Available online at www.sciencedirect.com

SciVerse ScienceDirect

Procedia Computer Science 00 (2018) 000–000

Procedia
Computer Science

www.elsevier.com/locate/procedia

The 7th International Workshop on Agent-based Mobility, Traffic and Transportation Models, Methodologies and Applications (ABMTRANS)

Comparison of Agent-based Simulation Frameworks for Unmanned Aerial Transportation Applications

Yazan Mualla[a,*], Wenshuai Bai[a], Stéphane Galland[a], Christophe Nicolle[b]

[a]*LE2I, Univ. Bourgogne Franche-Comté, UTBM, F-90010 Belfort, France*
[b]*LE2I, Univ. Bourgogne Franche-Comté, UB, F-21000 Dijon, France*

Abstract

Recently, the applications of Unmanned Aerial Vehicles (UAVs) in aerial transportation are gaining more interest. Due to operational costs, safety concerns and legal regulations, agent-based simulation frameworks are preferably used to implement models and conduct tests. With the abundance of such frameworks, this paper introduces a methodology to compare the most widely used frameworks. The methodology is based on the ISO software quality model, and uses a weighted sum scoring system to give points to the frameworks under study. The proposed criteria in the methodology consider agent-based simulation features and adapt specific features of unmanned aerial transportation. Preliminary comparison results and recommendations are provided and discussed.

Peer-review under responsibility of the Conference Program Chairs.

Keywords: Agent-based Simulation, Unmanned Aerial Vehicles, Software Quality Model, Frameworks Comparison

1. Introduction

Unmanned Aerial Vehicles (UAVs), most commonly known as drones, are becoming increasingly popular for civil applications in several domains such as surveillance, agricultural, transportation, products delivery, energy, disaster mitigation, environment preservation and infrastructure. Currently, a new era of civil UAVs that can autonomously fly outdoor and indoor is emerging. The features that make them interesting to use are their small dimensions, good manoeuvrability, ability to take-off and land vertically, simple mechanics and payload capability. The main drawback to mention is the high amount of energy consumed by these devices [1]. The future tendency is that UAVs will become more and more used, as new civil applications are developing in people's daily life in urban environments. According to Teal Group Corporation press held in Paris on 19 June 2017, France, the civil UAV promises to be the most dynamic growing sector of the world aerospace industry in the following years. The growing demand from governments and private consumers for civil UAVs have the potential to quadruplicate or more the civil UAV market over the next decade. Teal Group predicts that civil UAV production worldwide will reach $73.5 billion over the next decade.

* Corresponding author. Tel.: +0-000-000-0000; fax: +0-000-000-0000.
E-mail address: yazan.mualla@utbm.fr

1877-0509
Peer-review under responsibility of the Conference Program Chairs.

Moreover, autonomous flight presents great scientific and technical challenges related to the energetic cost of staying airborne and to the perceptual intelligence required to negotiate complex environments[2].

Research concerning the application of civil UAVs in transportation is advancing steadily[3] due to several advantages such as the high speed of movement in air compared to land and the overcoming of traffic jams in crowded areas. One mostly known example of the research in UAV transportation domain is the Amazon Prime Air[4,5] where UAVs are used to deliver packages. Another important example is the transportation of biological samples as, immediately after being collected, the biological samples need to be conveyed to the sites of testing which can be located at a very long distance from the collection site[6]. Even though there were some concerns about the safety of this procedure in such transportation environments, it was shown that UAV transportation systems are a viable option for the transportation of biological products[7].

However, to introduce UAVs in urban areas, there are some concerns such as the possible consequences, especially on people safety, of a mechanical failure that may cause a crash and the costs of such incident. To guarantee it is safe for UAVs to fly over people's heads and to reduce costs, different scenarios must be modeled and tested. However, some regulations restrict the use of UAVs in cities, so to perform tests with real UAVs, one needs access to expensive hardware and field tests that usually consume a considerable amount of time and require trained and skilled people to pilot and maintain the UAV. Moreover, on the field, it may also be hard to reproduce the same scenario several times[8]. In this context, the development of simulation frameworks that allow transferring real world scenarios into executable models using computer simulation frameworks, i.e. simulating UAVs activities in a digital environment, are mostly welcome.

