



COGKNOW

D2.5.1

Prototype Release of situation aware services

Version 1.0

WP 2 Technical Development, Services and Applications

Dissemination Level: Public

Lead Editor: Marino Bresciani, Mobi

17/11/2008

Status: Final

SIXTH FRAMEWORK PROGRAMME
PRIORITY IST-2005/2006-2.5.11



Unit H3 - eInclusion

Proposal/Contract no.: 034025

Context

WP 2	This WP undertakes the development of the necessary cognitive prosthetics to support the person with dementia.
WPL	Veiko Raime (MOBI)
Task 2.5	Multi-Modal Interfaces
TL	Marino Bresciani
Dependencies	This deliverables uses specifically the input of the deliverables D2.1.1 (Technical Specification of Systems and services) along with the feedback following the evaluation from Field Trial #1.
Starting date	18/04/2007
Release date	16/09/2008

Author(s)	Marino Bresciani, Mobi
Contributor(s)	Lhoutellier Vincent (Mobi), Mossaab Hariz (GETINT), Henri ter Tofte (TELIN)
Reviewers	Mounir Mokthari (GET INT)
Approved by:	PSG on 17/11/2008

History

Version	Date	Authors	Sections Affected
0.1	17/04/2008	Lhoutellier (Mobi)	
0.2	16/09/2008	Mossaab Hariz	II.B;II.C;II.D;II.E;III.B
0.3	01/11/2008	Marino Bresciani (Mobi)	Change of structure – Insertion of contributions from TELIN
0.4	05/11/2008	Marino Bresciani (Mobi)	Final checks
0.5	17/11/2008	Mounir Mokthari	Change of structure – Insertion of contributions from TELIN
1.0	17/11/2008	Ricardo Castellot	Moving XML section as Annex, cosmetic changes for final version

Table of Contents

- 1 Summary 4
- 2 Introduction..... 5
 - 2.1 The COGKNOW approach 5
- 3 Role of sensors..... 7
 - 3.1 Positioning and movement sensors..... 7
 - 3.2 Global Positioning System..... 8
- 4 Prototypes 9
 - 4.1 Sensors for Context Situation 9
 - 4.1.1 PwD is on the phone..... 9
 - Sensors available:..... 9
 - 4.1.2 PwD is eating lunch. 9
 - Sensors available..... 9
 - 4.1.3 PwD is sleeping or out of bed during the middle of the night..... 10
 - Sensors available..... 10
 - 4.2 Role of GPS: “Take Me Home” 11
- 5 Context engine 13
- 6 Conclusion..... 16
- References 17
- Annex A. XML Schema for Context Rules 18

1 Summary

This deliverable presents the prototype system which has been developed within the Project to offer persons suffering from mild dementia with a series of situation awareness services which will automate some services depending on particular context the user is involved.

In this document the work has involved the development of some particular examples of situation awareness services that have been used to help the user in some particular situations. Particular contexts trigger some warning or auxiliary service which will be shown to the user via the graphical interface of the system.

A demonstration of the situation awareness services can be accessed by viewing some specific sections of 2 DVDs related to this document.

The DVD related to Home Based Visualization services can be accessed using the following links.

- [https://cdh.project.ltu.se/main.php/Cogknow%20Field%20Trial%202%20\(Medium\).wmv?fileitem=27852809](https://cdh.project.ltu.se/main.php/Cogknow%20Field%20Trial%202%20(Medium).wmv?fileitem=27852809) (High quality)
- [https://cdh.project.ltu.se/main.php/Cogknow%20Field%20Trial%202%20\(Small\).wmv?fileitem=27852808](https://cdh.project.ltu.se/main.php/Cogknow%20Field%20Trial%202%20(Small).wmv?fileitem=27852808) (Low quality)

The DVD related to Multimodal Interfaces can be accessed using the following link.

- <http://office.mobi.ee/~marino/dw/Prototype%20Release%20Of%20MultiModal%20Interfaces.wmv>

2 Introduction

This deliverable presents the prototype system which has been developed within the Project to offer persons suffering from mild dementia with a set of context-awareness service they can use during their day.

