Architecture of a Security and Surveillance System

The benefits of an open and generic approach

Florian Segor, Axel Bürkle, Sven Müller, Rainer Schönbein, Matthias Kollmann
IAS – Interoperabilität und Assistenzsysteme
Fraunhofer IOSB
Karlsruhe, Germany
{florian.segor, axel.buerkle, sven.mueller, rainer.schoenbein, matthias.kollmann}@iosb.fraunhofer.de

***Abstract*—During the recent years, it has been increasingly shown that open and generic system platforms exhibit considerable advantages over closed systems in different areas of security technology. With systems that are flexible and adaptable to emerging demands, it can avoid expensive dedicated solutions becoming useless when requirements change. In this paper, a universal infrastructure is presented. It is shown exemplarily how easy and cost-effective new demands can be met by integrating new software and hardware components. The described architecture is designed to achieve maximum reusability.**

Keywords - generic; system architecture; security system; control station

I. INTRODUCTION

In order to be adaptable to a wide range of different requirements and applications, the complex surveillance system AMFIS [1] (**A**ufklärung mit **M**iniatur**f**luggeräten **i**m **S**ensorverbund) presented in this paper was developed as a mobile and generic system, which delivers an extensive situation picture in complex surroundings - even with the lack of stationary security technology. In order to achieve maximum flexibility, the system is implemented open and mostly generalized so that different stationary and mobile sensors and sensor platforms can be integrated with minimal effort, establishing interoperability with existing and future assets. The system is modular and can be scaled arbitrarily or be tailored by choosing the modules suitable to the specific requirements.

After a short introduction into the AMFIS framework an overview of the application scenarios is presented, followed by a detailed description of the AMFIS architecture in section IV. Section V introduces the integration of a new sensor. The paper closes with conclusions and future work.

II. THE AMFIS FRAMEWORK

The AMIFS system consists of a universal ground control station and a customizable set of sensors and sensor carriers (see Fig. 1). In addition, there are interfaces to external exploitation stations and control centers.

The ground control station is an adaptable prototype system for managing data acquisition with various sensors, mobile ad hoc networks and mobile sensor platforms. The main tasks of the ground control station are to work as an ergonomic user interface and as a data integration hub between multiple sensors and a super-ordinated control center. The sensors can be stationary or mounted on moving platforms such as micro UAVs [2], unmanned ground vehicles (UGVs) or underwater vehicles. The system includes means to control different kinds of mobile platforms and to direct them to potentially interesting locations especially in areas with no prior sensor equipment. The actual AMFIS system is highly mobile and operational at any location with relative ease. The sensor carriers of this multi-sensor system can be combined in a number of different configurations to meet a variety of specific requirements. The functions of the ground control station include: task

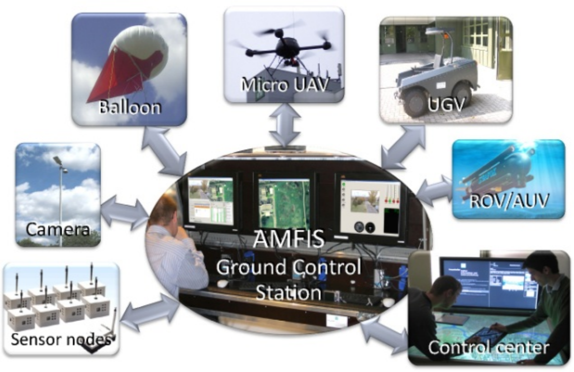

Fig. 1 Modular Ground Control Station AMFIS.

management, mission planning, control of sensors and mobile platforms, situation awareness, fusion and exploitation of sensor data, reporting, generation of alarms and archiving.

III. APPLICATION SCENARIOS

The sense of security of our society has significantly changed during the past years. Besides the risks arising from natural disasters there are dangers in connection with criminal or terroristic activities, traffic accidents or accidents in industrial environments. Especially in the civil domain in the event of big incidents, there is a need for a better data basis to support the rescue forces in decision making. The search for buried people after building collapses or the

 ISBN: 978-1-61208-184-7

clarification of fires at big factories or chemical plants are possible scenarios addressed by the AMFIS system.

Many of these events have very similar characteristics. They cannot be foreseen in their temporal and local occurrence so that situational in situ security or supervision systems are not present. The data basis, on which decisions can be made, is rather slim and therefore the present situation is very unclear to the rescue forces at the beginning of a mission. Precisely in these situations, it is extremely important to understand the context as fast as possible in order to initiate the suitable measures specifically and efficiently.

