J Intell Robot Syst (2009) 54:39–59
DOI 10.1007/s10846-008-9254-1

UAVs Integration in the SWIM Based Architecture for ATM

Nicolás Peña · David Scarlatti · Aníbal Ollero

Received: 15 March 2008 / Accepted: 30 June 2008 / Published online: 31 October 2008

Abstract The System Wide Information Management (SWIM) approach has been conceived to overcome the capacity and flexibility limitations of the current ATM systems. On the other hand the commercial applications of Unmanned Aerial Vehicles (UAVs) require the integration of these vehicles in the ATM. From this perspective, the unavoidable modernization of the ATM is seen as an opportunity to integrate the UAVs with the rest of the air traffic. This paper is devoted to study the feasibility and impact of the aggregation of UAVs on the future ATM supported by a SWIM inspired architecture. Departing from the existing technical documents that describe the fundamentals of SWIM we have explored the compatibility with a potential UAVs integration and also explored how the UAVs could help to improve the future ATM system. We will use the weather application as an example in both cases.

Keywords UAV · Air traffic management · System wide information management

1 Motivation and Objectives

The number of aircrafts in operation all around the world has grown steadily from the hundreds to the tens of thousands. As technology advanced, new systems and services were added to the Air Traffic Management (ATM) to improve safety and capacity, but due to the lack of a standard procedure to insert new functionalities in the ATM, these systems were designed independently, with very different interfaces

---

N. Peña (✉) · A. Ollero
University of Seville, Robotics, Vision and Control Group,
Avd. de los Descubrimientos s/n, 41092, Sevilla, Spain
e-mail: nicolas.grvc@gmail.com

D. Scarlatti
Boeing Research & Technology Europe – Cañada Real de las Merinas, 1-3,
Building 4-4th floor, 28042 Madrid, Spain

and had to be hard wired to each other in a specific way for every combination that needed to interoperate. As the number of present systems grew, the cost of inserting a new one was always higher.

The result of this trend is that the current ATM is a rigidly configured, complex collection of independent systems interconnected by very different technologies from geographically dispersed facilities. Then, they are expensive to maintain, and their modifications are very costly and time consuming. Future capacity demands require the implementation of new network-enabled operational capabilities that are not feasible within the current ATM systems. In fact, the safety, capacity, efficiency and security requirements to meet the expected demand require the application of new flexible ATM architectures. A new approach to face this future demand is the so-called System Wide Information Management (SWIM) [1, 2]. This system enables shared information across existing disparate systems for network-enabled operations, and improves air traffic operations by integrating systems for optimal performance [2, 4].

The commercial applications of Unmanned Aerial Vehicles (UAVs) require the integration of these vehicles in the ATM [5]. Currently, the UAVs operate in a completely segregated aerial space due to the absence of protocols for their integration in the Air Traffic Management systems. From this perspective, the planned modernization of the ATM is seen as an opportunity to integrate the UAVs with the rest of the air traffic in a single aerial space. In fact the implementation of the SWIM concept makes easier the integration of the UAVs in the ATM than the architecture in use. Moreover, the standardization of interfaces and the involved network centric concepts involved in SWIM are additional benefits for the integration of UAVs.

This paper is devoted to study the impact of the aggregation of UAVs on the future ATM supported by a SWIM inspired architecture. There are some publications and on going projects on the UASs integration in the aerospace. These usually focus the aspects that need to be improved before this integration happens such as the autonomous sense and avoid [6] or the safety requirements [7]. In [8] a new architecture for this integration is proposed.

This paper studies the integration of the UAVs in the ATM from the point of view of the actual plans for the future SWIM inspired, ATM [2, 3]. The effect at the different layers of the planned ATM structure is discussed. The paper will also explore the possibility that arises from the integration of the UAVs in the ATM that can be achieved in several layers of the proposed architecture. For example, regarding the network layer, which is heavily stressed by the ever growing aircrafts density, it is possible to consider stratospheric UAVs providing in a dynamic way additional bandwidth in areas with such requirements. Another example, in the application layer, is the service provided by the weather application that could be improved by UAVs acquiring information from areas with higher uncertainty on weather conditions.

On the other hand, for a proper integration in SWIM, the software and hardware architectures of the UAVs should be adapted. In terms of the hardware on-board, the limited payload should be considered to prioritize (in terms of safety) the on-board equipment included in the UAV. For instance, the ACAS (Airborne Collision Avoidance System) system should be included and integrated in the software on-board to allow an automated response for collision avoidance in the same way that manned aircrafts do.

We have centered our study in two general aspects of this integration. One is at the application level, where the functionality of the different services as surveillance and weather is offered to the SWIM clients. The other one looks at the layers below the application one to try to evaluate if the approaches proposed for some of the inners of the future ATM, like the proposed client server model and the data models, would be a problem or an advantage for the integration of the UAVs. The paper contains a section regarding to each of the two aspects studied by the authors.

2 Application Layer

This section presents some considerations regarding the SWIM application layer that are useful to put in context some concepts that will be presented later on.

The integration of the UAVs and their associated infrastructure in SWIM applications will be studied from two different approaches:

- UAVs using SWIM applications during their operation, and
- UAVs providing services intended to improve the performance of some SWIM applications.

Regarding the first approach, a common application, such as the weather application, will be selected in order to describe how an UAV will use the weather services. Thus, the components interacting with the UAV will be considered and the differences with respect to a conventional aircraft at the application level (i.e. automated periodic weather reporting instead of on pilot demand), as well as the requirements for the UAVs (on-board probabilistic local weather model, permanent link with a broker, required bandwidth, etc) will be examined. Figure 1 shows the proposed architecture of the weather application in a SWIM enabled NAS (National Airspace System). The elements shown in the Figure will be used in the rest of this section in a simplified manner referring to them as weather data producers, weather data repositories, weather brokers and finally, weather application clients.

The second approach considers UAVs providing an improvement in the applications of SWIM and even new services for manned aircrafts, UAVs and all clients in general. Some examples could be: provide weather information from locations with high uncertainty, response in emergencies, serve as flying repositories of pseudo-static information such as mid and long-term weather information, etc.

In this section, the weather application will be taken as a case study in order to clarify some aspects related to the integration of the UAVs in SWIM.

2.1 UAVs Using SWIM Applications During their Operation

In the application layer, it is relevant to distinguish between UAVs with minor human intervention and UAVs with a remote human pilot or operator. In this section, the analysis will be focused on the first one due to the following reasons:

- When the UAV is teleoperated, the human operator can play the role of a conventional pilot in terms of using the SWIM applications (i.e. processing the messages from a SWIM weather application). Therefore, in this case and from

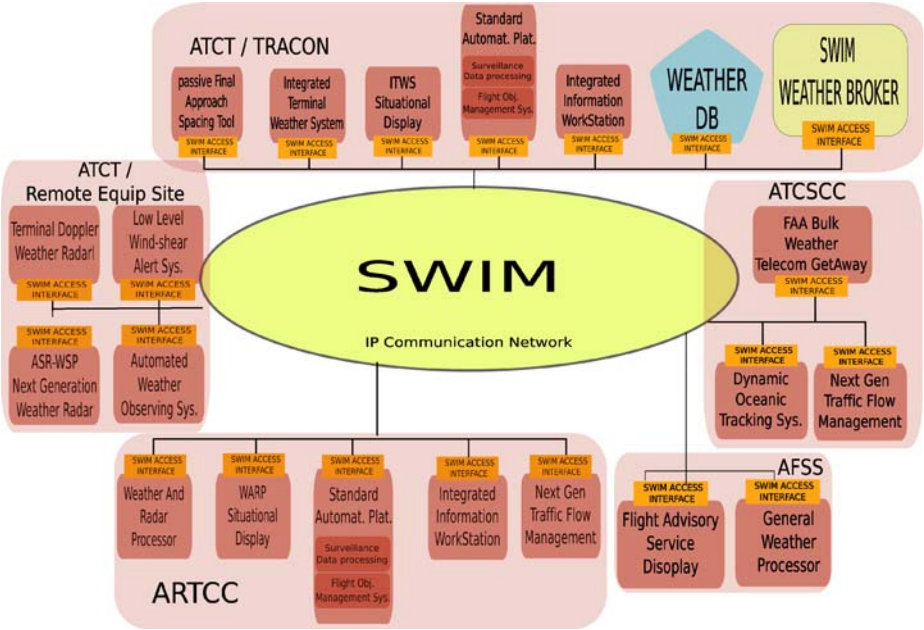


Fig. 1 The weather application elements adapted to the SWIM concepts

the point of view of the application layer, there is no relevant difference between an UAV and any other conventional aircraft. As it will be discussed later, in other layers several differences arise.

