GLOBUS

Globus An International Journal of Medical Science, Engineering and Technology
A Refereed Research Journal
Vol 9 / No 1 / Jan-Jun 2020 ISSN: 2319-8958

EMBEDDING ROBOTICS INTO THE DEFENCE ARCHITECTURE

*Varun Gupta
**Dr Sanjay Kumar

Paper Received: 04.01.2020 / **Paper Accepted**: 09.02.2020 / **Paper Published**: 10.02.2020
Corresponding Author: Varun Gupta; doi:10.46360/globus.met.320201002

Abstract

Robot architectures could be thought of as software engineering in some ways. The unique requirements of robot systems, however, set robot designs apart from other software architectures. From an architectural standpoint, the most crucial of them is that robot systems have to communicate asynchronously, in actual time, with a variable, frequently dynamic environment. Numerous robot systems must also react in a variety of time frames, from millisecond feedback controls to minutes or hours for complicated jobs. Many robot architectures have the ability to act in real time, control actuators as well as sensors, support concurrency, recognise and respond to exceptional circumstances, deal with uncertainty, and integrate a high-level (symbolic) preparing via low-level (numerical) control in order to meet these requirements.

Keywords: Robotics, Defence, Architecture, Software.

Introduction

There can be advantages to utilising one particular architectural style over another, even though the same capability is frequently possible using several architectural styles. Think about how a robot system's communication style may affect its dependability as an illustration. Message passing is the primary form of communication used by many robot systems. Client-server is a common kind of communication where a message delivery by the client is followed by a reply from the server.

Publish-subscribe is an alternative communication model in which information are published asynchronously and any modules that have expressed an earlier interest in receiving them are sent a copy. Modules often send a request when using client-server message passing, then block while awaiting the response. Deadlock can happen if the response rarely materialises (for instance, if the server module fails). The control flow still commonly anticipates a response even if the module doesn't block, which could provide unanticipated outcomes if the response never comes or if a response to another request happens to come in before it. Conversely, publish-subscribe systems tend to be more dependable since the control flow normally does not presume any specific sequence whereby messages are processed. As a result, missing and out-of-order messages typically have less of an impact.

Review of Literature

Darko (2019) [1] Research into robots is advancing significantly. The following issues, however, restrict the employment of present technologies in our daily tasks:

(i) robots work in uncharted surroundings;
(ii) robots cooperate with one another and even with humans; and
(iii) robots must never cause harm to individuals or property.

The research landscape is fractured as a result of the different angles from which those problems are being approached by researchers. We want to present a thorough and repeatable overview of the state for the art in managing risk with mobile robotic systems from the standpoint of the software

* *Research Scholar, Kalinga University, Naya Raipur, India.*
***Supervisor, Kalinga University, Naya Raipur, India.*

engineering. We chose 58 primary studies and analysed them using a systematically defined classification framework on an initial set of 1274 possibly relevant research publications. This is known as the systematic mapping methodology. This study makes four contributions:

(i) a system of classifications for managing safety when interacting with the software for mobile robotic systems (MSRs);
(ii) a map of present-day techniques and approaches for software safety for MRSs;
(iii) an explanation of new challenges as well as implications for future research; as well as
(iv) a replication put for independent replication as well as verification of this study. Our findings support the notion that the majority of technologies are not yet suitable for usage in routine situations. There is a need for turn-key solutions that are prepared to handle all of the aforementioned difficulties.

Estefo (2015) [3] the robotics middleware The de facto standard for creating robot apps is ROS. We hypothesised that ROS's low-quality code results in a bad user experience for users and poses significant obstacles to the development of robot software based on our experience using ROS. Through a study of duplicate code in launch files, we give a preliminary estimate of the code quality within the ROS ecosystem in this paper. Our previous work suggested that there is quite a bit of code duplication in these configuration files, and this investigation confirms this. We discover that clones are quite comparable and that twenty-five percent of packages that have numerous launch files contain duplicate code.

Analysis

To accomplish interoperability, robotics can be included into defence architecture. This has a number of benefits, including better situational awareness, more effective operations, and safer soldiers. The following are some crucial ideas and tactics for completing this integration:

Protocols for standardisation and interoperability must be established in order to guarantee flawless communication and coordination between various robotic systems. To achieve interoperability between multiple robots, sensors, and command systems, it is necessary to define interfaces, data formats, and communication protocols.

Unified Command and Control: Putting in place a centralised command and control system makes it possible to coordinate and manage a number of robotic assets. Across the robotic fleet, this system should offer a uniform interface that unifies data from various sensors and enables effective task assignment, monitoring, and decision-making.

Defence robotics frequently rely on different sensors, including cameras, LIDAR, radar, and other specialised equipment. Sensor fusion and data integration. A thorough and precise understanding of the environment can be obtained by successfully integrating sensor data using sensor fusion techniques. The outcomes of missions can then be improved by using the combined data.

Collaboration and coordination are promoted when collaborative autonomy is allowed between robotic systems. This entails creating algorithms and protocols that enable robots to exchange information, organise and carry out cooperative missions, and adjust to changing circumstances. Task distribution, sensor sharing, and cooperative behaviours are made easier by collaborative autonomy, which improves the robotic fleet's overall efficacy.

