

Complex Sensing Event Process of IoT Application Based on EPCglobal Architecture and IEEE 1451

*Chao-Wen Tseng, Chih-Ming Chang, and Chua-Huang Huang
Department of Information Engineering and Computer Science/
Intelligent Identification Technology and Research Center
Feng Chia University, Taichung, Taiwan*

2012.10.25

Outline

- Introduction
- IoT Framework and Management Platform
- Complex Sensing Event Process of EcoPark Application
- Implementation of EcoPark Application
- Conclusions and Future Works

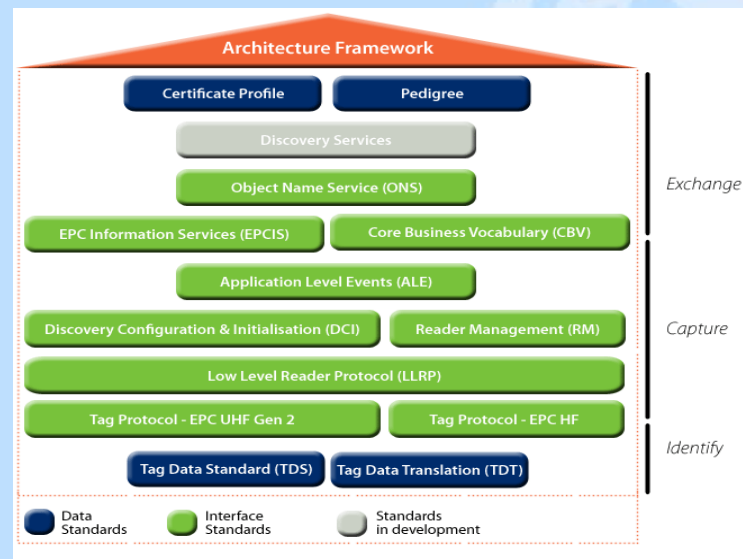
Introduction

- **For achieving intelligent capability, an IoT application system must be able to**
 - **Analyze Object Data**
 - **Reason Sensing Events**
 - **Trigger Responding Actions**
- **This paper presents :**
 - **An IoT framework based on EPCglobal architecture and IEEE 1451 standard**
 - **An IoT management platform to verify the IoT framework**
 - **An EcoPark application to simulate and demonstrate the mechanism of complex sensing event process**

Introduction

➤ EPCglobal Architecture Framework

- EPCglobal was acquired by GS1 in 2003 and has released a collection of standard documents
- ALE middleware has the functions of filtering and consolidating EPCs and related data from one or more data sources.
- EPCIS event repository is located between the capture layer and the exchange layer of EPCglobal architecture framework.

