Towards a Monitoring System for High Altitude Objects

Sébastien Jean[*]
University of Grenoble
LCIS laboratory, EA 3447
50 rue B. de Laffemas
26902 Valence, Cedex 9
France
sebastien.jean@iut-valence.fr

Kiev Gama[†]
University of Grenoble
Laboratoire LIG, UMR 5217
110 avenue de la Chimie
Domaine Universitaire de Saint-Martin-d'Hères
38041 Grenoble Cedex 9
France
kiev.gama@imag.fr

Didier Donsez[‡]
University of Grenoble
Laboratoire LIG, UMR 5217
110 avenue de la Chimie
Domaine Universitaire de Saint-Martin-d'Hères
38041 Grenoble Cedex 9
France
didier.donsez@imag.fr

André Lagrèze[§]
University of Grenoble
LCIS laboratory, EA 3447
50 rue B. de Laffemas
26902 Valence, Cedex 9
France
andre.lagreze@iut-valence.fr

ABSTRACT

High Altitude Objects (HAO), typically sounding balloons, are mobile objects that gather information (e.g. weather data) during their trip and send it to base stations using wireless communication. Once launched, these objects need to be tracked and recovered, and ideally monitored to exploit data in real-time. This paper discusses about middleware and embedded system concerns when monitoring such objects. The architecture that is presented in the following relies on both a monitoring middleware based on a modified RFID suite (part of the OW2 Aspire project, primarily targeting the management of objects in an *Internet of Things* for RFID-based and sensor-based applications), and on an embedded system (part of the HAO) with multimodal communication capabilities. This approach has been validated by two experiments consisting in a real time monitoring of a sounding balloon. The whole application is generic enough to be used to track and monitor other kinds of mobile objects, including sounding rockets and Unmanned Aerial Vehicles (UAVs).

[*]Assitant Professor, CTSYS group.
[†]PhD student, ADELE team.
[‡]Professor, ADELE team.
[§]Assitant Professor, COSY group.

Categories and Subject Descriptors

C.2.1 [**Computer-Communication Networks**]: Network Architecture and Design—*Wireless communications*; C.2.4 [**Computer-Communication Networks**]: Distributed Systems —*Distributed applications, Distributed databases*; H.3.5 [**Information Storage and Retrieval**]: Online Information Services —*Data sharing, Web-based services*

General Terms

Experimentation, Management, Measurement

Keywords

Internet of Things, mobile sensors, middleware, sounding balloon, High Altitude Objects

1. INTRODUCTION

High Altitude Objects (HAO), typically sounding balloons, can fly several hours (wind or motored powered) and reach the stratospheric layer. During the journey, these objects gather data (e.g. weather conditions) by the way of an embedded system including a set of sensors and send data via wireless communication links. Because the landing point of these objects can not be predicted with a high accuracy (simulation software using meteorological forecasts and atmosphere models can spot the landing point of a sounding balloon within a 20km wide area), HAOs need to be tracked in order to be recovered. Furthermore, since these objects can be completely lost, data they gather have to be ideally monitored in real-time.

The work presented in this paper intends to define and validate a real-time monitoring system for such objects. This monitoring system relies on a mobile embedded system and a monitoring middleware. The embedded system has been designed to fulfill some requirements related to energy consumption, extensibility, weight, and taking into account the fact that the HAO can often not be tracked only from a single station and using a single communication link. The monitoring middleware has been built by modifying a RFID middleware conforming to the *Internet of Things* philosophy.

The complete system has been validated by two experiments that have been consisting in the real-time monitoring of a sounding balloon. The last experiment is the result of a teamwork performed by forty high school students, five undergraduate computer science students, under the supervision of four researchers. This collaborative work, with the help of the CNES by the mean of a sponsoring association, has lead to the tracking of a 150km flight that lasted for 3 hours (2 hours ascending, 1 hour falling down). During the journey, the embedded system has been transmitting weather data (temperature, pressure) and GPS location to a base station using both very high frequency (VHF) wide range radio link and a GSM modem. Meanwhile, the middleware that has been adapted to be used in that context has been able to store sensor data and display in real-time monitoring information using graphical interfaces and maps.

The rest of the paper is organized as follows. Section 2 firstly discusses background and motivations. Section 3 presents the proposed architecture and details how a general-purpose RFID middleware has been enhanced to support HAO monitoring. Section 4 gives an overview of the experiments that have validated the approach, also describing the system embedded on the balloon. Section 5 concludes and presents some further work.

