ISPRS Int. J. Geo-Inf. **2015**, *4*, 385-399; doi:10.3390/ijgi4010385

OPEN ACCESS

ISPRS International Journal of
Geo-Information
ISSN 2220-9964
www.mdpi.com/journal/ijgi

Article

A Sensor Web-Enabled Infrastructure for Precision Farming

Jakob Geipel [1,*], Markus Jackenkroll [2], Martin Weis [3] and Wilhelm Claupein [1]

[1] Institute of Crop Science, University of Hohenheim, Fruwirthstr. 23, 70599 Stuttgart, Germany; E-Mail: wilhelm.claupein@uni-hohenheim.de

[2] Institute of Phytomedicine, University of Hohenheim, Otto-Sander-Str. 5, 70599 Stuttgart, Germany; E-Mail: m.jackenkroll@uni-hohenheim.de

[3] Centre for Geodesy and Geoinformatics, University of Applied Sciences (HFT) Stuttgart, Schellingstr. 24, 70174 Stuttgart, Germany; E-Mail: Martin.Weis@hft-stuttgart.de

* Author to whom correspondence should be addressed; E-Mail: jakob.geipel@uni-hohenheim.de; Tel.: +49-711-459-22938; Fax: +49-711-459-22297.

Academic Editors: Georg Bareth, Fei Yuan and Wolfgang Kainz

Received: 30 November 2014 / Accepted: 5 March 2015 / Published: 18 March 2015

Abstract: The use of sensor technologies is standard practice in the domain of precision farming. The variety of vendor-specific sensor systems, control units and processing software has led to increasing efforts in establishing interoperable sensor networks and standardized sensor data infrastructures. This study utilizes open source software and adapts the standards of the Open Geospatial Consortium to introduce a method for the realization of a sensor data infrastructure for precision farming applications. The infrastructure covers the control of sensor systems, the access to sensor data, the transmission of sensor data to web services and the standardized storage of sensor data in a sensor web-enabled server. It permits end users and computer systems to access the sensor data in a well-defined way and to build applications on top of the sensor web services. The infrastructure is scalable to large scenarios, where a multitude of sensor systems and sensor web services are involved. A real-world field trial was set-up to prove the applicability of the infrastructure.

Keywords: Sensor Web Enablement; Open Geospatial Consortium; precision farming; interoperable; open source; 52° N; sensor; UAS; web service

1. Introduction

The use of sensor technologies is more and more applicable in agriculture nowadays. In the domain of precision farming (PF), it is an inevitable aid for the generation of site-specific spatial and temporal information to support crop management strategies [1–3]. Within the last decade, several agricultural machinery and sensor construction companies have established a multitude of sensor systems for sensing soil- and plant-related parameters, as well as for sensing environmental impact factors, influencing the development of the cultivated plants [3]. Most of these sensor systems are designed for: (i) stationary use, e.g., soil moisture sensing networks [4,5]; (ii) hand-held use, e.g., fluorescence and hyper-spectral reflection sensors [6]; or (iii) mobile use on ground-based sensor platforms, e.g., fluorescence, hyper-spectral reflection and ultrasonic sensors, which are mounted on tractors [7–10]. Recent development added the possibility for (iv) mobile use on aerial sensor platforms, e.g., camera systems, which are mounted on unmanned aerial vehicles (UAVs) or unmanned aircraft systems (UASs) [11–13].

Most of these sensor systems are operated with vendor-specific control units, user interfaces and communication protocols. As this varies from sensor system to sensor system, using sensors from different vendors may quickly lead to complex, inconsistent and time-intensive procedures for sensor data storage, processing and distribution. Moreover, many sensor systems are integrated into decision support systems for site-specific online and offline applications and are implemented on tractor terminals, e.g., the Yara N-Sensor (Yara International ASA, Germany) and the GreenSeeker (NTech Industries Inc., Ukiah, CA, USA). Raw data access is not guaranteed in all circumstances, and users are commonly bound to vendor-specific processing routines in order to retrieve and analyze the collected sensor measurements.

To overcome this lack of standardized procedures for sensor control and access, as well as for sensor data encoding and distribution, Nash *et al.* [14] suggest utilizing standards from the Open Geospatial Consortium's (OGC) initiatives to automate agricultural sensor data processing. The OGC Sensor Web Enablement (SWE) initiative bridges the gap between sensors and processing applications, providing a suite of standards "[...] to enable all types of Web and/or Internet-accessible sensors, instruments, and imaging devices to be accessible and, where applicable, controllable via the Web" [15]. It consists of several definitions of "sensor related data in a self-describing and semantically enabled way" [16]. SWE, therefore, can be utilized as the basis for a sensor web, an infrastructure that hides the underlying architecture, the network communication mechanisms and the heterogeneous sensor hardware from the applications built on top [17]. Although most realizations of a sensor web originate in other fields of research and for large-scale scenarios, e.g., oil spill disasters [18], flood management [19] or general risk management [20], recent studies proved the adaptability for the agricultural domain, operating in even smaller contexts [21].

The first implementations for stationary wireless sensor networks (WSNs) proved the potential of this idea for precision agriculture. Some researchers describe improved concepts for decision making processes in agriculture by connecting WSNs with web services as part of a spatial data infrastructure (SDI), building on the SWE specifications [22–24]. Other researchers developed applications, based on these web services, e.g., for online spraying operations, utilizing a web feature service (WFS) on-the-fly [25]. Having a magnitude of possibilities to combine stationary and mobile, ground-based and aerial, as well as temporary and permanent sensor systems, current sensor networks have become more and more complex. As a consequence, the connection of sensor systems and entire sensor networks with a sensor web needs to be as flexible as possible to facilitate the integration of sensor data into web services and applications.

This study provides a simple, but effective method to embed various sensor systems into a sensor web approach, making their data accessible for applications using well-defined and interoperable standards of the OGC SWE initiative framework. The idea for establishing this method originates from the various field experiments, which were conducted at the agricultural research stations of the University of Hohenheim, Stuttgart, Germany. Many of these experiments involve sensor measurements, but lack a general work flow with standardized mechanisms for the control and access of sensors, as well as the storage and processing of their data. The authors show how to utilize open source software, provided by the 52° North Initiative for Geospatial Open Source Software GmbH (52° N), and adapt it to the needs of PF. A field trial environment was set-up to verify the method in a real use-case scenario for the adoption of SWE for PF-sensing.