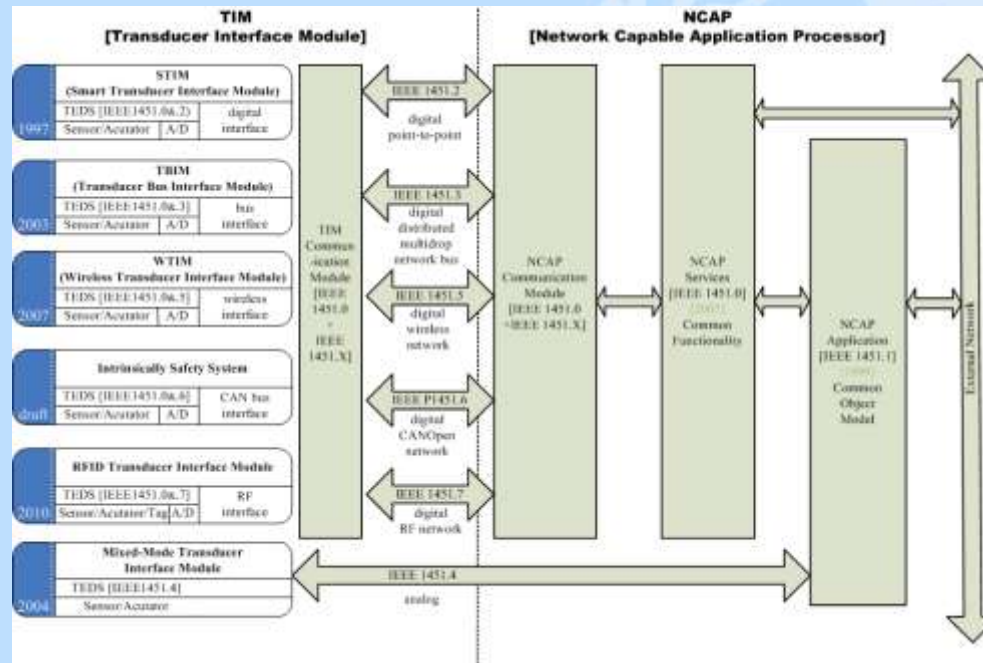


[Source: GS1]

Introduction

➤ IEEE 1451 Standard Series

- IEEE 1451 is a series of smart transducer standards to provide a unified way to access any type, manufacturer, and wired or wireless sensors.
- IEEE 1451 standard series can be divided into two main parts: Transducer Interface Module (TIM) and Network Capable Application Processor (NCAP).

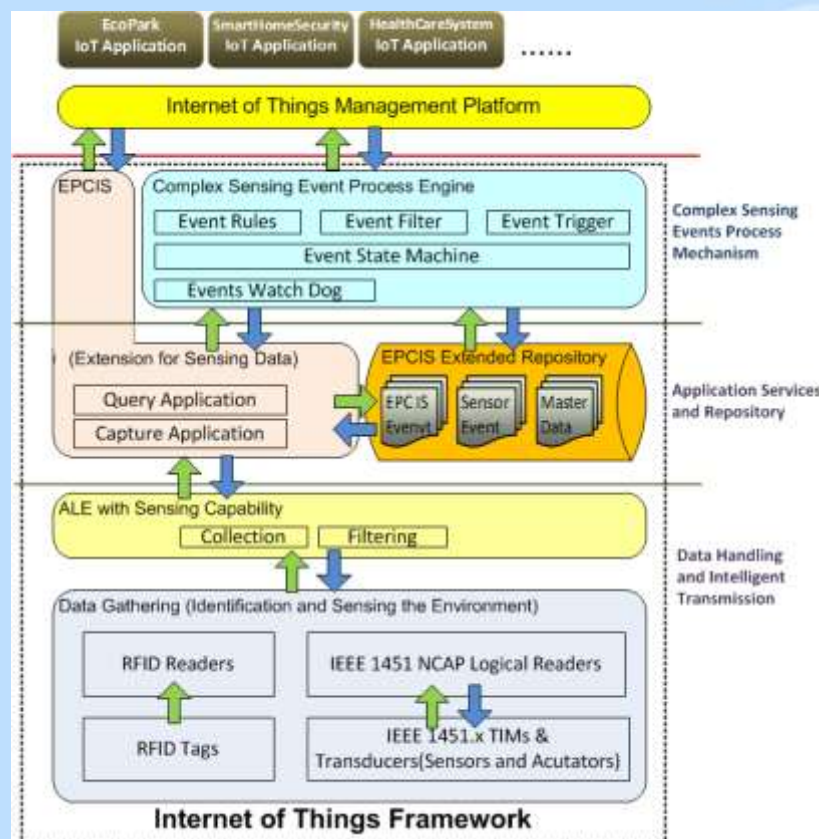


Introduction

- **Complex event processing is a technology originally developed for extracting information from distributed message-based systems and widely applied to various fields.**
- **The perspective of IoT is :**
 - ***One Paradigm, Many Visions***
 - **Most of RFID applications focus on identifying items but sensing the environment is lack of.**
 - **Transducers (sensors and actuators) and intelligent communication technology will also play the other crucial roles in the IoT systems.**

