# Unmanned Aerial Vehicles as MobileMulti-sensor Platforms

Matthes Rieke, Theodor Foerster, Arne Broering

Institute for Geoinformatics – University of Muenster, Germany
{m.rieke, theodor.foerster, arneb}@uni-muenster.de

## ABSTRACT

Unmanned Aerial Vehicles (UAVs) equipped with multiple sensors publish data in unpredictable manner using considerable bandwidth and missing spatial reference. Additionally, the available approaches only provide proprietary access to the data. Integrating sensors from UAVs into a synchronized data stream is required for publishing the transmitted data on the Web. This article describes a software framework, which simplifies such integration. In particular, it synchronizes registered sensor data streams and attaches a spatial reference through a geotagging mechanism. Moreover, the resulting data is published on the Web and made accessible in an interoperable way using Sensor Web technology. The presented framework is applied to a multi-sensor equipped UAV for a real-world scenario exploring the surface-near atmosphere of the earth.

## 1 INTRODUCTION

The goal of our use case is to analyse the surface-near atmosphere (the *Prandtl-Layer* ranging between 50 and 100 m above ground level, Oertel [2004]) and its meteorological inversions. Classic sensor platforms such as measurement towers or helicopters are limited in flexibility and their operating height. With respect to our use case these limitations point out the requirement ofan alternative approach. Likewise considering the high costs of classic measurement methods, we initiated the development of a highly flexible multi-sensor platform using an Unmanned Aerial Vehicle (UAV). Operating a multi-sensor equipped UAV requires integrating various low-level protocols of different vendors to gather sensor data. Currently, this integration is achieved pair wise and an efficient publishing of the sensor data on the Web in an interoperable way is not yet supported by the existing approaches. Therefore, this article contributes a framework a) for transmitting data gathered by sensors attached to UAVs in real-time to a groundstation and synchronizing those multiple data streams to one geotagged stream and b) for publishing the data on the web. The framework is applied to investigate different phenomena of the atmosphere with respect to the above mentioned use case and to make the sensor data accessible on the web using Sensor Web technology (Section 2). The presented work is applied to the Mikrokopter UAV[1].

At first we give an overview of existing UAVs and their processing software as well as describing the concept of the Sensor Web (Section 2). Subsequently the architecture (Section 3) and implementation (Section 4) of the framework is described. The article is closed by a conclusion and an outlook on future tasks (Section 5).

## 2 RELATED WORK

This section describes the fundamentals as applied in this work. In particular, it introduces UAVs and reviews the Sensor Web. The sensor web is used in this work to publish the data gathered by the UAV on the web. Moreover, we review the existing software frameworks for collecting sensor data from UAV and show that the framework described in this article is unique.

### 2.1 Unmanned Aerial Vehicles

During the last years, unmanned aerial vehicles have reached an unprecedented scale in application, both in the military and the civilian domain (Anibal [2007]). Especially the Vertical Take-Off and Landing (VTOL) units have been used in different non-military applications including power line inspection, airborne surveillance, fire prevention or agricultural applications (Valavanis [2007]). This development has been aided by the comparable cheap investments into such UAVs.

[1]Mikrokopter website: http://www.mikrokopter.de/.

The centerpiece of aVTOL is a microcontroller processing data provided by the Inertial Measurement Unit (IMU). The IMU observes the UAV's location in 3D-space by measuring the Earth's magnetic field with three accelerometers and magnetometers. The microcontroller reacts to changes by adjusting the propulsion of the UAV. Current available civil UAVs are equipped with a built-in GPS module and therefore are versatile for different geographic applications. Additionally most are equipped with a built-in wireless communication link (e.g. ZigBee[2]modules) for real-time tracking of the UAV bya connected ground station (e.g. a common Laptop).

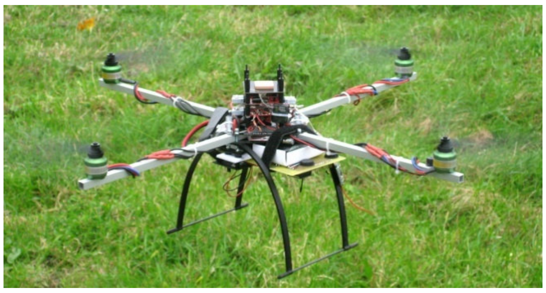


***Figure 1:*** 4-rotor Mikrokopter.

The applied Mikrokopter (Figure 1) is a low-cost UAV equipped with a VTOL using a modular architecture. The minimal version consists of an IMU controlling the UAVs flight attitude. It can be extended by a GPS module, a barometric height unit and a magnetic heading module (Buechi [2010]). Another civilian UAV equipped with a VTOL is the Microdrone[3]. This UAV aims at professional customers. Several customized sensors – mainly serving the digital video sector – can be mounted to the UAV. From the Arduino open-source community evolved the ArduCopter[4] UAV platform. Its architecture allows the user to attach multiple custom sensors. Table 1 summarizes the main characteristics of these UAVs. For all three platforms a data processing software is available, however, these lack a mechanism to integrate the carried sensors with an interoperable infrastructure, such as the Sensor Web.