2. Materials and Methods

This section gives background information about the principles and the implementation of an actual agricultural sensor infrastructure. The focus was set to publish sensor data to a remotely-distributed SWE infrastructure and make it accessible for researchers and user applications in a well-defined way. The sensor infrastructure of this study was based on the recommendations of Bröring *et al.* [18], who described the implementation of an extended sensor infrastructure stack. The infrastructure stack is shown in Figure 1 and will be explained in the following.

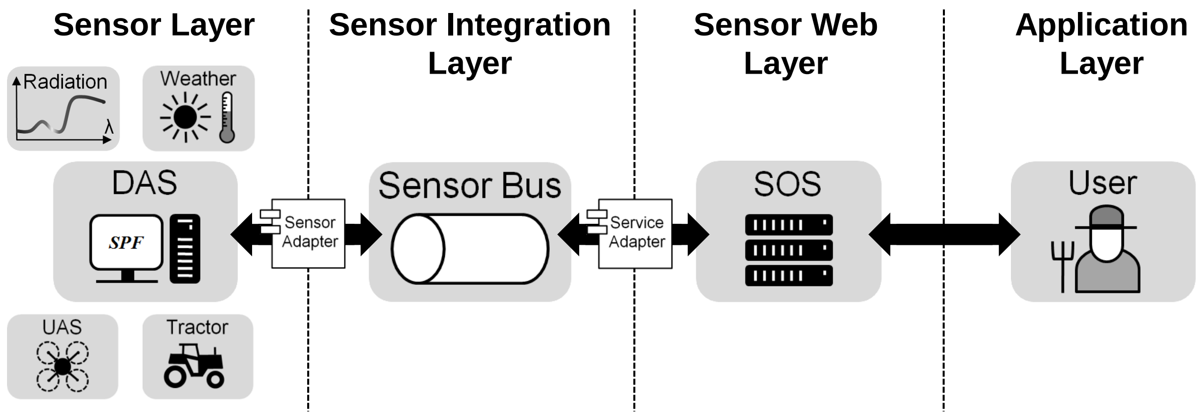


Figure 1. The extended sensor infrastructure stack as introduced by Bröring *et al.* [18]. It is based on three main layers for: (i) sensor control and communication (sensor layer); (ii) Sensor Web Enablement (SWE) services as part of a sensor web (sensor web layer); and (iii) end users and computers (application layer), which build applications on top of the SWE services. A fourth layer is an intermediary integration layer, facilitating the connection of sensors and services (sensor integration layer).

The extended sensor infrastructure stack is based on three main layers and one integration layer, covering all levels from sensor measurements to end-user applications. The sensor layer is the lowest level layer, managing the communication within sensor networks. It consists of the different sensor devices and one or several data acquisition systems (DAS), to control and access all sensor systems on-the-fly. The sensor integration layer is an intermediary layer between sensors and SWE services. Its idea is to establish an infrastructure that connects sensor web services, requesting specific sensor data, with sensors, delivering exactly the requested data, on-the-fly [26]. The sensor web layer consists of one or a multitude of SWE services. Each service is defined for special purposes, e.g., the sensor event service (SES), which offers a web interface to publish and subscribe to notifications from sensors [27], or the sensor observation service (SOS), which offers the discovery and retrieval of real-time or archived data, produced by any kind of sensor system [28]. The application layer is the highest level layer, where users or computer systems interact with the SWE services.

This study proposes an infrastructure that consists of a sensor layer, a sensor integration layer and a sensor web layer. An application layer was not part of this study. The following paragraphs give insight into the implementation of these layers.

2.1. Sensor Layer

The sensor layer represents the lowest level layer of the proposed infrastructure. It was set-up by four different sensor systems and a DAS, to control and access the sensor systems. Communication was enabled by a 2.4-GHz wireless local area network (WLAN) and a 3G mobile Internet connection.

2.1.1. Sensor Systems

The sensor layer involved: (i) a stationary HYT221 weather sensor (HYT221, IST AG, Wattwil, Switzerland) for measuring temperature and relative humidity; (ii) a stationary MMS1 NIR enhanced spectrometer (HandySpec Field, tec5 AG, Oberursel, Germany) for the registration of incident solar radiation; (iii) a tractor, equipped with a Multiplex fluorescence sensor (Multiplex, FORCE-A, Orsay, France) for the detection of within-field plant health; and (iv) Hexe, a prototype UAS, equipped with a PiCam RGB camera (Raspberry Pi Camera, Raspberry Pi Foundation, Caldecote, Cambridgeshire, UK), a self-assembled multi-spectral camera (D3, VRmagic Holding AG, Mannheim, Germany) and an MMS1 NIR enhanced spectrometer, for the detection of plants' spectral parameters [29]. The HandySpec sensor system was operated by a consumer notebook, which also served as the processing unit for the DAS. All other sensor systems were operated by individual Raspberry Pi Model B computers (Raspberry Pi Foundation, Caldecote, Cambridgeshire, UK), which were equipped with wireless adapters to enable communication with the DAS (see Figure 2).

All sensor systems were geo-referenced. The stationary sensor systems were placed at well-known locations, whereas the mobile platforms were equipped with a Global Navigation Satellite System (GNSS) to track their locations on-the-fly. The sensors were controlled by self-developed software routines, implementing vendor-specific application programming interfaces (APIs). The software routines were executed on the Raspberry Pi control units and the notebook.

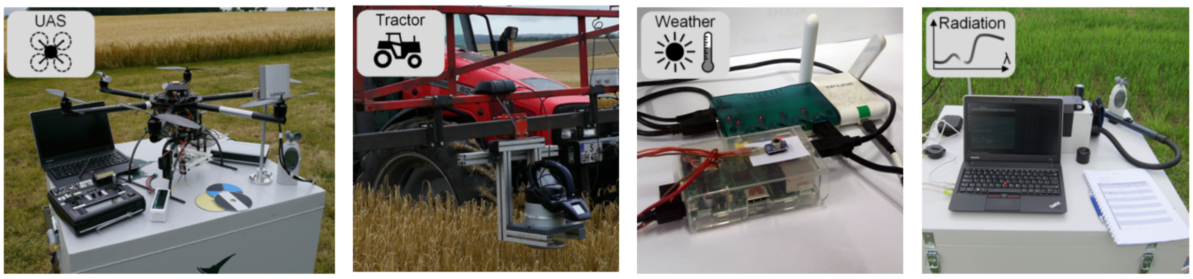


Figure 2. Overview of the sensor systems involved in the sensor layer. From left to right: Hexe (unmanned aircraft system (UAS)), Multiplex fluorescence sensor (tractor), HYT221 weather sensor (weather) and HandySpec Field spectrometer with base station (radiation).