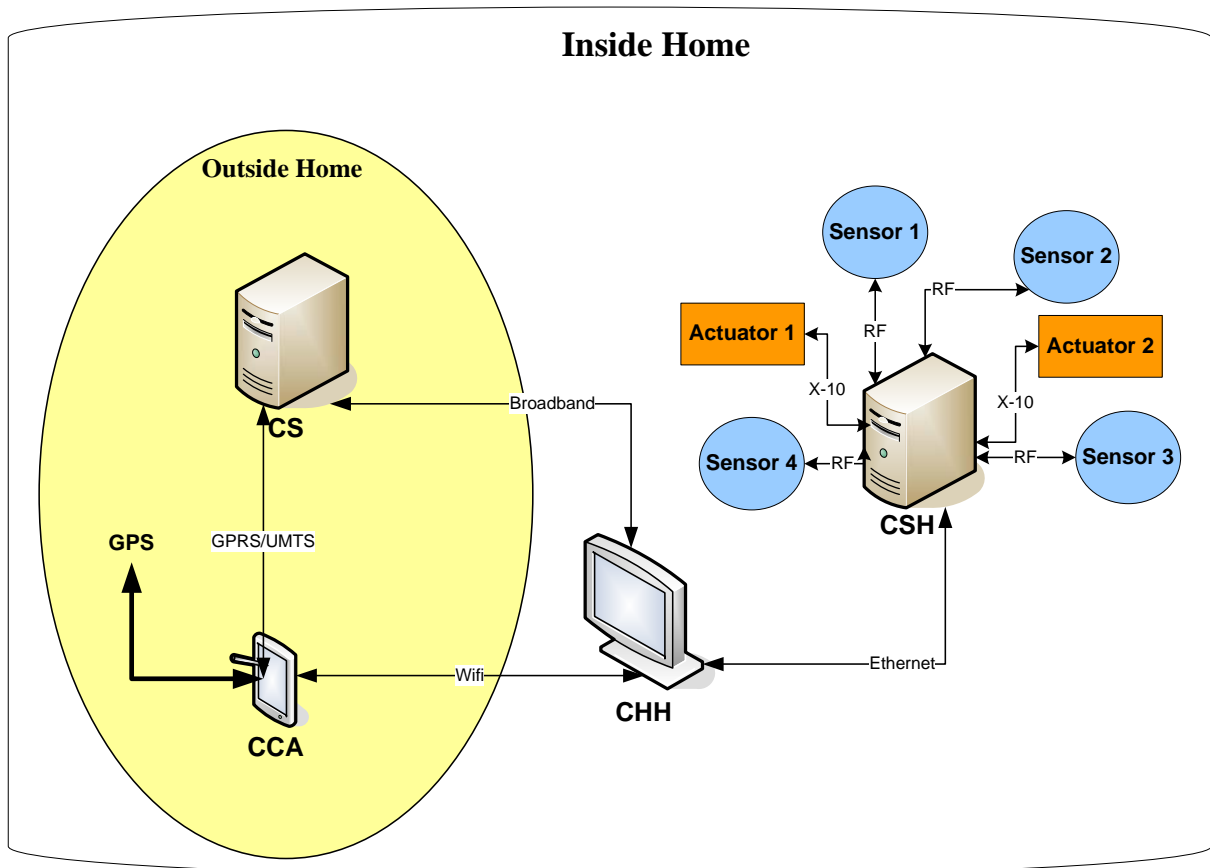
The complexity of modern networks raises several challenges in the design and development of communication services. The unbearable costs in configuration and management call for autonomic approaches, in which services are able to self configure and self-adapt their activities without human intervention. The need for ubiquity of service provisioning calls for the capability of services of adapting their behaviours depending on the current situation in which they are used.

Mobile devices such as personal digital assistants (PDAs) and mobile phones are in widespread use already today and converging to mobile smart phones. They take in account the characteristics of user's environments. Knowing the situation, it allows the system to better identify the information to be delivered to them and to choose the appropriate services with regard to their scope, which is referred to as service roaming. Going one step further, a situation-aware system (SAS) abstracts from the context dimensions by translating specific contexts into logical situations.

Such a concept is a really progress, especially when healthcare is involved. International projects, for instance "Nomadic Media" an Eureka-ITEA, have been studied and applied to provide mobile users with innovative healthcare services.

2.1 *The COGKNOW approach*

In case of COGKNOW project one can dissociate two global situations, i.e. Inside and Outside home. The graph bellow introduce those scenario and their own modal in interaction.



Several sensors and actuators are deployed in the patient home. The **Cogknow Sensered Home (CSH)** communicates with the sensor to get information about the user activity and to control the environment if needed. A context reasoner inside the **CSH** processes the sensor data to come out with a contextual situation. Based on his situation the user is provided with adapted services.

3 Role of sensors

It is anticipated that the home environment will be equipped with a number of devices which will have embedded sensory elements for example a fridge door sensor to indicate when the fridge was last opened or a sensor on the front door to indicate the last time the door was opened. When the diseased is going out home, the door's sensor detect the event, and the system is commute to outside mode.

Those informations are relayed to the monitoring centre and from there it will be possible to remotely monitor the activities of the person and assess if they are complying with their planned schedule of activities. To facilitate, these operations will require the configuration of sensory elements within the home environment to allow their operational status to be relayed or polled by the mobile device.

3.1 Positioning and movement sensors

PIR Movement Detector

Fitted in the hallway or living room, PIR's can be used for Activity Monitoring or Intruder Detection. If no movement is detected within a preset period an inactivity alarm will be raised. If movement is detected when a resident is "away" an intruder alarm will be reported.



Sensor Controller

This can be used with an in-bed sensor or chair sensor to raise an alarm if the user gets up and does not return within a preset period. Can also be used with the Remote Controller Mains Switching Kit to automatically turn lamps on/off. Can be mains or battery powered.



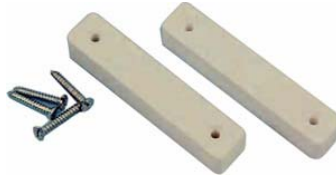
Bed/Chair Sensors

the bed sensor is placed under the mattress whereas the chair sensor is placed under the seat cushion. The Sensor Controller will calibrate itself for the users weight. An adapter is available for 2 sensors in a double bed.



Door Contacts

these can be fitted to any door, the Sensor Controller is used to set the monitoring periods. If the door is opened a lamp can be turned on and an alarm is raised to a local Carer.