IoT Framework and Management Platform

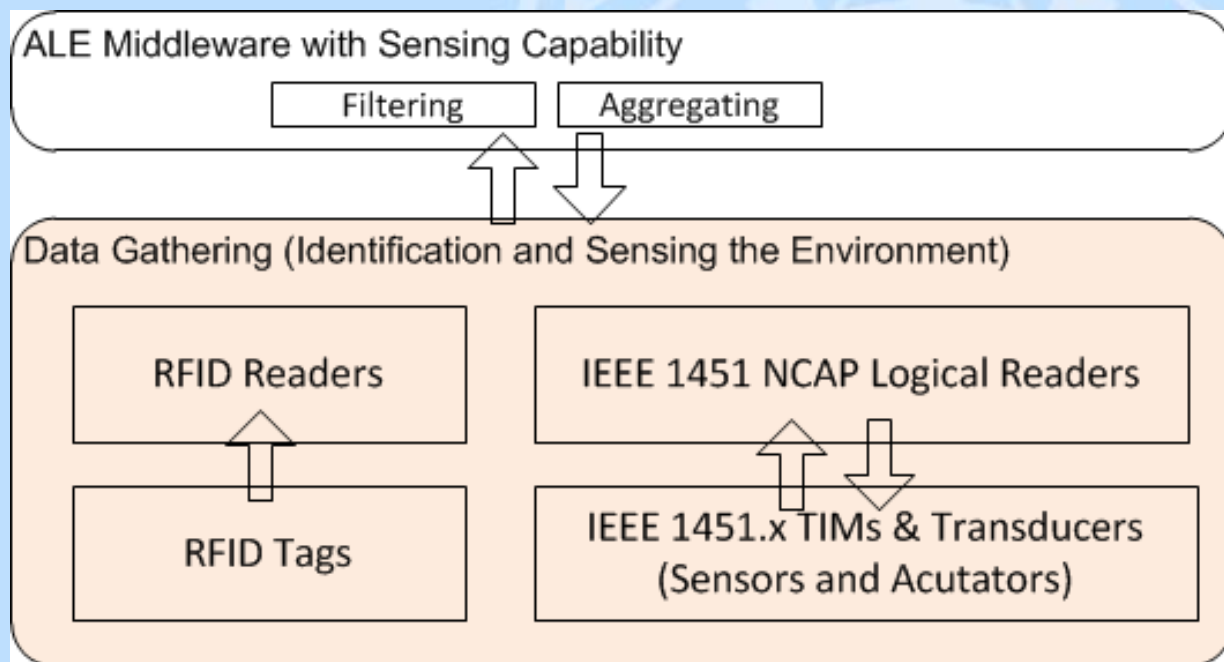
- Using the EPCglobal architecture framework as the foundation, and integrates with IEEE 1451 standards.
- The platform architecture can be built on a variety of network technologies and development tools.



IoT Framework and Management Platform

① Data Collection and Intelligent Transmission

- ✓ The abstraction of RFID items or transducers must be considered at this physical layer.
- ✓ For the general purpose of this framework, we assume all types of RFID tags can be used, not only C1G2.
- ✓ IEEE 1451 supports a plug and play mechanism for smart transducers.



IoT Framework and Management Platform

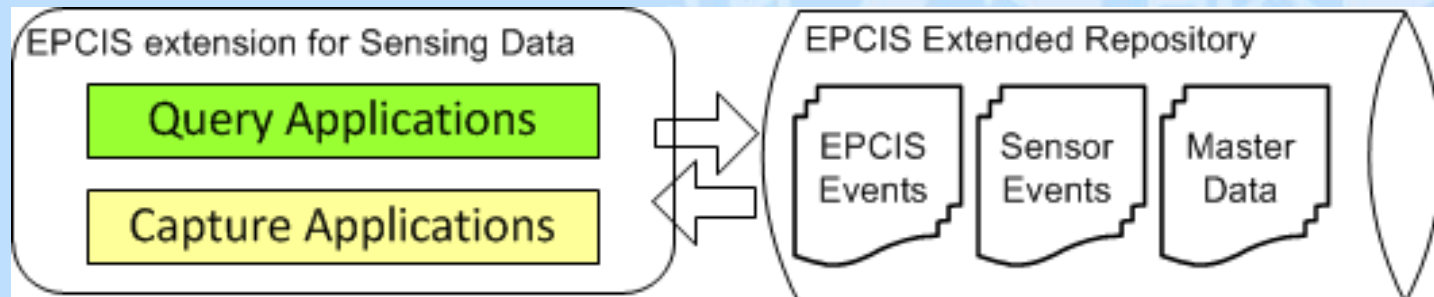
① Data Collection and Intelligent Transmission