2.1.2. Data Acquisition System

As DAS software, the authors chose the java-based and open source software framework "Sensor Platform Framework" (SPF, https://wiki.52north.org/bin/view/SensorWeb/SensorPlatformFramework). Its main purpose is to gather and, if needed, interpolate sensor data based on a periodic time interval or the availability of certain observations. Its generic architecture supports the inversion of control (IoC) design, offering extension points, which act as interfaces for input and output plugins [30].

Every connection of a sensor system with the DAS was realized by implementing an individual input-plugin and a plugin description document. As all sensor control units and the DAS share the same network, the input-plugins were configured: (i) to establish a network connection to the appropriate sensor control unit; (ii) to send configuration parameters; and (iii) to request sensor observations (see Figure 3).

The plugin description document describes the plugin's interpolation behavior, the sensor's observations and its meta data. The meta data were encoded in SensorML, a sensor description language, which is specified by SWE and used to describe sensors and processes [31]. Table 1 lists the most important parameters of each input plugin.

On the output plugins' side, three output mechanisms were of interest: a visual control of the geo-referenced sensor observations, a mechanism to forward the sensor observations into the sensor web and a simple data logger in case the DAS is disconnected from the sensor web. All of these mechanisms have already been established in three different output plugins, which can be downloaded from the 52°N website and are displayed in Figure 3. Visualization was done by the "SensorVis—Real Time Sensor Visualization" (https://wiki.52north.org/bin/view/SensorWeb/SensorVis) plugin, which allows live visualization of sensor data based on a 3D virtual globe environment [32]. Logging was realized using a slightly adapted version of the "File Writer Plugin", which is part of the standard SPF packages. As the forwarding mechanism, the "Sensor Bus Output Plugin", also distributed within the standard SPF packages, was used. It implements a sensor adapter for a logical bus for the standardized connection of sensor data and SWE services, which will be explained in the following paragraphs [18,26].

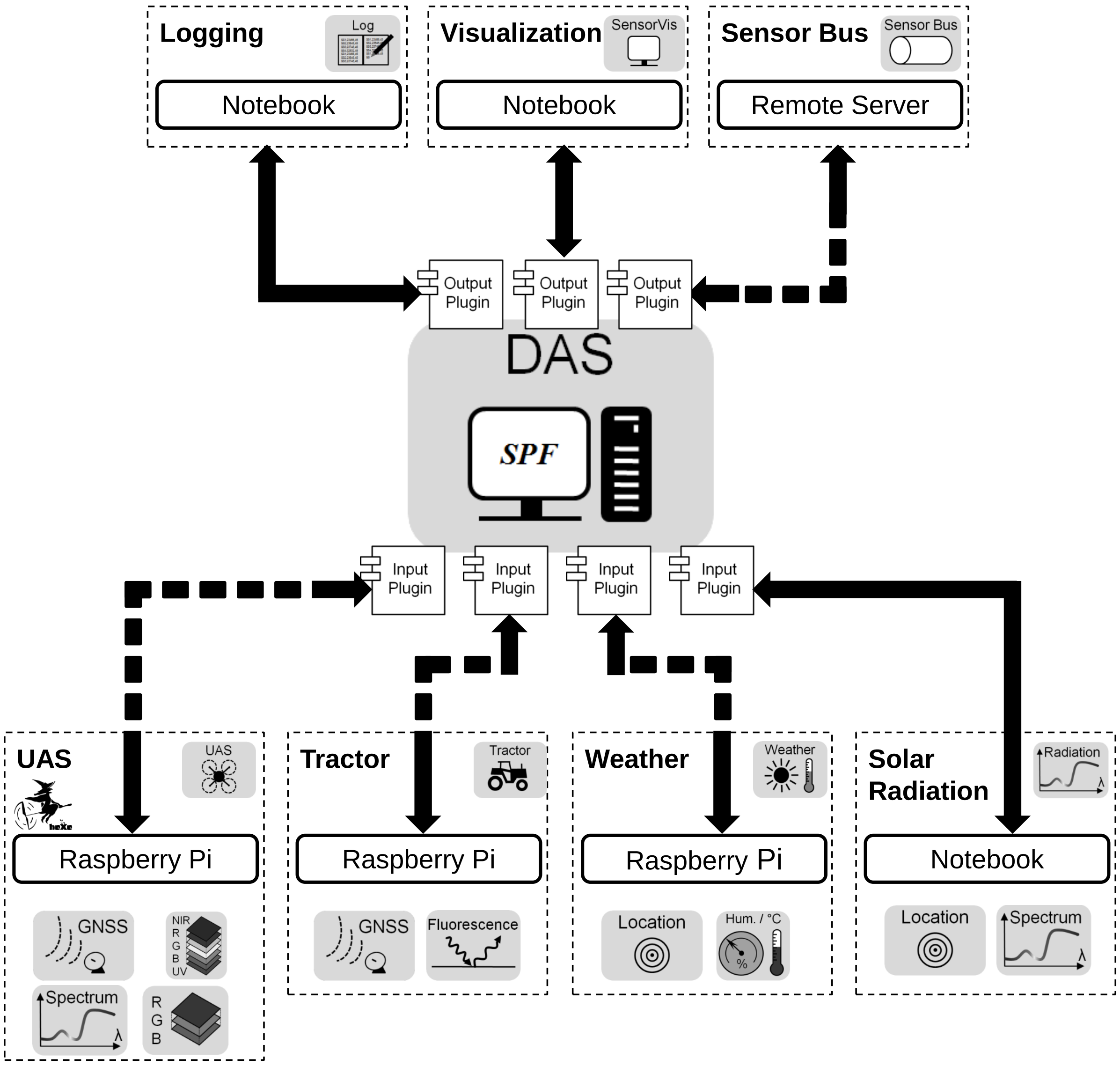


Figure 3. Overview of the input and output plugin architecture of the Sensor Platform Framework (SPF), which serves as the data acquisition system (DAS). Four input plugins were implemented to control and access all sensor systems individually. The Raspberry Pis and the notebook serve as control units, implementing vendor-specific sensor protocols. DAS and control units communicate with each other either through wireless (dashed lines) or wired connections (solid lines). Three output plugins were implemented for: (i) the live-visualization of sensor observations during measurement; (ii) for the local logging of received sensor data; and (iii) for the forwarding of the sensor data into the sensor bus. Visualization and logging were performed on the notebook, running the DAS. Forwarding data into the sensor bus was realized via a mobile Internet connection.