Intelligent software agents have been established as a suitable technique for implementing autonomous behavior and decision-making in computer systems[9]. The use of agent-based simulation frameworks (frameworks for short in the rest of the paper) for UAVs is gaining more interest in complex civil application scenarios where coordination and cooperation are necessary, like for example in the study of the swarms formation of multiple UAVs[10].

The objective of this paper is to analyze the existing frameworks by selecting specific criteria to evaluate them based on UAVs civil application considerations. As a first look, many frameworks are hypothetically able to support the representation of moving objects in the space. As time, efforts and sometimes money are spent on deciding which framework should be used in a specific research task, it is vital to clarify the features of each framework, and what are the differences between them in order to choose the most suitable ones for this task. The general purpose of this work is to find the best frameworks to model and simulate moving objects in the space in general and UAVs in specific. The proposed comparison methodology has 4 phases. First, list and provide the general features of the candidate frameworks. Second, define comparison criteria related to the UAVs civil applications as there are no predefined comparison criteria in the existing literature in the domain of UAVs simulation according to our knowledge. Third, compare the frameworks as per the defined criteria. Fourth, discuss the results found and provide recommendations. This work does not consider for now the run-time tests in the comparison. Seven well-known frameworks have been considered for this work: Gazebo[11], AirSim[12], Janus with the Jasim plugin[13], Repast Simphony[14], Flame[15], NetLogo[16] and JADE[17].

The rest of this paper is organized as follows: Section 2 discusses the related work. The main work is provided in Section 3. First, the software quality models are discussed in Section 3.1. Second, the general features of the frameworks under study are provided in Section 3.2. Third, the proposed ranking criteria are discussed in details in Section 3.3. Fourth, the comparison itself with the results and the discussions are provided in Section 3.4. Section 4 concludes this work and identifies future research perspectives.

2. Related Work

The comparison of agent-based simulation frameworks has been considered in the literature by some framework surveys. Some works focused on one application domain when comparing the frameworks, like for example energy consumption market applications[18], geospatial applications[19], or parallel and distributed applications[20]. However, there is no survey that focused on flying objects or UAVs in the transportation domain as per our knowledge. Other surveys have been conducted but are currently outdated[21,22].

Railsback et al.[23] implemented a simple scenario with 100 agents randomly moving on a small grid (100x100 cells) for measuring the performance of simulation frameworks. Their results were mostly limited to their perception and experience when implementing their model. Abar et al.[24] presented a comprehensive comparative survey of 85 frameworks. However, the comparison criteria included only general features, and some frameworks were not included like Gazebo and AirSim which we find related to UAV applications. Lorig et al.[8] provided a technical comparison of 4 frameworks, but they focused only on one metric which is scalability. In this paper, a wider comparison in terms of criteria is defined, with focusing on features related to UAV applications like physics and environment.

3. Comparison of Frameworks

3.1. Software Quality Model

In order to compare two frameworks, there is a need to measure the quality model of each framework. In the literature of software comparison, there is no agreed methodology of assessing software quality that is widely accepted. Several efforts have been conducted by researchers to define a model for software comparison. Behkamal et al.[25] gathered most of the widely used models as follows:

(i) McCall Model[26]: The main idea is to establish a relationship between quality features and metrics. Nonetheless, metrics are not necessarily objective in this model.

(ii) FURPS Model[27]: It organizes features in two different categories of requirements: Functional requirements defined by the input and the predicted output, and Non-functional requirements which are usability, reliability, performance and supportability. However, this model does not take into consideration some features like the portability of software products.

(iii) Dromey Model[28]: It seeks to enhance the relationship between the features and the sub-features of software quality. As a disadvantage in the model, the criteria for measurement of software quality is missing.