- ✓ Through RFID and the IEEE 1451 standards, the application can identify items and sense the environment in the IoT.
- ✓ We extend ALE middleware for gathering, collecting and filtering the data stream of identification and sensing the environment.
 - We stretch the ALE by viewing the NCAP as a customized reader to fit in the EPCglobal architecture framework.
 - These data collected from readers and NCAP are sent to the ALE middleware with sensing capability, which performs filtering and collecting operations of data.
 - ALE middleware will generate the ECRReport according to the event cycle definition, data obtained model (Subscribe, Poll, Immediate), and data filtering and gathering rules.

IoT Framework and Management Platform

② Application Services and Event Repository

- ✓ EPCIS Extended Repository
- ✓ Application Services
 - A. Capture application is used to analyze ECRReport that received from the ALE middleware and to create simple events to for EPCIS.
 - » For handling sensing data, sensor event data is appended to master data in event repository.
 - » It enhances the ability of recording the sensing events and the measured values from sensors.



IoT Framework and Management Platform

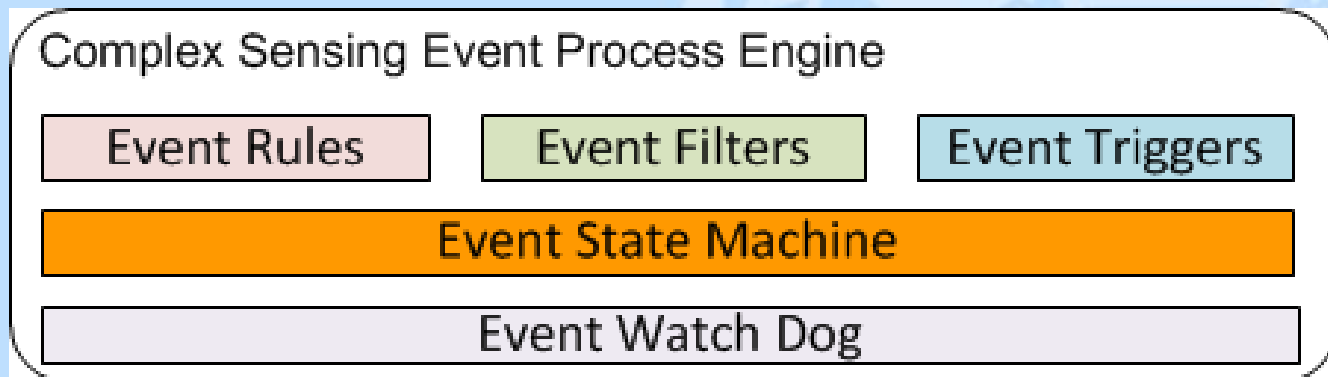
② Application Services and Event Repository

- B. Query application provides the relevant APIs to query events that are stored in the repository.
 - » *real time and historical events query by event process engine*
 - Query application search for simple events to form real time event stream immediately and continuously.
 - It provides event process engine a feature to perform real time analysis and event process.
 - It also provides historical data to the upper layer of applications that support statistical analysis of historical events.
 - » *business vocabulary query according to the master data*
 - According to the pre-defined master data in EPCIS repository, different types of event data can be generated and stored in the event repository.

IoT Framework and Management Platform

③ Complex Sensing Event Process Mechanism

- ✓ The mechanism of complex sensing event process uses the event watch dog to receive real time event stream from EPCIS.
- ✓ Complex event is an event that are generated or aggregated by other events.
- ✓ To find out the significant events (event pattern) from multiple events, and do the follow-up action if they satisfy the event pattern.



