IWSCFF 19-74

KNOWLEDGE-BASED SELF-RECONFIGURATION AND SELF-AWARE DEMONSTRATION FOR MODULAR SATELLITE ASSEMBLY

Mark Post[*] and Jim Austin[†]

In this research, we aim to develop a knowledge-based self-reconfiguring laboratory demonstrator made up of cubic modules for validating self-reconfiguration strategies for modular satellites. Modular satellite technologies have the potential to reduce the costs and complexity of on-orbit assembly and servicing operations, as well as reducing waste for end-of-life satellites and required mass to orbit. Modules are sized to be 10cm cubes comparable to the 1U CubeSat form factor, and can connect and communicate at will to form larger structures. Each module can carry different payloads and perform different functions, while retaining common abilities to execute software and store information. Laboratory testing of self-reconfiguration strategies for a modular structure focuses on a key challenge essential to reconfigurable satellites: the ability to autonomously determine and achieve appropriate configurations of structures in orbit without human input. We apply knowledge-based inference methods to determining a suitable configuration of modules for a modular satellite and reasoning methods for planning the sequence of actions required to implement this configuration. In addition to satellites and space structures, self-assembling hardware represents a platform that could potentially be used to build many kinds of terrestrial robots and autonomous systems.

INTRODUCTION

The assembly and reconfiguration of the International Space Station uses a robotic arm and automated docking with human oversight, but most satellites must be operated and repaired as single entities at great cost by human intervention. In a modular satellite architecture, smaller parts of spacecraft can be delivered using multiple launches or as needed for maintenance and upgrading activities, and assembled robotically into a flexible geometry that is not required to fit into a launch vehicle. Telescopes, communications satellites, and exploration spacecraft can all be assembled from self-contained component parts in space rather than being limited by single large payload capacities and complex deployment systems, and can also repair themselves by changing the mod ules used. The overlapping core robotics technologies needed to address the challenges of modular satellites include sensing and perception, GNC, microgravity mobility and mobile manipulation, autonomy and intelligence.[1] Autonomous self-assembly of free-flying modules is a very difficult dynamic problem in addition to adding mass and control overheads to each module, and at present the use of a robotic manipulators assumed for physical placement of the modules. Modular spacecraft configuration must use autonomous reasoning as communication delays and eclipses prevent reliable communication with humans.[2] The self-configuration process is driven by requirements

[*]Lecturer, Electronic Engineering,University of York, Heslington, York, YO10 5DD.
[†]Professor, Computer Science, University of York, Helsington, York, YO10 5GH.

for positioning specific modules on the satellite, the dynamics and structure, module power and communication equirements, and self-healing from faults. Machine learning can adjust the control parameters to compensate for the space environment. We apply a semantic reasoning process to the self-configuration problem as it allows the system to easily adapt itself to the addition of new hardware and software that was not originally programmed or designed for. Internal knowledge is provided by Ontology Web Language (OWL) files in XML format, which is parsed and reasoned over by a reasoning engine. The optimal configuration of modules is determined based on both module-level and system-wide information, and a plan is constructed as to how to move modules to their final positions. The reasoning and planning methods are considered together because there are elements of planning that affect reasoning (e.g. the necessity of moving hard-to-reach modules versus easier-to-reach ones).

BACKGROUND

Fractionation and Autonomic Computing for Robotics

One of the first steps toward fractionated architectures in general computing was made with the definition of Autonomic Computing. The concept of Autonomic Computing is central to dealing with complexity in large-scale systems and has been conceptually present for over 15 years, mainly focused on self-management of complex Information Technology systems such as corporate networks as addressed in IBM papers[3].[4] As envisioned, an autonomic system must "know itself" sufficiently well to embody four characteristics:

- Self-configuration: systems can install functions and set parameters following high-level rules
- Self-optimization: parameters can be tuned and performance improved autonomously over time
- Self-healing: localized software and hardware problems can be detected and repaired/bypassed
- Self-protection: large cascade failures or malicious attacks can be quickly identified and defended against