Table 1. Summary of the the sensor systems' observations, specified in the input plugins.

Sensor System	Sensors	Observations
Hexe	GNSS	Lon, Lat, Alt
	IMU	Nick, Roll, Yaw
	MMS1 NIR enhanced	256 reflection values
	PiCam RGB	Image identifier
	VRmagic Camera	Image identifier
Tractor	GNSS	Lon, Lat, Alt
	Multiplex	6 fluorescence indices
Weather	Preset location	Lon, Lat, Alt
	HYT221	Relative humidity
		Temperature
Solar Radiation	Preset location	Lon, Lat, Alt
	HandySpec	256 radiation values

2.2. Sensor Integration Layer

The authors chose the sensor bus to serve as the sensor integration layer in between sensor systems and remotely-connected sensor web services (see Figure 4). Although it is designed to enable a sensor plug and play infrastructure for a sensor web by incorporating semantic matchmaking functionality, a publish/subscribe mechanism and a generic driver mechanism [18], the available sensor bus output plugin is limited to messaging, based on the sensor bus protocol [26]. Therefore, matchmaking, publish/subscribe and driver issues were handled manually.

A driver mechanism to control and access the connected sensors was implemented for every SPF input plugin, individually. The sensor bus plugin was configured to publish all sensor data, gathered by the SPF, into an Extensible Messaging and Presence Protocol (XMPP) chat channel, which ran as ejabberd (https://www.ejabberd.im) software on an Internet-connected server at the University of Hohenheim (see Listing 1). The chat message format follows the sensor bus protocol specifications and offers a simple solution to distribute sensor data to a remote SWE service.

A sensor bus service adapter was implemented to forward the observations from the sensor bus to an SOS. It was realized as a python program. It subscribed and listened to the XMPP chat channel, which contained the published sensor data (see Listing 1). The service adapter was designed to parse the sensor data from the sensor bus protocol format to an SOS request Extensible Markup Language (XML) format. Related sensor observations were assembled and grouped following the predefined SensorML profiles. Subsequently, an *InsertObservation* request was composed to add the observations to the SOS [28]. The *InsertObservation* request is part of the transactional operations SOS profile. This optional transactional profile allows clients to register new sensors (*InsertSensor*) and add observations. Observations in the request are encoded in accordance with the Observations and Measurement (O&M) schema, a standard to describe all observations of a sensor system [33].

Listing 1: Exemplary listing of a sensor bus message, published by the HYT221 weather station. The sensor adapter broadcasts a message to register the sensor (*SensorRegistration*) and publishes all available sensor observations (*PublishData*), consequently.

```
(10:11:58) spf_user2: SensorRegistration>urn:sengis:id:HYT221>urn:sengis:id:HYT221 (stationary platform) connected via SPFramework>
urn:sengis:id:HYT221>
firstCoordinateName<latitude<secondCoordinateName<longitude<thirdCoordinateName<altitude>urn:ogc:def:crs:EPSG::4326>0.0>0.0>0.0>
humidity<%<altitude<m<longitude<deg<latitude<deg<temperature<Cel>SensorRegistration
(10:11:58) spf_user2: PublishData>urn:sengis:id:HYT221>2014−06−27T10:11:57.355+01:00>class java.lang.Double>36.2>humidity>
(10:11:58) spf_user2: PublishData>urn:sengis:id:HYT221>2014−06−27T10:11:57.355+01:00>class java.lang.Double>485.234>altitude>
(10:11:58) spf_user2: PublishData>urn:sengis:id:HYT221>2014−06−27T10:11:57.355+01:00>class java.lang.Double>8.9221>longitude>
(10:11:58) spf_user2: PublishData>urn:sengis:id:HYT221>2014−06−27T10:11:57.355+01:00>class java.lang.Double>48.7450>latitude>
(10:11:58) spf_user2: PublishData>urn:sengis:id:HYT221>2014−06−27T10:11:57.355+01:00>class java.lang.Double>18.54>temperature>
```