IoT Framework and Management Platform

③ Complex Sensing Event Process Mechanism

- ✓ The core handling modules in event process engine consist of:
 - A. Event Filter
 - » *When the simple event stream coming from middleware and does not meet any predefined event patterns, then they are dropped out from the simple events.*
 - B. Event Rules
 - » *Event process engine uses the rules language and the state machine model to describe the event processing rules and rule-based reasoning.*
 - C. Event State Machine
 - » *The results can be sent to event processing engine to facilitate further reasoning and decision making.*
 - D. Event Trigger
 - » *When an event pattern meets the specific conditions, action is triggered by pre-defined rules to make the appropriate response.*

Complex Sensing Event Process of EcoPark Application

➤ EcoPark Application

- It is designed to demonstrate the mechanism of complex sensing event process.
- This application of reusable energy resources in EcoPark is divided into electricity and water.
- It can reduce the manual operations of park management effectively and support the existing system.
- The complex sensing event process is implemented for supporting the IoT management platform.
- Two types of simple event are captured in EPCIS :
 - Object Events
 - Sensor Events

Complex Sensing Event Process of EcoPark Application

● Simple Events of EcoPark Application

Park Resources	Simple Event
Power Resource	<i>EA</i> ₁ : Detection of battery storage capacity. <i>EA</i> ₂ : Detection of LED street light on. <i>EA</i> ₃ : Detection of LED spotlight on. <i>EA</i> ₄ : Detection of start of fountain.
Water Resource	<i>EB</i> ₁ : Set tank water level detection. If the water reservoir level is high, the water flows to the ecological pond; if reservoir water level is low, the water supply switching device is switched to tap water.
Context-aware Environment	<i>EC</i> ₁ : Air temperature detection. <i>EC</i> ₂ : Soil moisture detection. <i>EC</i> ₃ : Detect pedestrians passing. <i>EC</i> ₄ : Detect ambient light intensity (day and night).

Complex Sensing Event Process of EcoPark Application

● Complex Sensing Event Operators

Operator	Notation	Meaning
Disjunction	$(E_1 \vee E_2)$	Event E_1 or E_2 occurs.
Conjunction	$(E_1 \wedge E_2)$	Events E_1 and E_2 occur.
Negation	$(!E)$	Event E does not occur.
Before	$Before(E, t)$	Event E occurs before time t .
After	$After(E, t)$	Event E occurs after time t .
Between	$Between(E, t_1, t_2)$	Event E occurs between time t_1 and t_2 .
Interval	$Interval(E, p)$	Event E occurs within time interval p .
Sequence	$(E_1; E_2)$	The sequence relationship between the events: event 1 occurs, then event 2.
List	(E_1, E_2, E_3)	There is no sequence relationship among the events: events 1, 2, 3 subject to the conditions of time occur.

Complex Sensing Event Process of EcoPark Application

- An example of the EcoPark EPCIS sensor event data

```
- <SensorEvent>
  <eventTime>2012-4-30T20:33:31.116-06:00</eventTime>
  <sensorID>A001</sensorID>
  <action>ADD</action>
  <sensorType>temperature sensor</sensorType>
  <measuredUnits>celsius</measuredUnits>
  <measuredValue>26.28</measuredValue>
  - <readPoint>
    <id>urn:epc:id:sgln:0614141.07347.124</id>
  </readPoint>
  - <evLocation>
    <id> urn:epc:id:sgln:0614141.07346.0</id>
  </evLocation>
</SensorEvent>
```