- Although nowadays almost all UAVs require in some extend the intervention of a human teleoperator, it is expected a transition towards full autonomy [9], allowing one operator to manage several UAVs. Furthermore, full autonomy would be even possible in some SWIM applications as weather remote sensing using stratospheric UAVs [10], reducing their operational cost.

Then, in the following, only autonomous UAVs will be considered. Therefore, in the weather application for example, problems related to autonomous weather messages processing and decision making should be addressed.

In the current ATM systems, after receiving one message, the human pilot has to “decode” and process it in order to make a decision taking into account other parameters such as the type of aircraft, payload, remaining fuel, etc. METAR (Meteorological Terminal Aviation Routine Weather Report) and TAF (Terminal Aerodrome Forecast) information are essential for flight planning and in-flight decisions. TAF messages are a very concise, coded 24-hour forecast for a specific airport that, opposed to a public weather forecast, only addresses weather elements critical to aviation as wind, visibility, weather and sky condition. A more recent type of message is the Transcribed Weather Enroute Broadcasts (TWEBs) which are composed by some Weather Forecast Offices (WFOs) and contain very similar information to a TAF but for a 50 mile wide corridor between two or three frequently connected airports.

As far as an autonomous UAV should perform this whole message processing by itself, the following requirements could be considered for the decisional level of an UAV:

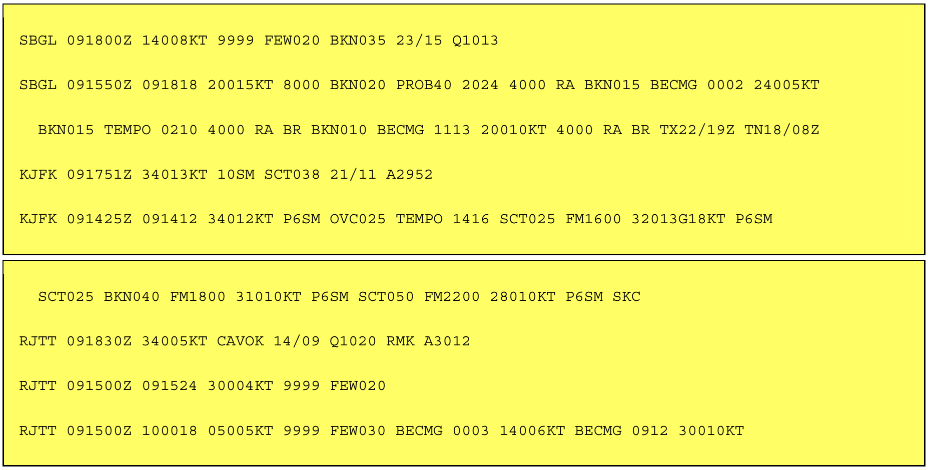
- Estimate when a weather report is required. If the uncertainty about the weather in a given area of the flight plan is higher than a given threshold, the UAV should ask for a weather report in order to reduce the uncertainty of its local weather model. The weather reports requests will have a set of parameters such as the area of interest for example. Although a periodic weather reports request scheme could be adopted, the capability to autonomously estimate when a report is required will decrease the limited available data bandwidth.
- Process and decode the standard weather messages formats. Some examples of those formats are shown in Fig. 2.

In the proposed SWIM data models exposed in [3], several approaches are possible regarding the weather messages:

(a) Embedding the formats currently used into SWIM weather messages: The UAV should have to decode and process those formats.

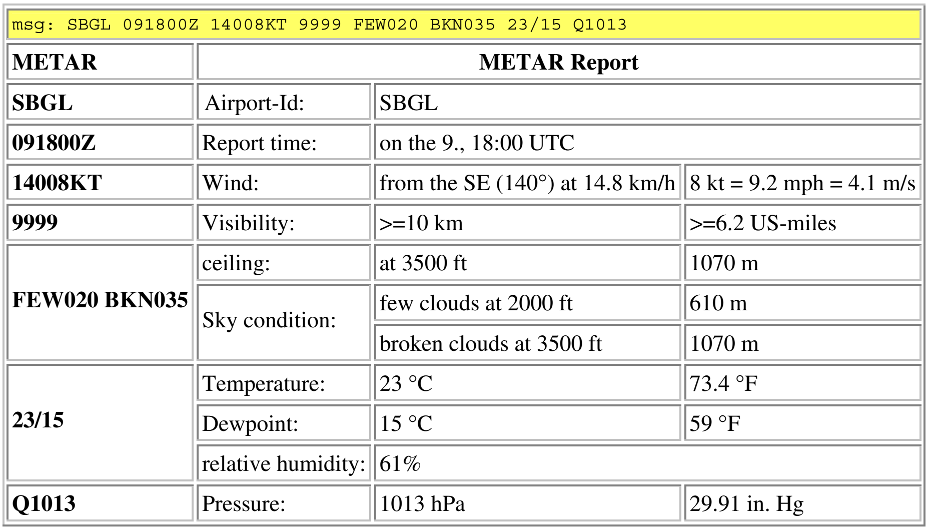
There are several software projects which deal with METAR and TAF messages and even web sites providing access to some of these messages in a human readable format as [11].

In particular, the metaf2xml software [12] parses and decodes aviation routine weather reports and aerodrome forecasts (i.e. METAR and TAF messages) and stores the components in XML. They can then be converted to plain language, or other formats (see an example in Fig. 3). Similar software could be running on board the UAV, parsing the SWIM weather messages. After the parsing process, the information should be used to update the local weather model. A similar operation is performed by a software module of the FlightGear project [13] which updates a global weather model from parsed weather messages.



```
SBGL 091800Z 14008KT 9999 FEW020 BKN035 23/15 Q1013

SBGL 091550Z 091818 20015KT 8000 BKN020 PROB40 2024 4000 RA BKN015 BECMG 0002 24005KT

  BKN015 TEMPO 0210 4000 RA BR BKN010 BECMG 1113 20010KT 4000 RA BR TX22/19Z TN18/08Z

KJFK 091751Z 34013KT 10SM SCT038 21/11 A2952

KJFK 091425Z 091412 34012KT P6SM OVC025 TEMPO 1416 SCT025 FM1600 32013G18KT P6SM

  SCT025 BKN040 FM1800 31010KT P6SM SCT050 FM2200 28010KT P6SM SKC

RJTT 091830Z 34005KT CAVOK 14/09 Q1020 RMK A3012

RJTT 091500Z 091524 30004KT 9999 FEW020

RJTT 091500Z 100018 05005KT 9999 FEW030 BECMG 0003 14006KT BECMG 0912 30010KT
```

Fig. 2 Some METAR and TAF messages from Rio (SBGL), New York (KJFK), and Tokyo (RJTT)



