковой сигнал или вибрация смартфона, можно завести блог о состоянии любимых растений и выдавать информацию там...

Останавливаемся на варианте, когда в этой ситуации отправляется СМС на телефон и выдается соответствующее предупреждение в общей информации о состоянии растений.

Итак, приходим к следующей схеме,
где аппаратными средствами
(устройствами) для нашей задачи
становятся:



1. Датчик влажности почвы со встроенным модем связи,
2. Смартфон для получения СМС и просмотра информации о состоянии растений
3. Настольный компьютер для просмотра информации о состоянии растений

Тут же возникает вопрос и об интерфейсах, с помощью которых пользователь может получать необходимую информацию:

Для Смс - это смартфон или мобильный телефон, привязанный через заданный номер

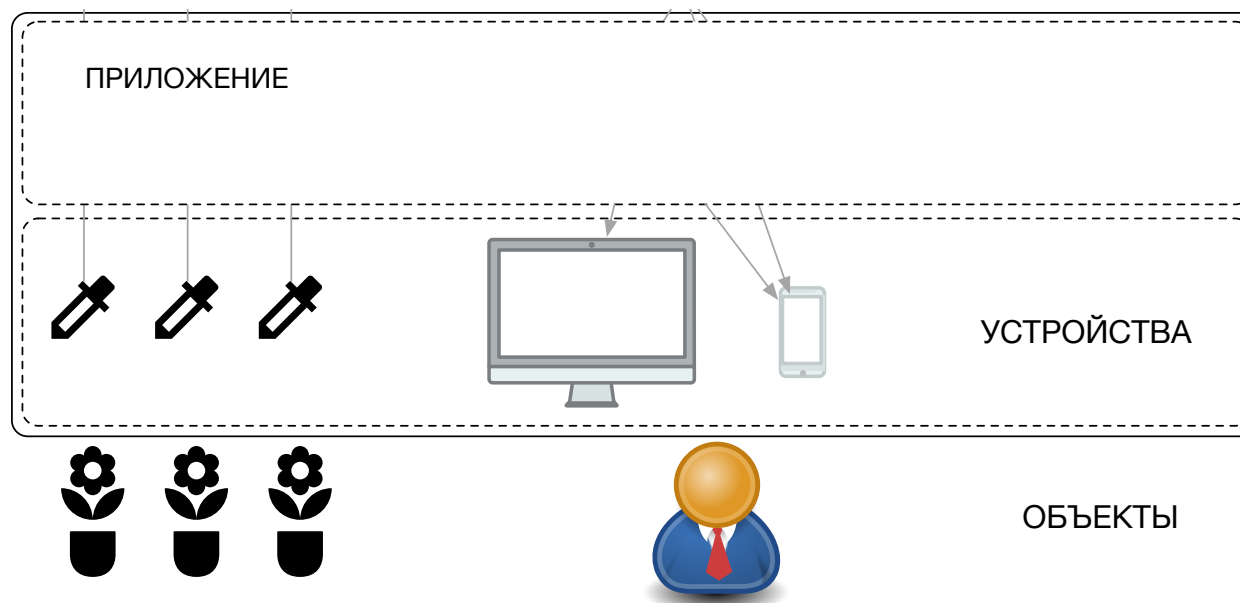
Для информации о состоянии растений для просмотра с настольного компьютера можно использовать веб-интерфейс.

Для просмотра информации со смартфона - можно как веб-интерфейс, так и мобильное приложение.

Но мобильное приложение подразумевает дополнительные затраты на разработку, кроме того оно создает дополнительные хлопоты для пользователя: его надо специально загружать, потом следить за обновлениями и пр.

Поэтому останавливаемся на веб-приложении.