Complex Sensing Event Process of EcoPark Application

● Trigger Events of EcoPark Application

Park Resource	Trigger Event
Power Resource	<p><i>trigger(EA_a, power_to_battery)</i>: Power save to battery .</p> <p><i>trigger(EA_b, battery_to_power)</i>: Additional power from the battery.</p> <p><i>trigger(EA_c, led_streetlamp_on)</i>: LED street light switch on.</p> <p><i>trigger(EA_d, led_spotligh_on)</i>: LED spotlight switch on.</p> <p><i>trigger(EA_e, start_fountain)</i>: Start fountain motor.</p>
Water Resource	<p><i>trigger(EB_a, rain_to_ecological_pond)</i>: Water flows to Eco Pond.</p> <p><i>trigger(EB_b, piped_water_on)</i>: Switching device to tap water.</p> <p><i>trigger(EB_c, sprinkler_on)</i>: Start sprinkler watering equipment.</p> <p><i>trigger(EB_d, humidifier_on)</i>: Spray mist humidifier device.</p>

Complex Sensing Event Process of EcoPark Application

- Behavior Description Using Event Process Clause

```
SELECT EVENT <event pattern>  
FROM E(e), event stream data  
WHERE <event parameters condition>  
TIME <Before/After/Between/Interval (time express)>  
TRIGGER ACTION <do something>
```

