Использование сенсоров в программных приложениях на платформе Sensor and Location вWindows 7

Выполнила:
Ткач О.В
Научный руководитель:
Киселев Г.Д.

Контекстно-зависимые системы

- Контекст – ситуация, в которой находится объект
- Контекстно-зависимые программы – изменяют свое поведение в зависимости от ситуации, в которой находится пользователь

Этапы работы контекстно-зависимых систем

Примеры

- GPS-навигаторы на телефоне
- Определение местоположения пользователя
- Контекстная информация о точках интересов (POI) вблизи пользователя (магазины, кафе, АЗС и т.д.)
- Изменение яркости изображения в зависимости от яркости окружающей среды
- Сенсоры освещенности/дождя в авто
- Изменение ориентации интерфейса при изменении положения телефона по отношению к земле

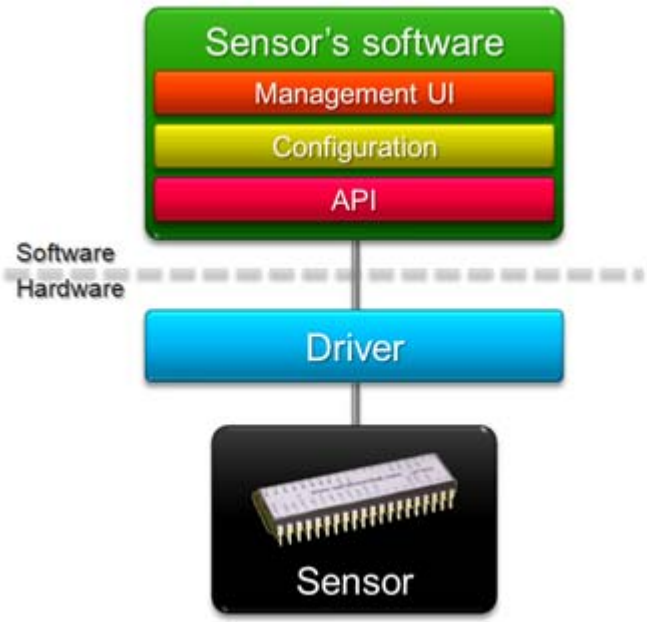
Что такое сенсоры?

- Сенсор – устройство, которое измеряет некоторую величину и преобразует ее значение в сигнал компьютеру
- Примеры:
- Сенсор освещенности
- Сенсор влажности
- Сенсор температуры
- Акселерометр

Проблематика

- Способ доступа к сенсорам
- Обычно – через COM-порты
- Доступ только из одного приложения
- Закрытые форматы данных
- Вопрос безопасности
- Поддержка работы нескольких сенсоров
- Например, GPS не работает внутри зданий – нужно одновременно работать на основе нескольких сенсоров
- Плотная интеграция с аппаратными решениями
- Приложения должны учитывать специфику аппаратных устройств