2. BACKGROUND & MOTIVATIONS

Existing HAO tracking software like the ones developed by the CNES [1] are not interoperable. They lack of flexibility for tracking and also of richness in data display. They are also not suitable for current applications that use web-centric technologies. Other similar domains, like sounding rocket experiments, share some common points with sounding balloons tracking. However, experiments with rockets [2] often focus only on tracking technologies like GPS and capturing sensor data (e.g. different types of solar rays) available in higher altitudes. Trips are usually of short duration and data is gathered without too much concern in online monitoring like in the approach presented here.

In the embedded system that would go with the high altitude object (HAO), there is a need for multimodal communication when sending data captured during the flight to a base station able to receive and exploit this data. Similar experiments with sounding balloons like [3] usually develop proprietary software targeted to the experiment scenario. That particular experience shares some elements with the approach to be further presented, particularly according to embedded system design. However, no information can be found about the ground system architecture and the communication interface redundancy. In our context, we investigate how to reuse off-the-shelves middleware that could store sensor data and also could enable the real-time tracking and monitoring of the HAO trip by using a web-centric environment. This data should be persistent as well, so it can be retrieved later after the experiment is finished.

Middleware used to track objects in supply chains [4] have a good potential for monitoring other types of objects. If such types of middleware could provide the necessary extensibility for adding and storing the information that needs to be tracked (like geographic positioning, altitude and pressure), they could be easily used in the context of HAOs. A broader vision for tracking objects and their information is found in the *Internet of Things*, a term initially coined by the MIT Auto-ID Center [5], where objects in a network (initially targeting a supply chain) would be individually identified with RFID tags. In a wider sense, this idea evolved [6] as to refer to an ubiquitous object society where different objects are connected, combining RFID, sensor networks and ubiquitous technologies.

In the *Internet of Things*, objects could be identified instantly and related information could be easily found. However, existing standards [7] mainly focus on supply chain objects equipped with RFID tags. Although it exists object identifiers translation standards [8], these ones are still focusing on supply chains, targeting product code schemes instead of ordinary object ”candidate identifiers” (e.g. GSM phone number, MAC address). Other approaches [9] also defend the idea that the IoT should not be RFID-centric. Also, proposals [10] aim at supporting other object identifiers like ISO 14443 or ISO 15693. Other approaches [11], promote IPv6 to identify physical objects inside the *Internet of Things*.

2.1 Architecture

The suitable architecture, depicted in Figure 1, should be flexible and general enough to be used in different contexts for monitoring other kinds of objects like sounding rockets, driftsonde balloons and Unmanned Aerial Vehicles (UAVs).

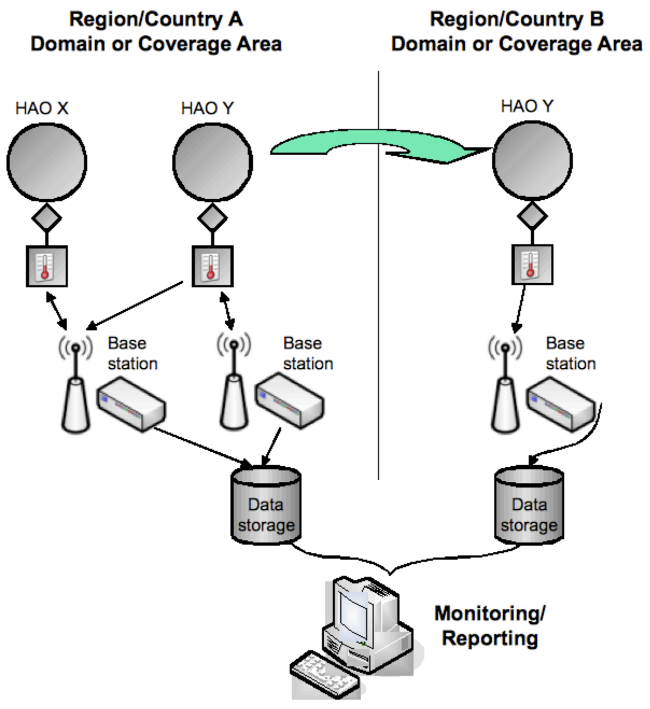


Figure 1: Functional Architecture Overview.

It would enable the collection of data from such types of objects moving across relatively large distances, with the requirement of data persistency in federated databases (a concept that comprises several databases connected through a network).

Due to the wideness of the covered geographical areas, we assume that these objects should be tracked and monitored by different organizations (aerospace security concerns are out of the scope of this paper). For example, if an HAO enters the aerospace of another region and keeps sending its data, another base station should be able to capture the data sent by the object and to store it in a database belonging to this region. Information related to the tracked object should also be shared across federated databases.