| | **Mikrokopter** | **Microdrones** | **ArduCopter** |
|---|---|---|---|
| Sizes (diameter) | 30-70 cm | 70-103 cm | 45-60 cm |
| Number of rotors | 4-8 | 4 | 4-8 |
| Maximum payload | 300-800 grams | 200-1,200 grams | 400-500 grams |
| Wireless downlink | Bluetooth or HF connection | 2,4 GHz quad-diversity receiver | XBee Pro |

***Table 1:*** UAV overview.

## 2.2 Sensor Web

The Sensor Web Enablement (SWE)[5]initiative of the Open Geospatial Consortium (OGC) standardizes the integration of sensors into the Sensor Web. As part of OGC's specification program the SWE initiative developed several standards for web services and data models forming an interoperable Sensor Web (Botts et al. [2008]). The basis for this infrastructure is the information model comprising a sensor description language (SensorML) as well as a data model for gathered sensor measurements (Observations & Measurements). These two concepts enable the interoperable exchange of sensor data and corresponding metadata between different services of the Sensor Web. These web services include the Sensor Observation Service (SOS), Sensor Planning Service (SPS), and the Sensor Event Service (SES). The SOS is designed for accessing real time as well as historic sensor data, and sensor metadata. Its operations allow users to request sensor data based on spatial, temporal and thematic filter criteria. Requested sensor data are encoded conforming to the Observations & Measurements standard and the sensor metadata are returned as SensorML documents. To control and task sensor platforms the Sensor Planning Service (SPS) can be used. In combination with UAVs, the SPS can serve as an interoperable interface for mission planning. A publish/subscribe-based delivery of

[2]ZigBee website: http://www.zigbee.org/.
[3]Microdrone company website: http://www.microdrones.com/.
[4]ArduCopter website: http://code.google.com/p/arducopter/.
[5]http://www.ogcnetwork.net/swe/.

sensor data in real-time is provided by the Sensor Event Service (SES). The SES incorporates a complex event processing (Luckham [2008]) technique to filter on sensor data streams. In an industry surveillance scenario, where a UAV with attached gas detection sensors could be utilized, the SES could determine the exceeding of pre-defined gas occurrence thresholds.

## 2.3 Similar Software Approaches

A software framework which is explicitly designed for the fast and flexible integration of heterogeneous sensors is the Global *Sensor Networks* (GSN) middleware (Aberer et al. [2006]). Virtual sensors provide the middleware with a sensor data stream as a sequence of time stamped data tuples. GSN filter and store these data streams internally using publish/subscribe mechanisms in contrast to this work where complex stream processing is delegated to the higher service layer. Another approach, the *Sensor Web Agent Platform* (SWAP, Moodley and Simonis [2006]), provides a three-layered architecture for integrating raw sensor data into a multi-agent system. The sensor layer holds multiple instances of sensor agents which access the sensor data directly or make use of different intermediary (web) services. Sensor agents pass their data on to the knowledge layer where re-usable simulation and processing methods are applied. The application layer has the purpose to present the data to a non-agent end-user or to establish connections to several web services. However, both GNS as well as SWAP lack a mechanism to synchronize multiple data streams which is a fundamental requirement to manage highly dynamic data streams coming from sensors carried by UAVs.

## 3 ARCHITECTURE OF THE PROPOSED FRAMEWORK

The software framework has been designed to fulfill the requirements of our real-world use case (Section1). To analyze the surface-near atmosphere, three different phenomena shall be observed*: air temperature*, *relative humidity* and *barometric pressure*. For referencing the local observations (with no location attached) in a geographic context (in 3-D space) a geotagging mechanism for each measurement is required. To meet these requirements we developed a ground station unit capable of processing any incoming data from the UAV.

In general, a software framework features both a reusable design along with a reusable implementation (Hohpeand Woolf [2003]) and a high coherence between the framework classes (Gamma et al. [1995]). Akey characteristic of software frameworks are *extension points* which act as interfaces for plugins(Buschmann [1998]). Extension points support the *inversion of control* paradigm. Our framework provides such extension points for data sources and output components(Figure 2). Plugins for such components can be written in a flexible manner so that the integration of arbitrary kinds of sensors attached to UAVs is supported through the framework. Besides, the components for supplying sensor data of a multi-sensor platform, the framework comprises components for coupling different phenomena and for providing processed data to a higher-level layer, such as the Sensor Web. These components are introduced in the following.

