Validating Knowledge-Based Framework through Mission-Oriented Sensors Array and Smart Sensor Protocol

Douglas Rodrigues, Rayner M Pires and Kalinka R L J C Branco
Institute of Mathematics and Computer Science, University of São Paulo, São Carlos, Brazil
E-mail: {douglasr, rayner, kalinka}@icmc.usp.br

Abstract. This paper addresses the problem of using Service-Oriented Architecture (SOA) in critical embedded systems, mainly in Unmanned Aerial Vehicles (UAVs). We present the use of a SOA approach to provide the integration of the payload in the UAV. The integration is provided by a plug and play protocol named Smart Sensor Protocol (SSP) that validates the SOA approach.

1. Introduction

Although the use of Service-Oriented Architecture (SOA) in the field of applications related to business and IT (Information Technology) is well established, several aspects should be considered in the field of critical embedded systems, especially for Unmanned Aerial Vehicles (UAVs). For instance, the availability of sufficient resources, such as processing power and memory capacity, is one of some issues to be considered. This class of systems can range from small to large, complex and multi-core systems. Therefore, in many of them, there are enough resources to be able to use Web services.

Then, we proposed the KBF (Knowledge-Based Framework for Dynamically Changing Applications) [1]. The KBF allows dynamic definition of Web services, depending on the current system configuration, the availability of Web services component, and a database with predefined context information. Specifically, the KBF uses both context and monitoring information to compose or generate services with different levels of reliability, security and performance, in a dynamic and intelligent way.

The MOSA (Mission-Oriented Sensor Array) [2], in turn, is a concept of sensors arrangement capable of performing a specific mission. The sensors to be used are determined according to the mission characteristics. Once these characteristics have been established, MOSA is then made representing a new specific sensor to that mission.

As described in [3], the SSP (Smart Sensor Protocol) is an application level protocol that was designed to rule the interaction between MOSA mission execution system and UAV system.

Thus, the architecture presented in this paper is designed to facilitate the communication between the MOSA and the KBF, which occurs using the SSP. The KBF is able to support the MOSA to perform missions autonomously, once KBF selects the service that best fits its system usage scenario, so the mission is performed in a more intelligent way.
2. Related works
Different implementations of UAVs are found in the literature and most of these implementations and developments typically use traditional approaches [4] [5]. The USAF roadmap [6] discusses and suggests the use of SOA in UAVs. This is a new approach, once these kind of systems have special requirements, like real-time performance.

There are also approaches that discuss the use of SOA in embedded systems [7] [8] [9]. Unlike those, the approach here presented discusses the use of SOA in a critical embedded system to achieve reusability, maintainability, interoperability, flexibility, and security using KBF and MOSA.

3. Communication between the KBF and the MOSA using the SSP
The operation of the KBF can be summarized as the enumerated steps illustrated in Figure 1. In this case, notice that the MOSA acts as the client.

(i) Initially, a service provider develops its service, describes its interface (e.g., WSDL) and publishes in a service registry (e.g., UDDI). After the publication, a classifier automatically classifies the service according to defined selection criteria (reliability, security and performance). This operation will be repeated for every new service published.

(ii) The resulting information from the classification are sent and added to the database.

(iii) A MOSA is connected to the UAV and sends a request to the broker, specifying necessary functionality features for defined mission.

(iv) The broker performs a search in the service registry, looking for an appropriate service, based on the features requested by the MOSA.

(v) Following the search, the broker gets a set of service interfaces with same functionality, changing only their selection criteria. i.e., among the services with same functionality, one of them may be more secure, another may have better performance, another may have greater reliability and so on.

Figure 1. Knowledge Based Framework for Dynamically Changing Applications (KBF).
(vi) With the obtained information from these interfaces, now the service selection and composition module queries the database to verify these services selection criteria, choosing the one that is the best for the situation, according to the defined mission.

(vii) Then the chosen service, its functionality and its selection criteria are related to the mission and sent to the reconfigurable matrix. As each mission is defined by a MOSA, the next time that same MOSA is connected, the service selection and composition module will perform a search on the reconfigurable matrix and return the service previously used, removing the need to repeat the search.

(viii) After performing this procedure, the broker finally sends the request to the chosen service.

(ix) After processing the request, the response is returned to the broker.

(x) Finally, the response is forwarded to the MOSA that initially requested the service.

In this case, MOSA is a client connected to the UAV, and every time that a MOSA board is connected to an UAV, it starts a series of requests to the aircraft system in order to decide about its qualification to accomplish a predefined mission stored in that MOSA (item iii illustrated in Figure 1).