In the architecture illustrated in Figure 1, different base stations, fixed or mobile (e.g. a support vehicle), are receiving data from HAOs and are storing it in their associated databases. Other base stations either in the same or in a different coverage area could get information about the same HAO. Even if the information is stored in a different database, monitoring and reporting tools would have access to a federated database. The monitoring and reporting tool could take advantage of web technologies to enable data visualization from any endpoint with Internet access.

2.2 Communications

The next subsections enumerate the different kinds of requirements that the system should fulfill, both under the embedded system and the communications perspectives.

2.2.1 *Embedded System Requirements*

Because the object can be completely lost, the embedded system should be low cost. Although it is not a mandatory feature, having GPS location could enhance monitoring, through the use of a map-based display.

The embedded system should be extensible. The hardware and software basis should be generic enough to be easily extended to support additional/different sensors or applications domains.

Because the HAO journey can last several hours and because its payload weight usually does not allow to use long-last (and consequently heavy) batteries, the embedded system should be energy efficient.

2.2.2 *Communication Requirements*

The communication of the HAO with the base station should have a certain degree of flexibility.

Because of fault-tolerance concerns (communication failures), but mainly as a countermeasure to prevent black-outs that usually and normally occur on some communication networks, several communication interfaces should be available. Wide range HAM radio link suffers for example from the loosing of direct sight when pointing an object far away and hidden behind landscape relieves. GSM operated networks also let some areas uncovered or weakly covered for some economical reasons. Having a set of complementary communication interfaces could consequently strengthen link availability. Nevertheless, the choice of a radio technology must often usually be done according to national rules that sometimes forbid upstream communication.

Communication interfaces used should be energy efficient. This requirement is shared with embedded system ones, for the same reason.

Because the HAO can move far away from its base station (in case of a fixed one), communication interfaces used should be wide range.

3. MIDDLEWARE

For storing and monitoring data, the proposed architecture relies on *AspireRFID RFIDSuite* that has been developed by the LIG laboratory and that is also available as part of the OW2 *AspireRFID* open source project [12]. This RFID middleware allows for storing in a federated database sensor data associated with an object. It also provides graphical applications that can be used as a monitoring console for data visualization. The middleware implements parts of the EPC Global standards [7] that are used in the so called *Internet of Things*. Figure 2 underlines (in the darker squares) parts of the architecture that match EPC Global standards used in the middleware.

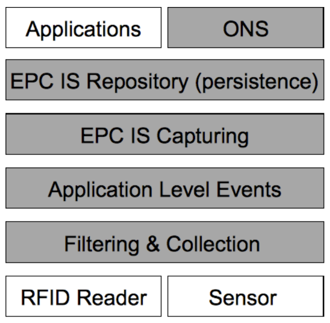


Figure 2: Logical Architecture Layers

RFID/NFC readers are attached to edge computers. The word edge refers to the nearness of the network frontier, where objects (e.g. sensors, RFID tags) can be found. Edges collect RFID tag information and can possibly add to it additional data coming from local sensors. The filtering and collection layer available in that part of the architecture implements minimal filtering capabilities that avoid to send duplicate tag readings (a tag can be scanned more than once in a very small interval of time). The scanned tag and associated data are sent as *Application Level Events* (ALE) reports to the *EPC Information System* (EPC IS) where data is usually stored with persistency. The ALE reports have a predefined set of information (e.g. tag IDs) that can be sent along with it but support extensions for additional data. In the case of our middleware, the ALE structure has been extended to support sensor information. If any sensor is attached to the edge application, the information is also sent along with the ALE report.

Each time a new tag is read, the EPC IS notifies the *Object Naming Service* (ONS), that stores object /EPC IS matching. The ONS standard is promoted by EPC Global and relies on the *Domain Name System* (DNS). The EPC Network ONS infrastructure is managed by Verisign and requires a paying subscription. This kind of monopoly prevents small organizations to benefit from the *Internet of Things* by sharing information about their objects. In addition, ONS mainly targets supply chain products rather than ordinary objects as we propose.

The RFID middleware on which our architecture relies has an ONS-like approach that could be called an *Object Naming Web Service* (ONWS) and that uses HTTP SOAP Web Services instead of DNS. This web services approach allows for easy and free sharing of information through the Internet by different organizations, if using the same middleware on both side.