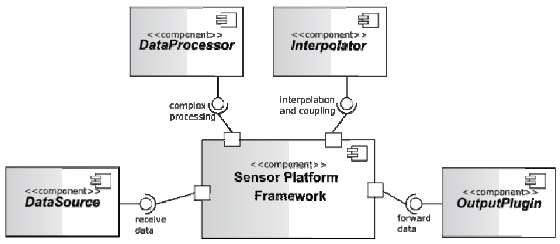


***Figure 2:*** Framework model with extension points.

First, data source plugins establish the connection to the multi-sensor platform and process the incoming data. It allows our software framework to access the measured data as well as its metadata (the *plugin description*). The metadata is based on SensorML (Section 2) for describing the inputs which act as the actual observed phenomena of the sensor platform. Besides these inputs, the meta data document characterizes the plugin-specific behaviour of the framework, especially the generation ofoutput data based on:

- a periodic time interval or
- the availability of a specific phenomenon.

Figure 3 exemplifies the two types of behaviour of an incoming sensor data stream. The *Period-Behaviour* indicates the generation of output every 20 seconds. In contrast, the *AvailabilityBehaviour* waits until new data of the phenomenon *temperature* has been collected. This example points out the need of an *interpolation mechanism* as the phenomena arrive at different points in time. Such an interpolation mechanism is described in the following section.

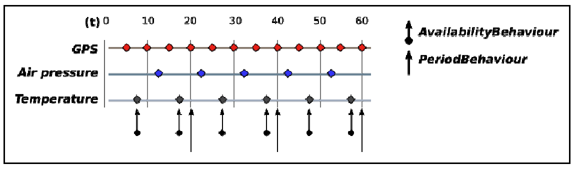


***Figure 3:*** Different types of behaviour.

The main task of the framework core is to manage the data of registered data sources. The framework collects available data the actual measurements of any registered data source (Figure 4).By modelling the architecture this way, we ensure the *inversion of control* paradigm, as extending component's functions are only triggered by the framework's control flow.

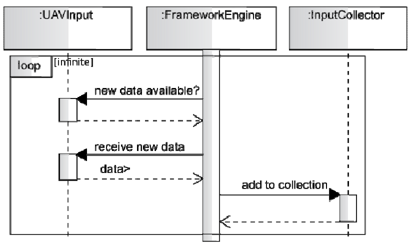


***Figure 4:*** Request of newly available data.

The framework manages the generation of output for each data source separately according to its description. Figure 5 depicts the actions performed after an output generation has been triggered. First of all the interpolation mechanism is called. As exemplarily illustrated by Figure 3 the phenomenon *temperature* arrives with a different timestamp than the GPS signals. Assuming the providing data source generates output based on the availability of *temperature* measurements, all other phenomenon data are interpolated for the timestamp of this measurement leaving the temperature measurements untouched. This part forms the function to synchronize the different phenomena and is another extension point of the software framework. A developer can substitute the built-in mechanism (a linear interpolation algorithm) by complex algorithms fitting individual requirements of possible scenarios. After the interpolation and the call of all registered data processor components, the processed data is forwarded to output plugins.

The output plugin interface is also designed as an extension point to the framework to support multiple output formats. An output plugin retrieves the processed data from the framework core and converts it to a specific format. For example, it can create connections to existing web services or a harddisk to store the data.

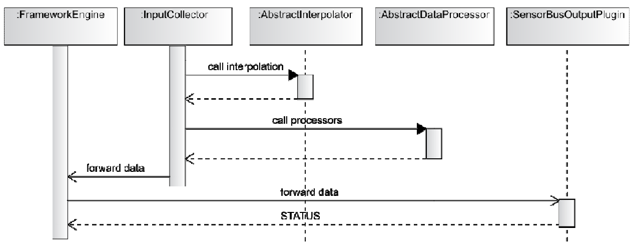


***Figure 5:*** Interpolation, processing and output of data.

For publishing the incoming sensor data streams in an interoperable way, as required in our use-case, we created an output plugin for the Sensor Web. Since the Sensor Web, as defined by OGC's SWE framework, comprises different kinds of web services such as SOS and SES (Section2) suitable for sensor data as provided by a UAV, we included an additional abstraction layer into the architecture, the so-called Sensor Bus (Broering et al [2010]). The Sensor Bus realizes a communication infrastructure which underlies the different Sensor Web services (Figure 6). The output plugin publishes the incoming sensor data on the Sensor Bus in a well defined format. These data messages are retrieved by service adapters which translate the sensor data into the service specific encoding. Consequently, by adapting the output plugin to the Sensor Bus and its well-defined communication protocol, the measured data can easily be published to all Sensor Web services which are listening on the SensorBus.