В результате получаем следующую концепцию предлагаемой к разработке системы:



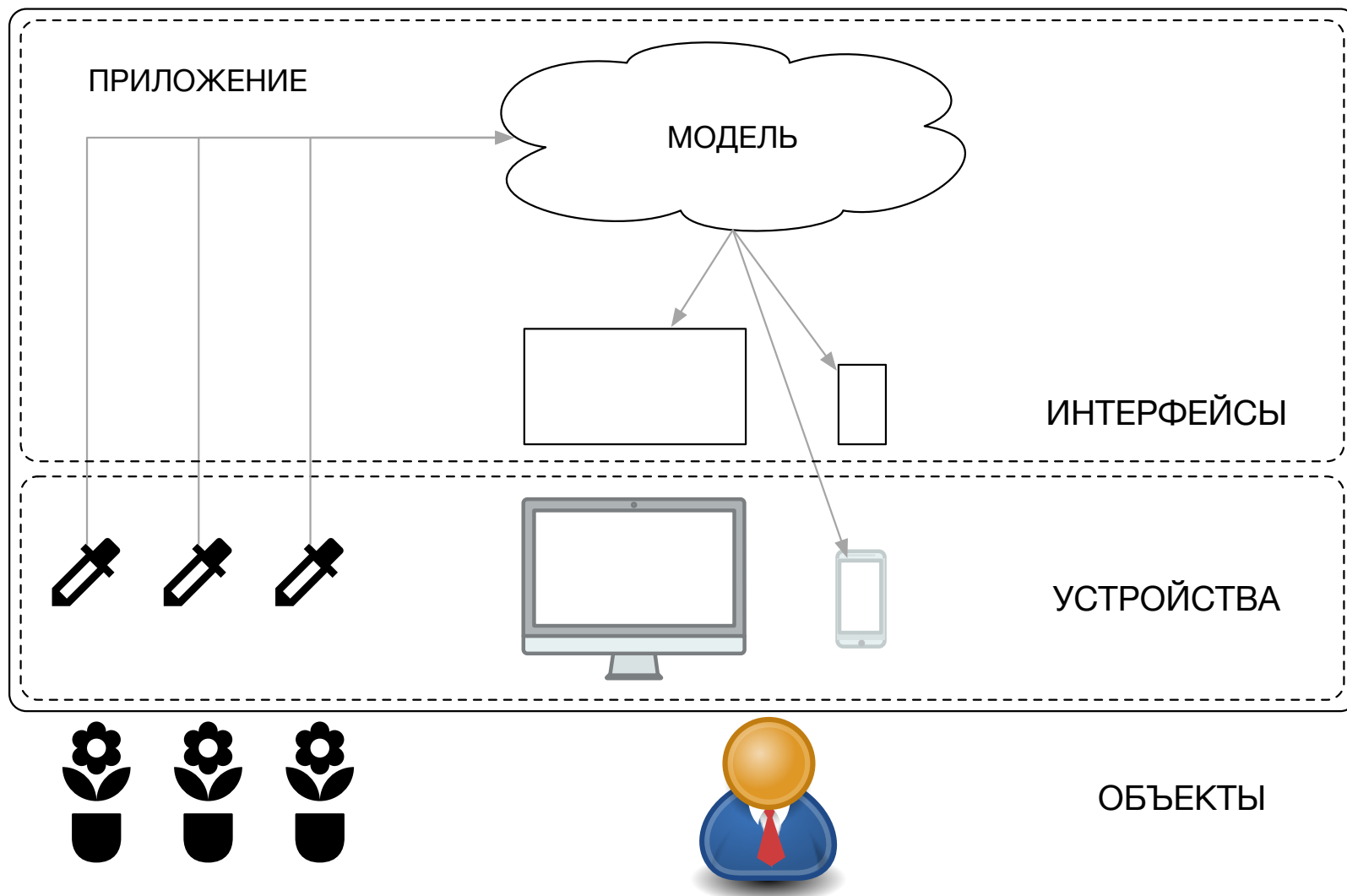
Надо разработать приложение, которое будет получать информацию с датчиков влажности почвы и выдавать ее пользователю через веб-интерфейс, доступный со смартфона и десктопа, а также отправлять пользователю СМС-сообщения при падении уровня важность ниже критического.