Being features of multiple, networked components, these aspects drive the design and behaviour of so-called Autonomic Elements, which function as fractionated components capable of managing themselves and their tasks within the system. Funabashi et al. predicts that by 2025, systems will need to learn and self modify in an automated fashion.[5] Other visions of how self-managing systems have been similarly formulated to address complexity and reliability such as that of Organic Computing, which focuses on the effects of emergent behaviours in massively connected self-organizing systems.[6] Another concept of computing related by the goals of system fractionability and complexity minimization is that of Wide Computing, in which concurrent tasks are spread horizontally across a network of visible, connected elements in a peer-to-peer fashion.[7] In the broad context of autonomic systems, four additional characteristics are often cited:[8]

- Self-awareness: the system is aware of its state, capabilities, and topology
- Context-awareness: the system is aware of its configuration, its goals, and its environment
- Open: localized software and hardware problems can be detected and repaired/bypassed

- Anticipatory: large cascade failures or malicious attacks can be quickly identified and defended against

These basic characteristics are used to define how a fractionated system should be designed. In order to provide self-awareness in its various forms such that the the other "self-*" characteristics are possible, it is necessary to make available all relevant information about the system in some sort of organized, machine-parseable format that can be used to perform autonomic tasks. This is where the need for semantic ontologies becomes central to autonomic systems. This paper focuses on the characteristics of self-configuration and self-healing as well as self-awareness.

Software and hardware architectures for robotics have gradually experienced a trend toward decentralization and fractionation. This is exemplified by the design of robotic middleware in recent years, which began with monolithic robotic software stacks and has evolved from server-based systems such as Player[9] through networked systems such as ROS[10] and now is focused on entirely decentralized software networks such as YARP[11] and ROCK.[12] The frequent interfacing and porting of software between these systems is testament to the value of being able to build heterogeneous systems that can interact seamlessly. Autonomic computing has been applied to the design of robots primarily as a means of fault tolerance but with the fractionation benefits of better modularity and flexibility as well.[13] Autonomic computing has been proposed for use in underwater vehicles as a means of applying a bio-inspired methodology of robotic self-management based on the human autonomic nervous system.[14] This work has been focused entirely on designing autonomic behaviours for robots, and does not address the problems of fractionation or reconfiguration at the system level. Other experimental improvements in autonomic-style self-management such as the ROS-based RoSHA multi-robot self-healing architecture[15] have been developed. However, these self-management features are implemented as add-ons such as failure diagnosis, coordination, and reconfiguration to the basic middleware and do not address the fractionation challenges in terms of underlying system design, which is ultimately necessary to failures in the underlying middleware and communications strategy.

In the field of aerospace robotics, similar trends are very evident as the use of fractionated architecture has become a subject of great interest. To increase reliability while minimizing complexity and retaining the advantages of a distributed architecture, Integrated Modular Avionics (IMA) architectures focus on the networking of standardized Line-Replaceable Units (LRUs) to perform a wide range of tasks in many different locations on a vehicle.[16] While static IMA networks designed and configured manually have already been successful in reducing complexity, power and mass, they do not realize the full potential of a fractionated system. The use of configuration-free "Plug & Fly" avionics[17] requires a common description of what each component's capabilities and connections are so that the system is sufficiently "self-aware" to perform this design and configuration autonomously given a set of basic rules. Fractionated architectures have a similar role in space robotic systems, promising improved responsiveness and tolerance to uncertainty, though qualification and use of fractionated systems in space is even more difficult due to the field being costly and highly risk-averse.[18] Autonomic computing concepts have been proposed as valuable for use in space[19] but largely for reasons of risk and complexity, no large-scale missions have yet made full use of them.