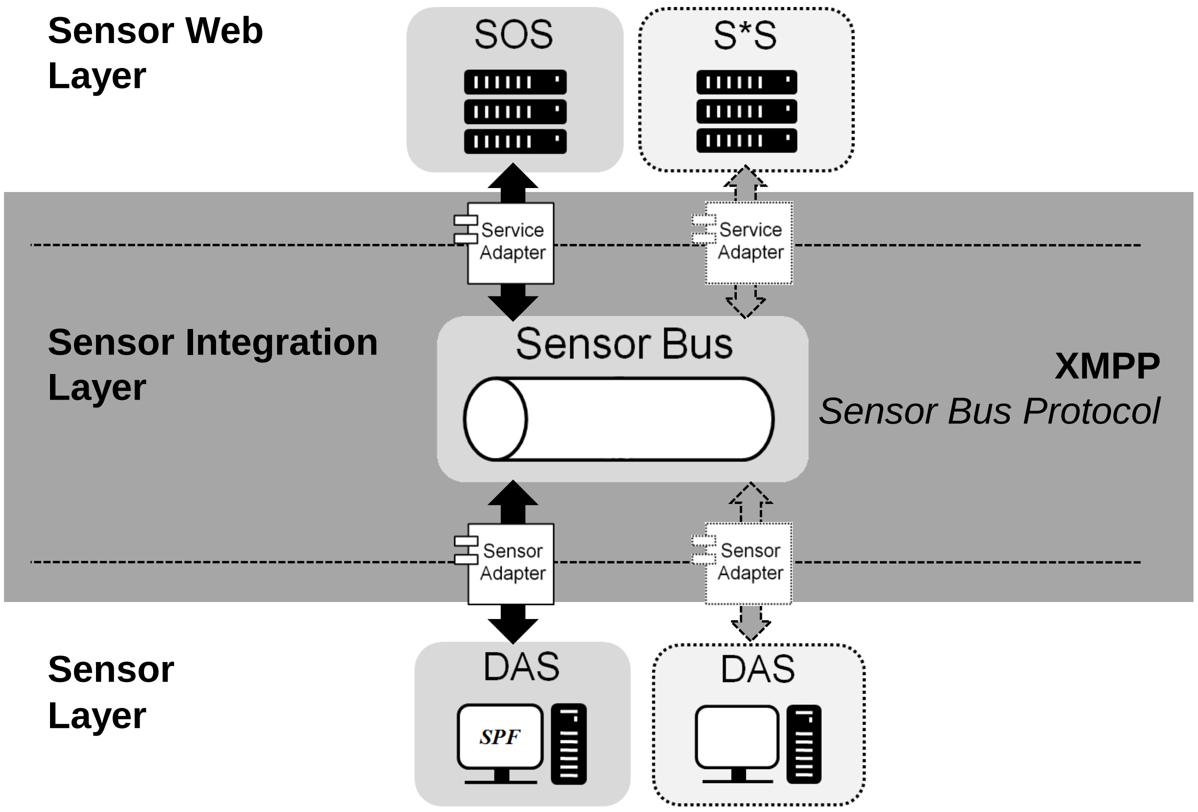


Figure 4. Overview of the sensor bus architecture, which is designed to facilitate the communication of sensor systems and SWE services. Any kind of sensor adapter can register to the bus and publish its sensor data according to the sensor bus message protocol. For subscription and receiving of sensor data, any kind of SWE services can register a service adapter, listening to the sensor bus. The architecture is scalable to scenarios where a multitude of sensor systems and SWE services participate.

2.3. Sensor Web Layer

The sensor web layer consists of an SOS. It is the most common SWE service and it was used in this study in its 52° N SOS 4.1 (https://wiki.52north.org/bin/view/SensorWeb/SensorObservationServiceIV) implementation, exclusively. It was set-up on a server, running at the University of Hohenheim. It offers a web interface for publishing operations, e.g., *GetCapabilities*, *GetObservation* and *DescribeSensor*, on the one hand, and for transactional operations, e.g., *InsertSensor* and *InsertObservation*, on the other hand. It builds on the technical frameworks of an Apache Tomcat 7 (http://tomcat.apache.org/tomcat-7.0-doc) servlet container, a PostgreSQL 9.3 (http://www.postgresql.org/docs/9.3) Database Management System (DBMS) and a PostGIS 2.1 (http://postgis.net/2013/08/17/postgis-2-1-0) support for geographic objects.

Based on the SensorML descriptions of every input plugin, each sensor system was registered once using the *InsertSensor* operation. After having registered the individual sensors, the sensor bus service adapter was able to perform *InsertObservation* operations on-the-fly, using the Service-Oriented Architecture Protocol (SOAP).

2.4. Field Trial

A typical PF field experiment served as test-bed for the proposed infrastructure. The field trial was conducted on 27 June 2014 and in clear skies in a field of winter-wheat (*Triticum aestivum* L.), located at Ihinger Hof (48.74°N, 8.92°E), a research station of the University of Hohenheim. The trial's aim was the acquisition and storage of sensor observations: (i) locally, on a notebook, running the DAS; and (ii) remotely, on an Internet-connected SOS.

The sensor systems were mounted on ground, on a tractor and on a UAS. The tractor and the UAS were configured to follow a predefined route in the field, whereas the weather station and the solar radiation sensor were set-up at fixed locations at the field's border. The consumer notebook, running the DAS, was set-up at the solar radiation sensor's location, together with a 2.4-GHz WLAN access point and a 3G mobile Internet connection, realized by mobile phone tethering. All sensor systems were operated simultaneously with a sampling interval of 1 Hz during a measurement period of approximately 6 min. Observation pull-requests were performed at the same rate via the 2.4-GHz WLAN connection. A maximum distance of 180 m in between the sensor system and notebook was reached by the UAS. The UAS covered a total area of 180×36 m.

Visualization and logging of the received observations took place on the notebook. Moreover, broadcasting was performed by the sensor bus plugin via the mobile Internet connection. The sensor bus messaging infrastructure was implemented as an ejabberd XMPP service on an Internet-connected server at the University of Hohenheim. In addition, this server hosted the SOS, as well as the sensor bus service adapter, which was listening to incoming messages of the XMPP chat channel. Figure 5 gives an overview of the complete infrastructure with a UAS observation example.

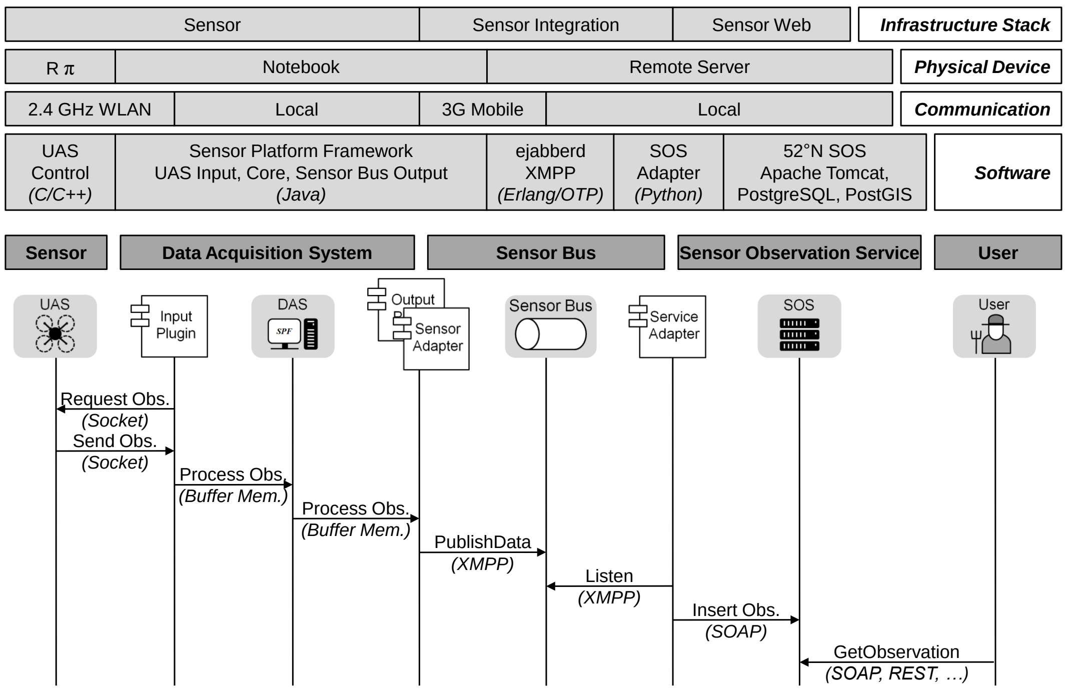


Figure 5. Sequence diagram of the processing of an exemplary UAS observation from acquisition to storage on an Internet-connected SOS (lower half). The upper half gives information about the realization of the different components of the infrastructure.