`msg: SBGL 091800Z 14008KT 9999 FEW020 BKN035 23/15 Q1013`

METAR	**METAR Report**		
SBGL	Airport-Id:	SBGL	
091800Z	Report time:	on the 9., 18:00 UTC	
14008KT	Wind:	from the SE (140°) at 14.8 km/h	8 kt = 9.2 mph = 4.1 m/s
9999	Visibility:	>=10 km	>=6.2 US-miles
FEW020 BKN035	ceiling:	at 3500 ft	1070 m
	Sky condition:	few clouds at 2000 ft	610 m
		broken clouds at 3500 ft	1070 m
23/15	Temperature:	23 °C	73.4 °F
	Dewpoint:	15 °C	59 °F
	relative humidity:	61%	
Q1013	Pressure:	1013 hPa	29.91 in. Hg

Fig. 3 Example of an automatic parsing and conversion to HTML of a METAR message

(b) Changing the formats currently used: One of the advantages of the proposed network-centric brokers based SWIM architecture is that allows fusing information from several sources to generate a response for a given request. The message with the response can have a new unified SWIM format designed to be easily decoded and processed. In fact, as the messages are generated by a broker merging several sources of information, the UAVs themselves only need to provide basic weather reports and a complex local weather model is not required.

The approach (a) would allow an easier and faster adaptation to SWIM (not for the UAVs integration in SWIM), but (b) takes full advantage of the SWIM architecture to provide better weather services.

- Autonomous decision making integrating the weather information with other parameters such as UAV dynamics, payload, remaining fuel, relevance of its mission, etc. Resulting plans should be directly reported with a proper data structure. In any case, changes in data structures reported periodically such the planned 4D trajectory, would also inform about the new local plan to other SWIM entities subscribed.

Regarding the UAV's autonomous decision functionality, it would be mandatory to periodically check that this mechanism is working properly. For example, the local plan or 4D trajectories reports could include the list of identifiers of weather messages used by the autonomous decision software. Furthermore, the decision making mechanism could be replicated in a ground facility in order to detect failures in the UAV operation (UAV malfunction), or eventually to protect against malicious intervention. In such a case, a human operator could take the control over the UAV through SWIM specialized channels that offer the CoS needed for teleoperation.

Finally, in order to illustrate an example of an UAV using the weather application over SWIM, the following storyboard based on Section 8 of [3] is presented. In a first stage (see Fig. 4), a weather database is updated with the messages sent from the weather data sources connected to SWIM. Following the publish/subscribe paradigm, a broker is acting as an intermediate element in the transaction, decoupling the data sources from the database itself.

In a second step, an UAV detects that its local weather model has a high level of uncertainty along its route and requests additional information. This query is managed by the nearest SWIM broker and sent to a weather data repository, where it is interpreted and processed (see Fig. 5). Another possible and more complex approach could involve a broker with the capability to process the query and translate it into a set of simple requests for different weather repositories. The two options differ in that the second one requires more intelligence in the brokers and less in the Data Repositories, taking the brokers a more than ‘middle man’ role. While the later can be seen as a more complex approach, it is more in line with the global SWIM philosophy of a really distributed, network centric system where data brokers for each service play a key role, isolating as much as possible the service clients from the inners of the backend elements of the service and the other way around.

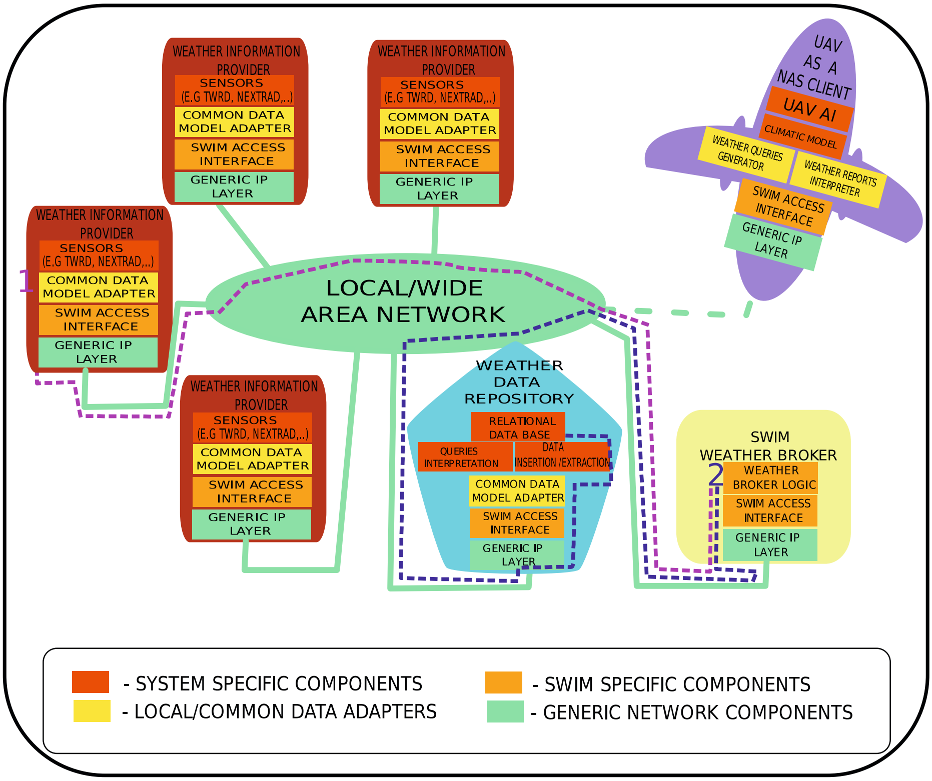


Fig. 4 A weather sensor connected to NAS and adapted according to SWIM reporting data. The information is sent to a database repository (or many) following the publish/subscribe paradigm

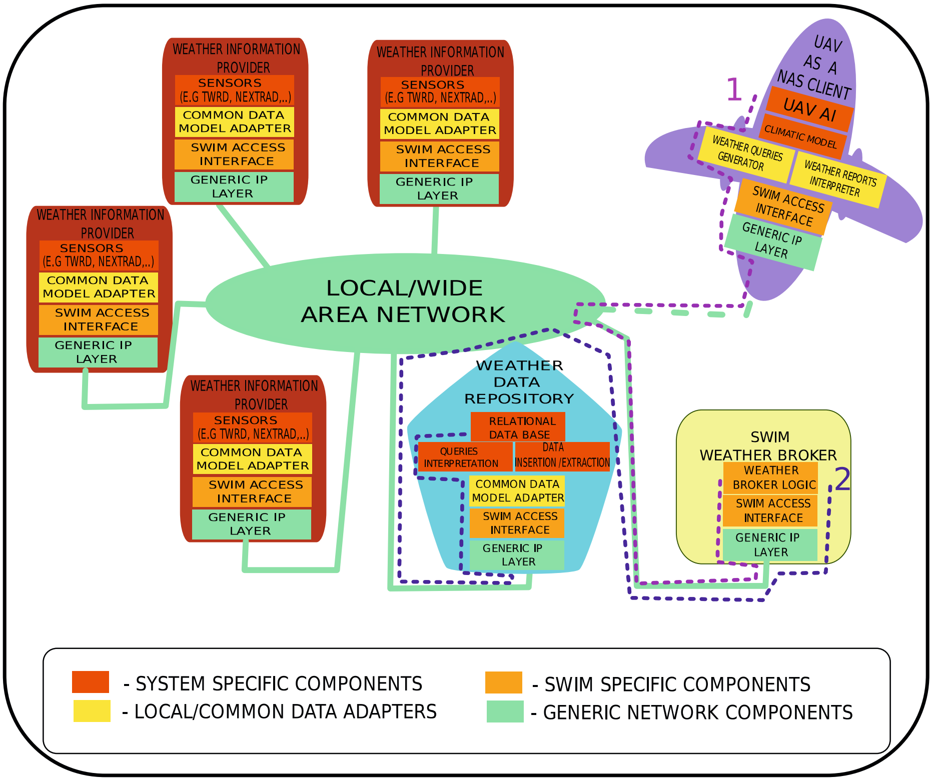


Fig. 5 When the UAV decision making core requires more weather information, a query is sent, via the closer broker, to a weather data repository

In the final stage (see Fig. 6), the required information is generated by the weather repository, formatted to be processed by the UAV and sent in a message to the broker, which just forwards it to the UAV. The message formatting task could have been also performed by the broker. Actually this would be obligatory in the ideal case of more intelligent brokers commented before. In any case, this would have been transparent for the rest of SWIM elements.