Ontologies for Robotics and Autonomous Systems

A logical and consistent way to define meanings within a system is a prerequisite for any kind of self-managing infrastructure. Traditionally, meanings in a system were programmed *implicitly*, such that variables and functions either did the same thing constantly or could be defined with internal meanings only relevant to that system. However, with the advent of the internet and highly-connected data infrastructures that must process a wide range of information, some way to define external meanings *explicitly* for machines to process has become vital, and this applies in equal measure to fractionated systems which must to some degree be able to handle any type of data or function within the system in any specific part.

In modern robotics engineering, the greatest challenge is currently autonomy and optimization, and the most common approach is to specify the task on a semantic level. Robotics ontologies are mostly task-based ontologies such as those used in KnowRob and CORA.[20] A comparison of ontologies related to robotics is given in,[21] which introduces an ontology for the robot programming domain. To extend the capabilities of ontological robotics, the IEEE "Ontologies for Robotics and Automation" working group studies industrial robotics, service robotics and autonomous robotics. In,[22] the authors focus on an ontology consisting of specific concepts and axioms with a systematic approach specifically for autonomous robotics. The awareness of each agent in the cooperation of multiple agents will increase the performance of autonomy, resolve task complexity, increase reliability and simplify in design.

The research into ontology-based implementation solutions for autonomous mobile robot systems are rare due to their diversity. The complexity of information exchange between robotic devices using different programming languages, operating systems, and data management standards restricts their ability to interact with each other. The OntAPIbot ontology[23] has been proposed for defining the formal specification of concepts to reduce the heterogeneity of robotic terms and facilitate the use of reasoning tools to correct knowledge interpretation and reuse. Reference[24] developed a domain ontology and ontology-based methodology to support the conceptual modelling of autonomous systems. The OASys and related methodologies have been applied to a robot control testbed and a process control testbed.

As semantic descriptions of objects must provide a means of reasoning with respect to related objects, the use of description logic has long been considered the basis of a reasoning system as a subset of first order logic that provides reasoning support.[25] By far the most popular means of defining semantics is by applying the Web Ontology Language (known as OWL), which is based on description logic and envisioned as a means of expressing ontologies in internet resources, the so-called "Semantic Web". OWL was created by synthesis of concepts from many previous efforts to characterize and perform inference from classes of objects, primarily the DAML+OIL language that was itself a synthesis of DAML-ONT and OIL.[26]

By applying OWL semantics to both data and functions, very flexible descriptions of resources in the fractionated system are possible. Semantic characterization of functions is related to the process of service description for internet services, and the Ontology Web Language for Services (OWL-S) has been developed to provide semantic specification for these kind of services. OWL-S is intended to provide semantic descriptions for use with the Web Services Description Language (WSDL) as such IDLs do not contain sufficient semantic information for an autonomous agent to find and utilize the interfaces in an automated manner.[27] While the interfaces in a fractionated system should be far more straightforward and automated than web interfaces, the requirement for clear semantic description of the inputs and outputs of a service are similar, and OWL-S is used to inspire the organization of semantic functional identification.

The actual structure and format of data within Semantic Web resources is traditionally defined by the Resource Description Framework (RDF), which is based on the concept of a triple of Subject-Predicate-Object to define semantic descriptors. RDF defines a schema (a set of accepted uses) of classes for defining subjects, objects, and predicates within resources. However, RDF is also schema-limited in the sense that set operations and logical combinations of classes are not supported and does not explicitly provide a means for the inference processes used in OWL, as many constructs within OWL cannot be represented using triples. This also restricts class resolution to the specific terms within the schema (as there is no easy way to resolve synonyms). As the expressive design of RDF is not a one-to-one fit with OWL, which itself has evolved into a language with nearly intractable scope, three "species" of OWL were developed (Full, DL, and Lite) in increasing restruction of expressivity to both increase compatibility (OWL-to-RDF but not the reverse).[26] These subsets of OWL are reformulated as EL (for efficient reasoning), QL (for large database queries), and RL (for scaleable reasoning) in the newer OWL2 language which extends OWL with additional features, built-in datatypes, and other constructs to facilitate practical development using the language.[28] Additionally, OWL languages have been extended beyond two-valued logic with fuzzy logic[29] and probabilistic[30] extensions.