Мы говорили о том, что концепция - это то, как «видит» систему пользователь, включая то, как она действует.

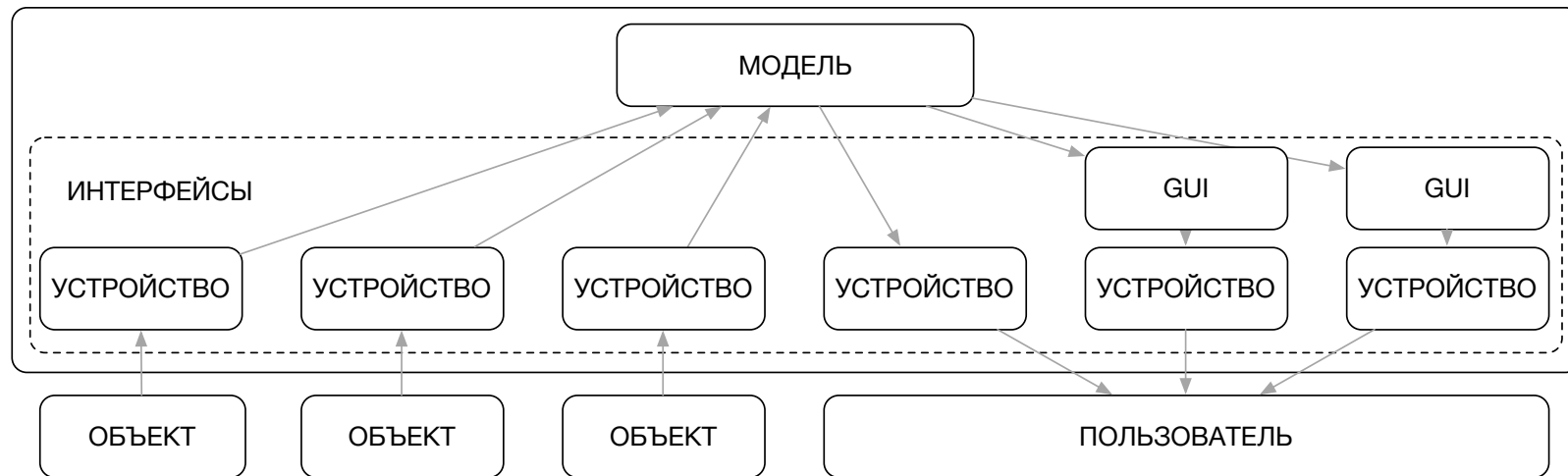
Соответственно, пользователь «ВИДИТ» что растения что-то отправляют в интернет, и там есть какое-то «приложение», которое что-то отправляет на телефон, и позволяет посмотреть информацию про растения «в интернете»

Смотря на концепцию глазами разработчика, мы видим набор объектов, которые как-то взаимодействуют друг с другом через приложение.