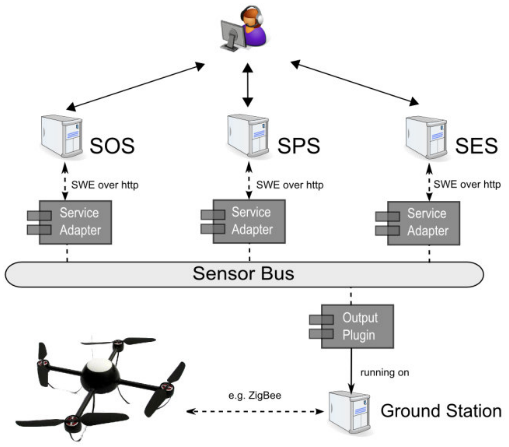
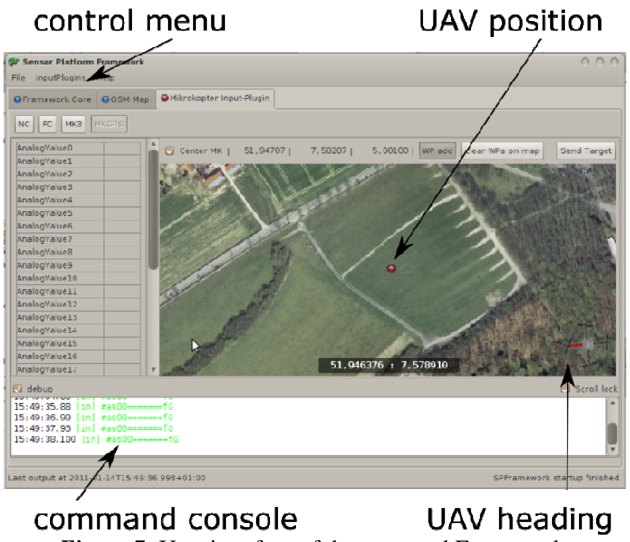


***Figure 6:*** High-level architecture.

## 4 IMPLEMENTATION AND EVALUATION

The control flow of the implemented prototype is maintained by the framework core according to *inversion of control*. It is responsible for managing the system memory and thereby ensures the stability of the software framework. In particular, it interacts with the extension points (data sources, output plugins) for retrieving or sending data. All retrieved measurements of a phenomenon are hold in a time stamped internal list as long as needed with respect to the behaviour of the output plugins.



```
15:49:35.88 [in] #as00=======f0
15:49:36.90 [in] #as00=======f0
15:49:37.93 [in] #as00=======f0
15:49:38.100 [in] #as00-------fG
```

***Figure 7:*** User interface of the proposed Framework.

The framework also includes a graphical user interface featuring a map view (based on OpenStreetMap[6]) where the current position, the magnetic heading and the raw data stream of the UAVis visualized (Figure 7). Thus, the user is able to trace the flight route and the heading of the UAV. For communicating with the Mikrokopter (Section 2), which is utilized in our use case, we implemented a particular data source component that is able to parse the incoming binary data streams (i.e. sensor

[6]OpenStreetMap project website: http://www.openstreetmap.org/.

data and location data separately). The used Mikrokopter is enhanced with a standalone computing board managing the interaction with the actual sensors (a Sensirion[7]SHT75 temperature/humidity sensor).Based on this approach we are able to easily integrate new sensors to the multi-sensor platform as it is completely independent of the UAV-specific electronics. However, the data source component has to process both the stream of the atmosphere-sensor board and the stream providing the GPS position of the Mikrokopter.

We evaluated the entire communication chain, including the Sensor Bus and two Sensor Web services(the SOS and SES), by conducting a stress test which iteratively scaled up the system parameters (the number of data tuples per second). In this scenario, we used a Sensor Bus implemented as an XMPP[8]multi user chat using an Openfire[9]server hosted on a 2.67 GHz Pentium IV machine with a DSL 2000 network connection. Figure 8 depicts the differences between two test series. Both series have been conducted with the same virtual phenomena measurements including *temperature, air pressure*, *humidity* and *GPS position*. The green graph displays a time series where no synchronization of measurements was applied in contrast to the red graph showing the same measurements synchronized based on timestamps of temperature measurements.