3. Results and Discussion

The infrastructure proved its ability to control all sensors, to access and forward their data and to store them in a well-defined, standardized SOS. The field trial showed that this sensor infrastructure is applicable to PF scenarios, although some hurdles still exist.

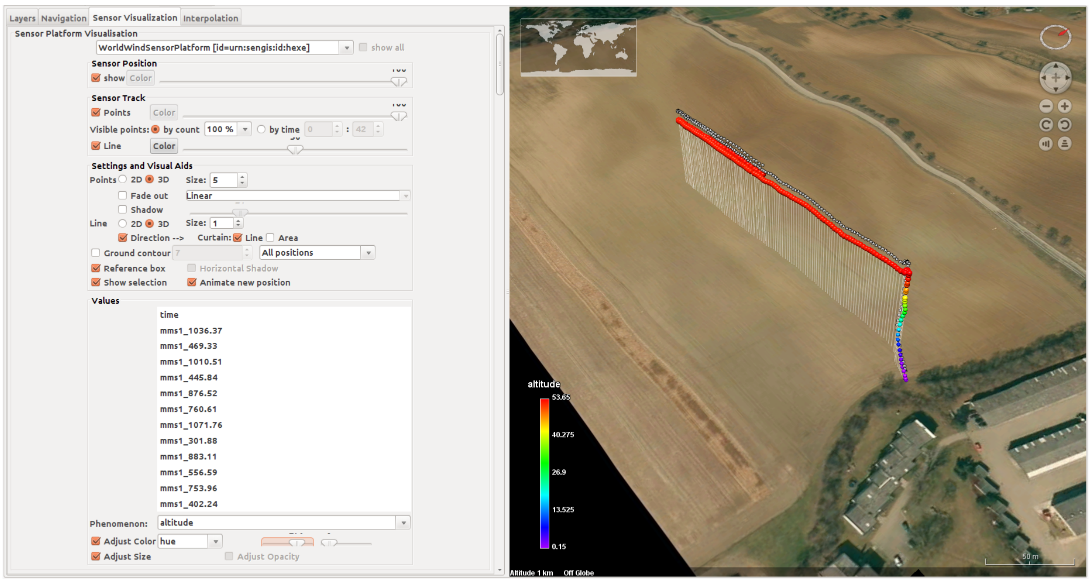
3.1. Sensor Layer

Despite having two connection losses of approximately 10 s due to instabilities of the WLAN, the sensor layer behaved as expected. Under stable network conditions, all sensor systems could be controlled flawlessly. Their data could be accessed by the DAS and forwarded to the sensor integration layer. The mobile Internet connection was stable throughout the whole test.

Intensive work had to be invested in the programming of the control unit software of all sensor devices. The software was designed to keep the sensors remotely controllable and accessible via network socket communication. Every software implementation had to cope with sensor-specific drivers and protocols. Although most sensor vendors offer APIs for software developers, some sensor protocols still have to be implemented by one's self, e.g., the Spectral Device Control and Transfer Protocol (SDCTP) for network control of the MMS1 NIR enhanced spectrometer. A generic driver mechanism, e.g., the sensor interface descriptor (SID) model, could overcome this intensive labor [34].

The SPF, which was used as DAS, served its purpose to integrate all sensor systems. Nevertheless, implementing correct input plugins and plugin descriptions had to be done carefully. Each input plugin was programmed to connect to a specific network socket to communicate with its according sensor control unit. Sensor data access was implemented with 1-Hz pull requests, which worked reliably, apart from two times of network instability. For configurable sensors, sensor control was realized via a graphical user interface (GUI). Sensor descriptions were realized in a standardized way with SensorML, defining the sensors' characteristics as part of a plugin description document. Moreover, the description document was used to specify the input plugins' interpolation behavior, as well as the input and output of observations. The output plugins worked as expected. Once registered for use, the visualization plugin was able to display all observations from every sensor on-the-fly (see Figure 6). The logging plugin logged all incoming observations to a .csv file. The size of the .csv file summed up to 1.3 MB during 6 min of measurement. The sensor bus output plugin worked flawlessly. It parsed the incoming observations to the sensor bus protocol format and forwarded the data into the XMPP chat channel.

The sensor layer implementation proved its practicability. A stable network and Internet connection is essential for this architecture. Despite potentially missing some of the sensed data due to unpolled pull mechanisms, instabilities may be also critical for near real-time applications in scenarios where data acquisition, data processing and application are performed online.


Layers | Navigation | Sensor Visualization | Interpolation

Sensor Platform Visualisation
WorldWindSensorPlatform [id=urn:sengis:id:hexe] ☐ show all

Sensor Position
☑ show Color

Sensor Track
☑ Points Color
Visible points: [✓ by count] 100 % ○ by time 0 : 42
☑ Line Color

Settings and Visual Aids
Points ○ 2D [✓ 3D] Size: 5
☐ Fade out Linear
☐ Shadow
Line ○ 2D [✓ 3D] Size: 1
☑ Direction --> Curtain: ☑ Line ☐ Area
☐ Ground contour 7 All positions
☑ Reference box ☐ Horizontal Shadow
☑ Show selection ☑ Animate new position

Values
time
mms1_1036.37
mms1_469.33
mms1_1010.51
mms1_445.84
mms1_876.52
mms1_760.61
mms1_1071.76
mms1_301.88
mms1_883.11
mms1_556.59
mms1_753.96
mms1_402.24

Phenomenon: altitude
☑ Adjust Color hue
☑ Adjust Size ☐ Adjust Opacity

Figure 6. Example of the SPF "SensorVis" output plugin [32] live-visualization of Hexe, a UAS sensor system, operating during the field trial. On the left side, visualization parameters can be selected and configured, depending on the available sensor observations. On the right side, the flight path and the selected sensor observation values are visualized by colored spheres, *i.e.*, indicating the received flight altitude information.