The concept behind this implementation is rather similar to the idea behind the ONS. Given an object identifier, the ONS returns a list of service endpoints related to this object. It is possible to get, for example, the historic of a given object whose information is spread accross several EPC IS. In our context, edges in base stations from different locations could for example receive information and send it to their corresponding EPC IS. In the case of different organizations (e.g. neighbor countries like France and Germany), as presented in Figure 3, with different EPC IS instances but under the same ONWS, information could easily be shared and the same object could be transparently tracked in the two different systems as if it was only stored in one.

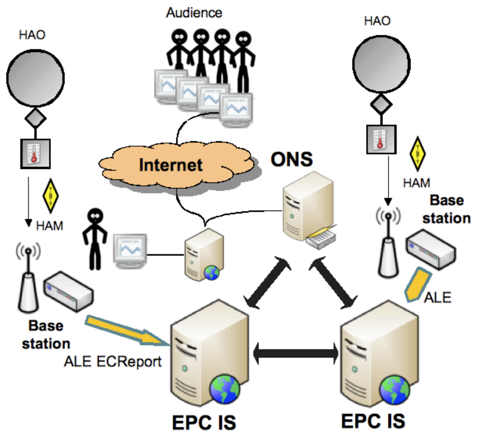


Figure 3: Physical Architecture Overview

The middleware has been designed with scalability in mind, so it is possible to deploy all its elements (filtering and collection, EPC IS, ONS, monitoring application) on one single machine if necessary (e.g. to set up an autonomous mobile station) or to distribute them over a set of networked machines. Its implementation targets the Java platform. In the edges (here, part of base stations), the OSGi platform is used to provide a dynamic component and service based environment for readers and sensors. The EPC IS runs on a *JOnAS* Java EE server that stores information in a relational database (by default an HSQL instance) and exposes part of its functionalities as HTTP Web Services. The ONS is deployed as a HTTP Web Service (that is called ONWS) in a Java EE application server as well. Communication between edge and EPC IS is performed via SMTP, Web Services or JMS (Java Message Service). The latter is used by default. Communication between EPC IS and ONS is performed via Web Services.

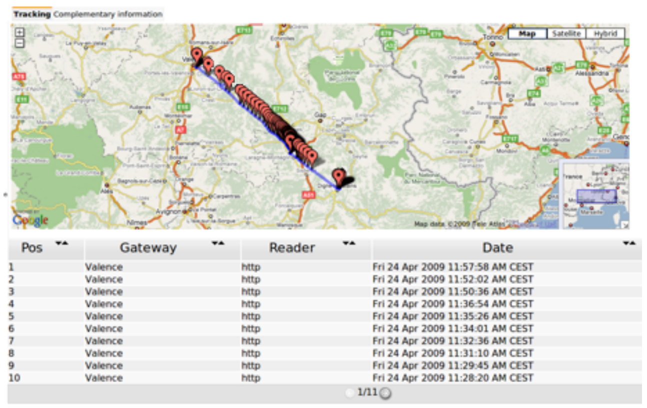
3.1 Adaptations

Sensors and RFID readers are managed in the edge system (i.e. in base station) by software components deployed in a dynamic service platform. In the case of the sounding balloon, all sensor data is however sent at once. Thus, a single frame sent by the balloon either via HAM radio or via an SMS message contains several sensor readings and a "tag scan" (actually being the GUID of the balloon) at once. There should be no sense in using a RFID tag as an identifier for the balloon since no RFID tag scan would take place. Therefore, some adaptations were needed in order to support an ordinary object without any RFID tag ID. In our case, the sounding balloon had to be considered as a pseudo-EPC identified object so it could seamlessly be integrated as a traceable object into the system.

A new module for the edge has been created in order to allow this integration. A web service approach has been used to expose an interface from the RFID middleware so external applications could communicate with it simulating RFID scans and sending sensor data as in the case of the sounding balloon. The choice of an object identifier had to be made, current standards only focusing on product identifiers like barcode to be translated into EPC compliant standards. The phone number associated to the sounding balloon's GSM SIM card has been used as the unique identifier required by the middleware (by translating it into a general product code). Tag scan and sensor readings were broken down into different events, sent separately to the filtering and collection layer, so it could be seamlessly integrated into the RFID middleware (with a behavior similar to the event-based approach regularly deployed).