Applications of the AMFIS system include support of fire-fighting work with a conflagration, clarification of the debris and the surroundings after building collapses and search for buried or injured people. Additionally, the system can be used to support the documentation and perpetuation of evidence during the cleaning out of the scene at regular intervals.

```
<message key=”string” type=”string”>
  <parentmessage key=”string” />
  <timestamp>YYYY-MM-DD hh:mm:ss</timestamp>
  <originator type=”SENSOR|SENSORNODE|USER|SYSTEM”>
    <sensornodeid>bigint</sensornodeid>
    <sensorid>bigint</sensorid>
    <userid>bigint</userid>
  </originator>
  <subject
type=”SENSORNETWORK|SENSORNODE|SENSOR|SYSTEM”>
    <sensornetwork>bigint</sensornetwork>
    <sensornodeid>bigint</sensornodeid>
    <sensorid>bigint</sensorid>
  </subject>
  <value>value</value>
</message>
```

Fig. 2 XML Example of an AMFIS Message.

IV. AMFIS Architecture

In order to create an open and generic system, one of the most important objectives was to design a sustainable architecture. One of the central demands for the system is the flexibility regarding new hardware, software or sensor components. The tasks, for which the AMFIS system is developed are varied and the technological development during the last years in related fields, e.g., mini and micro UAVs are enormous. Therefore it is obvious that the AMFIS architecture must be able to master new demands resulting from future assets.

To achieve this flexibility, the architecture was designed to not only be adaptable to components unknown today but also to be achieved with low expenditure.

The AMFIS ground control station's software architecture is basically 3-tiered following a pattern similar to the MVC (Model-view-controller) paradigm best known from web application development.

The central application is the so-called AMFIS Connector (see Fig. 3), which is a message broker responsible for relaying metadata streams within the network.

To be able to manage a vast amount of sensors and sensor carriers, the physical sensors and sensor carriers are logically mapped on the so-called sensorweb, a tree structure which contains virtual representatives of the actual units. The root node (sensorweb) is connected with a row of sensor networks, for example a set of PTZ cameras (Pan-Tilt-Zoom cameras). Each of these sensor networks consists of one or several sensor nodes, which correspond in each case to a physical sensor (for example a single PTZ camera). The sensor nodes themselves may again contain different sensors, for example a camera contains a compass.

The sensorweb is stored permanently in a database, from which an XML document is generated at runtime. This is also done by the AMFIS Connector, which can be seen as the central information service of the ground control station.

The communication protocol within AMFIS is based on XML strings (see Fig. 2), which are sent to TCP-Sockets. To simplify the use of this protocol and to provide different possibilities for software development, different implementations for different runtime environments (e.g. .NET, Java) are available, which enclose the XML management and allow an object oriented view of the messages to the user.

The implementation of the communication protocol is multicast-oriented; every incoming message is passed on by the Connector to all connected client applications. Each application decides itself if and how these messages are processed.

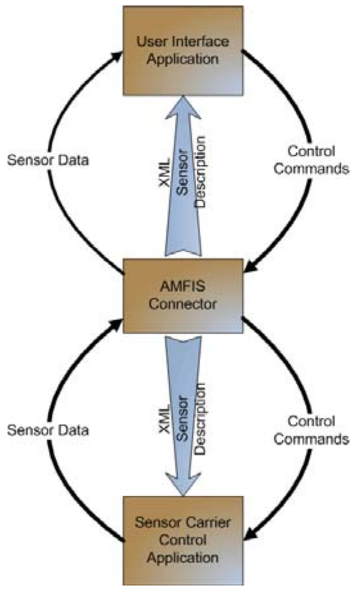

Fig. 3 Architecture of the AMFIS Framework.

If a client application connects to the AMFIS Connector, it first receives the XML structure of the sensorweb followed by a steady stream of XML messages. Each of these messages contains metadata (e.g., sensor status data or control information), which originates from one of the sensors in the sensorweb or is meant to change the status of one or more sensor nodes. After successful establishment of a connection, the Connector supplies the client application with a constant stream of live sensor data.

A client application in this context is any application that either includes one of the numerous AmfisCom implementations (.NET, Qt, Java) or implements the AMFIS message protocol directly.