(iv) Bayesian belief network (BBN) Model[29]: it is organized as a hierarchical structure. The root of the tree is the node 'Quality' connected to quality features nodes, and each node is connected to corresponding quality sub-features. Using this model, complex models can be designed that is difficult to design using other models.

(v) ISO Model[30]: The software product quality criteria are classified in a hierarchical tree structure of 6 features, that are further classified into 21 sub-features. This model provides the most wide range of comparison criteria in different aspects.

The ISO model is mainly adopted in this work as it includes several interesting criteria in the context of the proposed comparison like for example: 'Interoperability', 'Compliance', Understandability, 'Learnability', 'Operability', 'Adaptability', 'Installability', etc. However, some criteria are excluded like 'Efficiency' as run-time comparison tests have not been done in this work. Moreover, some other criteria[31] related to software development process metrics were added. Furthermore, we have introduced some extra criteria that are relatively important in the UAVs domain like the representation of gravity and magnetic fields, and the support of forced-based motion. The focus is on the criteria that are mostly related and associated with flying objects in the space and their environment.

3.2. Frameworks General Features

Table 1 provides the general features of the frameworks under study with the following abbreviations: Gazebo (GB), AirSim (AS), Janus with the Jasim plugin(JJ), Repast Simphony (RP), Flame (FL), NetLogo (NL) and JADE (JD). The consideration of project life span in the table is to analyze if the project that developed the software is still active, and at which rate it has been updated. One suggested way is to verify the time interval between updates or new versions released: Activity = Number of versions released ÷ Years of existence of the project.

3.3. Ranking criteria

The evaluation of the simulation frameworks is divided into 4 categories, with each category having different criteria. Simple ranking by giving ranks to the frameworks is impractical, as there is a need to quantify the level each framework is achieving a criterion. Therefore, a comparison system based on points (or scores) is used. S_j represents the set of all points given to the options that a criterion j can take, where $S_j \subset \mathbb{N}$ and $|S_j| \geq 2$. For example, if a criterion j has three options: Option1 (0p), Option2 (1p) and Option3 (2p), then $S_j = \{0, 1, 2\}$

Table 1. Frameworks general features

	GB	AS	JJ	RP	NL	FL	JD
Main domain	Robots	UAVs	General	General	General	General	General
License	Apache 2.0	MIT	Apache 2.0 (Janus), proprietary license (Jasim)	BSD	GPL	GNU, Academic license	LGPL v2
Open source	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Project life span	12.7	10	1.9	1.7	2.625	1.6	1.8
Programming languages	JavaScript	C++, Python, C#, Java	SARL, Java, JavaScript	Java, Python, C++, J#, Visual Basic, .Net, C#	NetLogo	C	Java
Operating systems	Linux, Mac Windows	Linux, Windows	Linux, Unix, Mac, Windows, Android	Linux, Mac Windows	Linux, Mac Windows	Linux, Mac Windows	Linux, Mac Windows

The criteria per category with the distribution of points are listed in the following:

A. *System Features:* The main system features of the framework.

1. Agent architecture: Support of either reactive or Belief–Desire–Intention (BDI) agent architecture (1p); Hybrid support of reactive and BDI architectures (2p).
2. Support the communication between agents: No communication (0p); Communication through the environment i.e. Stigmergy (1p); Stigmergy and direct communication between agents (2p).
3. Availability of template or example agent: No (0p); Yes (1p).
4. Support of sensors: No (0p); Yes (1p).
5. Support of Geographic Information System (GIS): No (0p); Yes (1p).

B. *Operation and Execution:* The main operational and executional features of the framework.

1. Installation ease: The installation method and the required IT skills to install. The simpler the method for the user is considered the better as follows: Command line install (0p); GUI installer (1p).
2. Operational ease: The support of features like Integrated Development Environment (IDE), command prompt and click & point. The evaluation is as follows: No IDE, and just a program compiler (0p); IDE with syntax coloring, and a dedicated compiler (1p); IDE with syntax coloring and helping features, such as auto-completion (2p); IDE with helping features, creation wizards and automatic code generation (3p).
3. Interaction with objects in the air (Human Controlled Aircrafts) and the ground: No interaction (0p); Interaction with objects in the air or the ground (1p); Interaction with objects in the air and the ground (2p).