3.2 Global Positioning System

Using GPS technology, the geographical position of a tracking device can be determined with considerable accuracy. The tracking device system used for this project was a HTC Touch Cruise GPS integrated mobile phone. The device was connected via wireless and 3G connection to the Cogknow system. It was used to obtain the geographical information of the handsets and to handle a functionality that permit to support the user navigating home when outside.

4 Prototypes

4.1 Sensors for Context Situation

Based on the context selection to be implemented context situation we have used the following set of sensors.

4.1.1 PwD is on the phone

Sensors available:

- Contact sensor on the phone set
- Presence sensor in the room.

Context:

- IF PwD is making a Phone call **THEN** delay any reminders
- IF sensor IS off Hook AND there is no phone call for 10 minutes **THEN** remind the pwd to replace Handset
- IF the phone call has ended AND there is no on hook message **THEN** remind the PwD to replace Handset
- IF the phone call has ended **THEN** remind PwD to continue previous task.

4.1.2 PwD is eating lunch.

Sensors available

- Presence sensor in the kitchen,
- Pressure sensor on the chair,
- Door sensor on the fridge
- Door sensor on the oven.

See section 8 on the DVD of Home Based Visualization Services for a demonstration of Warning Sensors related to the "Motivated to Eat" functionality

Context: Lunch reminder

- IF PwD IS in kitchen WITH range of reminder AND IS NOT having lunch **THEN** fire the reminder AFTER 20 second
- IF PwD IS NOT in kitchen AND IS NOT in living room **THEN** delay reminder
- IF PwD IS in kitchen AND is eating lunch **THEN** cancel reminder AND play Music

- IF PwD IS in kitchen AND is making lunch **THEN** delay reminder

4.1.3 PwD is sleeping or out of bed during the middle of the night.

Sensors available

- Pressure sensor
- Presence sensor

See section 9 on the DVD of Home Based Visualization Services for a demonstration of Warning Sensors related to the “Night Wandering” functionality

Context: **Person with Dementia (PwD)** wake up during the night

- IF PwD IS sleeping AND it IS dark AND they get out of bed **THEN** turn on bedroom and hall lights
- WHEN PwD returns to bed AFTER lights on events **THEN** turn off the lights

4.2 Role of GPS: “Take Me Home”

The GPS sensor in the mobile device was used in the “Take Me Home” function, which worked as follows for the users:

See section 6 on the DVD of Multimodal Interfaces for a demonstration of the Take Me home Functionality

1. When the home screen is visible, the user could press on the “Home” icon (the button in the lower right-hand corner)



2. The Take Me Home service started. Initially, the message “One moment please, while I find out where you are” (or an equivalent text in Dutch or Swedish, depending on the country where the trial was held) would blink on the screen, as long as the device did not have a GPS fix. When a GPS fix was obtained, directions were calculated (using TomTom 6 Navigator as engine and Western Europe maps v4.75) in “pedestrian” mode, also, the message “Please depart. Directions follow as you go.” Would blink for a few seconds.

- About 50m before a turning point was reached, directions were given visually (see a screendump directly below for an example) as well as an auditory navigation direction that went like “<PlingPlingPlong> After 50 meters, <Plong> turn right”. <PlingPlingPlong> and <Plong> were “earcons”, i.e. sounds we added to make sure we attracted auditory attention of the person with dementia before the content of the direction was delivered audibly.



- About 20m before a turning point was reached, directions were given again, this time “<Plong> Turn right.”



- The last two steps would repeat for each turning point.
- When the destination was approached within about 30 meters, the screen would look like the screen below and the message “<PlingPlingPlong > You have reached your destination” would sound.



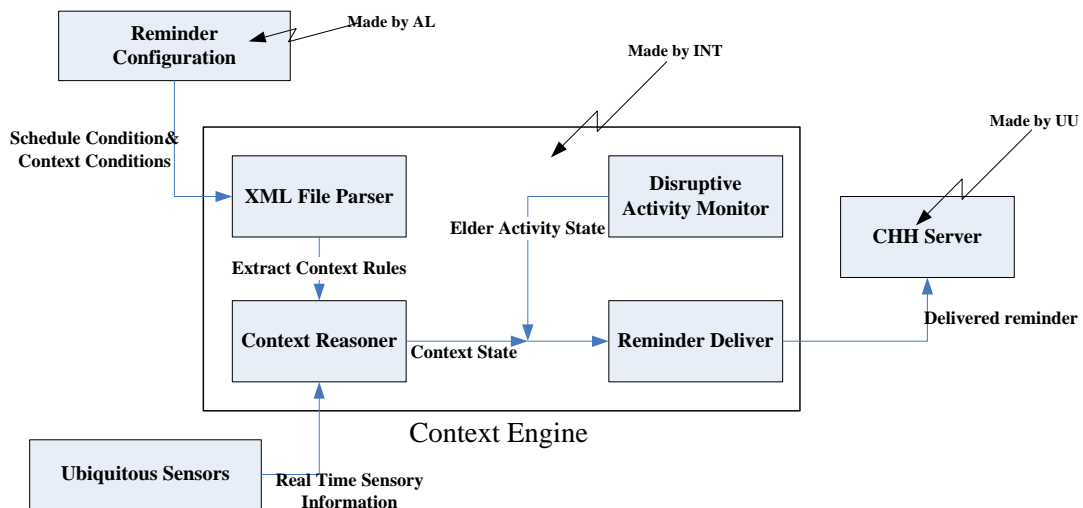
- Then, the screen would automatically return to the home screen:



8. If the screen was tapped while the destination was not yet reached, the user would be given a choice: with two buttons: “stop” or “continue” (with labels in the language of the user, i.e. Dutch, Swedish or English). The former button would end the navigation session and return to the home screen, the latter would resume the navigation session.