The communication library (AmfisCom) builds the object tree from the XML data, providing the application developer with a type-safe and object-oriented view of the network of sensors and sensor carriers.

All applications within the AMFIS system, from graphic user interfaces to background system services, are designed as client applications:

- The various GUI applications of the user interface, most importantly the analyst's interface, the situation overview, the Photo Flight [3] or the pilot's interface. Those applications offer a visual representation of received metadata to the user, for example by displaying the current geographical locations of the various sensor carriers in the map and transmit commands to the sensor carriers, e.g., a user-generated waypoint for a UAV.
- A number of services running in the background, notably the video server, offering time shifting and archiving for both video and metadata (more on video management within AMFIS see below) and the rule engine respectively the multi-agent system, both supporting the user by automating certain processes.
- Drivers for various sensor carriers, e.g., a dedicated control software for UAVs, which translates high-level flight commands like waypoints into the proprietary RS232-based control protocol of the respective drone and in turn generates metadata XML status messages containing the current position, heading, remaining flight time etc.
- Interfaces to third-party applications or networks, e.g., command and control centers.

While all metadata is sent as XML messages irrespective of their type (whether they are sensor measuring values, steering information or user generated announcements) the large amount of generated video data must be processed and stored differently.

To do so, the connector is tightly coupled with a server application called the video server. It is responsible for storing and distributing video streams, serving the dual purpose of providing time shifting capabilities to the network as well as reducing the load on the usually wireless links between sensor carriers and the ground control station. Since time shifting or archiving is not always required, this functionality was not integrated in the Connector itself in order to keep it as light-weight as possible. To store the generated mission data permanently, the video server is connected to a database, from which the data streams can be restored for playback.

In order to be able to transmit reconnaissance results to external systems, the stored video data or the live video streams can be accessed externally. The main disadvantage of this method is the lack of metadata generated in the AMFIS system along with the video stream. For reconnaissance tasks, additional data such as location, time, sensor carrier or sensor type are often vital. Hence, the video server offers the possibility to convert the video data into standardized video formats, which contain these metadata. This is done by the AMFIS Transcoder Process, which encodes the metadata and the video data into a STANAG 4609 compliant data stream. That way, not only the imagery but also corresponding additional informations are available to systems, which can handle this standardized data format. The video streams generated by the AMFIS Transcoder can be stored as video clips in a CSD (coalition shared database) [4] or be transferred in real-time to an exploitation system such as ABUL [5].

To be able to receive messages or data information requests from an external system on top of the possibility to publish information, a communication module was implemented, which is called the XMPP Client. The XMPP Client translates reconnaissance requests (e.g. a Region-Of-Interest) placed by external systems into the AMFIS message format.

V. SENSOR INTEGRATION

With the software architecture described in section IV, it is possible to integrate new sensors and sensor carriers as well as completely new technologies without changes in the system's basic structure.

The integration of a new technology of this structure is described exemplarily in the following sections. It explains the use of a smartphone as a new sensor in the AMFIS system.

The mobile device is integrated as a client and allows the user to access data of the AMFIS system. Additionally, it provides tools to acquire and generate sensor data and to feed the accumulated data into the overall system. Therefore, the person carrying the mobile device becomes a mobile sensor in AMFIS.

The functional structure of the new sensor is basically divided into three modules. Besides the sensor itself, it

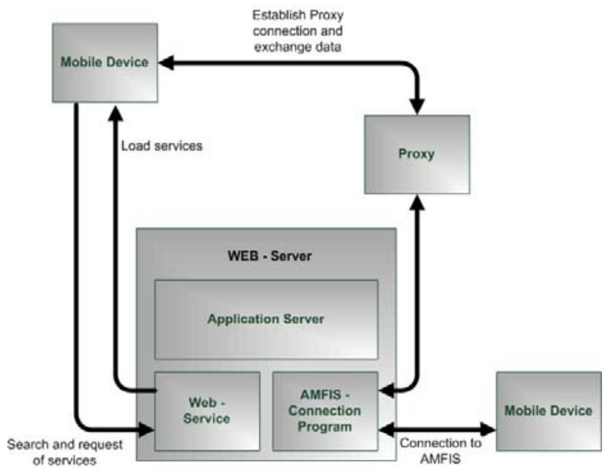

Fig. 4 Subsystem Architecture of the Smartphone.

affects the web server, which is responsible for supplying services and the application server, which structures the communication between the sensor and the system.