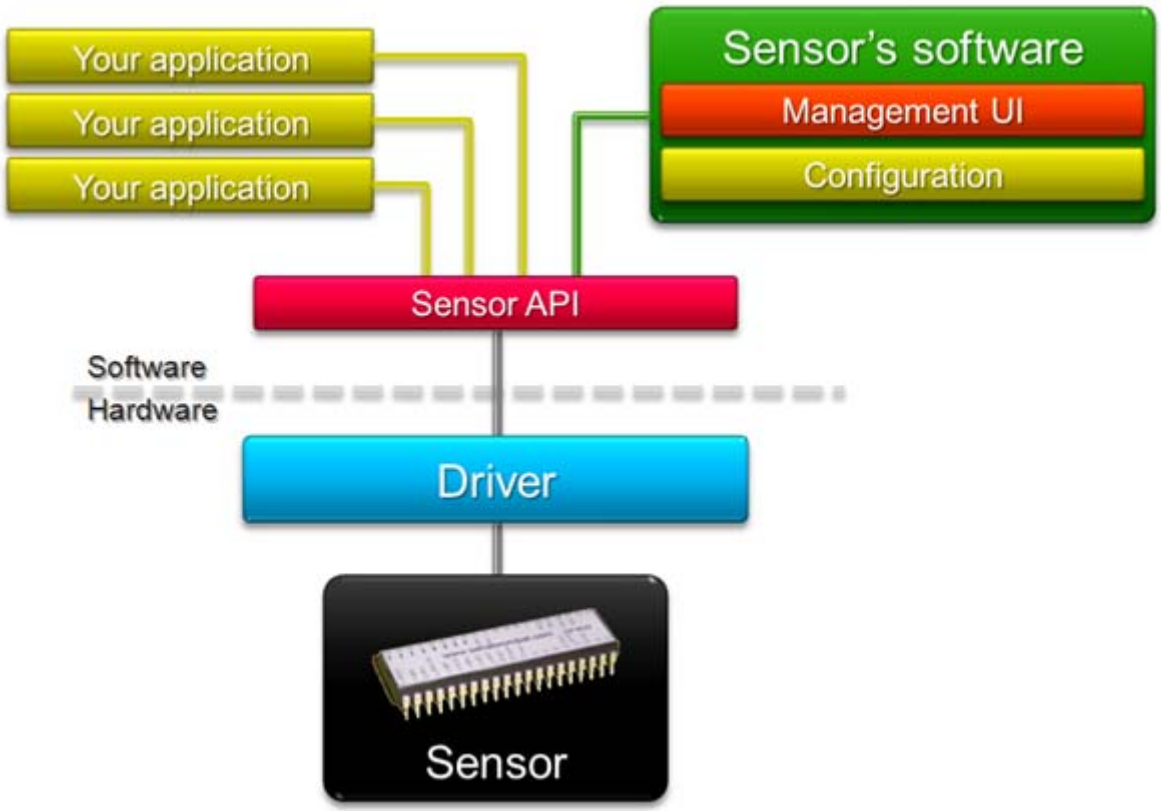
Работа с сенсорами



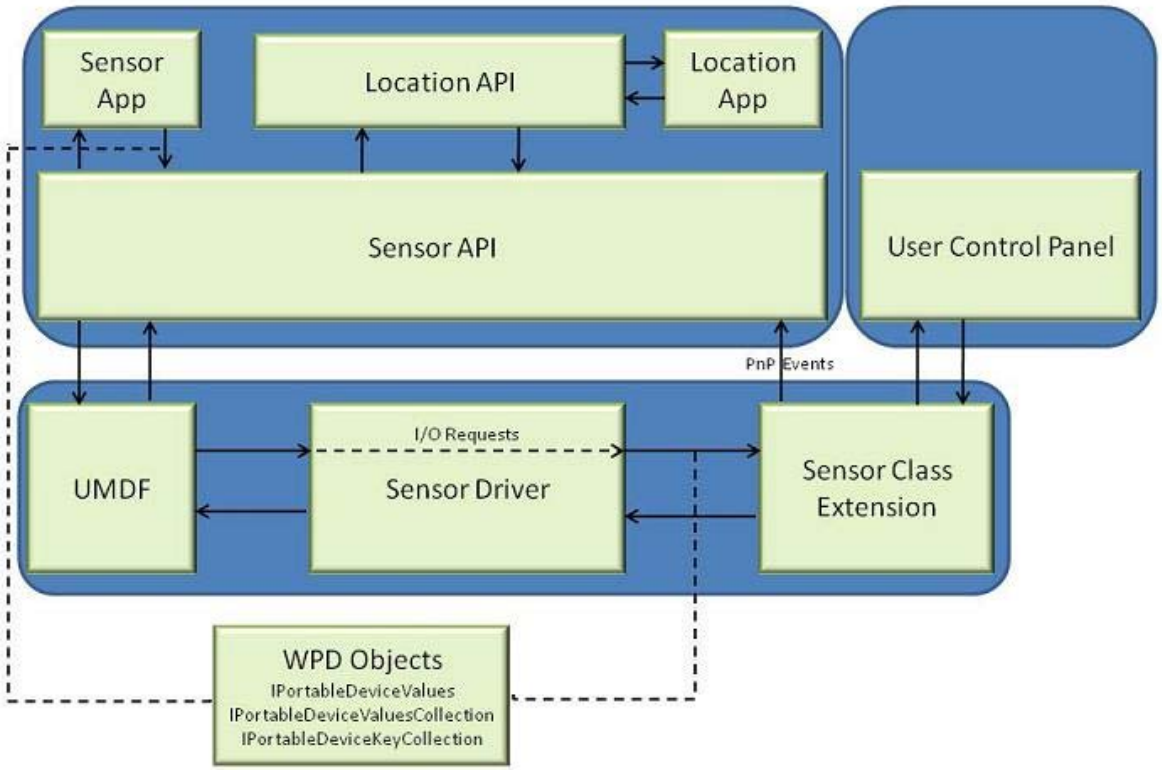
Sensor & Location Platform

- Единая модель API для доступа к устройствам
- физические сенсоры
- логические сенсоры
- Удобный API для получения информации
- Sensor API – полный доступ ко всей информации сенсора
- Location API – абстрактизированный API для доступа к данным о местоположении
- Пользователь имеет возможность управлять разрешениями на доступ приложения к сенсору
- Конкурентный доступ из множества приложения