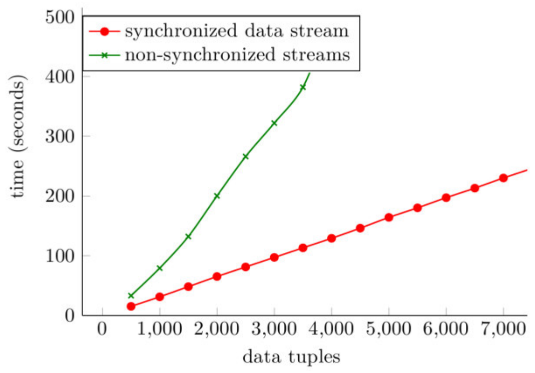


***Figure 8:*** Test results by comparison.

The test results showed that the system runs stable with up to 33 data tuples per second showingno scalability barrier when the framework´s synchronization mechanism is applied. Thus the framework using the Sensor Bus as the output mechanism has a linear scalability while being able to handle up to 33UAVs publishing data once a second on the same server instance. When bypassing the synchronization the output mechanism quadruples the outgoing Sensor Bus messages (one message per phenomenon measurement) resulting in a loss of performance.

## 5 CONCLUSION AND FUTURE WORK

The presented framework enables the communication with multi-sensor equipped UAVs.Supplied with a plugin-based architecture, it provides a generic approach regarding any kind of data source in the field of mobile sensor platforms. In combination with the Sensor Bus, as an interoperable communication infrastructure, the integration of mobile sensor platforms with the Sensor Web is possible. Thereby, the framework merges single unsynchronized data streams by interpolating the data based on timestamps. This synchronization mechanism is key in the field of highly dynamic data streams as retrieved from UAVs. The integration of a UAV for gathering atmospheric data has been evaluated with the proposed approach. The tests showed a linear scaling of the deployed system, reaching up to 33 data tuples per second.

In the long term, communicating with swarms of mobile sensor platforms (Rothermich et al. [2004])and managing their incoming data streams with the proposed software framework is an interesting field of research. Individual UAVs could create ad-hoc networks enabling a *swarm intelligence* while communicating with each other. Thereby such swarms are able for example to detect and trace gas plumes in industrial sites in an efficient way.

[7]Sensirion company website: http://www.sensirion.com/.
[8]XMPP website: http://xmpp.org/.
[9]Openfire website: http://www.igniterealtime.org/projects/openfire/.

## BIBLIOGRAPHY

Karl Aberer, Manfred Hauswirth, and Ali Salehi. A middleware for fast and flexible sensor network-deployment. In Proceedings of the 32nd VLDB Conference, Seoul, Korea, 2006.

OlleroAnibal. Multiple heterogeneous unmanned aerial vehicles.Springer, 2007.

M. Botts, G. Percivall, C. Reed, and J. Davidson. OGC Sensor Web Enablement: Overview and High Level Architecture. Lecture Notes In Computer Science, 4540:175–190, 2008.

A. Broering, T. Foerster, S. Jirka, and C. Priess. Sensor Bus: An Intermediary Layer for Linking Geosensor Networks and the Sensor Web. In COM.Geo '10: Proceedings of the 1st International conference on Computing for Geospatial Research and Application, pages 1–8, New York, NY,USA, 2010.

Roland Buechi. Faszination Quadrokopter. Verlag fuer Technik und Handwerk, 2010.

Frank Buschmann. Pattern-orientierte Software-Architektur: ein Pattern-System. Addison-Wesley,1998.

Erich Gamma, Richard Helm, Ralph Johnson, and John Vlissides. Design Patterns: Elements of Reusable Object-Oriented Software. Addison-Wesley, 1995.

G. Hohpe and B.Woolf. Enterprise integration patterns: Designing, building, and deploying messaging solutions. Addison-Wesley Longman Publishing, Boston, MA, USA, 2003.

David Luckham. The Power of Events: An Introduction to Complex Event Processing in Distributed Enterprise Systems. Addison-Wesley, 2008.

DeshendranMoodley and Ingo Simonis. A new architecture for the sensor web: The swap framework. In 5th International Semantic Web Conference ISWC, 2006.

Herbert Oertel. Prandtl's essentials of fluid mechanics.Springer, 2004.

Joseph A. Rothermich, M. IhsanEcemis, and Paolo Gaudiano.Distributes localization and mapping with a robotic swarm. Swarm robotics: SAB 2004 international workshop, 2004.

KimonValavanis. Advances in unmanned aerial vehicles: state of the art and the road to autonomy. Springer, 2007.