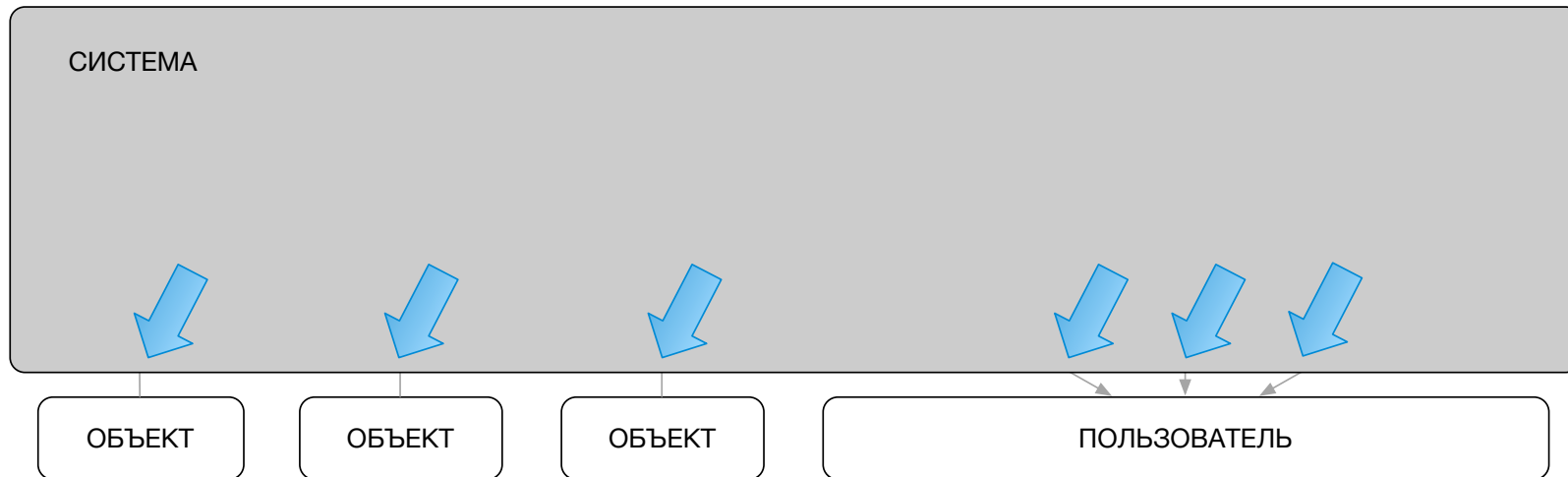
А чтобы, в свою очередь, было возможно взаимодействие объектов с моделью данных, организуются соответствующие интерфейсы.



В более формальном виде, общая структура системы будет выглядеть следующим образом:

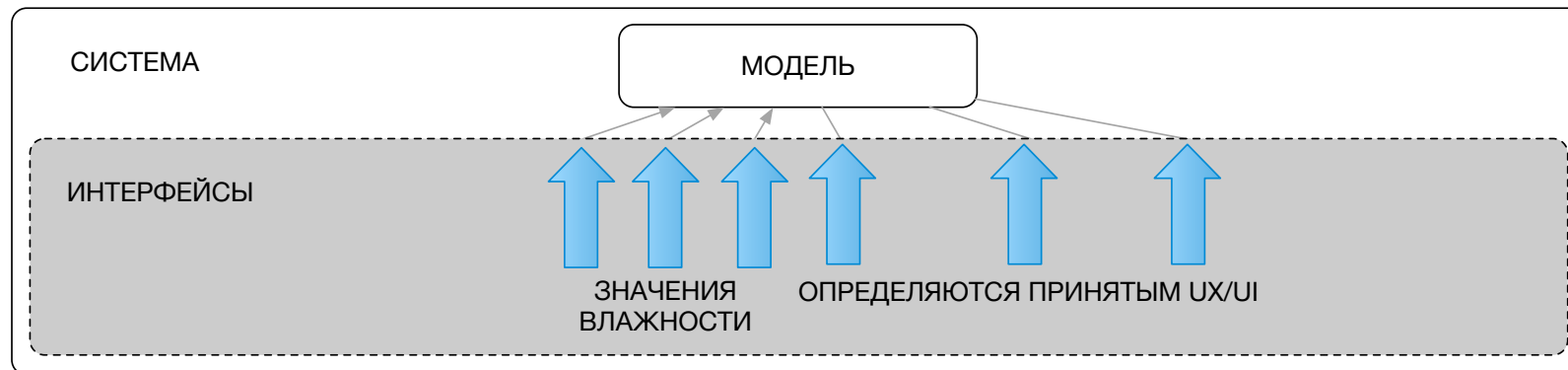


С точки зрения системного мышления, на этом этапе целевая система для нас еще «черный ящик», и мы помним из курса, что она определяется через ее взаимодействия, прежде всего, с окружающей средой (операционным окружением), в данном случае - с внешними по отношению к системе объектами.



И мы помним, что именно формат ЭТИХ взаимодействий и определяет системные требования (system requirements).