5 Context engine

1. Architecture of Context Engine



XML File Parser: this module is to parse the schedule and context conditions from the reminder configuration document which is a xml file, and extract context rules as input for Context Reasoner.

Context Reasoner: this module is to match the context rules with real time sensory data, and deliver satisfied context rules to Reminder Deliver.

Disruptive Activity Monitor: this module is to keep on monitoring the disruptive activity, and notify the Reminder Deliver when disruptive activity is detected.

Reminder Deliver: this module is to deliver reminder to CHH sever when all conditions are satisfied, it will also handle the conflicts between reminded activity and disruptive activity.

2. Context Rules

The important information need to offer to user:

- Is this reminder critical to you? (which means the reminder is important and must be delivered to him)
- During the activity (such as lunch), if you are interrupted by a call and go to answer the phone call, do you need to be reminded again when you finish the phone call (default

value is yes)

- when you are on sleep or on the phone, do you mind if the reminder prompt and interrupt you? (default value is no)

Reminders for eating

Lunch reminder:

- the elder has already began to have lunch => cancel
- Before 12:00 (11:30~12:00), the elder is in kitchen => prompt
- At 12:00,
IF the elder is not on the phone, not sleep => prompt
ELSE => delay, wait until the phone finished or the elder wakes up.
IF the time doesn't expire => prompt
ELSE
 - if the reminder is set to critical => prompt
 - if the reminder is set to uncritical => cancel

Dinner reminder:

- the elder has already began to have dinner => cancel
- Before 19:00 (18:30~12:00), the elder is in kitchen, => prompt
- At 19:00
IF the elder is not on the phone, not sleep => prompt
ELSE => delay, wait until the phone finished or the elder wakes up.
IF the time doesn't expire => prompt
ELSE
 - if the reminder is set to critical => prompt
 - if the reminder is set to uncritical => cancel

Personal activity

Watching TV reminder

- IF the reminder is set to critical
when time (set by user) is up => prompt
- IF the reminder is set to uncritical
IF The elder is at home, time is up (set by user), the elder is not on the phone,
not sleep => prompt
ELSE => delay, wait until the phone finished or the elder wakes up

- If the TV program ends => cancel
- If the TV program doesn't end => prompt

Brushing teeth reminder

When time is up (set by user)

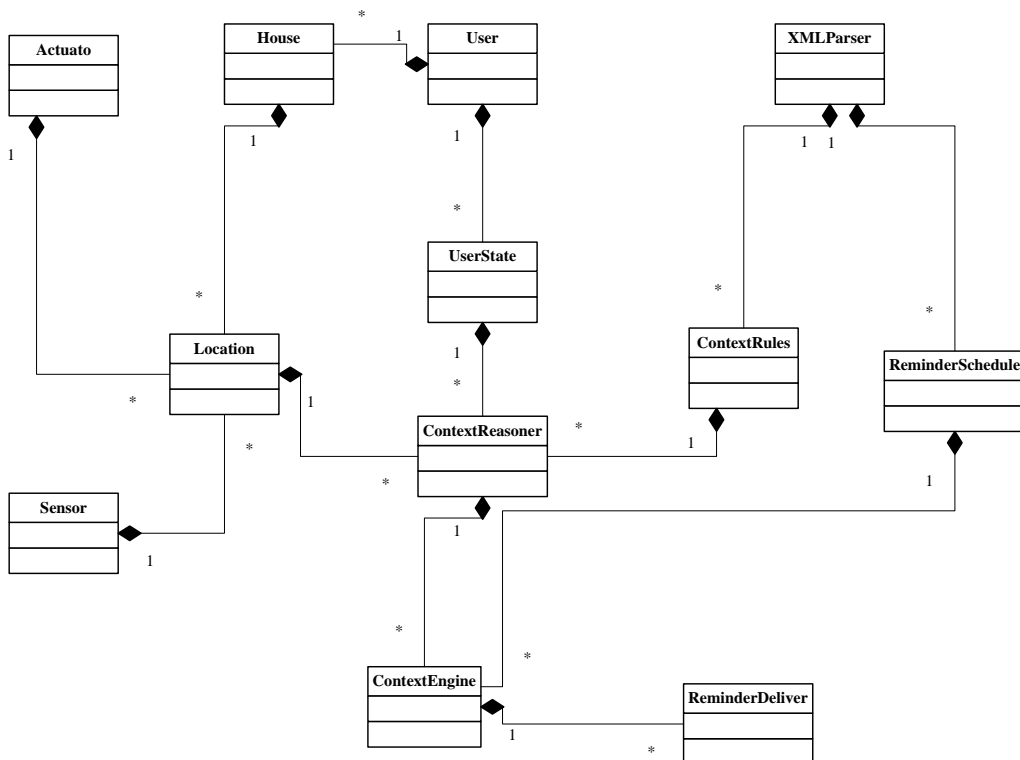
- IF the elder is at home, the elder is not on the phone, not sleep => prompt
- ELSE wait until the phone finished or the elder wakes up
 - IF the time doesn't expire =>prompt
 - Else the time expires
- If the reminder is set to critical => prompt
- If the reminder is set to uncritical => cancel

Appointment reminder

When time is up (set by user)

- IF the elder is at home, the elder is not on the phone, not sleep => prompt
- ELSE wait until the phone finished or the elder wakes up
 - IF the time doesn't expire =>prompt
 - Else the time expires
- If the reminder is set to critical => prompt
- If the reminder is set to uncritical => cancel

3. UML Diagram



6 Conclusion

This report has provided details of the context-awareness services with the CogKnow Project, within the main areas of cognitive reinforcement which are being addressed by work within the Project.