3.2 Monitoring Console

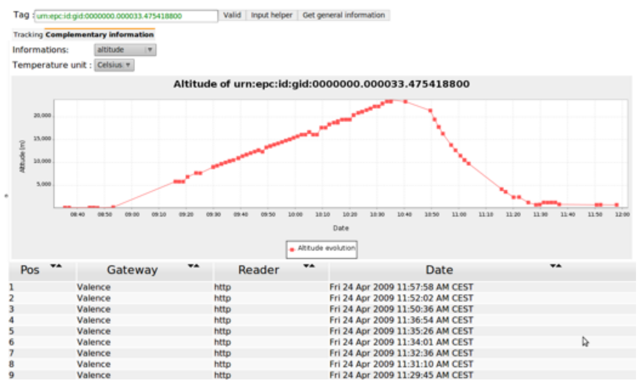
A web console application, that interrogates the ONWS and EPC IS, allows to query and locate object IDs as well as to visualize corresponding sensor readings. A map interface using the *Google Maps* API also allows to see the motion of the balloon, as illustrated in Figure 4.



Pos	Gateway	Reader	Date
1	Valence	http	Fri 24 Apr 2009 11:57:58 AM CEST
2	Valence	http	Fri 24 Apr 2009 11:52:02 AM CEST
3	Valence	http	Fri 24 Apr 2009 11:50:36 AM CEST
4	Valence	http	Fri 24 Apr 2009 11:36:54 AM CEST
5	Valence	http	Fri 24 Apr 2009 11:35:26 AM CEST
6	Valence	http	Fri 24 Apr 2009 11:34:01 AM CEST
7	Valence	http	Fri 24 Apr 2009 11:32:36 AM CEST
8	Valence	http	Fri 24 Apr 2009 11:31:10 AM CEST
9	Valence	http	Fri 24 Apr 2009 11:29:45 AM CEST
10	Valence	http	Fri 24 Apr 2009 11:28:20 AM CEST

1/11

Figure 4: Monitoring Console, Map Interface



Tag : urn:epc:id:gid:0000000.000033.475418800

Informations: altitude

Temperature unit : Celsius

Pos	Gateway	Reader	Date
1	Valence	http	Fri 24 Apr 2009 11:57:58 AM CEST
2	Valence	http	Fri 24 Apr 2009 11:52:02 AM CEST
3	Valence	http	Fri 24 Apr 2009 11:50:36 AM CEST
4	Valence	http	Fri 24 Apr 2009 11:36:54 AM CEST
5	Valence	http	Fri 24 Apr 2009 11:35:26 AM CEST
6	Valence	http	Fri 24 Apr 2009 11:34:01 AM CEST
7	Valence	http	Fri 24 Apr 2009 11:32:36 AM CEST
8	Valence	http	Fri 24 Apr 2009 11:31:10 AM CEST
9	Valence	http	Fri 24 Apr 2009 11:29:45 AM CEST

Figure 5: Monitoring Console, Sensor Data

If such web-centric application can be publicly accessed over the Internet, users located anywhere can monitor and follow-up the data related to the balloon journey providing a real-time experience during the flight. As the information is stored in a database, data can be visualized anytime after the experiment.

Other kinds of measurements like altitude and pressure are monitored via graphs sharing the same generic but adaptable format. Figure 5 provides a snapshot of the generic interface currently displaying the altitude.

4. EXPERIMENTAL VALIDATIONS

Two HAOs have been launched to validate the approach presented in sections 2 and 3. The first launch, in 2008, focused on setting up the embedded system and communication parts. The last launch, in 2009, focused on improving sensors accuracy and monitoring. These experiments have been sponsored by Planete-Sciences, a science promoting association mainly lead by CNES (french space agency). This sponsorship includes administrative arrangement (CNES manages flight permission and insurance), material support (Helium, flight chain elements not related to payload, HAM radio communication device), and technical consulting. The following presents these experiments and the lessons learned.

4.1 Shared Elements

The following describes some elements shared by the two experiments: the flight chain and the HAO hardware basis.

4.1.1 Flight Chain

CNES sponsorship, as it particularly includes insurance management, implies to fulfill drastic safety requirements. The nacelle must be cubic and built using soft materials (e.g. polystyrene), its weight must not exceed 2.5kg and its edges must be at least 30cm long. The communication system must run for at least 3 hours (which is a typical journey duration).

The flight chain, as depicted on Figure 6, consists in four elements:

- a latex balloon, inflated with $4m^3$ Helium gas,
- a radar deflector, used to avoid collision with airplanes,
- a parachute, used to decrease landing speed,
- a nacelle, containing the payload.