3.2. Sensor Integration Layer

The sensor integration layer was restricted to the sensor bus messaging mechanism, due to the limited functionality of the sensor bus output plugin. It was able to connect to the chat channel and broadcast all sensor data, collected by the DAS. Instead of broadcasting complete raster datasets, e.g., images, the captured raster data description was restricted to short image identifiers. As a consequence, all sensor datasets could be transmitted through the wireless Internet connection. The data transfer to the XMPP service was not encrypted. Generally, transfer encryption is desirable and available (transport layer security, TLS). If the channel communication should be kept private, it can be restricted to certain users and password authentication.

As this study utilizes only one sensor adapter and one service adapter, the sensor bus architecture is not exploited in all of its possibilities. Nevertheless, the introduced infrastructure offers the scalability of the sensor bus concept. It can be adapted to a multitude of sensor adapters and service adapters, e.g., for multiple SOS and SES, located at different institutions. Moreover, as it is a logical concept, messaging is not restricted to XMPP and can be replaced or extended by other communication protocols, e.g., Twitter and Internet Relay Chat (IRC) [18]. To enable sensor plug and play, mediating, publish/subscribe and driver mechanisms still have to be implemented.

3.3. Sensor Web Layer

The sensor web layer performed well. The Apache Tomcat server, as well as the PostgreSQL/PostGIS DBMS were installed smoothly, following the documented standard installation routines. The SOS package was delivered as a self-extracting file for the servlet container. The installation worked as expected. All needed databases were created automatically after SOS configuration. The SOS supported all operations of the implemented SOS service adapter. Here, *InsertSensor* and *InsertObservation* were used.

4. Conclusions

This work proved the applicability of the OGC SWE initiative framework definitions for the set-up of a sensor data infrastructure for PF applications. The proposed infrastructure guarantees a standardized collection and storage of spatio-temporal agricultural sensor data, accessible by SWE services and user applications. It is based on open source software, offering the possibility to deploy numerous sensor systems and SWE services. The DAS provides a consistent method for the control, access and forwarding of sensor observations. The sensor bus concept is scalable to more complex scenarios involving a multitude of sensor systems, DAS and SWE services. The implemented SOS is a first step towards a service-oriented architecture, based on further web services and OGC standards, offering functionalities of a holistic SDI for PF. In an SDI, web clients act as interfaces in between stored sensor data and a user, realizing the application layer of the infrastructure stack. It can be applied to machinery and sensor systems on the farm scale or be extended with data services offered by external parties. Moreover, as observations acquired by mobile or stationary systems share the same infrastructure, the applications and work flows built on top of it can themselves be built for mobile or stationary devices. Future research will be concentrated on establishing such an SDI for standardized sensor data distribution, processing and analysis in the PF domain.

Acknowledgments

The authors acknowledge the Carl-Zeiss-Foundation (Carl-Zeiss-Stiftung) for funding this work as part of the collaborative project SenGIS, at the University of Hohenheim, Stuttgart, Germany. Moreover, the authors acknowledge Arne Bröring, Matthes Rieke, Daniel Nüst, Christian Malewski and Jan Wirwahn for their advice and support in the implementation of this infrastructure.

Author Contributions

Jakob Geipel and Martin Weis proposed the idea for this work. Jakob Geipel wrote the manuscript, and programmed the control units and the input plugins of the sensor layer. Markus Jackenkroll set up the server, the DBMS and the SOS of the sensor web layer. Martin Weis programmed the sensor bus service adapter of the sensor integration layer. Jakob Geipel, Markus Jackenkroll, and Martin Weis performed the field work. Wilhelm Claupein helped with editorial contributions.

Conflicts of Interest

The authors declare no conflicts of interest.

References

1. Oerke, E.C.; Gerhards, R.; Menz, G.; Sikora, R.A.; Eds. *Precision Crop Protection—The Challenge and Use of Heterogeneity*, 1st ed.; Springer Verlag: Dordrecht, The Neatherlands, 2010.
2. Heege, H.J.; Eds. *Precision in Crop Farming: Site Specific Concepts and Sensing Methods: Applications and Results*, 1st ed.; Springer Science & Business Media: Dordrecht, The Neatherlands, 2013.
3. Peteinatos, G.G.; Weis, M.; Andújar, D.; Rueda Ayala, V.; Gerhards, R. Potential use of ground-based sensor technologies for weed detection. *Pest Manag. Sci.* **2014**, *70*, 190–199.
4. Phillips, A.J.; Newlands, N.K.; Liang, S.H.; Ellert, B.H. Integrated sensing of soil moisture at the field-scale: Measuring, modeling and sharing for improved agricultural decision support. *Comput. Electron. Agric.* **2014**, *107*, 73–88.
5. Li, Z.; Wang, N.; Franzen, A.; Taher, P.; Godsey, C.; Zhang, H.; Li, X. Practical deployment of an in-field soil property wireless sensor network. *Comput. Stand. Interfaces* **2014**, *36*, 278–287.
6. Peteinatos, G.G.; Geiser, M.; Kunz, C.; Gerhards, R. Multisensor approach to identify combined stress symptoms on spring wheat. In Proceedings of the 2nd International Conference on Robotics and Associated High-Technologies and Equipment for Agriculture and Forestry, Madrid, Spain, 21–23 May 2014; Gonzalez-de-Santos, P., Ribeiro, A., Eds.; pp. 131–140.
7. Martinon, V.; Fadailli, E.M.; Evain, S.; Zecha, C. Multiplex: An innovative optical sensor for diagnosis, mapping and management of nitrogen on wheat. In *Precision Agriculture 2011*, Proceedings of the ECPA, Prague, Czech Republic, 11–14 July 2011; Stafford, J., Ed.; Czech Centre for Science and Society: Prague, Czech Republic, 2011; pp. 547–561.
8. Tremblay, N.; Wang, Z.; Ma, B.L.; Belec, C.; Vigneault, P. A comparison of crop data measured by two commercial sensors for variable-rate nitrogen application. *Precis. Agric.* **2009**, *10*, 145–161.