Following development of these technical services, as presented within the report, evaluation of their perceived utility will be assessed during a series of three planned Field Trials in Northern Ireland, Lulea and Amsterdam. Based on the feedback from these evaluations the system will be adapted accordingly

References

- [1] Ref 1 : Yurong Xu, James C. Ford, Fillia S. Makedon, Dan Popa, Heng Huang, Li Shen. "In-home Localization for Home Care of Alzheimer's Disease Patients using Wireless Sensor Networks".
- [2] Ref 2 : *Fausto J. Sainz Salce, David England, Paul Vickers*. "HOUSEHOLD APPLIANCES CONTROL DEVICE FOR THE ELDERLY"

Annex A. XML Schema for Context Rules

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- edited with XMLSpy v2008 sp1 (http://www.altova.com) by kcassar (EMBRACE) -
->
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
elementFormDefault="qualified" attributeFormDefault="unqualified">
  <xs:element name="ALERTS">
    <xs:annotation>
      <xs:documentation>Reminder</xs:documentation>
    </xs:annotation>
    <xs:complexType>
      <xs:sequence>
        <xs:element name="ALERT" maxOccurs="unbounded">
          <xs:complexType>
            <xs:sequence>
              <xs:element
name="ATTACHMENTS">
                <xs:complexType>
                  <xs:sequence>
                    <xs:element
name="ATTACHMENT" maxOccurs="unbounded">
                      <xs:complexType>
                        <xs:simpleContent>
                          <xs:extension base="xs:base64Binary">
                            <xs:attributeGroup ref="ATTACHMENT_ATT_GR"/>
                          </xs:extension>
                        </xs:simpleContent>
                      </xs:complexType>
                    </xs:element>
                  </xs:sequence>
                </xs:complexType>
              </xs:sequence>
            </xs:element>
          </xs:sequence>
        </xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

```

        </xs:complexType>
    </xs:element>
    <xs:element name="EVENTS">
        <xs:complexType>
            <xs:sequence>
                <xs:element
name="EVENT" maxOccurs="unbounded">
                    <xs:complexType>
                        <xs:sequence>
                            <xs:element name="CONDITIONS">
                                <xs:complexType>
                                    <xs:all>
                                        <xs:element
name="SCHEDULE_CONDITION_GROUP" minOccurs="0">
                                            <xs:complexType>
                                                <xs:sequence>
                                                    <xs:choice>
                                                        <xs:element
name="SIMPLE" maxOccurs="unbounded">
                                                            <xs:complexType>
                                                                <xs:attribute
name="TIME_VARIANCE_ATT" type="xs:time" use="optional"/>
                                                                    <xs:attribute
name="DATE_TIME_ATT" type="xs:dateTime" use="required"/>
                                                                </xs:complexType>
                                                            </xs:element>
                                                        <xs:element
name="RECURRENCE">

```

<xs:complexType>

<xs:sequence>

<xs:element name="TIME" type="xs:time"/>

<xs:choice>

<xs:element ref="DAILY_RECURRENCE"/>

<xs:element ref="WEEKLY_RECURRENCE"/>

<xs:element ref="MONTHLY_RECURRENCE"/>

<xs:element ref="YEARLY_RECURRENCE"/>

</xs:choice>

<xs:element name="START_DATE" type="xs:date"/>

<xs:choice>

<xs:element name="NO_END_DATE"/>

<xs:element name="END_AFTER">

<xs:complexType>

use="required"/> <xs:attribute name="OCCURRENCES_ATT" type="xs:byte"

</xs:complexType>

</xs:element>

<xs:element name="END_BY">

<xs:complexType>

<xs:attribute name="END_BY_DATE_ATT" type="xs:date" use="required"/>

</xs:complexType>

</xs:element>

</xs:choice>

</xs:sequence>

<xs:attribute name="TIME_VARIANCE_ATT" type="xs:time" use="optional"/>

<xs:attribute ref="RECURRENCE_TYPE_ATT" use="required"/>

<xs:attribute name="END_TYPE_ATT" use="required"/>

<xs:simpleType>

<xs:restriction base="xs:string">

<xs:enumeration value="NO_END_DATE"/>

<xs:enumeration value="END_AFTER_X_OCCURRENCES"/>

<xs:enumeration value="END_BY_DATE"/>

</xs:restriction>

</xs:simpleType>

</xs:attribute>

</xs:complexType>

</xs:element>

</xs:choice>

</xs:sequence>

<xs:attribute

ref="SCHEDULE_TYPE_ATT" use="required"/>

</xs:complexType>

</xs:element>

<xs:element name="CONTEXT_CONDITION_GROUP"

