AARHUS
UNIVERSITET

ADVANCED PERVASIVE COMPUTING

Lecture 2: Distributed Context Awareness

Stefan Wagner
sw@eng.au.dk

AGENDA

- Context
- Context Awareness
- Context Modelling & Reasoning
- Distributed Context Awareness
- EnablingTechnologies
- PAN technologies
- LAN/WAN technologies
- Frameworks for Distributed Context Awareness:
- The Context Toolkit
- The Java Context Awareness Framework (JCAF)

AARHUS
UNIVERSITET

CONTEXT

› Context is derived from the Latin word: contextus
› from con- “together” and texere “to weave”.

Context is fundamental to human cognition, which is the act or process of knowing, including perceiving, recognizing, conceiving, and reasoning. Words, sentences, images, and experiences can be fundamentally differently interpreted when served to the user in a different context
McCracken & Wolfe (2004)

Context is the circumstances that form the setting for an event, statement, or idea, and in terms of which it can be fully understood
Oxford Dictionay (2012)

CONTEXT

› Plenty of definitions of context exists in pervasive computing:

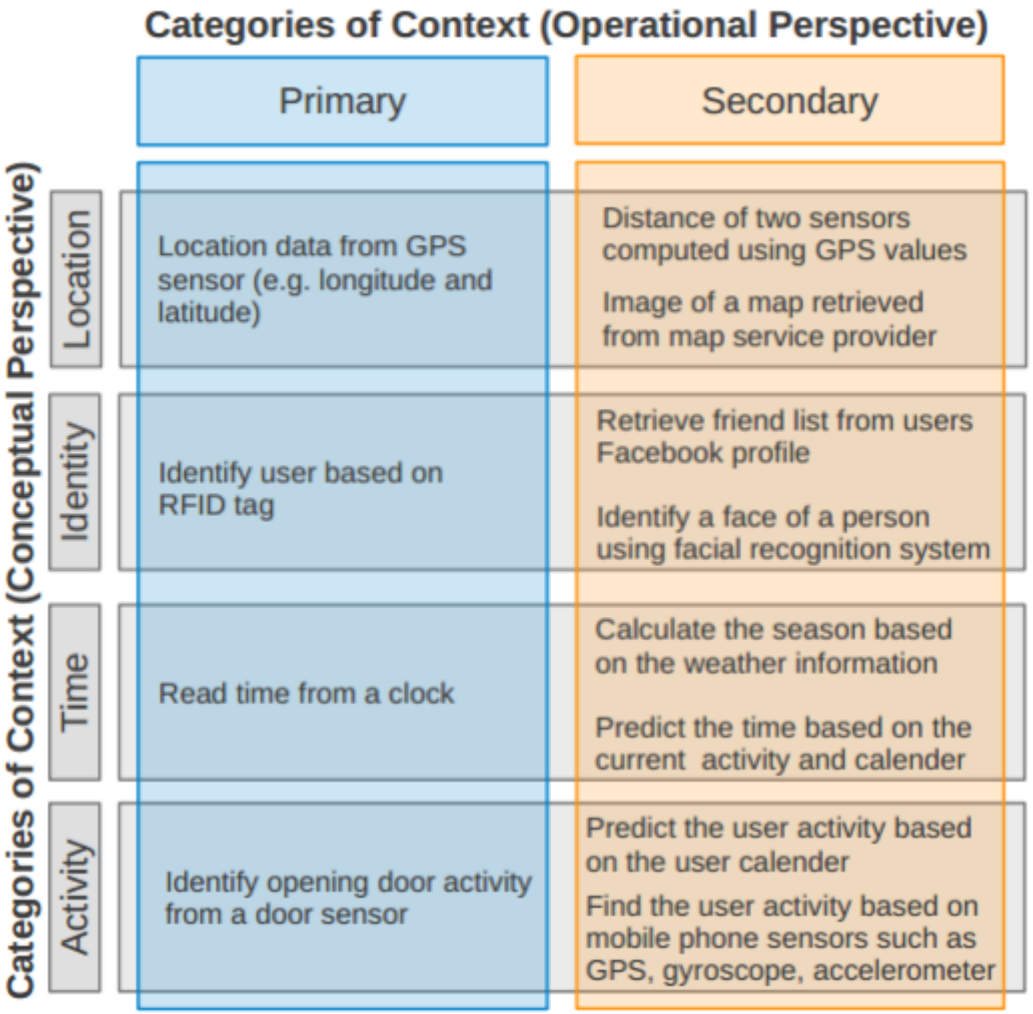
1. A context describes a situation and the environment a device or user is in.
2. A context is identified by a unique name
3. For each context a set of features is relevant.
4. For each relevant feature a range of values is determined (implicit or explicit) by the context.
Schmidt (1999)

A set of environmental states and settings that either determines an application's behavior or in which an application event occurs and is interesting to the user
Chen & Kotz (2000)

Any information that can be used to characterize the situation of an entity. An entity is a person, place, or object that is considered relevant to the interaction between a user and an application, including the user and the applications them-selves
Dey (2010)

The five W's (Who, What, Where, When, Why) is the minimum information that is necessary to understand context.
Abowd and Mynatt (2010)

CONTEXT


Categories of Context (Operational Perspective)
Primary
Secondary
Categories of Context (Conceptual Perspective)
Location
Location data from GPS sensor (e.g. longitude and latitude)
Distance of two sensors computed using GPS values
Image of a map retrieved from map service provider
Identity
Identify user based on RFID tag
Retrieve friend list from users Facebook profile
Identify a face of a person using facial recognition system
Time
Read time from a clock
Calculate the season based on the weather information
Predict the time based on the current activity and calender
Activity
Identify opening door activity from a door sensor
Predict the user activity based on the user calender
Find the user activity based on mobile phone sensors such as GPS, gyroscope, accelerometer

CONTEXT AWARENESS

› Again, several definitions exists:

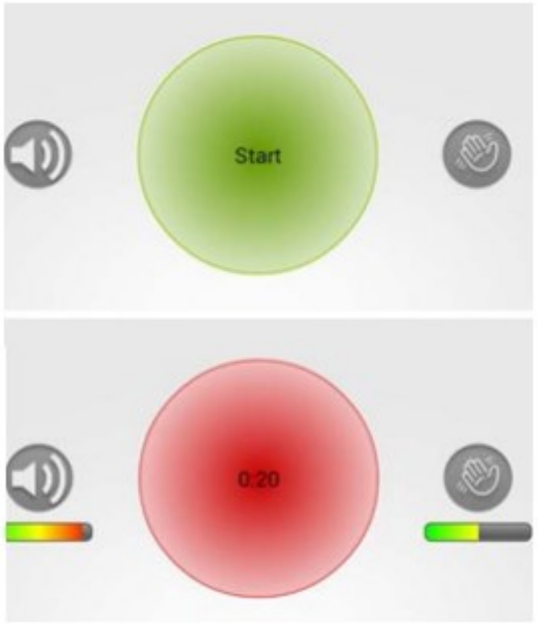
Context-aware computing is the ability of a mobile user's applications to discover and react to changes in the environment they are situated in. In our system mobile users run software that is constantly monitoring, or subscribing to information about the world around them.
Schilit & Thiemer (1994)

A system is context-aware if it uses context to provide relevant information and/or services to the user, where relevancy depends on the user's task.
Dey (2010)

CONTEXT AWARENESS

› Three features that context-aware applications may support:
› 1) Presentation of information and services to a user
› 2) Automatic execution of a service
› 3) Tagging of context to information for later retrieval

Abowd et al. (1999)



AARHUS
UNIVERSITET

CONTEXT AWARENESS

› Two main categories of context awareness:
› 1) Infrastructure-based context awareness
› 2) Self-contained context awareness

Varshney (2009)

Date	Adherence Approved	Systolic mmHg	Diastolic mmHg	Pulse BPM	Time Seated seconds	Talk Detected % of time
11-2012 09:46:48	☐	140	66	73	263	0 %
11-2012 09:47:53	☑	130	61	72	328	0 %
11-2012 09:48:48	☑	126	61	70	383	0 %
11-2012 09:52:22	☐	122	81	99	103	0 %
11-2012 09:53:02	☐	114	80	98	144	0 %
11-2012 09:53:40	☐	110	80	96	181	0 %
11-2012 09:54:18	☐	118	86	99	219	0 %

AARHUS
UNIVERSITET

CONTEXT TAGGING

Hi you have just received the following results from the Adherence Logger

Patient id:mia
Start date: Thu Oct 25 2012 Start time: 22:39
End date: Wed Oct 24 2012 End time: 23:03
Email sent to: dk.iha.sensorlogger@gmail.com

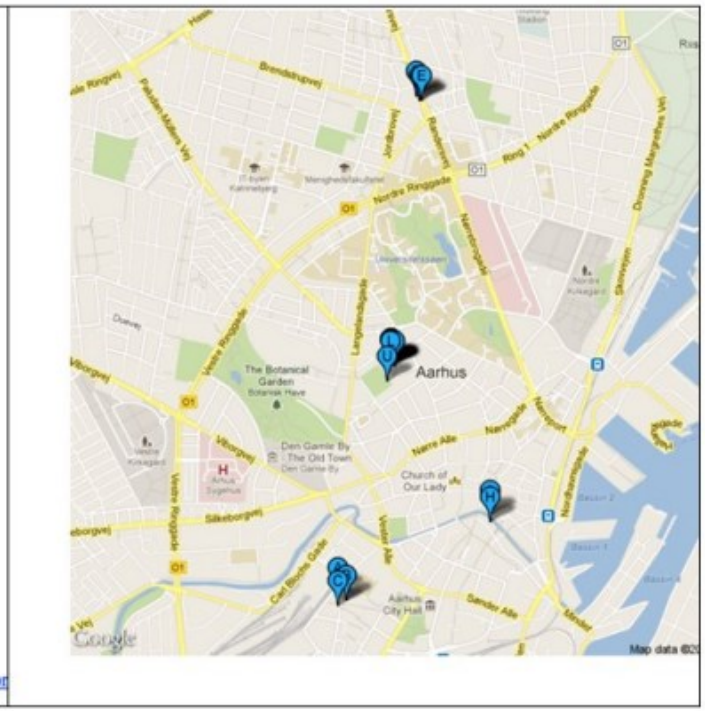
The table below shows your movement for the past 24 hours

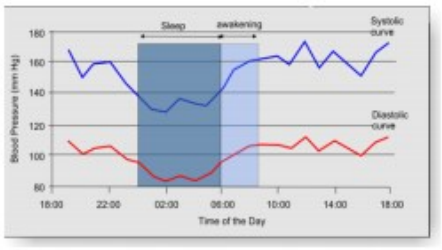
Time	Battery	Movement	Position
22:39	9	Low	NA,NA
22:24	9	Low	NA,NA
22:09	10	Low	NA,NA
22:03	12	Low	56.1632,10.20001

EDITED TO REDUCE SIZE

00:24	58	Low	NA,NA
00:09	59	Low	NA,NA
00:03	61	Low	56.16329,10.19996
23:39	61	Low	NA,NA
23:24	62	Low	NA,NA
23:09	64	High	NA,NA
23:03	65	High	56.16331,10.19985

Click on below link for map view of your movement for the past 24 hours:
http://maps.googleapis.com/maps/api/staticmap?&zoom=auto&size=600x300&maptype=roadmap&markers=color:blue%7Clabel:A%7C56.1632,10.20001&markers=color:blue%7Clabel:B%7C56.16329,10.20035&markers=color:blue%7Clabel:C%7C56.16319,10.20012&markers=color:blue%7Clabel:D%7C56.16332,10.19989&markers=color:blue%7Clabel:E%7C56.16312,10.19999&markers=color:blue%7Clabel:F%7C56.16329,10.19998&markers=color:blue%7Clabel:G%7C56.16323,10.21116&sensor





CONTEXT MODELS

A context model identifies a concrete subset of the context that is realistically attainable from sensors, applications and users and able to be exploited in the execution of the task. The context model that is employed by a given context-aware application is usually explicitly specified by the application developer, but may evolve over time.