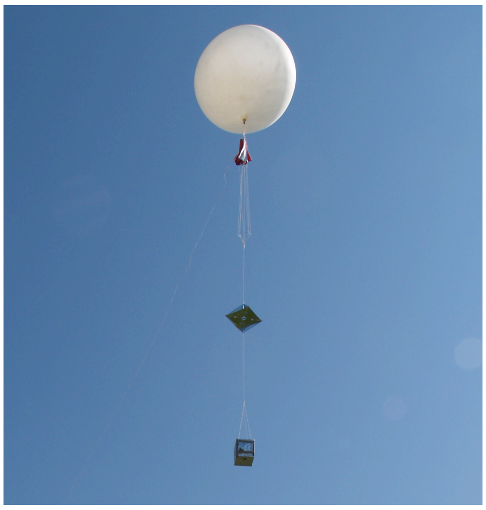


Figure 6: Flight Chain

The flight chain is assembled with the help of a CNES engineer, on the launch day, and the flight can occur if and only if the flight chain fulfills building requirements and if the payload works properly (i.e. data can be received reliably).

4.1.2 HAO Hardware Basis

HAO hardware embedded in the nacelle during the two flights consists in three parts, as depicted in Figure 7:

- a PIC micro-controller,
- a set of analog and digital sensors,
- a multimodal communication interface.

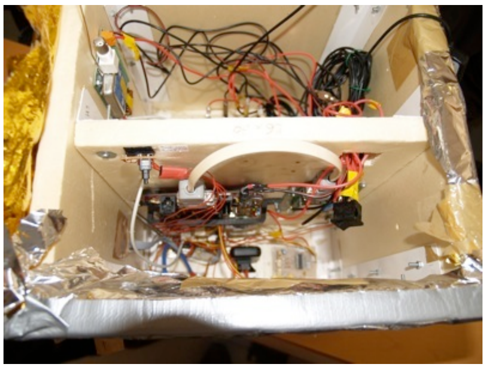


Figure 7: HAO hardware

The PIC architecture has been adopted as a basis for the embedded system because it fulfills the requirement in terms of energy consumption, extensibility (sensors, communication interface), and because of the wideness of the associated open source community.

The program run by the micro-controller polls sensors every minute and sends data as a variable length ASCII string (further called frame) using the multimodal communication interface. The frame format is compliant with the one specified by CNES (it uses the same delimiters), and received frames can consequently be further decoded using CNES free software if needed. The frame contains a time stamp generated on the embedded system, allowing for further correlation of frames received by different stations which are not time synchronized.

The set of sensors includes:

- temperature sensors placed inside and outside the nacelle, connected to the micro-controller using digital or analog links,
- pressure sensors, placed inside the nacelle, connected to the micro-controller using analog links,
- a GPS receiver, connected to the micro-controller using a RS232 serial line.

The multimodal communication interface consists in two links, respectively related to operated and non operated networks:

- a GSM interface, used to send and receive SMS (the upstream mode is used to force the immediate sending of data),
- a HAM radio interface, used only in downstream mode.

When using the GSM interface, the frame is directly mapped to a SMS. When using the HAM radio interface, digital data are firstly sent at a rate of 600 Baud on a RS232 serial line to a FSK hardware modulator (using 900/1200Hz frequencies). Then, analog signal is transmitted by a frequency modulation emitter (called Kiwi Millenium, and provided by CNES) using 137.950MHz as carrier signal frequency.

There are two separate sets of batteries, representing 25% of the payload weight, in order to ensure the complete system to last enough. The first one provides a 9V / 250 mA power supply for micro-controller, sensors (including GPS) and GSM interface. The other is dedicated to the radio interface, which requires approximately 200mA to work properly. Each set having a total capacity of 5000mAh, the system can run up to 20 hours (which is much more than the journey duration but helpful if HAO recovery takes time).

4.2 First Launch

The following gives some details about the first launch which occurred in Easter 2008.

4.2.1 HAO and Stations

During this experiment, only digital sensors have been used. Temperatures (internal/external) have been measured using LM75 sensors, linked to the micro-controller via an I2C bus. Pressure has been measured using a MS55534A sensor, linked to the micro-controller via an SPI bus.

Two stations have been used to track the flight:

- a base station using CNES hardware and software to handle radio frames, and using a GSM modem and a console-based application to handle SMS. Cartography has been managed separately by pointing each frame on *Google Earth*,
- a mobile station consisting in a car without Internet connection, equipped with two HAM radio receivers and a GSM. Radios receivers have been used without FSK demodulators (not available on time), forbidding onboard frame decoding.

4.2.2 Flight and Results

The HAO made a 210 minutes journey, climbing up to 32 km high, landing 150 km away.

The fixed station could received radio frames during the first 150 minutes of the flight but lost the balloon as it disappeared under the horizon during the landing phase (at an altitude of approximately 8km).