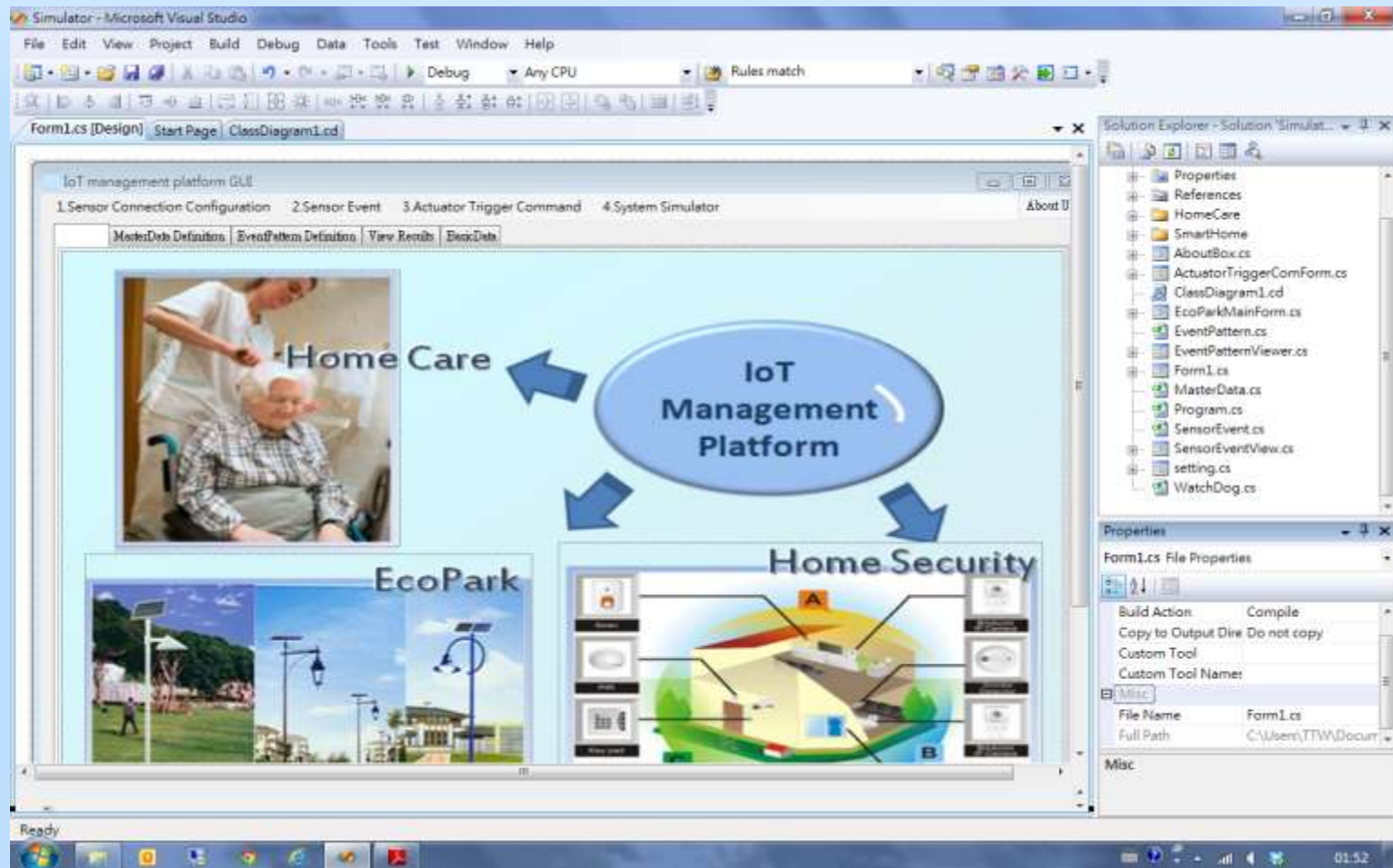
- For Example

Every 15 minutes, the system automatically checks whether the relative humidity of soil moisture in garden is below 20%. If the above criteria meet, it starts the sprinklers.

```
SELECT EVENT (EC2)  
FROM event stream data  
WHERE EC2.d<20 //relative humidity below 20%  
^ (EBc.evloc=EC2.evloc)  
TIME Interval 15 // (minute)  
TRIGGER ACTION trigger(EBc, sprinkler_on)
```

Implementation of EcoPark Application

➤ The IDE of IoT Management Platform



Implementation of EcoPark Application

➤ Core Classes of IoT Management Platform

The image displays a screenshot of the Visual Studio IDE showing the class hierarchy for the EcoPark application. The classes are organized into a grid of panels, each representing a class with its fields and methods.

- MasterData Class**
 - Fields: VocabularyElement
 - Methods: LoadFromXml, MasterData, ToSimpleXML, ToXML, ToXmlReader
- WatchDog Class**
 - Fields: eventPattern, preSensorEvent, TriggerXml, vocabularyElement
 - Methods: getEventPattern, initEventPattern, readEvent, WatchDog
- EventPattern Class**
 - Fields: SELECT_statement, SELECT_statement_status, TIME_statement, TIME_statement_status, TRIGGER_ACTION_stateme..., WHERE_statement, WHERE_statement_status
 - Methods: EventPattern, LoadEventPattern
- EventPatternViewer Class**
 - Fields: (empty)
 - Methods: Dispose, EventPatternViewer, InitializeComponent, ViewInTreeView
- SensorEvent Class**
 - Fields: action, eventTime, evLocation, measuredUnits, measuredValue, readPoint, sensorID, sensorType
 - Methods: (empty)
- SensorEventView Class**
 - Fields: (empty)
 - Methods: AddNode, Dispose, InitializeComponent, listBox1_SelectedI..., SensorEventView, ViewInTreeView, viewInWebbrower
- ActuatorTriggerComForm Class**
 - Fields: (empty)
 - Methods: ActuatorTriggerComForm, button1_Click, Dispose, InitializeComponent
- TRIGGER_ACTION_statement Class**
 - Fields: trigger_actuator_id, trigger_method, WHERE_statement
- EcoParkMainForm Class**
 - Fields: (empty)
 - Methods: (empty)

Implementation of EcoPark Application

➤ User Interface of Master Data Definition

The screenshot displays the 'EcoPark IoT management platform GUI' with the 'MasterData Definition' tab selected. The interface is divided into several sections:

- Navigation:** 1.Sensor Connection Configuration, 2.Sensor Event, 3.Actuator Trigger Command, 4.System Simulator, and About Us.
- Sub-tabs:** MasterData Definition, EventPattern Definition, and View Results.
- Form Fields:**
 - Vocabulary Type: SensorEventMapping (dropdown)
 - Sensor ID: EA1 (text input)
 - Sensor Type: power sensor (text input)
- Buttons:** Add, Delete, Save, and Load Master Data.
- Table:** A table listing sensor data with columns 'SensorID' and 'SensorType'. The row for EA1 (power sensor) is highlighted.
- XML Preview:** A text area showing the generated XML document structure, including the root element <EPCISMasterDataDocument> and a list of vocabulary elements for each sensor ID.
- Footer Buttons:** Setting and Exit.