A context attribute is an element of the context model describing the context. A context attribute has an identifier, a type and a value, and optionally a collection of properties describing specific characteristics
Henricksen (2003)

CONTEXT MODELS

› Key-Value models

› These models represent the simplest data structure for context modeling. They are frequently used in various service frameworks, where the key-value pairs are used to describe the capabilities of a service. Service discovery is then applied by using matching algorithms which use these key-value pairs.

› Markup scheme models

› All markup based models use a hierarchical data structure consisting of markup tags with attributes and content.

› Object oriented models

› Modeling context by using object-oriented techniques offers to use the full power of object orientation (e.g., encapsulation, reusability, inheritance). Existing approaches use various objects to represent different context types (such as temperature, location, etc.), and encapsulate the details of context processing and representation.

CONTEXT MODELS

› Logic based models

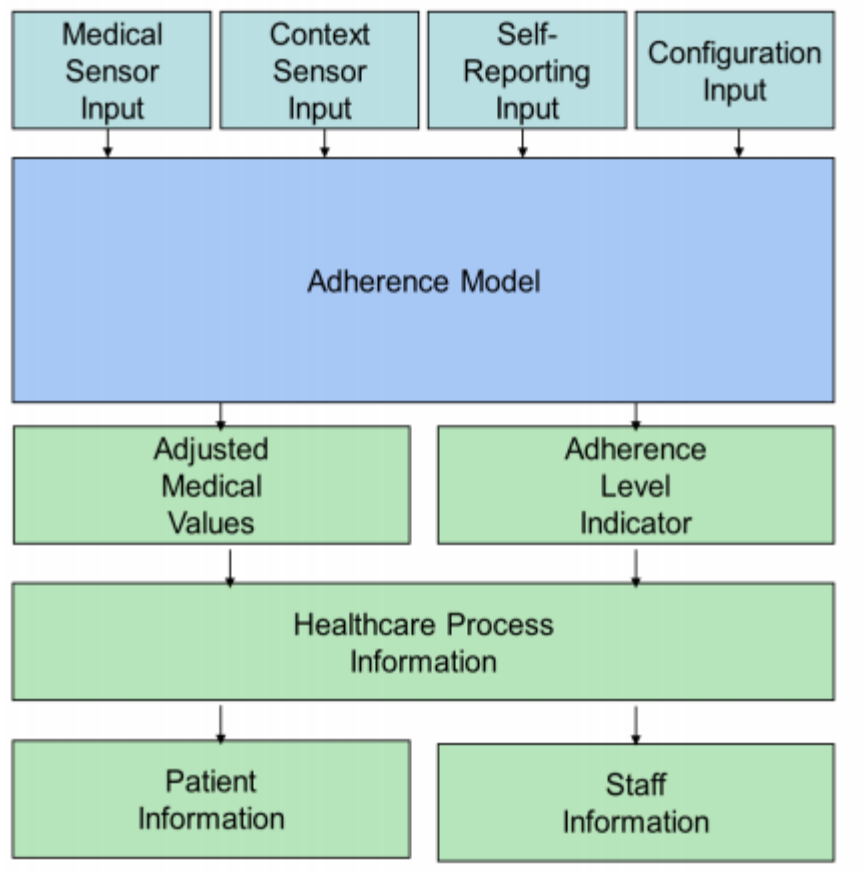
› Logic-based models have a high degree of formality. Typically, facts, expressions and rules are used to define a context model. A logic based system is then used to manage the aforementioned terms and allows to add, update or remove new facts. The inference (also called reasoning) process can be used to derive new facts based on existing rules in the systems. The contextual information needs to be represented in a formal way as fact.