Providing the services by using a web server allows an easy addition of new functions to the sensor. By the separation between processing of data and the communication infrastructure it becomes possible to insert additional and more computationally intensive functions, which can be transferred to another computer system if necessary.

Splitting the architecture into component-based modules enables the development of the subsystems without affecting the already existing services. In addition, platform independence is achieved by using the functionalities of a web service. Thus, different types of smartphones with different operating systems can be used (e.g., Android, iOS, Windows phone, etc.) without having to change the data processing functions. Only the client software on the mobile device has to be adapted.

A. Smartphone

To gain a maximum degree of independence from suppliers and other external influence and to be able at the same time to benefit from the possibilities of an up-to-date smartphone, the Samsung Nexus S was used as a prototype mobile device for AMFIS. It runs the Android™ 2.3 ("Gingerbread") operating system and has a 1 GHz processor.

In the first stage of development the functionality of an off-line GIS (Geographical Information System) [6] was integrated. This offers the possibility of a spatial orientation to the user regardless of a data connection to the internet or the AMFIS control station. In addition, the data communication is designed so that information from AMFIS can be shown on the map application when a connection is available. This concerns, for example, the geographical positions of other sensors or sensor carriers and additional information or instructions such as the request to move to a certain location for reconnaissance. Furthermore, it is possible to feed back own sensor data into the AMFIS system. At present, this includes own position data and the generation of images with the corresponding metadata.

To establish a connection between the smartphone and AMFIS, the available communication possibilities of the mobile device are used. In many scenarios it can be assumed that the user remains in immediate vicinity of the ground station AMFIS. In these cases the connection is established by using Wi-Fi. However, a shortfall of connectivity cannot be ruled out. Hence the data transfer via GSM/UMTS is realized as an alternative communication method.

Nevertheless, the complete loss of the connection has to be taken into account, too. Therefore, the sensor can also function as an independent stand-alone sensor. In this case the accumulated data is buffered and transferred automatically with a re-established connection.

B. Web server

Since the new sensor and its architecture are bound to already existing infrastructure and the independency from any operating system should be preserved, a service-oriented approach was chosen.

By implementing this solution a later integration of other platforms such as the Apple iPhone or a Windows phone device is possible since only the application has requires adaption and/or compilation for the new platform. The functions and processes of the web service or the connection management system for AMFIS do not require any changes.

The web server provides the connection to the internet and offers the runtime environment for the application server. It can be used to pre-filter packages or carry out the authentication for the mobile device. The basic infrastructure is thereby provided by an Apache Http server.

Among other features two web services were implemented to provide the basic functions.

One service is used to transmit the user's data (e.g., photos) to AMFIS. The other one is an update service, which can be used by the smartphone to establish a connection.

The first service for transferring user's data (e.g., photos and text news) is designed as a one way connection to be used exclusively by the smartphone to send data to the application server only. The application server sends an acknowledgment for the incoming data or messages. A transmission of data or messages accumulated by the application server is not possible with this web service. This service is only invoked if required, e.g., if an image or a text message needs to be transferred.

The update service for the representation of the positions of sensors and sensor carriers is used to dispatch data from the application server to the smartphone (see Section C). The smartphone uses this possibility to transfer not only its own position, but also any additional data accumulated so far. Another difference to the first web service consists of the fact that the smartphone calls this service periodically.

The AMFIS Connector software is therefore providing the suitable time slots according to the number of devices to be served. The smartphone is using the allocated time slot to change its own communication configuration accordingly. This is done to prevent a capacity overload, which might result in a data jam or even a breakdown of the communication infrastructure.

Within the update service, not only messages dedicated to be read by the mobile device are fetched but also other information is transferred; for example the position data of the mobile device. The position information of the mobile device is communicated with every existing proxy connection but at the latest with an established connection to the update service.

C. Application Server

The application server provides the web services as well as the connection management to AMFIS. It contains the runtime environment for these functions and is realized by an Apache Tomcat server. The application server offers different web services, which can be accessed by the smartphone. As soon as the smartphone has requested a service, a proxy is generated between the application server and the smartphone. This proxy runs only until the acknowledgement of the service and is terminated thereafter.