SensorID	SensorType
EA1	power sensor
EA2	led streetlamp sensor
EA3	led spotlight sensor
EA4	fountain sensor
EB1	water sensor
EC1	temperature sensor
EC2	humidity sensor
EC3	pedestrian detect sensor
EC4	light sensor

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<epcismd:EPCISMasterDataDocument
xmlns:epcismd="urn:epcglobal:epcis-masterdata:xsd:1"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
schemaVersion="1"
creationDate="2011-05-14T14:05:36.0Z">
<EPCISBody>
<VocabularyList>
<Vocabulary
type="urn:IOT:EcoPark:SensorEventMapping">
<VocabularyElementList>
<VocabularyElement id="EA1">power
sensor</VocabularyElement>
<VocabularyElement id="EA2">led streetlamp
sensor</VocabularyElement>
<VocabularyElement id="EA3">led spotlight
sensor</VocabularyElement>
<VocabularyElement id="EA4">fountain
sensor</VocabularyElement>
<VocabularyElement id="EB1">water
```

Implementation of EcoPark Application

➤ Simulation Result of EcoPark IoT Application

The screenshot displays the EcoPark Management System interface. On the left, there are three buttons labeled 'Case 1', 'Case 2', and 'Case 3'. A flow diagram shows 'EventPattern' leading to 'Sorting rules'. A red circle highlights a log window with the following text:

```
AM 06:00:00 and 2012/5/8 PM 06:00:00  
Event Filter occurred! => EC3  
Rules match: EC4 >200(lux)  
Rules match: Time Between 2012/5/8  
AM 06:00:00 and 2012/5/8 PM 06:00:00  
Rules match: EC1 >33(celsius)  
Rules match: Time Between 2012/5/8  
AM 06:00:00 and 2012/5/8 PM 06:00:00  
Rules match: Time Within 60,second  
Event pattern match!
```

Below the log, the text '1.Event Rules Matched!' is displayed. The main data area contains two tables. The first table is titled 'Data Source : NCAP' and 'Sensor Type : Sensor'. It has columns for eventTime, sensorID, action, sensorType, measuredUnits, measuredValue, readPoint, and evLo. The second table is titled 'Data Source : NCAP' and 'Sensor Type : Actuator'. It has columns for trigger_actuator_id, trigger_method, and trigger_condition. A red circle highlights the row in the second table where trigger_actuator_id is 'EBd', trigger_method is 'humidifier_on', and trigger_condition is 'trigger_condition: EBd.t=time INTERVAL 10minute'. The text '2.Trigger an Action!!!' is displayed next to this row.

eventTime	sensorID	action	sensorType	measuredUnits	measuredValue	readPoint	evLo
2012-05-08T07:37:50	EC1	ADD	temperature sensor	celsius	30	um:epc:id:sgln:0614141.07347.124	um:ep
2012-05-08T07:38:15	EC3	ADD	pedestrian detect sensor	bit	1	um:epc:id:sgln:0614141.07347.130	um:ep
2012-05-08T07:39:08	EC4	ADD	light sensor	lux	800	um:epc:id:sgln:0614141.07347.131	um:ep
2012-05-08T07:39:12	EC1	ADD	temperature sensor	celsius	34	um:epc:id:sgln:0614141.07347.124	um:ep
*							

trigger_actuator_id	trigger_method	trigger_condition
EBd	humidifier_on	trigger_condition: EBd.t=time INTERVAL 10minute
*		

Conclusions and Future Works

- This paper proposed IoT framework based on EPCglobal and IEEE 1451 standards and toward to an open environment.
- The EcoPark application based on IoT management platform shows the feasibility of IoT framework and the mechanism of complex sensing event process.
- The detailed extension on design issues :
 - RFID and Sensor Integration
 - ALE with Sensing Capability
 - EPCIS Repository Extension for Transducer