SELF-AWARE FRACTIONATED ARCHITECTURE

Rather than technological limitations, the capabilities of next-generation autonomous systems will be for the most part limited by the designer's ability and the ability of the system itself to cope with complexity. As the complexity and need for reliability in robots and autonomous systems increases to meet ever greater functional requirements, there is great interest in the use of fractionated architectures for autonomous applications, in which software tasks can be split up into simpler units and spread arbitrarily across a distributed network of similar hardware processors for parallelism and redundancy. At the same time, these distributed systems can have a vast number of potential configurations and interactions between components, and explicitly planning and predicting for all potential future uses and configurations quickly becomes impractically difficult. Modern robots and autonomous vehicles can contain tens to hundreds of algorithms running in parallel, each operating as part of a larger task on different data types and using different hardware requirements. In cases where unforeseen additional functionality must be added or the system must be re-configured quickly and reliably in response to problems or changed goals, a system designer can be easily overwhelmed with the number of changes that can be made.

Semantic information provides one important piece of the solution to this problem of complexity. By providing machine-parseable context to each part of the system in the form of ontologies, and applying logical reasoning rules to determine what each part contributes and how it operates, machines can be made "self-aware" at least within the fixed boundaries of an ontology. Inference algorithms can then be used to remove the responsibility of thinking ahead and planning for all possibilities from the designer, as semantic inference provides a means by which to make logical decisions on-the-fly regarding system configuration and data management. Widespread use of ontologies in Big Data analytics has already made the Internet a revolutionary resource for information gathering that now pervades all aspects of modern society, and the widespread use of ontologies in autonomous systems and robots is key to revolutionizing their capabilities to operate without human supervision.

To fulfil the requirements for successes of real-time self-configuration and self-monitoring of fractionated systems in the fields like information technology infrastructure, semantic data characterization, and robotics, a new semantic organization method is proposed in this paper. While many problem-specific ontologies have been created to define and classify systems, very few have actually been used to create and configure systems via inference and reasoning. No previous architecture for real-time autonomous robotic and aerospace systems has considered the use of general semantic self-awareness for self-configuration, and yet this is a vital part of many autonomous reasoning processes that would be carried out within such a system. Such a method allows a fractionated system to be fully self-aware and incorporate its own configuration into learning and reasoning processes performed during autonomous operation.

Using semantic reasoning to organize modules following Autonomic Computing principles at a low level within the system is particularly valuable in robotics applications in domains that are subject to more challenging constraints than the majority of electronic and computing systems, specifically regarding environmental tolerances, real-time response, and resiliency to faults and failures. In these domains, the capacity to self-organize based on general available information is a vital part of the tolerance to unexpected situations in which case obtaining general functionality is more important than adherence to a specific structure.

The flow of semantic information and instructions within the proposed conceptual architecture is shown in Fig. 1. For a fractionated autonomous system, it is necessary for a human to provide the functions within the system that perform key tasks, the classifications of data required for these functions, and the goals of the system in terms of these functions and data, which in this case is primarily the end result as defined as a function. The system itself is then able to arrange the functions into a complete chain of processes to accomplish the high-level goal, and place the functions dynamically on appropriate autonomic elements with data transferred between them. The key to accomplishing this is appropriate semantic association of functions, data, and system elements with common concepts that connect them logically. A logical reasoning process is then used to determine how the system can be organized In upcoming research work, this architecture will be applied to modular hardware so as to refine the use of "self-*" characteristics within a robotic system and to evaluate the effectiveness of using them in autonomous processes.