Queries for the UAV characteristics are done in the first step of the SSP protocol, named Negotiation, when the $\text{REQ}$ primitive requests a predefined list of features. The default list of queries requests information related to the aircraft physical capabilities.

Once the MOSA is compatible with the KBF, it must try to discover KBF services in the first step. The second primitive, named $\text{SEND}$, was designed in [3] for this purpose. It is used to encapsulate data exchanged between client and provider of Web services. This exchange will happen not only in the Negotiation step, but also during the whole period of mission execution.

A software architecture describes the system components and the way they interact at a high level. In this application, the components are MOSA and KBF, and they interact via SSP messages, as illustrated in Figure 2.

![Figure 2. The MOSA-SSP-KBF software architecture.](image)

SOA is a design pattern in which application components provide services to other components via a communication protocol and, in this scenario, SSP is the appropriate protocol. Through primitives $\text{SEND}$ and $\text{ACK}$, both MOSA and KBF will exchange data for the SOA Find-Bind-Execute Paradigm.

In the Negotiation step, that primitive peer will be firstly used to detect KBF presence on UAV side. Once KBF announces its presence, both primitives $\text{SEND}$ and $\text{ACK}$ will encapsulate all messages for services discovery on KBF registry.

A non-exhausted example of services enumeration that is pertinent to this scenario and is in the KBF registry is listed in Table 1.

While the mission is being executed, KBF will eventually serve MOSA with its services. All data exchanged will be encapsulated over SSP messages again, by means of that last peer of primitives. The SSP protocol specification defines the format of $\text{SEND}$ and $\text{ACK}$ messages, when they are triggered, and how much data can be put in the message payload.

We simulated a mission using a MOSA board plugged in a computer (as an UAV) and exchanging information about the mission. The mission was implemented using a specific grammar [3] and was based on a real flight. This simulation using the MOSA board and the KBF is illustrated in Figure 3.
Table 1. Example of services that can be consumed by MOSA application.

| Service Description         | Static services | Dynamic services | Assembly | Autonomy | Status |
|-----------------------------|-----------------|------------------|----------|----------|--------|
| Inform cruising altitude    | Inform minimum stall speed | Inform/define current altitude | Inform/define current weight |
| Inform maximum weight capacity | Inform/define current weight | Inform/define current weight | Inform/define current weight |
| Inform cruising speed       | Inform maximum speed | Inform current GPS position | Inform current GPS position |

Figure 3. MOSA board (right), connected to an Arduino (left) that makes the interface with the computer.

4. Conclusions
This paper presents the validation of SOA in a specific critical embedded system (UAV). The communication between the UAV and MOSA was described. The results show that is possible to fly an UAV using SOA and a MOSA for a specific mission.

As future work, we expect to show a complete mission implemented using SOA and MOSA. The takeoff and landing and all the services necessary to complete a entire mission using SOA is the next step of this work.

Acknowledgments
The authors acknowledge the support granted by FAPESP, processes 2011/06086-9 and 2011/044161.

References
[1] Rodrigues D, Pires R M, Estrella J C, Vieira M, Correa M, Camargo J B, Branco K R L J C and Trindade O 2011 Communications in Computer and Information Science 206 345–354
[2] Rodrigues D, Pires R M, Estrella J C, Marconato E A, Trindade O and Branco K R L J C 2011 Proceedings of the 2011 IEEE International Conferences on Internet of Things (iThings), and Cyber, Physical and Social Computing (CPSCom) pp 733–738
[3] Pires R, Chaves A and Branco K 2014 ICUAS '14: International Conference on Unmanned Aircraft Systems pp 1300–1310
[4] DoD 2007 Unmanned systems roadmap 2007-2032 U.S. Department of Defense. Office of the Secretary of Defense
[5] DoD 2009 Unmanned systems integrated roadmap FY2009-2034 U.S. Department of Defense. Office of the Secretary of Defense
[6] USAF 2009 United states air force unmanned aircraft systems flight plan 2009-2047 Headquarters, United States Air Force
[7] Thramboulidis K C, Doukas G and Koumoutsos G 2008 EURASIP Journal on Embedded Systems - Embedded System Design in Intelligent Industrial Automation 2008 1–15 ISSN 1687-3955
[8] Lee M H, Yoo C J and Jang O B 2008 International Journal of Advanced Science and Technology 1 55–64 ISSN 2005-4238
[9] Moritz G, Prüter S, Timmermann D and Golatowski F 2008 ETFA '08: IEEE International Conference on Emerging Technologies and Factory Automation pp 432–435