Аналогично, сама модель данных как подсистема целевой системы, в свою очередь, также определяется через свои взаимодействия

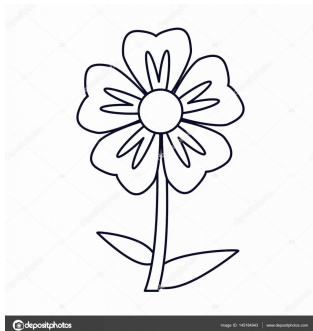


Прежде, чем зафиксировать концепцию системы, обратим внимание еще на один момент: система является человеко-ориентированной, поэтому для нее важно определить необходимые взаимодействия с пользователем, то, что сейчас часто называют UX, а также детализировать интерфейсы пользователя - UI



Но наш аналитик провел
необходимое исследования,
определил оптимальный сценарий и
предложил вид пользовательского
интерфейса.

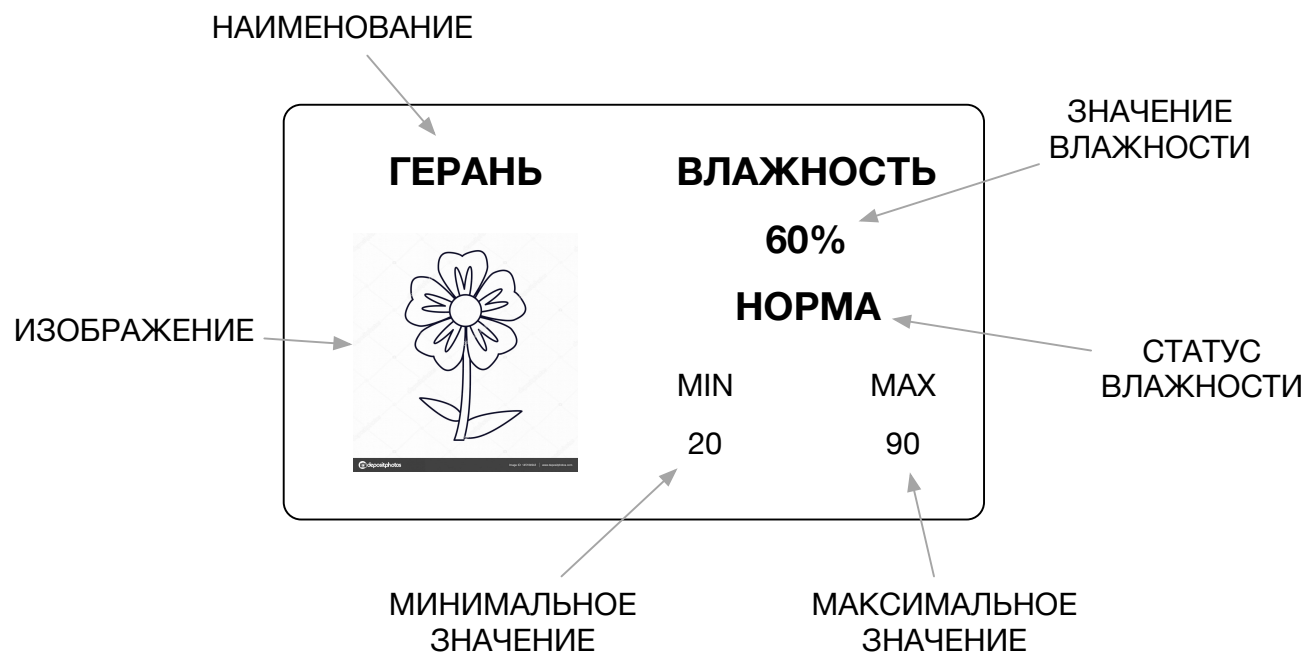
Мы согласовали его с заказчиком и
далее будем ориентироваться на
следующий его вариант

ГЕРАНЬ	ВЛАЖНОСТЬ	
	60%	
	НОРМА	
	MIN	MAX
	20	90

Но мы видим, что на этом этапе у нас появились новые взаимодействия и, соответственно, поменялись системные требования.