UPCOMING EXPERIMENTS

Some successful small satellite assembly demonstrators have been built,[31] but they are not assembled autonomously by robotic means. In our work, the scaled lab demonstrator (Fig. 2 and 3) will be used to test and validate the self-configuration software methods by re-configuring modules using a robotic manipulator. Sensor fusion software based on that developed in the InFuse project for ESA PERASPERA will be used for the sensing.[32] Frames for these modules will be rapid-prototyped using additive manufacturing, and will communicate with each other through optical as well as short-range radio links. The lab demonstrator considers hardware elements to be general processing nodes with program code and operating system installed, and they are operated in a fully distributed fashion with no single point of failure. The complexity of knowledge-based self-configuration is derived from the constraints in size and power of modules, the harsh conditions of space, and the need to consider a wide variety of information transferred, all of which are encapsulated in RDF-style semantic metadata. Hard-coding many rules for dealing with this complexity makes the system unnecessarily complicated, while semantic reasoning allows flexibility and robustness in the underlying automation, with system-related complexity moved into the realm of semantic metadata.

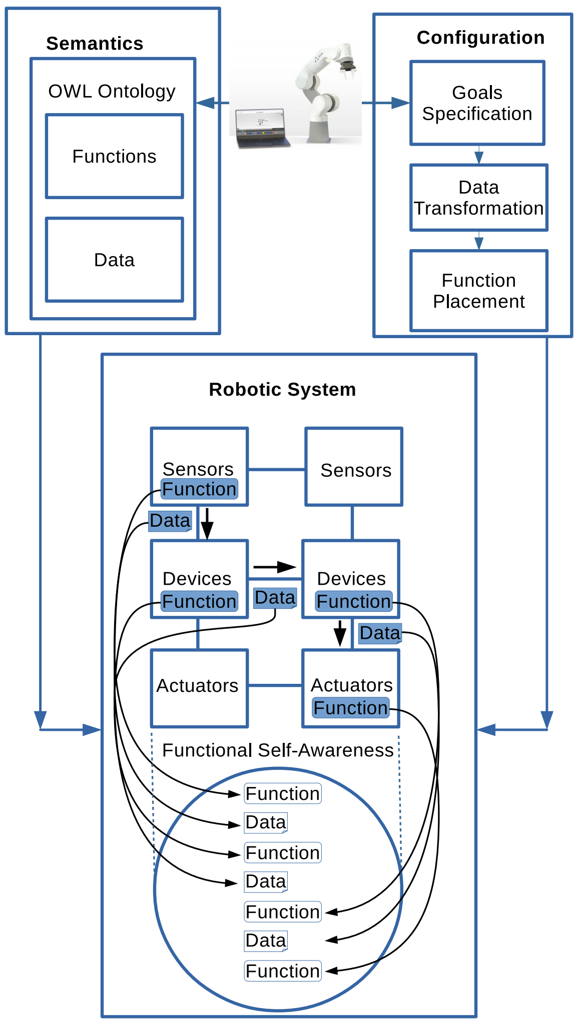

Figure 1. Diagram of the dataflow for a semantic method for self-assembly and self-reconfiguration of a robotized modular satellite system.

Demonstrating these advantages is the goal of the upcoming laboratory experiments, and will lead to commercial sustainability of this branch of research.

REFERENCES

[1] A. Nanjangud, P. Blacker, S. Bandyopadhyay, and Y. Gao, "Robotics and AI-Enabled On-Orbit Operations With Future Generation of Small Satellites," *Proceedings of the IEEE*, Vol. 106, No. 2, 2018, pp. 1–11.

[2] W. F. Truszkowski, M. G. Hinchey, J. L. Rash, and C. A. Rouff, "Autonomous and autonomic systems: A paradigm for future space exploration missions," *IEEE Transactions on System, man and cybernetics*, Vol. 36, No. 3, 2006, pp. 279–291.

[3] P. Horn, "Autonomic computing: IBM's Perspective on the State of Information Technology," 2001.

[4] J. O. Kephart and D. M. Chess, "The vision of autonomic computing," *Computer*, Vol. 36, No. 1, 2003, pp. 41–50.