Once the response has arrived to the UAV the new weather information would be finally inserted in the local weather model satisfying the need that initially caused the request.

2.2 UAVs Improving the Performance of SWIM Applications

The adoption of the SWIM concept provides a unique opportunity to improve existing services and provide new ones. The current architecture has independent subsystems and services, and different physical connections for each type of information, whereas the SWIM architecture is network-centric and designed to grow and adapt to future demands. Scalability and connectivity are inherent features in SWIM and the cost of adding new services is reduced when comparing with the current architecture [14].

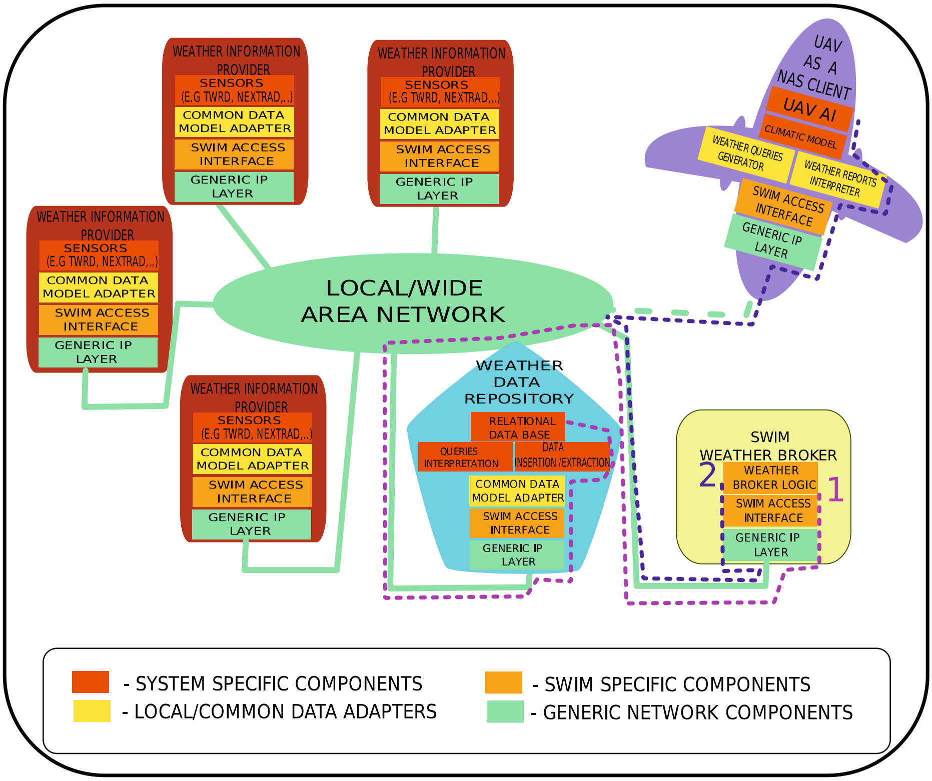


Fig. 6 The information requested is sent to the UAV

Taking into account the inherent scalability of SWIM and the emerging UAV technologies, it is expected to have UAVs supporting SWIM applications and even new UAV based applications. In the next subsections, several examples are provided assuming a flexible SWIM architecture with data fusion capabilities consistent with the COP concept [3]. Moreover, it is assumed that the brokers can manage “abstract” requests (not only forwarding raw data provided by sensors).

2.2.1 UAVs Acting as Weather Sensors

A team of stratospheric UAVs could be used to gather weather information [10] from areas with a weather estimation uncertainty higher than a given threshold. The autonomy of this type of UAVs is being improved by the recent developments in photovoltaic technology with the goal of achieving unlimited autonomy [15].

If it is expected to have traffic in a zone with high weather uncertainty, the weather servers could compute a list of waypoints to be visited for gathering weather data. Those servers would have access to the future traffic requesting this information to the corresponding brokers. Therefore, the waypoints could be prioritized depending on the expected routes in this zone.

The waypoints list could be sent to a stratospheric UAVs team that will autonomously allocate the waypoints among themselves using distributed algorithms trying to optimize some criteria, such as minimizing the total mission time or cost.

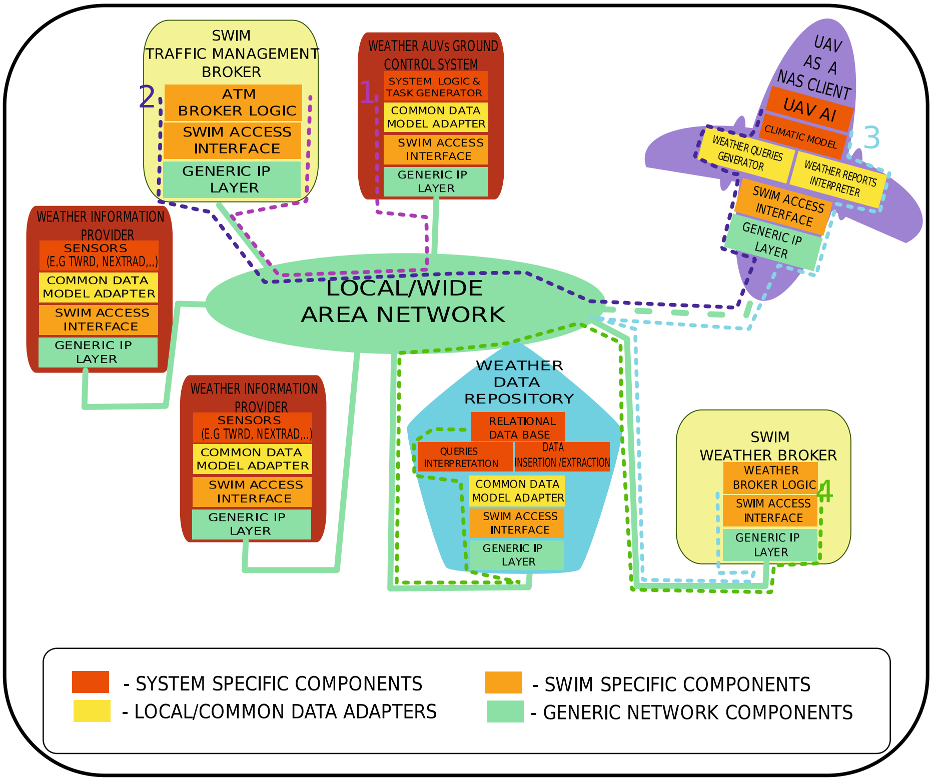


Fig. 7 UAV acting as a weather sensor. It is possible to insert this service in the NAS without changing the SWIM design

Figure 7 shows how the UAVs, acting as weather data sources, fit well in the proposed SWIM architecture. In this figure, four stages have been identified by different types of connecting lines in the information flow. In the first phase, composed by two stages, the UAV receives (via an ATM broker) a request consisting in a "goto" command. The UAV executes it and go to the specified location to gather weather information with the sensors on-board. In the second phase, the UAV reports weather data to the closest SWIM weather broker and then send the information to the weather repository.