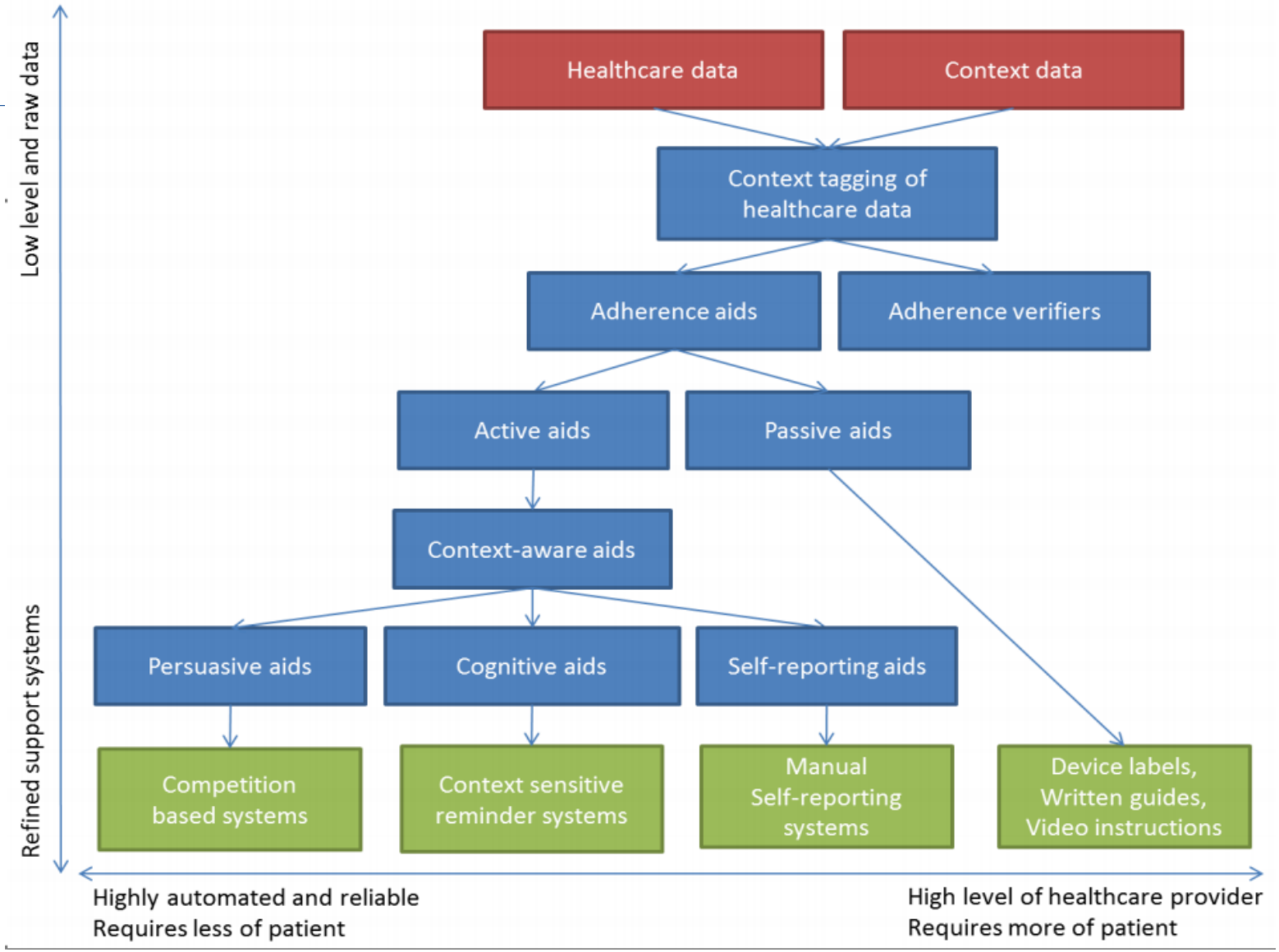
› Ontology based models

› Ontologies represent a description of the concepts and relationships. Therefore, ontologies are a very promising instrument for modeling contextual information due to their high and formal expressiveness and the possibilities for applying ontology reasoning techniques. Various context-aware frameworks use ontologies as underlying context models

CONTEXT MODELS

› Adherence Strategy Engineering Toolkit (ASET)




Low level and raw data
Refined support systems
Healthcare data
Context data
Context tagging of healthcare data
Adherence aids
Adherence verifiers
Active aids
Passive aids
Context-aware aids
Persuasive aids
Cognitive aids
Self-reporting aids
Competition based systems
Context sensitive reminder systems
Manual Self-reporting systems
Device labels, Written guides, Video instructions
Highly automated and reliable
Requires less of patient
High level of healthcare provider
Requires more of patient

CONTEXT MODELS

› Adherence Model Markup Language (AMML)

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns="AMML" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
targetNamespace="AMML">
    <xsd:complexType name="T_AdherenceModel">
        <xsd:sequence>
            <xsd:element ref="PatientDiseaseManagements"/>
        </xsd:sequence>
    </xsd:complexType>
       <xsd:complexType name="T_PatientDiseaseManagements">
        <xsd:sequence>
            <xsd:element ref="DiseaseManagement"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="T_Patient">
        <xsd:sequence>
            <xsd:element ref="PatientID"/>
            <xsd:element ref="Name"/>
            <xsd:element ref="Adress"/>
            <xsd:element ref="DateOfBirth"/>
            <xsd:element ref="City"/>
            <xsd:element ref="Country"/>
        </xsd:sequence>
    </xsd:complexType>
       …
       <xsd:element name=" AdherenceModel" type="T_ AdherenceModel"/>
       …
  </xsd:schema>
```

AARHUS
UNIVERSITET

CONTEXT MODELS

› AMML sample document

```
<HealthCareProcesses>
    <HealthCareProcess>
        <StartDate>2012-12-12T00:00:00</StartDate>
        <EndDate>2012-12-15T00:00:00</EndDate>
        <EstimatedDuration/>
        <Name>Diagnostic Home Blood Pressure Self-Measurement Session</Name>
        <InterventionDescription>For a period of 3 days self-measure your ....</InterventionDescription>
        <Recommendations>Please adhere to the recommendations of the ....</Recommendations>
        <Actions>
            <Action>
                <Name>blood pressure self-measurement morning</Name>
                <Description>Take blood pressure with contextual ... </Description>
                <DataUnits>
                    <DataUnit xsi:type="ContextData">
                        <Description>Time of day service performed</Description>
                        <Key>Time of day</Key>
                        <Value>6.34909E+17</Value>
                        <Registered>2012-12-12T14:00:52.9600512</Registered>
                        <SelfReported>false</SelfReported>
                        <Adhered>false</Adhered>
                    </DataUnit>
                    <DataUnit xsi:type="MedicalMeasurementData">
                        <Name>Systolic Blood Pressure</Name>
                        <Description>Measured Systolic blood pressure value</Description>
                        <Key>Systolic</Key>
                        <Value>147</Value>
                        <UnitSI>mmHg</UnitSI>
                        <Registered>2012-12-12T14:00:52.9600512</Registered>
                        <SelfReported>false</SelfReported>
                        <Adhered>false</Adhered>
                    </DataUnit>
    </HealthCareProcess>
</HealthCareProcesses>
```

CONTEXT MODELS

› Using AMML with ASET

```
<Rules>
    <Rule>
        <CollectedDataOnly>false</CollectedDataOnly>
        <EvaluationValueInIntervalOnly>true</EvaluationValueInIntervalOnly>
        <EvaluationValueNotInIntervalOnly>false</EvaluationValueNotInIntervalOnly>
        <RuleName>Time of day</RuleName>
        <EvaluationValues>
            <EvaluationValue>
                <ValueKey>Time of day</ValueKey>
                <ValueData>6.34908935E+17</ValueData>
                <ValueDataFormatted>07:00:00</ValueDataFormatted>
                </EvaluationValue>
                <EvaluationValue>
                <ValueKey>Time of day</ValueKey>
                <ValueData>6.34909E+17</ValueData>
                <ValueDataFormatted>10:00:00</ValueDataFormatted>
```