Работа с сенсорами через Sensor & Location Platform



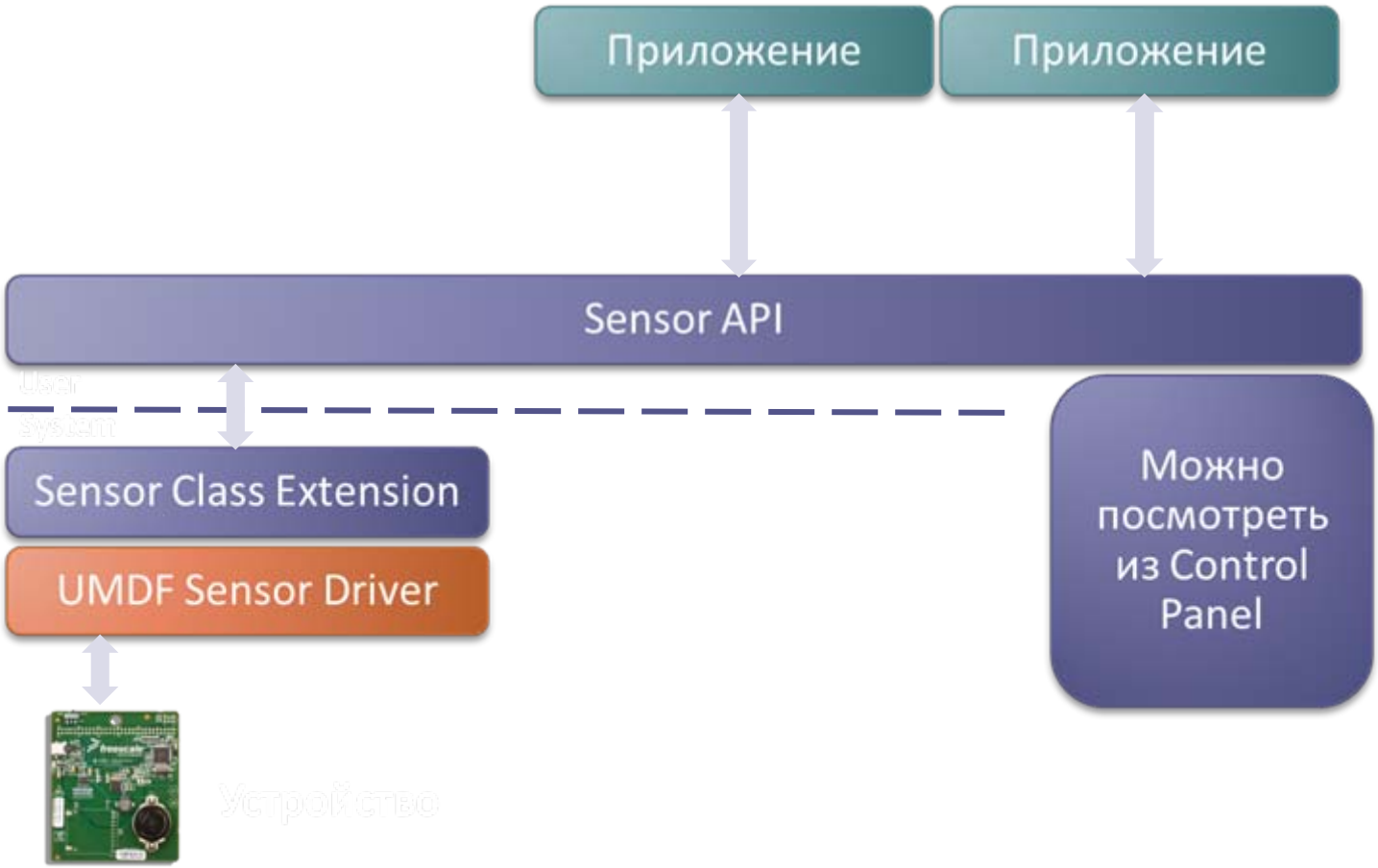
Архитектура



Компоненты

- DDI обеспечивает стандартный способ сенсорам подключаться к компьютеру и передавать данные;
- Sensor API предоставляет набор методов, свойств и событий для работы с подключенными сенсорами и данными, которые поступают с них;
- Location API предоставляет набор программных объектов для работы с информацией о местоположении;
- Location and Other Sensors Control Panel позволяет администраторам системы устанавливать сенсоры для каждого пользователя.