GSM signal, and consequently SMS frames receiving, was lost 15 minutes after the launching, at an altitude of approximately 3km. GPS signal was lost at an altitude of 24km (as predicted by constructor's datasheet). Due to a software bug, the lack of GPS signal induced a radio black-out of 45 minutes that ended as the HAO fell down under 24km. The mobile station started tracking the flight 90 minutes after the launching, driving from the base station location. After the HAO has been lost by the base station, the mobile station (too far from the HAO position to receive radio signal) succeeded in getting position via SMS, at an altitude of 2km. No further SMS request succeeded, since the balloon had entered a GSM black-out area. Radio signal has been gotten back near the last known position, and the balloon has been retrieved 2 hours later (8 hours after launching) by performing triangulation with the two portable HAM radio receivers and directional antennas.

Due to casing issues, received temperatures were not realistic and could not be exploited. Pressure sensor suffered of malfunction immediately after the launching (it may have been unplugged). GPS data collected permitted to further compute ascending and descending speed and, by extrapolation, to determinate the highest point reached.

4.3 Second Launch

From the lesson learned from the first one, the second launching, which has been made in Easter 2009, focused on improving mobile station as well as middleware.

4.3.1 HAO and Stations

For this experiment, analog temperature sensors made by the high school students group have been added, linked to the micro-controller via an ADC line. Digital pressure sensor has been replaced by analog ones, also linked to the micro-controller via an ADC line. A tiny camera (designed for RC planes, and with a weight of 38g) with built-in SD card storage has also been used to take still pictures of the ground.

Two stations have been used to track this flight:

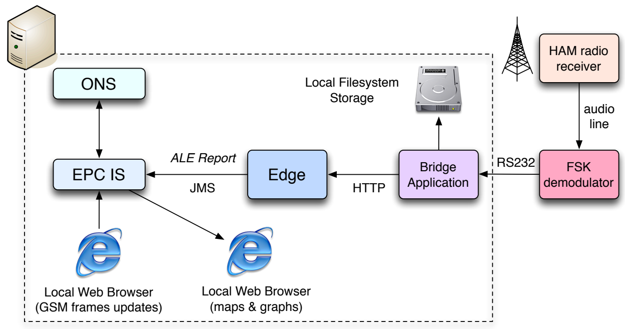


Figure 8: Station Hardware and Software Overview

- a fixed station, whose architecture overview is illustrated on Figure 8, using CNES hardware and in which has been deployed the entire middleware described in section 3 (edge, EPC IS and ONWS). An intermediate open source software has been developed (in Java for portability) as a minimal core application for autonomous mobile stations. It provides console-based display, local text-based filesystem storage and generic TCP-based forwarding. This software has been included in the base station to send frame coming out of the receiver serial line to the edge's HTTP Web Service Module. The GSM part has not been coupled with the software suite, it has been handled separately, with a GSM phone.

- a mobile station consisting in a car with a 3G Internet connection, equipped with a HAM radio receiver coupled to a FSK demodulator and a GSM phone.

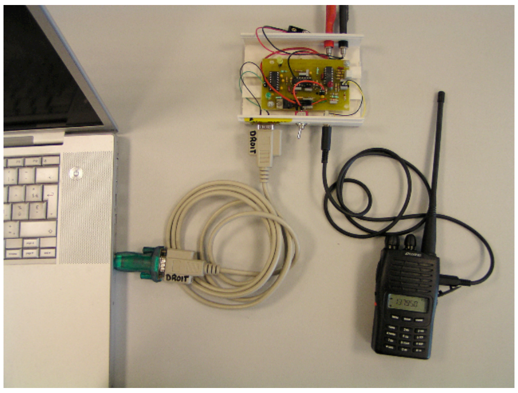


Figure 9: Mobile Station

The mobile station system used, which is presented on Figure 9 consists in a portable HAM radio receiver, a FSK hardware demodulator, and a laptop. The signal coming from the receiver (by the way of speaker connector) is demodulated and frames are finally sent to the laptop via a RS232 serial line. The mobile station was here a kind of "mobile fixed station" in the sense that the entire middleware, light enough, has been deployed on the laptop (instead of acting as an edge and forwarding the events to the fixed station's EPC IS).

4.3.2 Flight and Results

The HAO made a 180 minutes journey, climbing up to 26 km high, landing approximately 150 km away (10km from first launch's landing point).

Radio frames were never lost during the flight, as the mobile station (started earlier than the first time) could receive frames all along the trip. GSM signal was lost at an altitude of approximately 5km. The balloon landed in a covered area, SMS requests could consequently be sent successfully after the landing. Balloon recovery was much faster because of the knowledge of landing GPS location.

Some sensors did not work properly, but pressure and external temperature data could be exploited and correlated with theoretical values. Still pictures taken by the onboard camera could be further correlated with GPS locations.