minOccurs="0">

<xs:complexType>

<xs:sequence maxOccurs="unbounded">

<xs:element

name="CONTEXT_CONDITION">

<xs:complexType>

<xs:attribute

name="CONTEXT_ID_ATT" type="xs:string" use="required"/>

name="TRIGGER_VALUE_ATT" type="xs:byte" use="required"/> <xs:attribute

name="TRIGGER_CONDITION_ATT" use="required"> <xs:attribute

<xs:simpleType>

<xs:restriction base="xs:string">

<xs:enumeration value="Equals"/>

<xs:enumeration value="Below"/>

<xs:enumeration value="Above"/>

</xs:restriction>

</xs:simpleType>

</xs:attribute>

</xs:complexType>

</xs:element>

</xs:sequence>

</xs:complexType>

</xs:element>

<xs:element
name="PREDECESSOR_CONDITION_GROUP" minOccurs="0">

<xs:complexType>

<xs:sequence maxOccurs="unbounded">

```

name="PREDECESSOR_CONDITION">
    <xs:element
        <xs:complexType>
            <xs:attributeGroup
                ref="PREDECESSOR_CONDITION_ATT_GR"/>
            </xs:complexType>
        </xs:element>
    </xs:sequence>
</xs:complexType>
</xs:element>
</xs:all>
</xs:complexType>
</xs:element>
<xs:element name="PAYLOADS">
    <xs:complexType>
        <xs:sequence>
            <xs:element
                name="PAYLOAD"
                maxOccurs="unbounded">
                    <xs:complexType>
                        <xs:sequence>
                            <xs:element
                                name="PAYLOAD_PARAMETERS">
                                    <xs:complexType>
                                        <xs:sequence>

```


name="PAYLOAD_PARAMETER" maxOccurs="unbounded">

<xs:element

<xs:complexType>

<xs:attributeGroup ref="PAYLOAD_PARAMETER_ATT_GR"/>

</xs:complexType>

</xs:element>

</xs:sequence>

</xs:complexType>

</xs:element>

</xs:sequence>

ref="PAYLOAD_ATT_GR"/>

<xs:attributeGroup

</xs:complexType>

</xs:element>

</xs:sequence>

</xs:complexType>

</xs:element>

</xs:sequence>

<xs:attribute name="SYNCH_ATT" use="required">

<xs:simpleType>

<xs:restriction base="xs:string">

```

        <xs:enumeration value="ADDED"/>

        <xs:enumeration value="MODIFIED"/>

        <xs:enumeration value="DELETED"/>

    </xs:restriction>

</xs:simpleType>

</xs:attribute>

<xs:attribute name="EVENT_DESCRIPTION" type="xs:string" use="required"/>

<xs:attribute name="EVENT_ID" use="required"/>

</xs:complexType>

</xs:element>

                                </xs:sequence>
                                </xs:complexType>
                                </xs:element>
                                </xs:sequence>
                                <xs:attribute      name="SYNCH_ATT"
use="required">
                                <xs:simpleType>
                                    <xs:restriction
base="xs:string">
                                        <xs:enumeration
value="ADDED"/>
                                        <xs:enumeration
value="MODIFIED"/>
                                        <xs:enumeration
value="DELETED"/>
                                    </xs:restriction>
                                </xs:simpleType>
                                </xs:attribute>
                                <xs:attribute name="ALERT_ID_ATT"/>
                                <xs:attribute
name="ALERT_DESCRIPTION_ATT"/>
                                </xs:complexType>

```

```

        </xs:element>
    </xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="DAILY_RECURRENCE">
    <xs:complexType>
        <xs:attribute name="DAILY_PATTERN_ATT" use="required">
            <xs:simpleType>
                <xs:restriction base="xs:string">
                    <xs:enumeration
value="EVERY_X_DAYS"/>
                    <xs:enumeration
value="EVERYWEEKDAY"/>
                    <xs:enumeration
value="EVERYWEEKENDDAY"/>
                </xs:restriction>
            </xs:simpleType>
        </xs:attribute>
        <xs:attribute name="EVERY_NUMBER_DAYS_ATT"
type="xs:byte" use="optional"/>
    </xs:complexType>
</xs:element>
<xs:element name="WEEKLY_RECURRENCE">
    <xs:complexType>
        <xs:attributeGroup ref="WEEKDAYS_ATT_GR"/>
        <xs:attribute name="EVERY_NUMBER_WEEK_ATT"
type="xs:byte" use="required"/>
    </xs:complexType>
</xs:element>
<xs:element name="MONTHLY_RECURRENCE">
    <xs:complexType>
        <xs:choice>
            <xs:element name="DAY">
                <xs:complexType>
                    <xs:attribute ref="DAY_ATT"
use="required"/>
                    <xs:attribute
name="EVERY_NUMBER_MONTH_ATT" type="xs:byte" use="required"/>
                </xs:complexType>
            </xs:choice>
        </xs:complexType>
    </xs:element>
</xs:complexType>
</xs:element>