```
    </Rule
</Rules>
```

```
        static void Main(string[] args)
        {
            //Create an instance of the AMML Conversion Utility
            var converter = new AMMLConversionUtil();
            //Validate an existing document
            if (converter.ValidateAMMLDocument(@"RAM.xml", @"AMMLSchema.xsd", "AMML"))
            {
                //Convert it to an object model in order to illustrate the tool
                var aset = converter.DeserializeFromXML(@"RAM.xml");
                //And convert the object model back to AMML XML
                converter.ConvertToAMML(aset, @"RAM3.xml");
            }
```

CONTEXT REASONING

› Context Reasoning and Interpretation

- › Raw context data is often too noisy and too vague to provide the true context state of an entity
- › By combining current context event data with historic events from one or mulitple context sensors, this will allow us to determine a ”hidden state” based on the sensor fusion of one or more ”visible states”
- › Various approaches possible: Rule based logic, Hidden Markov Models, Bayesian Networks, Artificial Neural Networks, Support Vector Machines, Ontology based, Fuzzy reasoning, k-nearest neighbor

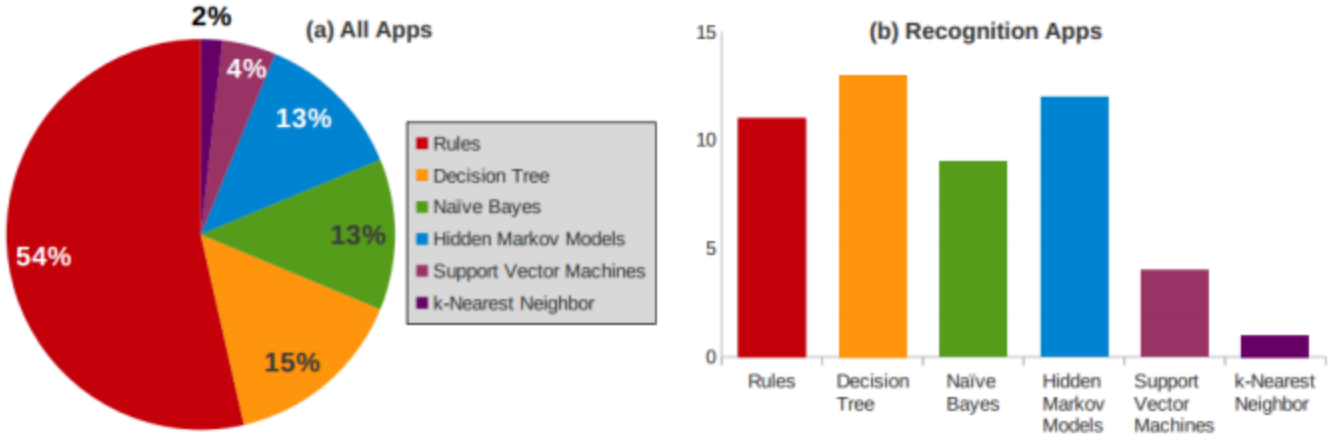
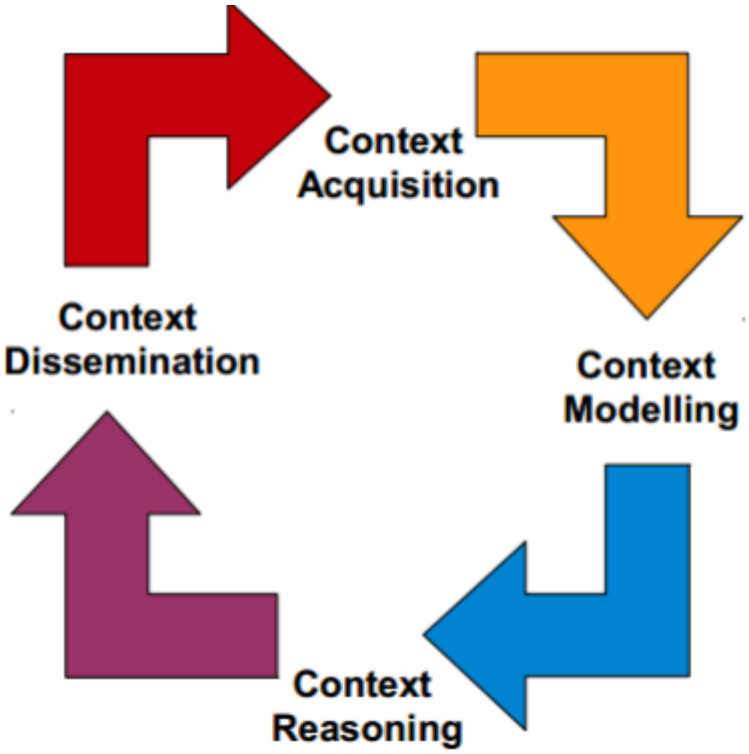


Fig. 7. (a) Counts of model types used in 109 of 114 reviewed context-aware applications. (b) Counts for 50 recognition applications; classifiers are used most often for applications that do recognition [108].

CONTEXT LIFECYCLE



AARHUS
UNIVERSITET

ENABLING TECHNOLOGIES

- Sensors
- Movement, proximity, distance, sound, temperature, light, gas, pressure, acceleration, gyroscope, magnetometer, altimeter

AARHUS
UNIVERSITET

ENABLING TECHNOLOGIES

› Sensor Platforms

› Personal computer, smart phone, tablet, smart watch, mobile sensor node, smart space computer (car, room, street-light)

ENABLING TECHNOLOGIES

- Programming API's
- Easy access to tablet/smart phone/smart watch sensors
- Easy access to computer based prototyping with context sensors

```
public class HelloAndroid extends Activity implements SensorEventListener {
        private SensorManager sensorManager;

        @Override
        public void onCreate(Bundle savedInstanceState){

                super.onCreate(savedInstanceState);
                setContentView(R.layout.main);

                sensorManager=(SensorManager)getSystemService(SENSOR_SERVICE);
                // add listener. The listener will be HelloAndroid (this) class
                sensorManager.registerListener(this,
                                sensorManager.getDefaultSensor(Sensor.TYPE_ACCELEROMETER),
                                SensorManager.SENSOR_DELAY_NORMAL);

       public void onSensorChanged(SensorEvent event){

                // check sensor type
                if(event.sensor.getType()==Sensor.TYPE_ACCELEROMETER){

                        // assign directions
                        float x=event.values[0];
                        float y=event.values[1];
                        float z=event.values[2];