Так, если мы начинали с того, что нам нужно выдать заказчику значение влажности и предупреждение, если оно ниже нормы, то сейчас список параметров расширился:

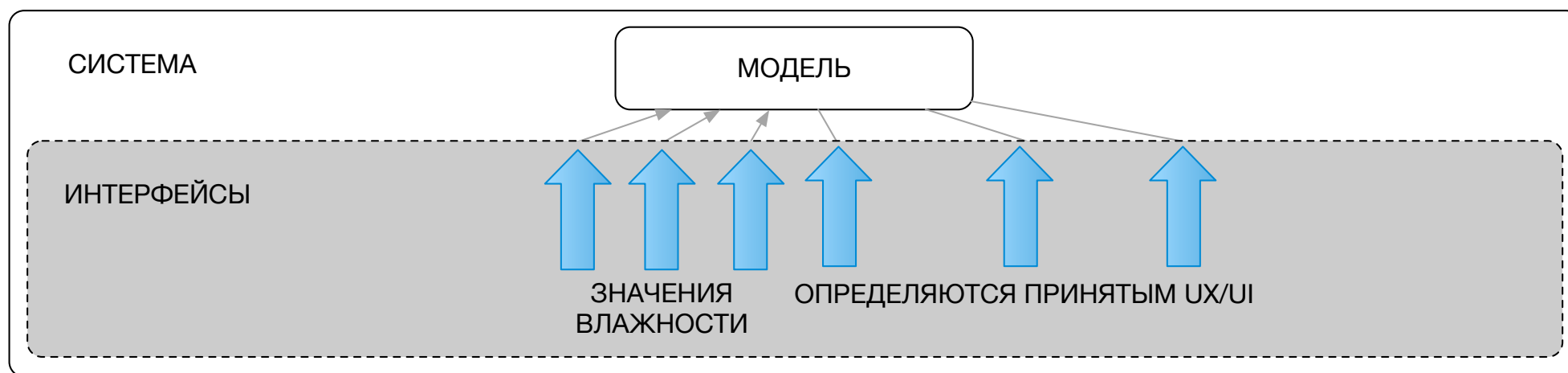
Нам надо выдать теперь название растения и его изображение, а также сопоставить текущее значение влажности с некими минимальным и максимальным значениями.



А чтобы их выдать - их сначала надо как-то задать

Поэтому, возвращаясь к системным требованиям и определению приложения как системы, мы видим, что информация от наших растений как от объектов осталась прежней, а вот взаимодействие с пользователем поменялось

Итак, как наша модель данных - пока
еще как черный ящик - будет
выглядеть через свои
взаимодействия?



Его входы со стороны объектов - значения влажности с датчиков по числу объектов.

А вот со стороны пользователя - его входы теперь:

- наименование растения
- изображение растения
- минимальное значение влажности
- максимальное значение влажности

И логично их заводить в систему также через веб-интерфейс.