```

```

        </xs:element>
        <xs:element name="MONTHLY_THE">
            <xs:complexType>
                <xs:attribute ref="CARDINALITY_ATT"
use="required"/>
                <xs:attribute ref="WEEKDAY_ATT"
use="required"/>
                <xs:attribute
name="EVERY_NUMBER_MONTH_ATT" type="xs:byte" use="required"/>
            </xs:complexType>
        </xs:element>
    </xs:choice>
    <xs:attribute name="MONTHLY_PATTERN_ATT"
use="required">
        <xs:simpleType>
            <xs:restriction base="xs:string">
                <xs:enumeration
value="DAY_X_OF_EVERY_X_MONTHS"/>
                <xs:enumeration
value="THE_X_DAY_OF_EVERY_X_MONTHS"/>
            </xs:restriction>
        </xs:simpleType>
    </xs:attribute>
</xs:complexType>
</xs:element>
<xs:element name="YEARLY_RECURRENCE">
    <xs:complexType>
        <xs:choice>
            <xs:element name="EVERY">
                <xs:complexType>
                    <xs:attribute ref="DAY_ATT"
use="required"/>
                    <xs:attribute ref="MONTH_ATT"
use="required"/>
                </xs:complexType>
            </xs:element>
            <xs:element name="YEARLY_THE">
                <xs:complexType>
                    <xs:attribute ref="CARDINALITY_ATT"
use="required"/>

```

```

use="required"/>
<xs:attribute ref="WEEKDAY_ATT"
use="required"/>
<xs:attribute ref="MONTH_ATT"
</xs:complexType>
</xs:element>
</xs:choice>
<xs:attribute name="YEARLY_PATTERN" use="required">
<xs:simpleType>
<xs:restriction base="xs:string">
<xs:enumeration
value="EVERY_MONTH_DATE"/>
<xs:enumeration
value="THE_X_DAY_OF_MONTH"/>
</xs:restriction>
</xs:simpleType>
</xs:attribute>
</xs:complexType>
</xs:element>
<xs:attribute name="WEEKDAY_ATT">
<xs:simpleType>
<xs:restriction base="xs:string">
<xs:enumeration value="MONDAY"/>
<xs:enumeration value="TUESDAY"/>
<xs:enumeration value="WEDNESDAY"/>
<xs:enumeration value="THURSDAY"/>
<xs:enumeration value="FRIDAY"/>
<xs:enumeration value="SATURDAY"/>
<xs:enumeration value="SUNDAY"/>
<xs:enumeration value="WEEKDAY"/>
<xs:enumeration value="WEEKENDDAY"/>
<xs:enumeration value="DAY"/>
</xs:restriction>
</xs:simpleType>
</xs:attribute>
<xs:attribute name="MONTH_ATT">
<xs:simpleType>
<xs:restriction base="xs:string">

```

```

        <xs:enumeration value="JANUARY"/>
        <xs:enumeration value="FEBRUARY"/>
        <xs:enumeration value="MARCH"/>
        <xs:enumeration value="APRIL"/>
        <xs:enumeration value="MAY"/>
        <xs:enumeration value="JUNE"/>
        <xs:enumeration value="JULY"/>
        <xs:enumeration value="AUGUST"/>
        <xs:enumeration value="SEPTEMBER"/>
        <xs:enumeration value="OCTOBER"/>
        <xs:enumeration value="NOVEMBER"/>
        <xs:enumeration value="DECEMBER"/>
    </xs:restriction>
</xs:simpleType>
</xs:attribute>
<xs:attribute name="DAY_ATT">
    <xs:simpleType>
        <xs:restriction base="xs:byte">
            <xs:enumeration value="1"/>
            <xs:enumeration value="31"/>
        </xs:restriction>
    </xs:simpleType>
</xs:attribute>
<xs:attribute name="CARDINALITY_ATT">
    <xs:simpleType>
        <xs:restriction base="xs:string">
            <xs:enumeration value="FIRST"/>
            <xs:enumeration value="SECOND"/>
            <xs:enumeration value="THIRD"/>
            <xs:enumeration value="FOURTH"/>
            <xs:enumeration value="LAST"/>
        </xs:restriction>
    </xs:simpleType>
</xs:attribute>
<xs:attribute name="SCHEDULE_TYPE_ATT">
    <xs:simpleType>
        <xs:restriction base="xs:string">

```

```

        <xs:enumeration value="SIMPLE"/>
        <xs:enumeration value="RECURRENCE"/>
    </xs:restriction>
</xs:simpleType>
</xs:attribute>
<xs:attribute name="RECURRENCE_TYPE_ATT">
    <xs:simpleType>
        <xs:restriction base="xs:string">
            <xs:enumeration value="DAILY"/>
            <xs:enumeration value="WEEKLY"/>
            <xs:enumeration value="MONTHLY"/>
            <xs:enumeration value="YEARLY"/>
        </xs:restriction>
    </xs:simpleType>
</xs:attribute>
<xs:attributeGroup name="WEEKDAYS_ATT_GR">
    <xs:attribute name="MONDAY_ATT" type="xs:boolean" use="optional"
default="0"/>
    <xs:attribute name="TUESDAY_ATT" type="xs:boolean"
use="optional" default="0"/>
    <xs:attribute name="WEDNESDAY_ATT" type="xs:boolean"
use="optional" default="0"/>
    <xs:attribute name="THURSDAY_ATT" type="xs:boolean"
use="optional" default="0"/>
    <xs:attribute name="FRIDAY_ATT" type="xs:boolean" use="optional"
default="0"/>
    <xs:attribute name="SATURDAY_ATT" type="xs:boolean"
use="optional" default="0"/>
    <xs:attribute name="SUNDAY_ATT" type="xs:boolean" use="optional"
default="0"/>
</xs:attributeGroup>
<xs:attributeGroup name="COGKNOW_MESSAGE_ATT_GR">
    <xs:attribute name="MESSAGE_ID_ATT" type="xs:string"
use="required"/>
    <xs:attribute name="VERSION_NO_ATT" type="xs:byte"
use="required"/>
    <xs:attribute name="CHECKSUM_ATT" type="xs:byte"
use="required"/>
    <xs:attribute name="DATETIME_STAMP_ATT" type="xs:dateTime"
use="required"/>