2.2.2 Surveillance in Specific Locations and Emergency Response

Nowadays, UAVs are mainly used in surveillance missions taking aerial images from a given area. Those missions are mainly for military purposes [9], but some civil applications can also be found [16, 17]. In the future, those surveillance functionalities could be integrated in SWIM providing services such as autonomous surveying of a disaster area or assistance to identify aircrafts present in the surveillance system that are not responding to the radio.

There are several companies that provide satellite images of specific areas during a given period. Low orbit satellites can provide sequences of images during a period of time limited by their orbital speed. Those companies could adopt the use of UAVs to provide more flexible services at a lower cost. Their clients could even have access in real time to the images via a web browser in a computer connected to SWIM. A quote from [18]: “The progressive implementation of the SWIM principles in AIM (Aeronautical Information Management) is in fact AIM’s evolution to IM, or Information Management that is fully SWIM based and which is the ultimate goal”.

A fully SWIM compatible system could be easily interconnected with other network-centric platforms allowing to increase the number of services provided. Furthermore, as far as the amount of shared information and resources will be increased, the cost of implementing new services will decrease.

In order to illustrate an application that could exploit the benefits of interconnecting SWIM to other existing networks, the following situation will be considered, as a generalization of the application of UAVs for fire monitoring [19]. A fire station in Madrid receives a fire alarm from the sensors located in a given building. Then, an autonomous visual confirmation process starts sending a request to GIM (General Information Management) for images from the building area. GIM is part of a global network integrated by many networks including SWIM and therefore the request is finally routed to a SWIM broker. Several manned and unmanned aircrafts are flying over the city and the broker selects a proper one equipped with a camera on-board to establish a direct link between the camera and the fire station. The camera pan&tilt is pointed to the building area and the video streaming is received in the fire station allowing confirm/discard the fire (this streaming is allowed in the third broker model presented in [3]).

3 Middleware and the Network-Centric Nature

3.1 Introduction

In this section, the impact of the UAVs insertion in the SWIM middleware is studied. As far as SWIM is being designed to be very scalable and generic with the stated goal of easing the integration of new elements, the following analysis can be considered as an indicator that it is well designed for it, at least in the UAVs case.

In the current SWIM specification, it is described how piloted aircrafts (or any other data source/sink) can publish their information and subscribe to any authorized data channel. In this section it will be analyzed if those procedures would also allow the integration of UAVs in a transparent way for other SWIM users. For this purpose, the main SWIM architecture concepts are revised in the next subsection to check if they are general enough for this seamless integration. In the next subsections, required changes or additions for the UAVs integration that had been spotted during our analysis are listed and explained.

3.2 SWIM Main Concepts and UAVs

3.2.1 Network-centric Nature

One of the main ideas present in SWIM is to connect all the systems integrating ATM in a uniform manner with well defined common interfaces, all connected to a network with information flowing from the sources to the sinks sharing generic routing channels. This information can be also processed in any intermediate subsystem

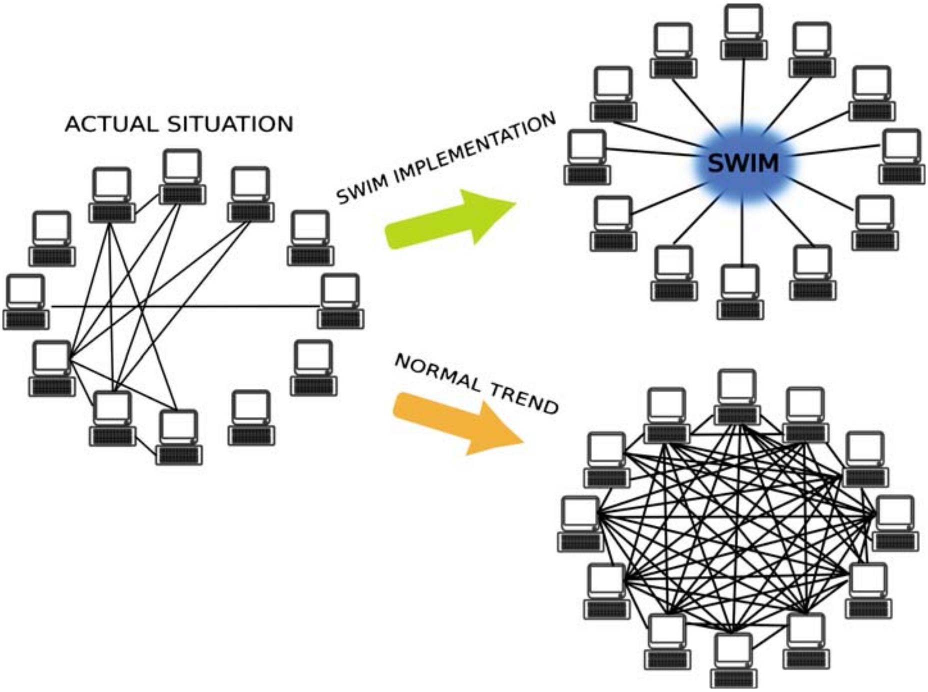


Fig. 8 SWIM network-centric approach versus point to point dedicated data link connections

demanding it. This concept represents a real revolution in the civil aviation as far as nowadays, ATM is mainly composed by many independent subsystems connected by dedicated channels in a very rigid way (see Fig. 8). Therefore, any change such as inserting a new subsystem has a significant associated cost.

On the other hand, in the recent research area of multi-UAV systems the architecture adopted has usually been network-oriented as it is the natural solution that leads to a flexible and cost effective interconnection among multiple systems. Regarding the physical communication layer, it is possible to find solutions with different channels for different types of information, i.e. high bandwidth analog channels for images transmission [20] or dedicated channels for teleoperation [21]. But the principal trend nowadays is to have one unique digital channel (or more if fault tolerance is a requirement) shared by different types of information. The progress in telecommunications technology is making this approach feasible.

Therefore, as the recent developments in multi-UAV systems follow a network-centric approach close to the SWIM architecture, from this point of view the introduction of the UAVs in SWIM is possible and even can be considered as natural.

3.2.2 The Publish/Subscribe Paradigm

Once a network-centric approach is adopted, the elements connected to the network can interchange information according to different models. Some of them are listed in Table 1, extracted from [3], which compares them with respect to the degree of decoupling between data producers and consumers.

Table 1 Different data distribution models compared w.r.t the degree of decoupling between information producers and consumers

Abstraction	Space decoupling?	Time decoupling?	Synchronization decoupling?
Message Passing	No	No	Publisher side
RPC/RMI	No	No	Publisher side
Async. RPC /RMI	No	No	YES
Notifications	No	No	YES
Shared Spaces	Yes	Yes	Publisher side
Message Queuing	Yes	Yes	Publisher side
Publish Subscribe	Yes	Yes	Yes

A high degree of decoupling leads to more robust solutions and to lower costs for the adaptation or insertion of new elements, which are properties required in SWIM. In general, the publish/subscribe paradigm allows a total decoupling between the sources and the sinks of information. Furthermore, new sources or sinks can be added or removed dynamically with a minor impact on the operation of the network. In fact, the publish/subscribe paradigm has been already adopted in other areas such as wireless sensor networks or multi-robot research [17, 22, 23], where the properties mentioned above are also relevant. Moreover, the performance of the network scales well against changes in the demand of a given type of information if dynamic replication techniques are applied.