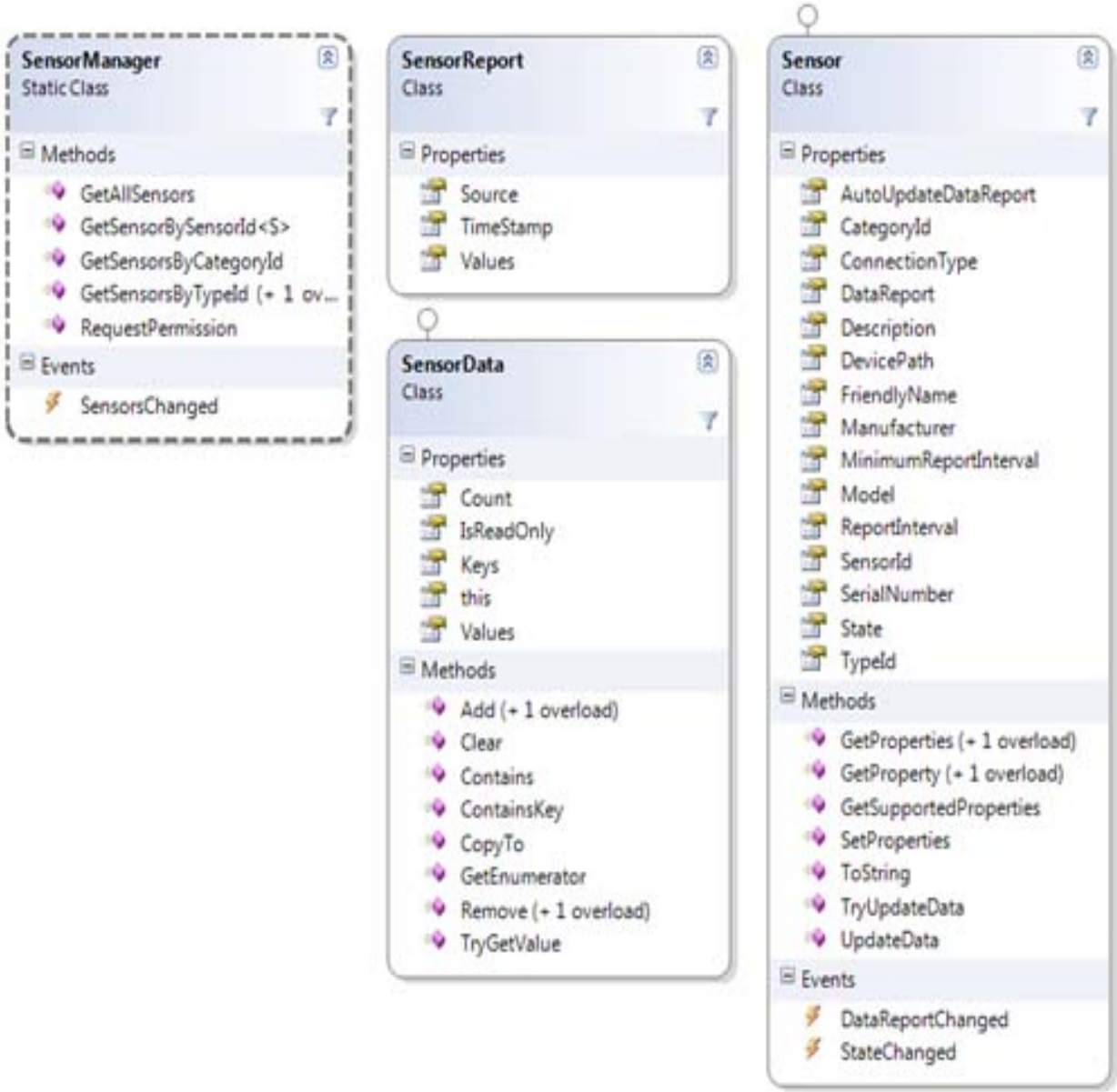
Архитектура доступа к сенсору



Представление сенсоров

- Поиск сенсора по категории и типу
- категория – набор связанных какими-то параметрами между собой сенсоров (например, категория погодная станция могла бы включать скорость ветра, атмосферное давление, температура и т.д.)
- тип – определение физических явлений, которые будут отображаться (скорость)
- Свойства – метаданные, которые может содержать сенсор
- read-only (название сенсора, его серийный номер)
- read-write (интервал опрашивания сенсора)
- Данные
- поля с данными, единицами измерения и типами
- События
- Состояние – определение возможных рабочих состояний