C. *Environment:* The features of the reproduced environment and how the UAV interacts with it.

1. Environmental model: No model (0p); One type of models, either 1d, 2d or 3d (1p); Two types of models (2p); Three types of models (3p).
2. Support of dynamic environment objects: The ability to reproduce the objects that can be found on a city environment like traffic lights or objects connected to the Internet. The evaluation is as per the support of those objects as follows: No support (0p); Simple dynamic environment objects with the algorithm being part of the simulator (1p); Complex dynamic environment objects with the support of co-simulation (2p).
3. Simulation of environmental dynamics: The ability to simulate environmental dynamics like wind and rain of different intensities, different visibility conditions etc. The evaluation is as follows: No support (0p); Simple simulation of environmental dynamics (1p); Complex simulation of environmental dynamics (2p).

D. *Physics:* The simulation of the laws of physics in a realistic manner, and how well the exact position of an object in the space is determined.

1. Representation of gravity field: No (0p); Yes (1p).
2. Representation of magnetic field: No (0p); Yes (1p).
3. Simulation of collision: No (0p); Yes (1p).
4. Support of forced-based motion: No (0p); Yes (1p).

To normalize the importance of the proposed criteria (i.e. to make sure all the criteria have the same impact on the total score, the concept of weighted sum is introduced by applying weights to criteria, as the maximum score of one criterion may be different than the maximum score of another criterion. Equation 1 defines the normalized weight for each criterion. For every framework, the total weighted score is calculated as per Equation 2.

$$w_j = \frac{1}{\max(S_j)} \tag{1}$$

$$T_i = \sum_{j=1}^{C} w_j . s_{ij} \tag{2}$$

Where: w_j is the weight of criterion j; $\max(S_j)$ is the maximum score of criterion j; T_i is the total weighted score of framework i; C is the total number of criteria; s_{ij} is the score of framework i for criterion j.

Table 2. Frameworks comparison

		GB	AS	JJ	RP	FL	NL	JD	$\max(S_j)$
A	Agent architecture	2	2	2	2	2	2	2	/2
A	Communication between agents	1	1	2	1	2	0	2	/2
A	Example agents	1	1	1	1	1	1	1	/1
A	Sensors	1	1	1	1	1	0	1	/1
A	GIS	1	1	1	1	1	1	0	/1
	System Features Weighted Score	**4.50**	**4.50**	**5.00**	**4.50**	**5.00**	**3.00**	**4.00**	**/5.00**
B	Installation ease	0	0	1	1	0	1	0	/1
B	Operational ease	2	2	2	3	3	3	2	/3
B	Interaction with objects (air and ground)	2	2	2	2	2	2	2	/2
	Operation and Execution Weighted Score	**1.67**	**1.67**	**2.67**	**3.00**	**2.00**	**3.00**	**1.67**	**/3.00**
C	Environmental model	1	1	3	2	2	2	0	/3
C	Dynamic environment objects	2	2	2	2	1	1	2	/2
C	Environmental dynamics	2	2	1	1	1	1	0	/2
	Environment Weighted Score	**2.33**	**2.33**	**2.50**	**2.17**	**1.67**	**1.67**	**1.00**	**/3.00**
D	Gravity	1	1	1	0	1	1	1	/1
D	Magnetic	1	1	0	0	0	1	1	/1
D	Collision	1	1	1	1	1	1	1	/1
D	Forced-based motion	1	1	0	1	1	0	0	/1
	Physics Weighted Score	**4.00**	**4.00**	**2.00**	**2.00**	**3.00**	**3.00**	**3.00**	**/4.00**
	TOTAL WEIGHTED SCORE	**12.50**	**12.50**	**12.17**	**11.67**	**11.67**	**10.67**	**9.67**	**/15.00**