                }
        }
}
```

PHIDGETS OS SUPPORT

Operating System	Drivers and Libraries	Direct Control	Remote Network Control	Supported Version
Desktop OSes				
Windows	Quick Downloads	✓	✓	XP SP3 or Newer
OS X	Quick Downloads	✓	✓	OS X 10.4 or newer
Linux	Quick Downloads	✓	✓	Kernel 2.6 or newer
Mobile/Wireless OSes				
Phidget SBC	Quick Downloads	✓	✓	All versions
Android	Quick Downloads	✓	✓	3.1 and newer, with USB port
Android	Quick Downloads	X	✓	1.5 to 3.0
iOS	Quick Downloads	X	✓	3.0 or newer
Windows CE	Quick Downloads	✓	✓	5.0 or newer

Language	Libraries	API	Code Samples†	Events	Logic Code	Use via Direct USB*	Phidget WebService	Native Library**	Phidget User Base	OS Support
Core Languages										
C#	Quick Downloads	All Devices	All Devices	✓	✓	✓	✓	✓		
C/C++	Quick Downloads	All Devices	All Devices	✓	✓	✓	✓	✓		
Java	Quick Downloads	All Devices	All Devices	✓	✓	✓	✓	✓		
Python	Quick Downloads	All Devices	All Devices	✓	✓	✓	✓	✓		
Cocoa	Quick Downloads	All Devices	All Devices	✓	✓	✓	✓	✓		
Visual Basic 6.0	Quick Downloads	All Devices	All Devices	✓	✓	✓	✓	✓		
.net Visual Basic .NET	Quick Downloads	All Devices	All Devices	✓	✓	✓	✓	✓		
Mobile Languages										
iOS	Quick Downloads	All Devices	InterfaceKit Only	✓	✓	X	✓	✓		
Android Java	Quick Downloads	All Devices	InterfaceKit Only	✓	X	Some Devices	✓	✓		
Scripting										
Applescript	Quick Downloads	All Devices	All Devices	✓	✓	✓	✓	✓		
AutoIt	Quick Downloads	All Devices	None	✓	✓	✓	✓	✓		
Ruby	Quick Downloads	All Devices	All Devices	✓	✓	✓	✓	✓		
Science and Math										
LabVIEW	Quick Downloads	All Devices	All Devices	✓	✓	✓	✓	✓		
MATLAB	Quick Downloads	All Devices	Some Devices	X	✓	✓	✓	X		
Simulink	Quick Downloads	All Devices	InterfaceKit Only	X	✓	✓	✓	X		
Multimedia										
Adobe Director	Quick Downloads	All Devices	InterfaceKit Only	✓	✓	✓	✓	✓		
Flash AS3	Quick Downloads	All Devices	All Devices	✓	✓	✓	✓	✓		
LiveCode	Quick Downloads	InterfaceKit Only	InterfaceKit Only	✓	✓	✓	X	✓		
Max/MSP	Quick Downloads	All Devices	All Devices	✓	✓	✓	✓	✓		
Other Languages										
C# (.NET Compact)	Quick Downloads	All Devices	Some Devices	✓	✓	✓	✓	✓		
.net Visual Basic (.NET Compact Framework)	Quick Downloads	All Devices	None	✓	✓	✓	✓	✓		
Visual Basic for Apps	Quick Downloads	All Devices	Some Devices	✓	✓	✓	✓	✓		
Visual Basic Script	Quick Downloads	All Devices	Some Devices	✓	✓	✓	✓	✓		
Delphi	Quick Downloads	All Devices	Some Devices	✓	✓	✓	✓	✓		

PHIDGETS PROGRAMMING

```
static void Main(string[] args)
{
    try
    {
        RFID rfid = new RFID(); //Declare an RFID object

        //initialize our Phidgets RFID reader and hook the event handlers
        rfid.Attach += new AttachEventHandler(rfid_Attach);
        rfid.Detach += new DetachEventHandler(rfid_Detach);
        rfid.Error += new ErrorEventHandler(rfid_Error);

        rfid.Tag += new TagEventHandler(rfid_Tag);
        rfid.TagLost += new TagEventHandler(rfid_TagLost);
        rfid.open();

        //Wait for a Phidget RFID to be attached before doing anything with
        //the object
        Console.WriteLine("waiting for attachment...");
        rfid.waitForAttachment();

        //turn on the antenna and the led to show everything is working
        rfid.Antenna = true;
        rfid.LED = true;

        //keep waiting and outputting events until keyboard input is entered
        Console.WriteLine("Press any key to end...");
        Console.Read();

        //turn off the led
        rfid.LED = false;