Классы


SensorManager
Static Class
Methods
GetAllSensors
GetSensorBySensorId<S>
GetSensorsByCategoryId
GetSensorsByTypeId (+ 1 ov...
RequestPermission
Events
SensorsChanged
SensorReport
Class
Properties
Source
TimeStamp
Values
SensorData
Class
Properties
Count
IsReadOnly
Keys
this
Values
Methods
Add (+ 1 overload)
Clear
Contains
ContainsKey
CopyTo
GetEnumerator
Remove (+ 1 overload)
TryGetValue
Sensor
Class
Properties
AutoUpdateDataReport
CategoryId
ConnectionType
DataReport
Description
DevicePath
FriendlyName
Manufacturer
MinimumReportInterval
Model
ReportInterval
SensorId
SerialNumber
State
TypeId
Methods
GetProperties (+ 1 overload)
GetProperty (+ 1 overload)
GetSupportedProperties
SetProperties
ToString
TryUpdateData
UpdateData
Events
DataReportChanged
StateChanged

О работе с классами

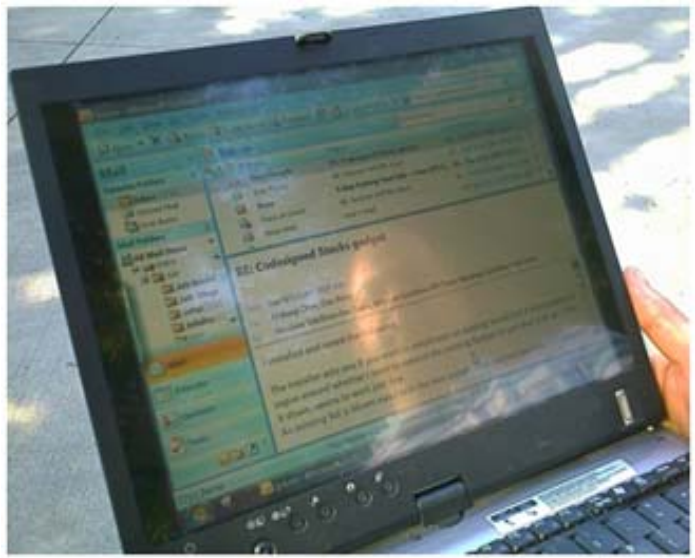
- при изменении состояния сенсора (подключен /отключен/активен/и т.д.) генерируется событие StateChanged
- после установления связи с сенсором генерируется событие DataReportChanged для получения данных
- для считывания состояния сенсора используется метод GetProperty (передаем идентификатор свойства, которое необходимо считать, в параметрах)

Безопасность

- Пользователь должен разрешить работу с сенсором
- Все сенсоры по умолчанию выключены
- Для включения сенсора требуются полномочия администратора
- Сенсоры могут быть настроены для каждого пользователя отдельно
- Диалог включения сенсора инициируется приложением

Оптимизация интерфейса с использованием датчика освещенности

- Увеличение масштаба для улучшения четкости и интерактивности



- Стиль шрифта также может варьироваться для оптимизации четкости и читабельности

Шрифт без засечки

Example text

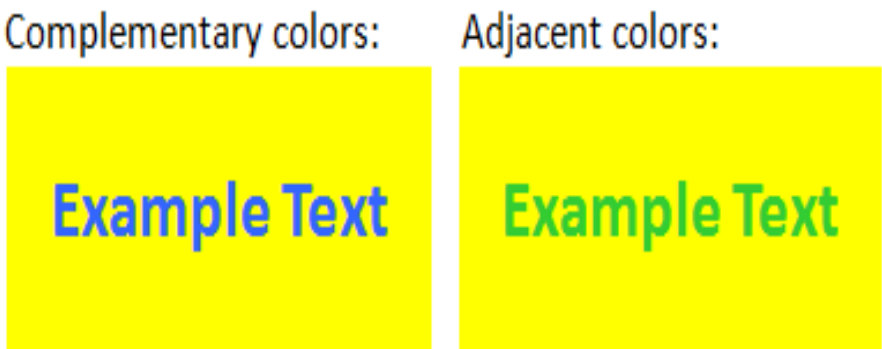
(Verdana, 28pt.)

Шрифт с засечкой

Example text

(Times New Roman, 28 pt.)

- Использование дополнительных цветов как способ улучшения четкости в неблагоприятных условиях



Спасибо за внимание!