9. Andújar, D.; Weis, M.; Gerhards, R. An ultrasonic system for weed detection in cereal crops. *Sensors* **2012**, *12*, 17343–17357.
10. Escolá, A.; Andújar, D.; Dorado, J.; Fernández-Quintanilla, C.; Rosell-Polo, J.R. Weed detection and discrimination in maize fields using ultrasonic and lidar sensors. In Proceedings of the International Conference of Agricultural Engineerig CIGR, Valencia, Spain, 8–12 July 2012.
11. Thenkabail, P.S.; Lyon, J.G.; Huete, A. Advances in hyperspectral remote sensing of vegetation and agricultural croplands. In *Hyperspectral Remote Sensing of Vegetation*, 1st ed.; Thenkabail, P.S., Lyon, J.G., Huete, A., Eds.; CRC Press Inc.: Boca Raton, FL, USA, 2012; pp. 4–35.
12. Berni, J.; Zarco-Tejada, P.; Suarez, L.; Fereres, E. Thermal and narrowband multispectral remote sensing for vegetation monitoring from an Unmanned Aerial Vehicle. *IEEE Trans. Geosci. Remote Sens.* **2009**, *47*, 722–738.
13. Geipel, J.; Link, J.; Claupein, W. Combined spectral and spatial modeling of corn yield based on aerial images and crop surface models acquired with an unmanned aircraft system. *Remote Sens.* **2014**, *6*, 10335–10355.
14. Nash, E.; Korduan, P.; Bill, R. Applications of open geospatial web services in precision agriculture: A review. *Precis. Agric.* **2009**, *10*, 546–560.
15. Botts, M.; Percivall, G.; Reed, C.; Davidson, J. OGC sensor web enablement: Overview and high level architecture. In *GeoSensor Networks*; Nittel, S., Labrinidis, A., Stefanidis, A., Eds.; Springer: Berlin/Heidelberg, Germany, 2008; Volume 4540, pp. 175–190.
16. Botts, M.; Percivall, G.; Reed, C.; Davidson, J. *OGC Sensor Web Enablement: Overview and High Level Architecture (White Paper) (OGC 06-050r2)*; OGC Implementation Specification: Wayland, MA, USA, 2013.
17. Bröring, A.; Echterhoff, J.; Jirka, S.; Simonis, I.; Everding, T.; Stasch, C.; Liang, S.; Lemmens, R. New generation sensor web enablement. *Sensors* **2011**, *11*, 2652–2699.
18. Bröring, A.; Maué, P.; Janowicz, K.; Nüst, D.; Malewski, C. Semantically-enabled sensor plug & play for the sensor web. *Sensors* **2011**, *11*, 7568–7605.
19. Bröring, A.; Beltrami, P.; Lemmens, R.; Jirka, S. Automated integration of geosensors with the sensor web to facilitate flood management. In *Approaches to Managing Disaster—Assessing Hazards, Emergencies and Disaster Impacts*; Tiefenbacher, J., Ed.; InTech: Rijeka, Croatia, 2012; pp. 65–86.
20. Klopfer, M.; Ioannis, K.; Eds. *Orchestra—An Open Service Architecture for Risk Management*; ORCHESTRA Consortium, 2008. Available online: http://www.eu-orchestra.org/docs/ORCHESTRA-Book.pdf (accessed on 13 November 2014).
21. Wiebensohn, J.; Jackenkroll, M. Evaluation and modelling of a standard based spatial data infrastructure for precision farming. In Proceedings of the EFITA-WCCA-CIGR Conference, Turino, Italy, 24–27 June 2013; p. C0107.
22. Polojärvi, K.; Koistinen, M.; Luimula, M.; Verronen, P.; Pahkasalo, M.; Tervonen, J. Distributed system architectures, standardization, and web-service solutions in precision agriculture. In Proceedings of the 4th International Conference on Advanced Geographic Information Systems, Applications and Services, Valencia, Spain, 30 January–4 February 2012; pp. 171–176.

23. Sawant, S.; Adinarayana, J.; Durbha, S.; Tripathy, A.; Sudharsan, D. Service oriented architecture for wireless sensor networks in agriculture. In *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, Proceedings of the ISPRS Congress, Melbourne, Australia, 25 August–1 September 2012; pp. 467–472.
24. Kubicek, P.; Kozel, J.; Stampach, R.; Lukas, V. Prototyping the visualization of geographic and sensor data for agriculture. *Comput. Electron. Agric.* **2013**, *97*, 83–91.
25. Kaivosoja, J.; Jackenkroll, M.; Linkolehto, R.; Weis, M.; Gerhards, R. Automatic control of farming operations based on spatial web services. *Comput. Electron. Agric.* **2014**, *100*, 110–115.
26. Bröring, A.; Foerster, T.; Jirka, S.; Priess, C. Sensor bus: An intermediary layer for linking geosensors and the sensor web. In *COM.Geo '10*, Proceedings of the 1st International Conference and Exhibition on Computing for Geospatial Research & Application, Bethesda, MD, USA, 21–23 June 2010; ACM: New York, NY, USA; pp. 12:1–12:8.
27. Echterhoff, J.; Everding, T. *OpenGIS Sensor Event Service Interface Specification (Discussion Paper) (OGC 08-133)*; OGC Implementation Specification: Wayland, MA, USA, 2008.
28. Na, A.; Priest, M. *Sensor Observation Service (OGC 06-009r6)*; OGC Implementation Specification: Wayland, MA, USA, 2007.
29. Geipel, J.; Peteinatos, G.G.; Claupein, W.; Gerhards, R. Enhancement of micro Unmanned Aerial Vehicles to agricultural aerial sensor systems. In *Precision Agriculture '13*, Proceedings of the ECPA, Lleida, Spain, 7–11 July 2013; Stafford, J., Ed.; Wageningen Academic Publishers: Wageningen, The Neatherlands, 2013; pp. 161–167.
30. Rieke, M.; Foerster, T.; Bröring, A. Unmanned Aerial Vehicles as mobile multi-sensor platforms. In Proceedings of the 14th AGILE International Conference on Geographic Information Science, Utrecht, The Neatherlands, 18–21 April 2011.
31. Botts, M.; Robin, A. *OpenGIS Sensor Model Language (SensorML) (OGC 07-000)*; OGC Implementation Specification: Wayland, MA, USA, 2007.
32. Nüst, D. Visualising interpolations of mobile sensor observations. In Proceedings of the GeoViz, Hamburg, Germany, 5–8 March 2013.
33. Cox, S. *Observation and Measurements—XML Implementation (OGC 10-025rl)*; OGC Implementation Specification: Wayland, MA, USA, 2011.
34. Bröring, A.; Below, S.; Foerster, T. Declarative Sensor Interface Descriptors for the Sensor Web. In Proceedings of the WebMGS 2010: 1st International Workshop on Pervasive Web Mapping, Geoprocessing and Services, Como, Italy, 26–27 August 2010.