3.4. Results and Discussions

The comparison results of all simulation frameworks are shown in Table 2. The last column represents the maximum score a framework can achieve for a specific criterion. For each category, the weighted scores of all frameworks are calculated, and the total weighted scores are provided at the end. It can be noticed from the table that Gazebo and AirSim score the best total weighted score with 12.50/15.00 to be the best two frameworks for the agent-based simulation of UAVs applications in general. However, they are not the best frameworks in all categories considered individually. For instance, and considering only the operation and execution category, the best two frameworks are Repast Simphony and NetLogo, with the best runner up to be Janus. Additionally, it is noted from the table that the total weighted score of JADE is low compared to other frameworks, which makes it the last option in our opinion when it comes to UAVs simulation.

It is worth mentioning that although the comparison is prepared as per the considerations of UAVs, the results can be generalized to include other flying objects in the space.

4. Conclusions and Perspectives

In this paper, a methodology to compare agent-based simulation frameworks is defined focusing on features of unmanned aerial transportation applications. The preliminary results show that Gazebo and AirSim are the most suitable frameworks for simulating UAVs applications with slightly a small better score than their runner up (Janus with the Jasim plugin). As a future work, a detailed run-time comparison of these frameworks will be conducted considering scalability in both abstraction and detailed levels. Moreover, some specific application criteria are to be considered, e.g. drone patrolling, drone delivery systems, that may affect the weights. Therefore, there will be a need for a good methodology for determining the weights depending on the specific application domain. This will enhance and refine the preliminary results of this paper. Furthermore, the subjective criteria will be considered by establishing frameworks tests conducted by several groups of users, and preparing questionnaires to acquire results.

Acknowledgements: This work is supported by the Région Bourgogne Franche-Comté (RBFC, France) project UrbanFly 20174-06234/06242, and by China Scholarship Council Grant 201701810074.