[5] M. Funabashi, T. Kawabe, and A. Nagashima, "Towards post embedded systems era," *2009 ICCAS-SICE*, 2009, pp. 466–469.

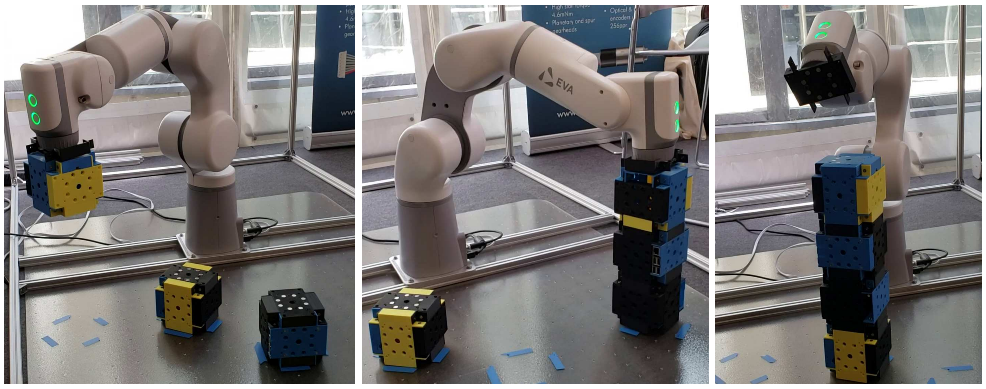

Figure 2. Prototype Testing of Modular Satellite Assembly Demonstrator. The robot arm is able to assemble modules into a desired configuration given information provided by the modules themselves.

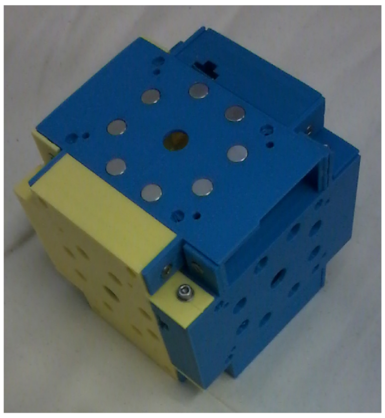

Figure 3. A Prototype Module used for the Demonstrator. Each side is itself a modular "tile" that can contain a different function of the module (e.g. power, processing, actuation) which are encapsulated in self-knowledge to facilitate efficient construction of modules with different specializations.

[6] H. Schmeck, "Organic computing-a new vision for distributed embedded systems," *Object-Oriented Real-Time Distributed Computing, 2005. ISORC 2005. Eighth IEEE International Symposium*, IEEE, 2005, pp. 201–203.

[7] L. J. Dickson, *Crawl-Space Computing: Cooperating Programs That Don't Hide Your Data While They Are Working On It.* Fierce Press, April 2014.

[8] M. Parashar and S. Hariri, "Autonomic computing: An overview," *Unconventional Programming Paradigms Conference*, pp. 257–269, Springer, 2005.

[9] B. Gerkey, R. T. Vaughan, and A. Howard, "The player/stage project: Tools for multi-robot and distributed sensor systems," *Proceedings of the 11th international conference on advanced robotics*, Vol. 1, 2003, pp. 317–323.

[10] M. Quigley, K. Conley, B. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, and A. Y. Ng, "ROS: an open-source Robot Operating System," *ICRA workshop on open source software*, Vol. 3, Kobe, Japan, 2009, p. 5.

[11] G. Metta, P. Fitzpatrick, and L. Natale, "YARP: yet another robot platform," *International Journal of Advanced Robotic Systems*, Vol. 3, No. 1, 2006, p. 8.

[12] S. Joyeux and J. Albiez, "Robot development: from components to systems," *6th National Conference on Control Architectures of Robots*, 2011, pp. 1–15.

[13] N. A. Melchior and W. D. Smart, "Autonomic systems for mobile robots," *Autonomic Computing, 2004. Proceedings. International Conference*, IEEE, 2004, pp. 280–281.