In the publish/subscribe paradigm, the flow of information between sources and sinks is managed dynamically by one or several intermediate entities, usually called data brokers (see Fig. 9). The middleware is composed by these intermediate entities and their communication protocols. As a result, there is no direct communication between sources and sinks and if a new subsystem has to be added to the network, it is only necessary to develop its interface with the data brokers. Therefore, compatibility tests between this subsystem and other existing subsystems in the network are not necessary, decreasing the integration cost and time. In the design of SWIM, the publish/subscribe architecture was adopted from the beginning to allow an easy integration of new subsystems in the NAS, such as the UAVs themselves.

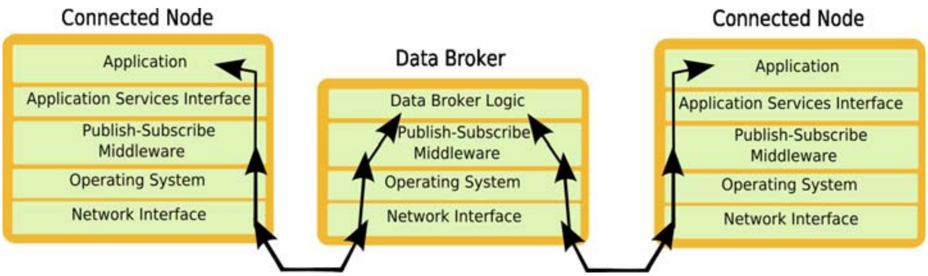


Fig. 9 Pure publish/subscribe model, with no direct interaction between clients

3.2.3 Architectures for the Brokers

There are several possible architectures for the data brokers, depending on the required services and capabilities for the whole communication system. Form the architectural point of view, the simplest one could be similar to the solution adopted in [17], where there is no independent element managing the information exchanges. The implementation of the middleware interface in the clients provides this functionality in a distributed way as shown in Fig. 10. This architecture does not follow a pure publish/subscribe paradigm, but allows low latencies and it is a good solution when real time operation is required.

On the other hand, the most complex architecture could use different protocols to offer pure publish/subscribe functionality for services requiring low bandwidth and a direct connection oriented protocol for streaming data services.

There are three different brokers architectures proposed for SWIM in [3] and all the information related to the SWIM middleware design is consistent with the analysis provided in this paper. Table 2, extracted from the executive summary of [3], shows the main differences between the architectures proposed for the SWIM brokers. It should be pointed out that, due to the network centric nature of SWIM, more than one model can operate at the same time in the same network depending on the application and data types involved in the communication.

On the other hand, the complexity of the communication system, and hence the fault probability, would be increased, at least linearly, with the number of different broker models implemented. Therefore, this number should be kept as low as possible and the integration of the UAVs services should be adapted to the models proposed in Table 2, which are general enough:

- Model "Pub/Sub Broker": It follows the strict publish/subscribe paradigm, so the UAVs will only have to communicate with the brokers and the integration would be easier. The latency associated with this model makes it incompatible with some of the most common UAV applications nowadays, such as teleoperation. Furthermore, there are also services offered by the UAVs such as surveillance video streaming that generates a massive amount of information and the use of data brokers could represent a bottleneck, increasing the latency and decreasing the quality of service.

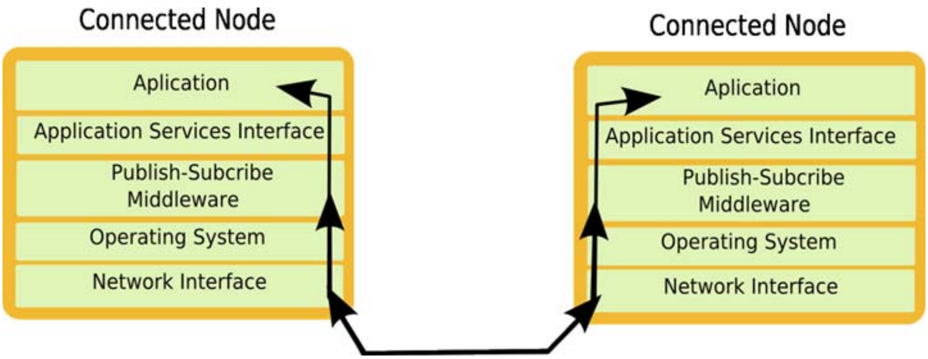


Fig. 10 Distributed version of the publish/subscribe paradigm

Table 2 Three different architectures proposed for the SWIM brokers

Broker Model	Description	Advantages	Disadvantages
Pub/Sub Broker	Implements the full set of functions needed to offer a pure Pub/sub middleware so that all operations between publishers and subscribers are completely decoupled in time, space and synchronization	Changes in publishers or subscribers completely transparent to each other; unified management of exchanged information	Possible extra latency in the process as data flows through the broker; This fact also means that the broker is a potential bottle neck limiting the ability to handle large streams of data
Lightweight Broker	Implements a subset of the functions needed to offer a pure Pub/sub middleware and implements the rest using traditional messaging mechanisms. This mixture means that not all operations between publishers and subscribers are completely decoupled	Supports implementation of several variants of Pub/Sub schemes	Publishers and Subscribers are not fully decoupled; predefined data channels need to be established
VC Broker	VC Broker is a superset of the functionalities offered by the pure Pub/Sub model. It can offer completely decoupled communication between the endpoints but also implements the primitives needed to establish a Virtual Circuit between to of them to achieve better latency or higher bandwidth	Broker approach is tailored to data type; Can offer different QoS for different data types	Extra complexity in information management functions such as monitoring; When connected by virtual circuits the Publishers and Subscribers are not fully decoupled

- Model “Lightweight Broker”: This model is intended to simplify the adoption of SWIM by using more classical approaches that would allow reusing some of the current interfaces. But the integration of the UAVs would require to implement those interfaces as special cases that do not follow the publish/subscribe policy implemented in the brokers. This option can be cost effective in the short term, but it is not as uniform or powerful as the other two models.
- Model “VC Broker”: In this solution, the broker offers a different kind of service depending on the data type. It works as the first model when it is enough to fulfill the service requirements. But when low latency or high bandwidth is required, the broker provides a virtual connection between the data producer and consumer to prevent the potential bottleneck due to centralized brokers. This model is general enough to fulfill all the requirements from the point of view of the UAVs integration. For most of the services, the UAV could use the

publish and subscribe pure model, as any other SWIM client. But for latency sensible applications such as teleoperation [24] or high bandwidth requirements as real time surveillance video transmission, virtual circuits between the UAV and the client application can be created by the broker dynamically.

3.2.4 Access Solutions Proposed for the SWIM Clients

Figure 11 shows three different models proposed to connect to SWIM. In the first model there is a SWIM interface software which is running on the hardware on-board and interacting with the brokers. This model is preferred for the UAVs integration as far as it is the most flexible solution in terms of the potential spectrum of services that can be offered and, additionally does not require increasing the payload. The downside of this solution is that the interface is not so decoupled from the UAV specific software and hardware as in the other options, and special tests could be required to check the implementation and performance of the SWIM interface in order to avoid security problems.

The second model is based on specific hardware to support the connection of currently NAS integrated subsystems to SWIM. Those subsystems have specific interfaces that require a hardware adaptation (see Section 8 in [3]). This model is not necessary for the UAVs as far as the hardware on board is quite flexible and updatable. In fact, the next generation of UAVs could be designed with SWIM hardware compatibility. Anyway, this model allows addressing the security issues much better than the others because the SWIM interface is based on a specific hardware that can be designed following "trusted computing" principles.