```

```

use="required"/>
<xs:attribute name="PATIENT_ID_ATT" type="xs:string"
use="required"/>
<xs:attribute name="MESSAGE_ATT" type="xs:string"
use="required"/>
<xs:attribute name="MESSAGE_TYPE_ATT" use="required">
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:enumeration value="PUSH"/>
      <xs:enumeration value="PULL"/>
    </xs:restriction>
  </xs:simpleType>
</xs:attribute>
<xs:attribute name="ACKNOWLEDGE_ATT" type="xs:boolean"
use="required"/>
<xs:attribute name="ERROR_MESSAGE_DESCRIPTION_ATT"
type="xs:string" use="optional"/>
<xs:attribute name="ERROR_MESSAGE_DETAIL_ATT"
type="xs:string" use="optional"/>
</xs:attributeGroup>
<xs:attributeGroup name="PAYLOAD_ATT_GR">
  <xs:attribute name="PAYLOAD_TYPE_ATT" use="required">
    <xs:simpleType>
      <xs:restriction base="xs:string">
        <xs:enumeration value="EMAIL"/>
        <xs:enumeration value="VIDEO"/>
        <xs:enumeration value="MUSIC"/>
        <xs:enumeration value="WARNING"/>
        <xs:enumeration value="REMINDER"/>
        <xs:enumeration value="ALARM"/>
        <xs:enumeration value="SMS"/>
      </xs:restriction>
    </xs:simpleType>
  </xs:attribute>
  <xs:attribute name="PAYLOAD_DEVICE_ATT" type="xs:string"
use="required"/>
</xs:attributeGroup>
<xs:attributeGroup name="PAYLOAD_PARAMETER_ATT_GR">
  <xs:attribute name="TYPE_ATT" use="required">
    <xs:simpleType>

```



```

<xs:restriction base="xs:string">
    <xs:enumeration value="TITLE"/>
    <xs:enumeration value="NAME"/>
    <xs:enumeration value="CONTENT"/>
    <xs:enumeration value="SENDER"/>
    <xs:enumeration value="RECEIVER"/>
    <xs:enumeration value="NUMBER OF
REPEATS"/>
    <xs:enumeration value="ATTACHMENT_ID"/>
    <xs:enumeration value="MESSAGE"/>
    <xs:enumeration value="GATEWAY"/>
    <xs:enumeration value="DELAY OF
MESSAGE"/>
    <xs:enumeration value="TIME BETWEEN
REPEATS"/>
    <xs:enumeration value="DESCRIPTION"/>
    <xs:enumeration value="PICTURE IMAGE"/>
    <xs:enumeration value="VIDEO IMAGE"/>
    <xs:enumeration value="MUSIC IMAGE"/>
    <xs:enumeration value="PICTURE FILENAME"/>
    <xs:enumeration value="MUSIC FILENAME"/>
    <xs:enumeration value="VIDEO FILENAME"/>
</xs:restriction>
</xs:simpleType>
</xs:attribute>
<xs:attribute name="VALUE_ATT" type="xs:base64Binary"
use="required"/>
</xs:attributeGroup>
<xs:attributeGroup name="ATTACHMENT_ATT_GR">
    <xs:attribute name="SYNCH_ATT" use="required">
        <xs:simpleType>
            <xs:restriction base="xs:string">
                <xs:enumeration value="ADDED"/>
                <xs:enumeration value="MODIFIED"/>
                <xs:enumeration value="DELETED"/>
            </xs:restriction>
        </xs:simpleType>
    </xs:attribute>

```

```

<xs:attribute name="FORMAT_ATT">
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:enumeration value="JPEG"/>
      <xs:enumeration value="BMP"/>
      <xs:enumeration value="WAV"/>
      <xs:enumeration value="MPEG"/>
      <xs:enumeration value="TXT"/>
      <xs:enumeration value="DOC"/>
    </xs:restriction>
  </xs:simpleType>
</xs:attribute>
<xs:attribute name="ID_ATT" type="xs:string"/>
<xs:attribute name="LENGTH_ATT" type="xs:byte"/>
<xs:attribute name="SIZE_ATT" type="xs:byte"/>
</xs:attributeGroup>
<xs:attributeGroup name="PREDECESSOR_CONDITION_ATT_GR">
  <xs:attribute name="EVENT_ID_ATT" type="xs:string"
use="required"/>
  <xs:attribute name="EVENT_STATUS_ATT" use="required">
    <xs:simpleType>
      <xs:restriction base="xs:string">
        <xs:enumeration value="User Confirmed"/>
        <xs:enumeration value="User Rejected"/>
        <xs:enumeration value="User Pending"/>
        <xs:enumeration value="User Not Confirmed"/>
        <xs:enumeration value="System Confirmed"/>
        <xs:enumeration value="System Rejected"/>
        <xs:enumeration value="System Not
Confirmed"/>
        <xs:enumeration value="System Pending"/>
      </xs:restriction>
    </xs:simpleType>
  </xs:attribute>
</xs:attributeGroup>
</xs:schema>

```