References

1. Rodić, A., Mester, G.. Modeling and simulation of quad-rotor dynamics and spatial navigation. In: *Intelligent Systems and Informatics (SISY), 2011 IEEE 9th International Symposium on*. IEEE; 2011, p. 23–28.
2. Floreano, D., Wood, R.J.. Science, technology and the future of small autonomous drones. *Nature* 2015;**521**(7553):460–466.
3. George, A.. Forget roads, drones are the future of goods transport. *New Scientist* 2013;**219**(2933):27.
4. Bamburry, D.. Drones: Designed for product delivery. *Design Management Review* 2015;**26**(1):40–48.
5. D'Andrea, R.. Guest editorial can drones deliver? *IEEE Transactions on Automation Science and Engineering* 2014;**11**(3):647–648.
6. Lippi, G., Mattiuzzi, C.. Biological samples transportation by drones: ready for prime time? *Annals of translational medicine* 2016;**4**(5).
7. Amukele, T., Ness, P.M., Tobian, A.A., Boyd, J., Street, J.. Drone transportation of blood products. *Transfusion* 2017;**57**(3):582–588.
8. Lorig, F., Dammenhayn, N., Müller, D.J., Timm, I.J.. Measuring and comparing scalability of agent-based simulation frameworks. In: *German Conference on Multiagent System Technologies*. Springer; 2015, p. 42–60.
9. Wooldridge, M., Jennings, N.R.. Intelligent agents: Theory and practice. *The knowledge engineering review* 1995;**10**(2):115–152.
10. Bürkle, A.. Collaborating miniature drones for surveillance and reconnaissance. In: *Proc. of SPIE Vol*; vol. 7480. 2009, p. 74800H–1.
11. Koenig, N., Howard, A.. Design and use paradigms for gazebo, an open-source multi-robot simulator. In: *Intelligent Robots and Systems, 2004.(IROS 2004). Proceedings. 2004 IEEE/RSJ International Conference on*; vol. 3. IEEE; 2004, p. 2149–2154.
12. Shah, S., Dey, D., Lovett, C., Kapoor, A.. Airsim: High-fidelity visual and physical simulation for autonomous vehicles. In: *Field and Service Robotics*. 2017, URL: `https://arxiv.org/abs/1705.05065`. `arXiv:arXiv:1705.05065`.
13. Galland, S., Gaud, N., Demange, J., Koukam, A.. Environment model for multiagent-based simulation of 3d urban systems. In: *the 7th European Workshop on Multiagent Systems (EUMAS09), Ayia Napa, Cyprus*. 2009, .
14. North, M.J., Howe, T.R., Collier, N.T., Vos, J.R.. The repast simphony runtime system. In: *Proceedings of the agent 2005 conference on generative social processes, models, and mechanisms*; vol. 10. ANL/DIS-06-1, co-sponsored by Argonne National Laboratory and The University of Chicago; 2005, p. 13–15.
15. Kiran, M., Richmond, P., Holcombe, M., Chin, L.S., Worth, D., Greenough, C.. Flame: simulating large populations of agents on parallel hardware architectures. In: *Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems: volume 1-Volume 1*. International Foundation for Autonomous Agents and Multiagent Systems; 2010, p. 1633–1636.
16. Wilensky, U., Evanston, I.. Netlogo: Center for connected learning and computer-based modeling. *Northwestern University, Evanston, IL* 1999;**4952**.
17. Bellifemine, F.L., Caire, G., Greenwood, D.. *Developing multi-agent systems with JADE*; vol. 7. John Wiley & Sons; 2007.
18. Zhou, Z., Chan, W.K.V., Chow, J.H.. Agent-based simulation of electricity markets: a survey of tools. *Artificial Intelligence Review* 2007; **28**(4):305–342.
19. Castle, C.J., Crooks, A.T.. Principles and concepts of agent-based modelling for developing geospatial simulations. Centre for Advanced Spatial Analysis (UCL), UCL (University College London), Centre for Advanced Spatial Analysis (UCL); 2006.
20. Rousset, A., Herrmann, B., Lang, C., Philippe, L.. A survey on parallel and distributed multi-agent systems. In: *Padabs 2014, 2nd Workshop on Parallel and Distributed Agent-Based Simulations, in conjunction with Euro-Par 2014*; vol. 8805. Springer; 2014, p. 371–382.
21. Serenko, A., Detlor, B., et al. Agent toolkits: A general overview of the market and an assessment of instructor satisfaction with utilizing toolkits in the classroom. *Research and working paper series of the Michael G DeGroote School of Business* 2002;(455):43.
22. Allan, R.J.. *Survey of agent based modelling and simulation tools*. Science & Technology Facilities Council; 2010.
23. Railsback, S.F., Lytinen, S.L., Jackson, S.K.. Agent-based simulation platforms: Review and development recommendations. *Simulation* 2006;**82**(9):609–623.
24. Abar, S., Theodoropoulos, G.K., Lemarinier, P., O'Hare, G.M.. Agent based modelling and simulation tools: A review of the state-of-art software. *Computer Science Review* 2017;.
25. Behkamal, B., Kahani, M., Akbari, M.K.. Customizing iso 9126 quality model for evaluation of b2b applications. *Information and software technology* 2009;**51**(3):599–609.
26. Fitzpatrick, R.. Software quality: definitions and strategic issues. *Reports* 1996;:1.
27. Khosravi, K., Guéhéneuc, Y.G.. A quality model for design patterns. *University of Montreal, Tech Rep* 2004;.
28. Dromey, R.G.. A model for software product quality. *IEEE Transactions on Software Engineering* 1995;**21**(2):146–162.
29. Stefani, A., Stavrinoudis, D., Xenos, M.. Experimental based tool calibration used for assessing the quality of e-commerce systems. In: *e-Business and Telecommunication Networks*. Springer; 2006, p. 100–106.
30. for Standardization, I.O., Commission, I.E.. *Software Engineering–Product Quality: Quality model*; vol. 1. ISO/IEC; 2001.
31. Kan, S.H.. *Metrics and models in software quality engineering*. Addison-Wesley Longman Publishing Co., Inc.; 2002.