The AMFIS Connector management software runs on the application server. The software contains an encoder, which re-encodes the data received by the smartphone in suitable AMFIS messages and transmits these data to AMFIS. This process works also conversely using a suitable decoder to provide information to the mobile sensor. The management process is therefore able to decode AMFIS messages and to route the information to the receiving smartphone. As it is to be expected that the mobile device will not be able to sustain a connection under all circumstances, an agent object for each registered smartphone is created. Within this agent a stack of “messages to be transferred” exists, which contains all messages encoded in the smartphone application-readable format. Without any further transformation, necessary data can be transmitted faster in the event that a connection can be established. The messages remain in the list until reception of the data has been acknowledged. Even though, this increases the duration of the data exchange as well as the amount of the data to be exchanged, this is necessary to prevent a possible loss of messages, which might be critical. If the connection is disturbed or terminated during the transfer process, the data is not deleted but further provided and transferred with the next possible connection. In addition to these services, the agent object provides a certain supervision function. The agent is equipped with a timer, which is reset on each connection between the agent and the smartphone. If no connection to the smartphone can be established before timeout, the system can be informed about the connection state to the mobile device, which can be used for example, to notify the AMFIS system and the user about the lost link.

VI. CONCLUSIONS AND FUTURE WORK

The current state of the development is the first attempt to integrate a smartphone into the heterogeneous sensor network of the security framework AMFIS. The functions realized so far are only the most essential and most basic processes within this subsystem to make it usable.

Beside generating and providing reconnaissance results or reports in the form of images or text messages, the next logical step is to create and distribute video data. The essential problem to be solved is the higher amount of data if one or several sensors start to transmit video data simultaneously.

However, not only electro-optical sensors are conceivable. The array of sensors utilized in smartphones is constantly growing (compass, acceleration sensors etc.). Also, standardized interfaces, such as Bluetooth, can be used to link additional and more specialized sensors to the system.

Conceivable external sensors could be portable gas detectors or systems for detecting radiological and perhaps in the near future, also biological dangers. Moreover, the evaluation and processing of the data could already occur on-site. However, the ascertained results can be transmitted immediately to the ground station including the accompanying geo-information.

Furthermore, functions as the supervision of vital signs or an indoor localization of task forces, using a combination of runtime calculation of the signal, the values of the acceleration sensors and the build-in compass is conceivable.

In addition to these options, the use of the mobile device as an extra access point to distribute data without having to rely on an external connection is a benefit. A sort of multi-hopping network could be built up, which routes the information even under difficult communication conditions so that accumulated data can reach the ground station and other sensors carriers in the overall system.

ACKNOWLEDGMENT

The authors would like to thank Torsten Großkurth and Judy Lee-Wing for their contribution.

REFERENCES

[1] S. Leuchter, T. Partmann, L. Berger, E.J. Blum, and R. Schönbein, “Karlsruhe generic agile ground station,” in: Beyerer J. (ed.) Future Security, 2nd Security Research Conference, 12th-14th September 2007, Karlsruhe, Germany, Fraunhofer Defense and Security Aliance, pp. 159-162. Universitätsverlag, Karlsruhe (2007).

[2] A. Bürkle, "Collaborating miniature drones for surveillance and reconnaissance", in Proceedings of SPIE 7480, 74800H, Berlin (2009); doi:10.1117/12.830408.

[3] F. Segor, A. Bürkle, M. Kollmann, and R. Schönbein, “Instantaneous Autonomous Aerial Reconnaissance for Civil Applications - A UAV based approach to support security and rescue forces,” in: The 6th International Conference on Systems ICONS 2011, pp.72-76, St. Maarten, The Netherlands Antilles (2011).

[4] B. Essendorfer and W. Müller, “Interoperable sharing of data with the Coalition Shared Data (CSD) server,” in: North Atlantic Treaty Organization (NATO)/Research and Technology Organziation (RTO): C31 in Crisis, Emergency and Consequence Management, page 12 - 24, Bucharest (2009).

[5] N. Heinze, M. Esswein, W. Krüger and G. Saur, "Image exploitation algorithms for reconnaissance and surveillance with UAV", in Proceedings of SPIE 7668, 76680U, Orlando (2010); doi:10.1117/12.852555.

[6] M-H. Tsou and J. Smith “Free and Open Source Software for GIS education,” A White Paper, (2011), http://www.iapad.org/publications/ppgis/tsou_free-GIS-for-educators-whitepaper.pdf <retrieved: 11, 2011>.