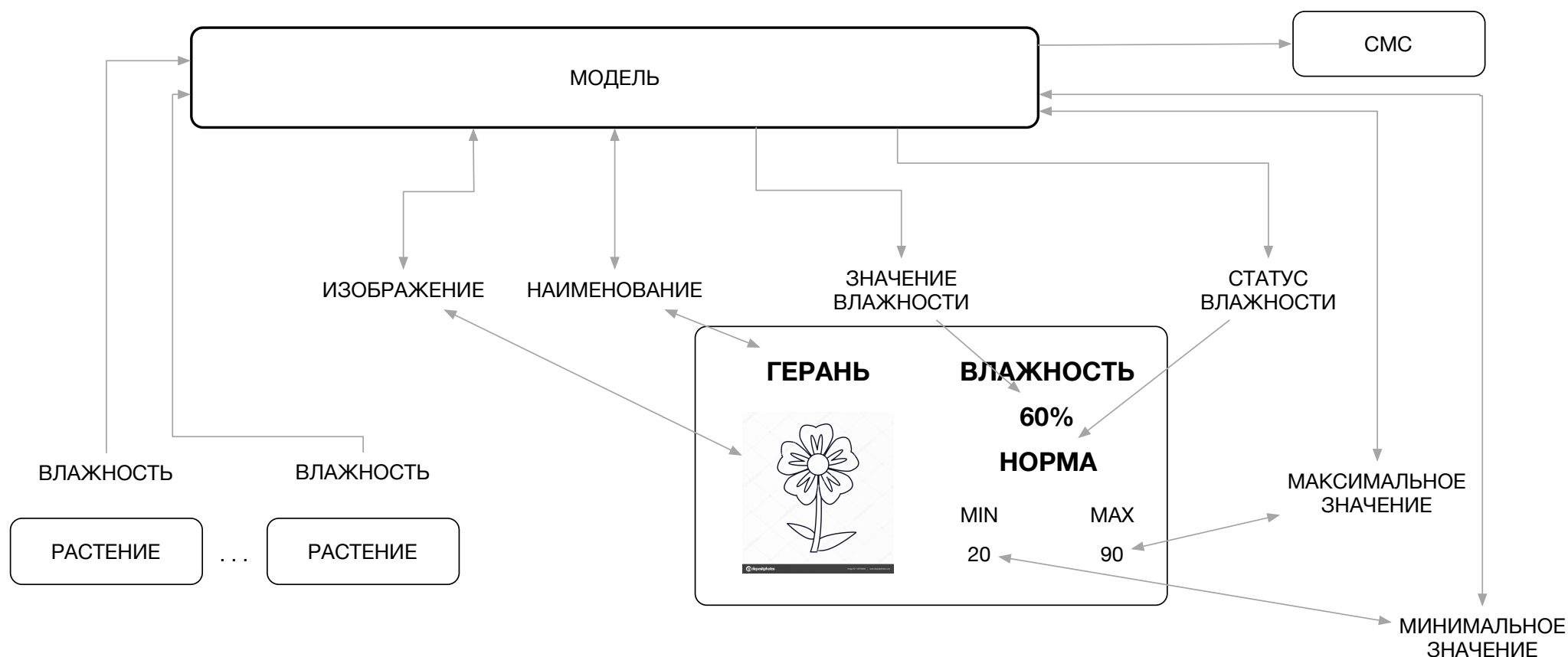
А какие выходы?

Через СМС мы по-прежнему передаем предупреждение о критическом уровне влажности

А через веб-интерфейс:

- наименование
- изображение растения
- текущее значение влажности
- его статус, а также
- минимальное и максимально значение

Итак, мы приходим к следующей структуре приложения



Обратите внимание: у нас изменение пользовательского интерфейса стало следствием изменения сценария взаимодействия пользователем.

Причем этот сценарий взаимодействия стал содержать бизнес-правила, определяющих изменение статуса влажности в веб-интерфейсе и отправку СМС. Итак, мы показали сам подход, как, исходя из требований к системе сформировать ее концепцию.

Такая концепция будет показывать, как как бы «со стороны» будет выглядеть и как вести себя в процессе решения поставленной задачи.

Скажем, для приведенного в примере приложения мониторинга, мы опишем, каким образом система будет предоставлять необходимую пользователю информацию, какую информацию она сама должна получать для этого и так далее.

И тут мы говорим, как система устроена (но не внутри, поскольку она для нас пока черный ящик), а как выглядит и как она себя ведет - на что реагирует и пр.

Но, естественно, понимаем, что внешний образ и внутреннее устройство всегда взаимосвязаны...

Таким образом, исходя из требований к системе, мы получаем определение системы (system definition), или описывает, в частности как система будет вести себя в окружающем мире.

Само поведение системы прописывается как комплекс ее реакций на изменение «внешних условий».

В данном случае - это выдача информации пользователю при изменении значений влажности, а также изменение параметров интерфейса и реакций системы при задании соответствующих значений пользователем.

Собрав, зафиксировав и систематизировав эти требования, как было рассмотрено в предыдущем разделе, мы можем перейти к детализации архитектуры системы.

Об этом поговорим в следующем видео.