        //close the phidget and dispose of the object
        rfid.close();
        rfid = null;
```

AARHUS
UNIVERSITET

DISTRIBUTED CONTEXT AWARENESS

› Self-contained context awareness is limited
› - using context-aware sensors on a local device can only update part of the model
› - often only simple causality updates – when person present -> do task
› › - hard to make elaborate and accurate models – e.g. of activitites of daily living

› Infrastructure-based context awareness
› - enables sensor fusion of multiple context sensor types
› - environment based, body worn, virtual,
› - requires support for distribution of sensor data

AARHUS
UNIVERSITET

ENABLING TECHNOLOGIES

› PAN Technologies
› Z-Wave, ZigBee, Bluetooth, ENOCEAN

› LAN/WAN Technologies
› TCP/IP, HTTP/SOAP/WS/Web services, CORBA, Java RMI, ICE, DDS, Thrift

WEB SERVICES

› Provides a simple and heterogeneous distribution model
› 1. Standard HTTP/SOAP based Web services (WS-*) o
› 2. HTTP/JSON/XML Web services (REST)

› Wide spread support
› - Language and platform access transparency through open standards
› - Location transparency through HTTP protocol
› - Built-in firewall traversal through HTTP
› - Security, scalability, and reliability services with WS-*

› Easy to design a distributed context aware service infrastructure
› Alternative to existing distributed context awareness frameworks

DISTRIBUTION MIDDLEWARES

› Alternatives to Web services:

› 1. Basic socket (UDP/TCP/IP/HTTP): extremely high heterogenity, but error-prone and no built-in services. Scalability/security/reliability very problematic.
› 2. Java Remote Method Invocation (Java RMI): tight coupling with Java, Java Standard Edition only, fast but requires router access for WAN communications. Limited features and only one vendor. Security problematic.
› 3. Common Object Request Broker Architecture (CORBA): heterogenity high, large number of vendors, impressive number of services and features available, steep learning curve, large differences in products.
› 4. Internet Communication Engine (ICE): comparable to CORBA but faster and simpler, license can be an issue, and only one version.
› 5. Data Distribution Service (DDS): comparable to CORBA but event based and with quality of service parameters. Limited open source availability.

› Exercise:

› Make a matrix of the technologies above (use google) in order to identify which distribution technology is best suited in your opinion

CONTEXT FRAMEWORKS

1. Context Toolkit, Dey et al. (2001)
2. Gaia project (Roman et al., 2002)
3. Context Managing Framework, Korpipää et al. (2003)
4. Context Broker Architecture (CoBrA) Chen et al. (2003)
5. Service-Oriented Context-Aware Middleware (SOCAM), Gu et al. (2004)
6. Context-Awareness Sub-Structure (CASS), Fahy & Clarke (2004)
7. Java Context Aware Toolkit (JCAF), Bardram (2006)
8. Hydra, Badii et al. (2009)
9. MidSen Patel et al. (2009)
10. COPAL, Li et al. (2010)
11. ComiHOC, Wibisono et al. (2010)
12. Octopus, Firner et al. (2011)

› In the following we will discuss 1 and 7 in detail

COMMON FRAMEWORK CHALLENGES

- **Most available context awareness frameworks and middleware's face a range of challenges**:
- 1. Available in Java only or supported on a single platform only
- 2. Overly complex and not ”real world operational”
- 3. Limited flexibility in the design
- 4. Limited security/scalability/reliability transparency
- 5. Limited context sensor support
- 6. Limited documentation and tutorials
- 7. Firewall transversal limited or not possible
- 8. Dependent on a centralized server architecture

- In conclusion: are they really worth the hassle?
- Consider starting from scratch, or using a very basic low level framework
- Context frameworks may act as “best practice” guides

AARHUS
UNIVERSITET

CONTEXT TOOLKIT

› Toolkit for distributed context-aware apps

› - framework for acquiring and handling context
› - Standard components

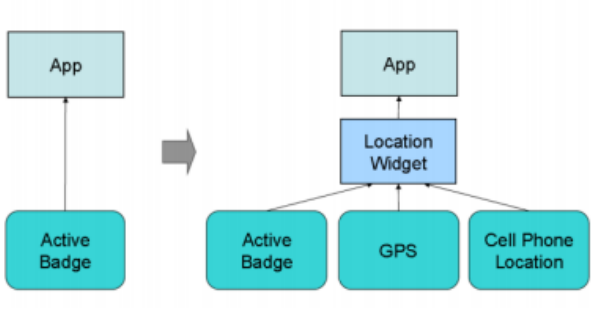
› Three key abstractions

› – Widgets, Interpreters, and Aggregators

CONTEXT TOOLKIT

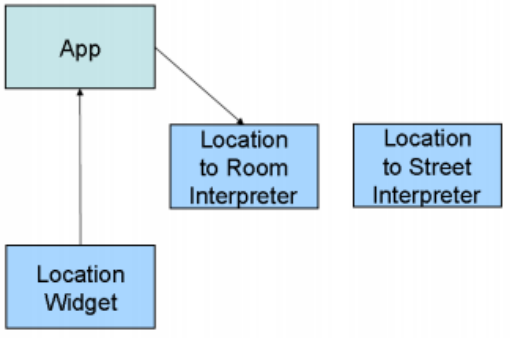
› Context Widget

› Is a software component that:
› - provides application access to context information
› - encapsulates context acquisition



› Context Interpreters

› Converts context to higher level information

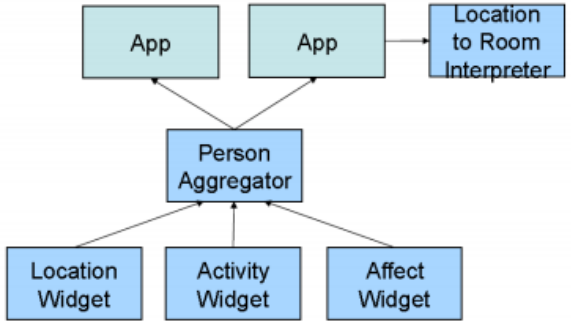


› Context Aggregators

› Collects context relevant to particular entities

› Context Services

› Component for executing services (actuators)



› Context Discoverers

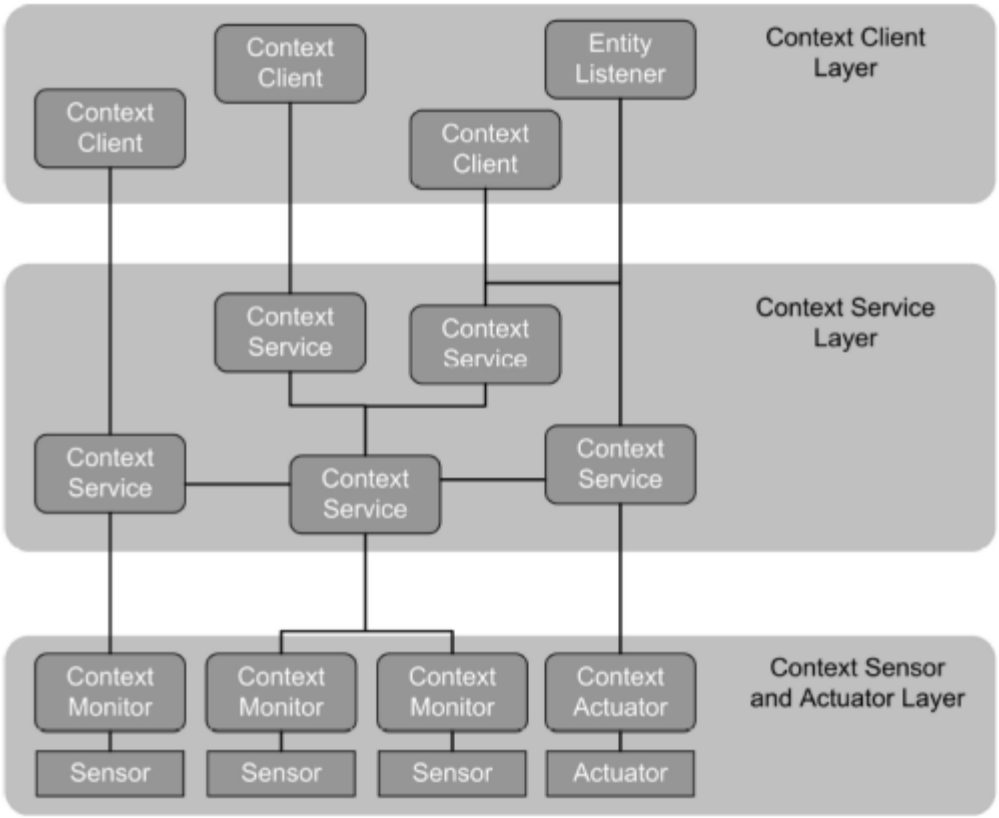
› Registry of capabilities in the framework

CONTEXT TOOLKIT

- **Technical design:**
- Written in Java
- Custom HTTP protocol
- Basic port to C#, VB, Flash
- No security/reliability/scalability features
- No ”out of the box” sensors supported
- http://contexttoolkit.sourceforge.net/

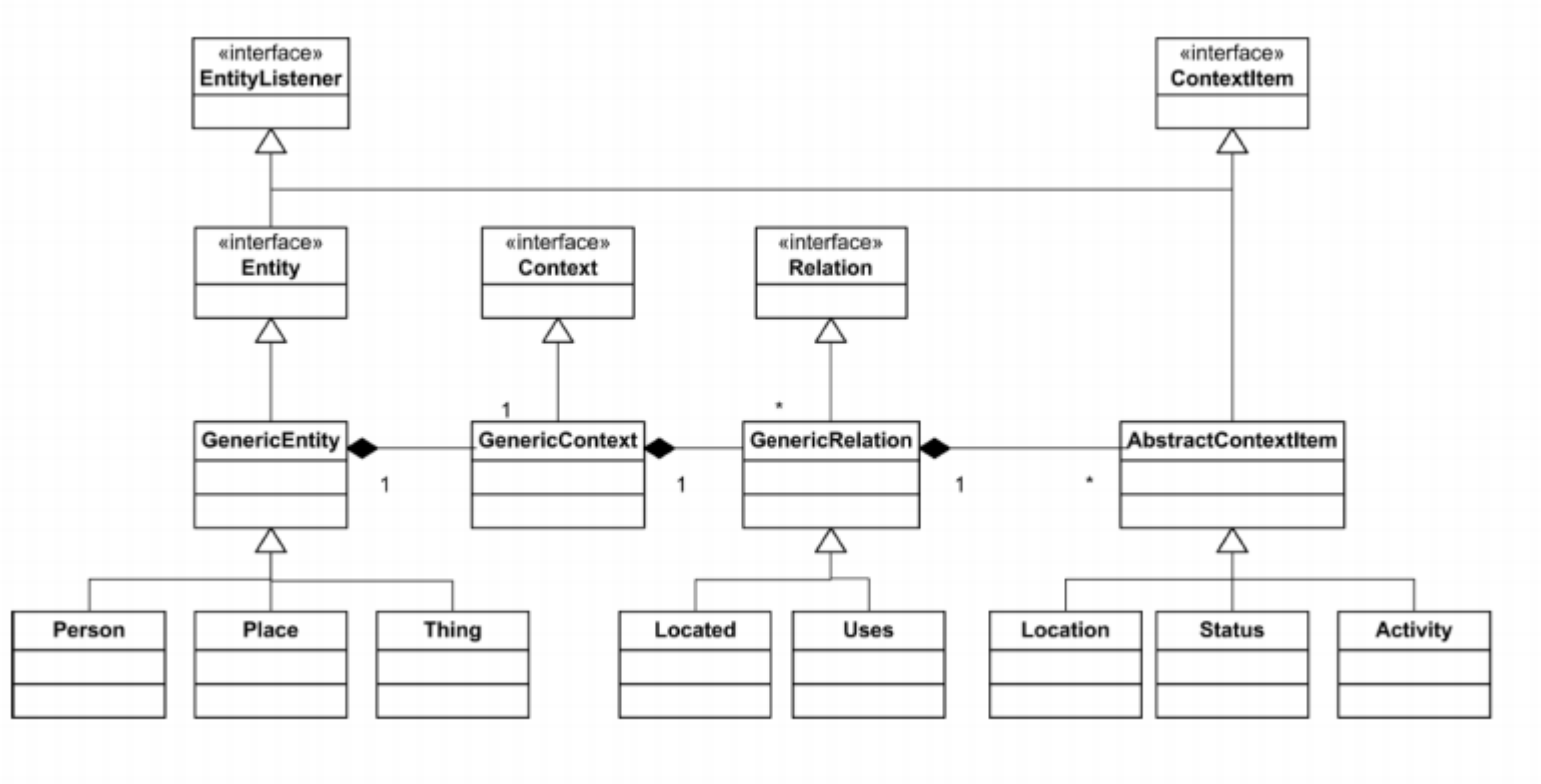
JAVA CONTEXT-AWARENESS FRAMEWORK

› JCAF Runtime Infrastructure



JAVA CONTEXT-AWARENESS FRAMEWORK

› JCAF Context Model



JAVA CONTEXT-AWARENESS FRAMEWORK

- **Technical Design:**
- Developed in Java
- Requires
- - Java RMI for distribution model
- - JAAS for security
- Limited heterogeneity (only Java Standard Edition)
- Firewall issues
- No ”out of the box” sensors supported
- http://sourceforge.net/projects/jcaf/

EXERCISE AND HAND-IN

› Exercise 2:

- Learn how to contact a distributed fixed sensor via a Web service and acquire its data for use in a context aware system.
- Learn how to contact a mobile wireless sensor via Bluetooth and acquire its data for use in a context-aware system.

› Hand-in 2:

- Make a UML Deployment diagram of a pervasive system consisting of a bed sensor, and four movement sensors (in the four rooms of a sample home), where all five sensors sends a context event to a central computer (gateway). Also, describe in ½ a page of text: 1) which specific technologies can be used to implement this (hardware nodes, network, middleware), and 2) how these sensors may be used to detect serious incidents - such as serious falls, and detecting dementia patients suffering from insomnia
- Must be solved in groups of 3-4. Must be handed in on Campusnet. 3) consider how a wearable sensor node (such as the Shimmer) can be used to supplement the ambient infrastructure based sensors.