Communication Infrastructure: To ensure seamless connectivity between robots, command centres, and other military assets, a strong and secure communication infrastructure must be established. To guarantee data security and integrity, this infrastructure should support real-time data transfer, have enough bandwidth, and include encryption and authentication techniques.

Training and Human-Robot Interaction: To successfully integrate robotics into defence architecture, operators and soldiers must receive the necessary training. Interfaces and protocols for human-robot interaction should be simple to use so that staff may easily operate, observe, and gather data from robots.

Testing and assessment: To guarantee the dependability, efficiency, and safety of integrated robotic systems, rigorous testing and assessment are essential. To find flaws, hone algorithms, and enhance system performance, realistic field trials, simulation exercises, and operational evaluations are conducted.

Scalability and Modularity: Considering scalability and modularity while designing robotic systems enables growth and the incorporation of new technologies in the future. Robotic components can be added to or replaced using modular architectures, and the robotic fleet can expand without major redesign thanks to scalable designs.

Security in networks: Given our reliance on networked systems, cybersecurity is crucial. Defence robotics are protected from cyber attacks by implementing strong cybersecurity measures, such as intrusion detection systems, encryption, secure authentication methods, and regular system updates.

Integrating robotics into defence architecture necessitates taking into account ethical and legal issues. This entails developing accountability systems, limiting autonomous decision-making, defining rules of engagement, and analysing the moral ramifications of utilising lethal force through autonomous systems.

Defence organisations may successfully incorporate robots into their architecture while attaining interoperability, boosting operational capabilities, and maximising the advantages of robotic systems in defence situations by carefully weighing these variables and putting relevant strategies into place.

Conclusion

Robotic software systems can be integrated and interoperable with other robotic systems within the defence domain by utilising the Department of Defence (DoD) High Level Architecture (HLA). The DoD created the HLA standards and guidelines for distributed simulation systems in order to facilitate simulation-based learning and acquisition. The following procedures would need to be taken into consideration in order to apply the HLA to robotics software systems:

Identify the objects, attributes, and interactions that the robotic system needs to model by defining the object model. Determining the robot entities, sensors, actuators, and other pertinent parts is part of this process.

Constructing the Federation Determine the various robotic systems that will take part in the federation and the accompanying software components. Define the protocols, data exchanges, and interfaces that will be utilised for system-to-system communication.

Create the Federates: Put into practise the federates, which are the software elements that stand in for the many robotic systems that make up the federation. Each federate shall abide by the HLA requirements and recommendations for information sharing, synchronisation, and communication.

Define interaction class libraries and object class libraries: Make libraries that outline the interaction and object classes that are utilised within the federation. The features, behaviours, and data types essential for optimal communication and interaction amongst robotic systems should be included in these libraries.

Conflict of Interest

There is no conflict of interest between the authors in this manuscript.

References

1. Bozhinoski, Darko, Davide Di Ruscio, Ivano Malavolta, Patrizio Pelliccione, Ivica Crnkovic (2019). Safety for mobile robotic systems: A systematic mapping study from a software engineering perspective. Journal of Systems and Software, 151, 150-179.
2. Gajbhiye, B. N. and Prasad, V. Kamakshi (2013). Model for Defence R&D Scientific and Engineering Software Development. Lecture Notes on Software Engineering, 1(3), 319-322.
3. Estefo, P., Robbes, R. and Fabry, J. (2015). Code duplication in ROS launchfiles. 2015 34th International Conference of the Chilean Computer Science Society (SCCC), Santiago, Chile, 1-6, doi: 10.1109/SCCC.2015.7416575, 2015.
4. Maguri, Dr. Ramesh (2015). A Quick Review on Cloud Computing and Related Security Issues. Cosmos An International Journal of Management, 4(2), 1-4.
5. Singh, Navdeep (2014). A Study on Cooperative Defense Against Network Attacks. Cosmos Journal of Engineering & Technology, 4(2), 1-4.
6. Kumar, Praveen (2014). A Study on Cloud Computing. Cosmos Journal of Engineering & Technology, 4(2), 1-3.
7. Anuradha (2015). Study in Technological Challenges in Digital Libraries. Cosmos An International Journal of Art & Higher Education, 4(2), 9-11.
8. Goel, Agarwal, Nidhi (2008). A Global Change in Education through Information Technology and Communication. Enterprises Information Systems & Technology, Mac Millan Advanced Research Series, ISBN: 13: 978-0230-63516-6, pp. 124-126.
9. Mishra, Dr. Pramod and Sharma, Pradeep Kumar (2018). Digital Literacy Competencies in The 21st Century. Globus Journal of Progressive Education, 8(2), 1-3.
10. Kumar, Puneet (2008). A Comparative Study of Information System's Security by using Graphs. Enterprise Information Systems & Technology, MacMillan India Ltd., 222-227, ISBN 0230-63516-4.
11. Vashishtha, Dr. Sangeet and Sharma, Pooja (2018). Big Data- New Trend of Change in Complex Corporate World.

Globus An International Journal of Management & IT, 10(1), 4-6.
12. DeMarinis, N., Tellex, S., Kemerlis, V., Konidaris, G. & Fonseca, R. (2018). Scanning the internet for ROS: A view of security in robotics research. arXiv preprint arXiv:1808.03322.
13. Dieber, B., Breiling, B., Taurer, S., Kacianka, S., Rass, S. & Schartner, P. (2017). Security for the robot operating system. Rob. Auton. Syst., 98, 192-203.