Finally, the third model is based on standard web browsers, whose services can be "upgraded" by new web 2.0 technologies in the near future. In the last years, web browsers have been used for teleoperation applications in robotics and even in some

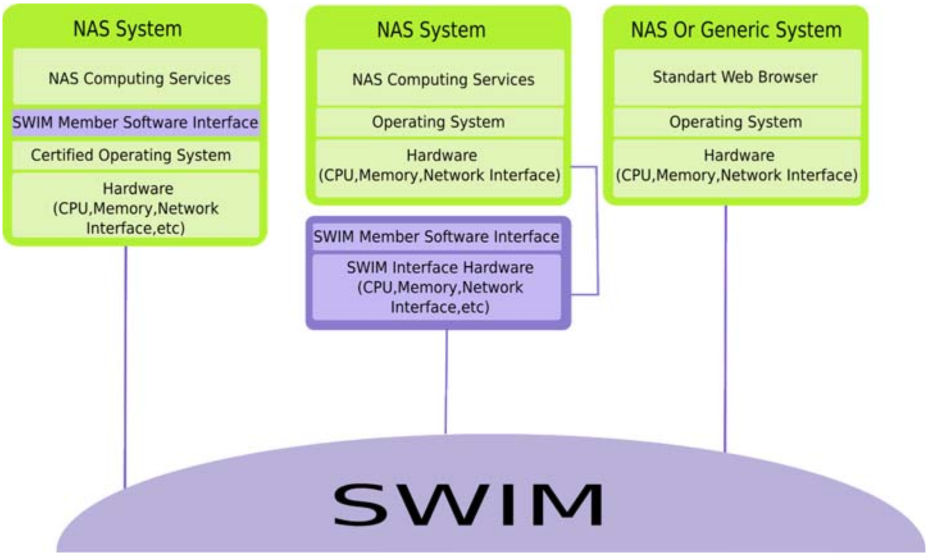


Fig. 11 Different models to connect to the SWIM network

UAVs control centers. Therefore, it is possible to have SWIM clients based on web browsers providing access to the UAV services and, in the opposite direction, the UAVs could also have access to information servers, such as map servers, through a web interface. In any case, limitations due to the latency and associated overhead involved in this model should be considered in order to select which interface to use for each service provided or required. The advantage of using this third model is that it is easier to develop and that the web browser/server protocols are well tested and designed with security in mind.

A combination of the first and third models (for non critical data queries) could be the best option.

3.2.5 UAVs Interfaces for Data and Services Access

The SWIM data model is described in the Appendix G of [3], where a detailed study of the different factors to be taken into account in the specification of the SWIM data structures is provided. Moreover, the impact of the data model on the flexibility and complexity of adding new subsystems is also presented.

Regarding the data model, the main design decisions made up to now and found in the used references can be summarized as follows:

- All the information should have a unified format, which has been called "SWIM Common Data Model" (see Fig. 12). Therefore, all the messages must be embedded in a standard data container, whose header should include standard fields independent of the type of message, such as its source, the data time stamp, etc. This mandatory header required in all the SWIM messages contains a set of fields which are usually referred as "SWIM Common Data Fields". During the development of SWIM it is expected to have sub-headers corresponding to subclasses of data types leading to a tree of standard data structures. For every type of message, this tree will contain the list of mandatory fields and their associated

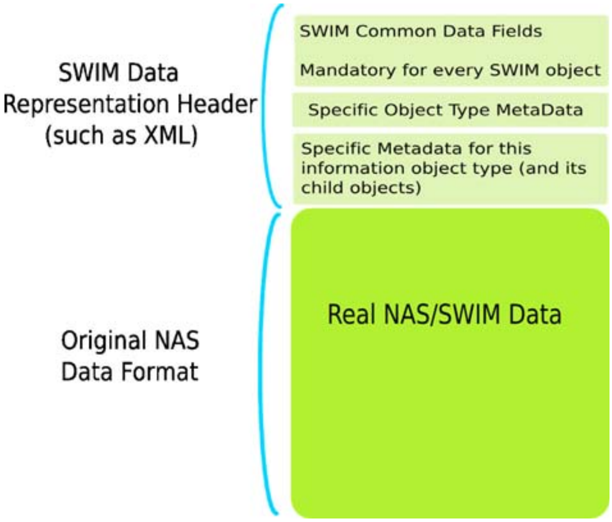


Fig. 12 Common data model diagram

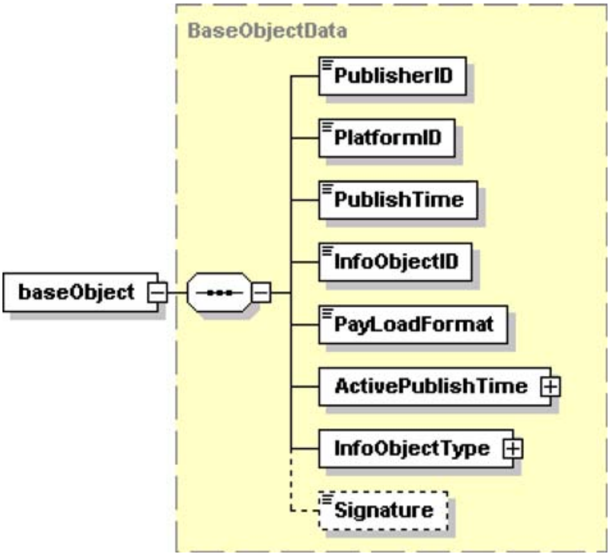


Fig. 13 Example of a SWIM common data format base class, exposing examples of possible fields of the Common Data Fields

data structures. An example of a SWIM common data format base class is shown in Fig. 13.

- The message structure should be embedded in the messages themselves, allowing a client to process a message without any previous knowledge about its internal data structure. This feature can be implemented using metalanguages or markup languages, such as XML (eXtensible Markup Language), and provides flexibility in the messages generation. For example, if a field is not applicable in a given message, this field is not included. Other advantage when using markup languages is the low probability for bad interpreted messages. When a new message is received, only the fields with known identifiers are processed whereas the rest of fields are ignored. This characteristic also makes easier the migration from an old type of message to a new one. During the transition both the old and the new fields are sent in order to keep offering the same services.
- In an initial stage, the NAS messages in their current format should be embedded into the new SWIM messages to allow an easier migration of the existing subsystems. In fact, once the original NAS message has been extracted from the new message, the rest of the process will remain the same. Later, those old data structures can be fully substituted for XML messages.

Almost all the decisions summarized above makes easier the integration in SWIM of the UAVs, as any other kind of new NAS clients, due to the fact that they were adopted with the flexibility criteria in mind. Of course, this statement does not apply to the last point, which is only oriented to decrease the migration cost.

3.3 Aspects Requiring some Adaptation

In the previous subsection, it has been mentioned that in contrast to the rigid nature of the current NAS architecture, the flexible network-centric design of SWIM allows an easy integration of the UAVs. But there are several aspects that could require further considerations due to the particular characteristics of the UAVs. It should be pointed out that the required adaptations presented in the next subsections do not come from the autonomous nature of the UAVs. In fact, the software on-board can make this autonomy characteristic nearly transparent for the rest of NAS subsystems.

In the following subsections, several aspects that could require some adaptation to tackle with some particular characteristics of the UAVs are presented. Moreover, possible solutions built on top of the basic functionality provided by SWIM are also depicted. Those solutions are based on elements that could be also added to the current SWIM specification in order to provide services requiring high flexibility.

3.3.1 Dynamic Brokers

The current UAV applications usually involve taking-off and landing from temporal or improvised locations. From the proposed architecture for the hierarchical connection and distribution of SWIM brokers [3], it seems that the idea is to have at least one broker in every airport that manages the insertion of new aircrafts into SWIM.

If the integration process of the UAVs implies to operate from conventional airports, their functionally will be drastically reduced. Therefore, an UAV taking-off from a given location needs a procedure to signal that it has started to operate and requires SWIM services (even if it is far away from an airport) such as receiving the "clear for take-off" message. The whole procedure requires a connection to a broker.