The entire flight has been followed in realtime by forty high school students in an amphitheater, using graphical user interfaces (maps and graphs presented in Figure 4 and Figure 5) available from the middleware. It could also have been followed from any Web browser.

5. CONCLUSIONS & FURTHER WORK

As part of the middleware infrastructure supporting the *Internet of Things*, we can find federated databases that can store and can enable object information monitoring. Although these objects are basically RFID tagged products on a supply chain, a RFID middleware can easily be enhanced to be suitable for ordinary objects tracking. In the context of HAOs, such adaptation combined with mobile technologies has allowed to enable sounding balloon monitoring inside the *AspireRFID RFIDSuite.*

The pseudo EPC identifier used for the sounding balloon is not compliant with the naming scheme as defined by EPC Global. Thus, in a global context requiring to connect to the ONS paying service managed by Verisign, our approach would possibly have integration problems. In the context of our middleware with multiple EPC IS connected with the ONWS, the location of non-RFID identified objects works without any issue, as it has been validated by experiments. Since we found a practical scenario where an alternative to the product code oriented environment is needed outside the scope of supply chain contexts, we can suppose that standards that govern the *Internet of Things* would evolve in such a way that soon any object could be supported. We see no issue to conform to these to-be-defined naming schemes.

The approach presented in this paper is generic enough for further use in different contexts like monitoring data captured by Unmanned Aerial Vehicles (UAV), sounding rockets, driftsonde balloons or other objects that use mobile technologies for sending data similarly to what has been described here.

As further work, we plan to launch another sounding balloon in April of 2010 with multiple base stations feeding different databases. Thanks to the web-centric approach adopted by the middleware, the monitoring of the balloon trip will be available through the Internet in real-time from anywhere. It is also planned to avoid the use of a hardware FSK-demodulator by replacing it by a software demodulation done using the built-in audio card available on every laptop/machine. This would make the following of the balloon by HAM radio operators easier since it would no longer require additional hardware.

6. ACKNOWLEDGMENTS

Part of this work has been carried out in the scope of the Aspire project (http://www.fp7-aspire.eu) which is co-funded by the European Commission in the scope of the FP7 programme under contract number 215417. Help and contributions from all partners of the project and also the OW2 AspireRFID community are acknowledged.

The authors thank the CNES, all students (high school and undergraduate) and other people that have contributed to this project and participated in the assembly and launching of the sounding balloons, as well as HAM radio operators that have achieved the recovery process.

7. REFERENCES

[1] Logiciels de Réception de Télémesures. http://www.planetesciences.org/espace/basedoc/ Logiciels_de_réception_des_télémesures

[2] Montenbruck O., Markgraf M., Turner P., Engler W., Schmitt G.; GPS Tracking of Sounding Rockets - A European Perspective; NAVITECH'2001, 10-12 Dec. 2001, Noordwijk.

[3] Flight 2 - High Altitude Object. http://www.natrium42.com/halo/flight2/

[4] Kefalakis, N., Leontiadis, N., Soldatos, J., Gama, K., and Donsez, D. 2008. Supply chain management and NFC picking demonstrations using the AspireRfid middleware platform. In Proceedings of the ACM/IFIP/USENIX Middleware '08 Conference Companion (Leuven, Belgium, December 01 - 05, 2008). Companion '08. ACM, New York, NY, 66-69. DOI= http://doi.acm.org/10.1145/1462735.1462751

[5] AUTO-ID Labs. http://autoid.mit.edu/cs/

[6] Yan, L., Zhang, Y., Yang, L.T., Ning, H. (Eds). 2008. The Internet of Things; from RFID to the next-generation pervasive networked systems. Auerbach Publications.

[7] EPC Global Inc. http://www.epcglobalinc.org

[8] EPC Global Inc. Tag Data Translation Standard. http://www.epcglobalinc.org/standards/tdt

[9] Williams, B. What is the Real Business Case for the Internet of Things? In: Synthesis Journal, ITSC, 2008.

[10] Schmidt, L., Mitton, N., Simplot-Ryl, D. Towards Unified Tag Data Translation for the Internet of Things. In: Proceedings of Wireless Communication Society, Vehicular Technology, Information Theory and Aerospace & Electronics Systems Technology (VITAE'09), Aalborg, Denmark, 2009

[11] Cheekiralla, S., Engels, D.W., An IPv6-Based Identification Scheme, ICC '06. IEEE International Conference on Communications, vol.1, no., pp.281-286, June 2006

[12] OW2 AspireRFID. http://wiki.aspire.ow2.org