[14] C. C. Insaurralde, "Autonomic computing technology for autonomous marine vehicles," *Ocean Engineering*, Vol. 74, 2013, pp. 233–246.

[15] D. Kirchner, S. Niemczyk, and K. Geihs, "RoSHA: A multi-robot self-healing architecture," *Robot Soccer World Cup*, Springer, 2013, pp. 304–315.

[16] G. B. Tagawa and M. Souza, "An overview of the integrated modular avionics (IMA) concept," *Brazilian Conference on Dynamics, Control and Applications*, 2011, pp. 277–280.

[17] B. Annighoefer, M. Riedlinger, and O. Marquardt, "How to tell configuration-free integrated modular avionics what to do?!," *Digital Avionics Systems Conference (DASC), 2017 IEEE/AIAA 36th*, IEEE, 2017, pp. 1–10.

[18] O. Brown and P. Eremenko, "Fractionated space architectures: A vision for responsive space," tech. rep., Defense Advanced Research Projects Agency Arlingtion VA, 2006.

[19] W. F. Truszkowski, M. G. Hinchey, J. L. Rash, and C. A. Rouff, "Autonomous and autonomic systems: A paradigm for future space exploration missions," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, Vol. 36, No. 3, 2006, pp. 279–291.

[20] F. Ramos, A. Olivares-Alarcos, A. S. Vázquez, and R. Fernández, "What Can Ontologies Do for Robot Design?," *ROBOT 2017: Third Iberian Robotics Conference*, 2017, pp. 465–476.

[21] I. Plauska, "Ontology for robot programming domain," *In XV Internaltinal PhD Workshop OWD*, 2013.

[22] B. Bayat *et al.*, "Requirements for building an ontology for autonomous robots," *Industrial Robot: An International Journal*, Vol. 43, No. 5, 2016, pp. 469–480.

[23] F. Zeshan, R. Mohamad, N. Ahmad, Mohammad, and A. Hussain, Syed, "Ontology for autonomous mobile robot system," *New Trends in Software Methodologies Tools and Techniques*, 2014, pp. 1073–1086.

[24] J. Bermejo-Alonso, R. Sanz, M. Rodriguez, and C. Hernadez, "An ontological framework for autonomous systems modelling," *International Journal on Advances in Intelligent Systems*, Vol. 3, No. 3 and 4, 2010, pp. 211–225.

[25] F. Baader, *The Description Logic Handbook: Theory, Implementation and Applications*. Cambridge university press, 2003.

[26] S. Bechhofer, "OWL: Web ontology language," *Encyclopedia of database systems*, pp. 2008–2009, Springer, 2009.

[27] D. Martin *et al.*, "Bringing semantics to web services: The OWL-S approach," *International Workshop on Semantic Web Services and Web Process Composition*, Springer, 2004, pp. 26–42.

[28] B. C. Grau, I. Horrocks, B. Motik, B. Parsia, P. Patel-Schneider, and U. Sattler, "OWL 2: The next step for OWL," *Web Semantics: Science, Services and Agents on the World Wide Web*, Vol. 6, No. 4, 2008, pp. 309–322.

[29] F. Bobillo and U. Straccia, "Fuzzy ontology representation using OWL 2," *International Journal of Approximate Reasoning*, Vol. 52, No. 7, 2011, pp. 1073–1094.

[30] Z. Ding and Y. Peng, "A probabilistic extension to ontology language OWL," *System Sciences, 2004. Proceedings of the 37th Annual Hawaii international conference*, IEEE, 2004, pp. 10–pp.

[31] C. Underwood and S. e. a. Pellegrino, "AAReST Autonomous Assembly Reconfigurable Space Telescope Flight Demonstrator," *Proceedings of 69th International Astronautical Congress*, 2018, pp. 1–17.

[32] M. Post, R. Michalec, and et.al, "InFuse data fusion methodology for space robotics," *Proceedings of 69th International Astronautical Congress*, 2018.