Nowadays, the ground control stations of the UAVs are continuously connected to them, and in some aspects, act as airports control towers (ATCT) for the UAVs. Then, it seems a natural solution to equip the ground control station with a broker that can link the UAV with other NAS elements. This solution involves dynamic brokers that are subscribing and unsubscribing to SWIM continuously from different locations.

3.3.2 High Bandwidth Channels on Demand

The teleoperation of the UAVs or some specific UAVs services such as surveillance could require the transmission of high bandwidth data in real-time during certain periods. Therefore, protocols to establish dedicated communication channels on demand are also required.

3.3.3 Collaborative UAVs Surveillance System

Regarding surveillance, due to the small dimensions and furtive nature of some UAVs, all the UAVs should be equipped with GPS and should continuously broadcast their positions at a given rate. This positional information could be merged with the rest of the surveillance related information (as the one provided by the primary and secondary radars) by the related data brokers. Given that the GPS infrastructure has been improved in recent years with the goal of making it more reliable and useful for the FAA this possibility should be easy to implement.

New elements designed to increase the usefulness of the GPS system for ATM applications are:

- The Wide Area Augmentation System (WAAS) [24]: Created by the Federal Aviation Administration (FAA) to augment GPS with additional signals for

increasing the reliability, integrity, accuracy and availability of GPS for aviation users in the United States.
- The Local Area Augmentation System (LAAS): Created to allow GPS to be used for landing airplanes. LAAS is installed at individual airports and is effective over just a short range. This system should help the autonomous landing and take off of UAVs in normal airports.

3.3.4 Special Communication Technologies

As far as satellite links are intended to be used for global coverage in SWIM (Satellite enhanced CNS), the payload and budget limitations of some types of UAVs in terms of communication equipment on-board could be a limitation. In such cases, it could be the responsibility of the UAVs associated ground station to provide the communications link with its UAVs. As this is not the best, most general solution, it is the most straight forward one as the only link between the UAVs and the rest of the SWIM components would by its associated ground control station. This is coherent with the proposed figure of dynamic data brokers presented in Section 3.3.1.

4 Conclusions

The transition from the current rigidly configured ATM approach to the SWIM based ATM architecture will represent an important change in ATM concepts and procedures. This transition could represent a good opportunity to facilitate the introduction of the UAVs in non-segregated aerial spaces, which has been recognized as one of the main barriers for commercial UAV applications.

The UAV integration is examined in this paper at different layers of the ATM structure from global concepts, as the network centric nature and the publish/subscribe paradigm, to the particular interfaces, broker and data models required to implement SWIM.

Furthermore, the UAVs integration could also help to improve basic ATM services, such as the weather information, and to offer new services such as on demand surveillance.

Then, it can be seen that the required extensions to include UAVs in the air traffic management of non-segregated aerial spaces are minimal and compatible with the proposed SWIM based ATM architecture.

References

1. Boeing Technology: Phantom works. Advanced Air Technology Management. http://www.boeing.com/phantom/ast/61605_08swim.pdf. Accessed 24 October 2007
2. SWIM Program Overview: http://www.swim.gov (redirects to www.faa.gov). Accessed 9 May 2008
3. System-Wide Information Management (SWIM) Architecture and Requirements. CNS-ATM Task 17 Final Report, ITT Industries, Advanced Engineering and Sciences Division, 26 March 2004
4. Jin, J., Gilbert, T., Henriksen, S., Hung, J.: ATO-P (ASD 100)/ITT SWIM Architecture Development. CNS-ATM, 29 April 2004

5. Koeners, G.J.M., De Vries, M.F.L., Goossens, A.A.H.E., Tadema, J., Theunissen, E.: Exploring network enabled airspace integration functions for a UAV mission management station. 25th Digital Avionics Systems Conference, 2006 IEEE/AIAA, Oct. 2006, pp. 1–11
6. Carbone, C., Ciniglio, U., Corraro, F., Luongo, S.: A novel 3D geometric algorithm for aircraft autonomous collision avoidance. In: 45th IEEE Conference on Decision and Control (CDC'06), pp. 1580–1585. San Diego, California, December 2006
7. UAV safety issues for civil operations (USICO), FP5 Programme Reference: G4RD-CT-2002-00635
8. Le Tallec, C., Joulia, A.: IFATS an innovative future air transport system concept. In: 4th Eurocontrol Innovative Research Workshop, December 2005
9. UAV Roadmap 2005–2030 – Office of the Secretary of Defense, August 2005
10. Everaerts, J., Lewyckyj, N., Fransaer, D.: Pegasus: design of a stratospheric long endurance UAV system for remote sensing. In: Proceedings of the XXth ISPRS Congress, July 2004
11. Weather Information Interface: http://aviationweather.gov/. Accessed 9 April 2007
12. metaf2xml: convert METAR and TAF messages to XML. Project web site: http://metaf2xml.sourceforge.net/. Accessed 25 May 2006
13. FlightGear Flight Simulator Project Homepage: http://www.flightgear.org/. Accessed 5 May 2008
14. Meserole, C.: Global communications, navigation, & surveillance systems program – progress and plans. In: 5th Integrated CNS Technologies Conference & Workshop, May 2005
15. Romeo, G., Frulla, G.: HELIPLAT: high altitude very-long endurance solar powered UAV for telecommunication and earth observation applications. Aeronaut. J. **108**(1084), 277–293 (2004)
16. Merino, L., Caballero, F., Martínez-de Dios, J.R., Ollero, A.: Cooperative fire detection using unmanned aerial vehicles. In: Proceedings of the 2005 IEEE, IEEE International Conference on Robotics and Automation, Barcelona (Spain), April 2005
17. Ollero, A., Maza, I.: Multiple Heterogeneous Unmanned Aerial Vehicles. Springer Tracts on Advanced Robotics. Springer, Berlin (2007)
18. Aeronautical Information Management Strategy, V4.0. EUROCONTROL, Brussels, Belgium March 2006
19. Merino, L., Caballero, F., Martínez-de Dios, J.R., Ferruz, J., Ollero, A.: A cooperative perception system for multiple UAVs: application to automatic detection of forest fires. J. Field Robot **23**(3), 165–184 (2006)
20. Beard, R.W., Kingston, D., Quigley, M., Snyder, D., Christiansen, R., Johnson, W., McLain, T., Goodrich, M.: Autonomous vehicle technologies for small fixed-wing UAVs. J. Aerosp. Comput. Inform. Commun. **2**(1), 92–108 (2005)
21. Alcázar, J., Cuesta, F., Ollero, A., Nogales, C., López-Pichaco, F.: Teleoperación de helicópteros para monitorización aérea en COMETS (in Spanish). XXIV Jornadas de Automática (JA 2003), León (Spain), 10–12 Septiembre 2003
22. Sørensen, C.F., Wu, M., Sivaharan, T., et al.: A context-aware middleware for applications in mobile AdHoc Environments. In: Proceedings of the 2nd Workshop on Middleware for Pervasive and Ad-hoc Computing Table of Contents. Toronto (2004)
23. Soetens, H., Koninckx, P.: The real-time motion control core of the Orocos project Bruyninckx, Robotics and Automation. In: Proceedings. ICRA '03. (2003)
24. Lam, T.M., Mulder, M., van Paassen, M.M.: Collision avoidance in UAV tele-operation with time delay, conference on systems, man and cybernetics. ISIC. IEEE International. Montreal, October 2007
25. Loh, R., Wullschleger, V., Elrod, B., Lage, M., Haas, F.: The U.S. wide-area augmentation system (WAAS). Journal Navigation **42**(3), 435–465 (1995)