DOI: 10.13009/EUCASS2022-7352
9TH EUROPEAN CONFERENCE FOR AERONAUTICS AND SPACE SCIENCES (EUCASS)

Reference Architecture for Space Production Systems

David Sanderson, Adam Brown**, James Leadbetter***, Andrew Gordon***, Emma Shires*, and Svetan Ratchev**
**Institute for Advanced Manufacturing,*
University of Nottingham, England, UK, NG7 2GX
{Firstname.Lastname}@nottingham.ac.uk
***The Manufacturing Technology Centre,*
Pilot Way, Ansty Park, Coventry, UK, CV7 9JU
{Firstname.Lastname}@the-mtc.org
****BAE Systems Operations Ltd*
james.leadbetter@baesystems.com · andrew.gordon2@baesystems.com

Abstract

In this paper, we propose a conceptual framework and reference architecture for space production systems (SPS). A review of the existing digital architecture approaches and standards for space systems identifies a number of potential avenues. Based on identified gaps, we extend one of these standards to meet production system requirements and define a reference architecture for space production systems. An initial design is presented, then validated through implementation in a self-contained ground-based robotic demonstration cell that can assemble a variety of hexagonal panels into a number of different designs.

1. Introduction

The SMARTER project was a UK collaborative R&D project that funded a feasibility study of manufacturing in space. The project focussed on how reconfigurable autonomous robotic technologies could be used to automatically manufacture components, assemble large structures, and service or repair existing space assets [1]. The work described in this paper is a development of the SMARTER project, specifically the manufacturing integration plan, which investigated how production resources could be integrated into space platforms. In this paper, we summarise existing digital architectural approaches from the space domain, and identify some gaps in these approaches (Sections 2 and 3). Based on these gaps, we propose a conceptual framework and reference architecture for space production systems (RASPS). This architecture is described through a number of architectural viewpoints and followed by a general design for space production systems (Sections 4 and 5). Finally, Section 6 describes an implementation of the RASPS in a physical ground-based robotic demonstrator system.

2. State of the art

2.1 In-space production

A desire for production to occur in space has a long history, whether this is the historically manual assembly and repair of space assets [2] or the planned autonomous additive manufacture of human habitats on other planets [3]. Organisations such as NASA have devoted a large amount of resources to developing the technologies for in-space production. An example of this is the NASA's Space Technology and Exploration Directorate strategy on In-Space Assembly, Construction, and Operations based out of the Langley Research Centre: it focusses on "On-orbit autonomous assembly and aggregation of large space structures, modular designs, and interfaces to meet the Agency's science and exploration future mission needs" [4].

There is a large body of work on the technologies to enable in-space production, often through the use of autonomy and robotics [5–12]. Other work considers how existing technology can be used to enable that capability [13,14], or how the software for such a capability could be designed and verified [15–17].

Similarly to the integration of in-situ resource utilisation (ISRU) technologies, any use of in-space production technologies will require deep integration with existing space architectures [18]. Although some work is evident in the literature on how this might be accomplished for ISRU, particularly in Lunar or Martian mission scenarios [19–23],

DOI: 10.13009/EUCASS2022-7352

there is little in terms of the detailed integration with software systems, or how this might be accomplished for platforms dedicated to autonomous production operations.

2.2 Digital architectures for space platforms

A review of the existing digital architecture approaches and standards for space systems brings a number of avenues to the fore.

The **core Flight System (cFS)** developed by NASA is a software platform and framework along with a set of reusable application modules that are intended for use as satellite flight control software [24]. The cFS has a layered and component-based design and uses a dynamic runtime environment. This enables reusability and provides the ability to change system components during operation. This is particularly relevant to any desires for a flexible in-space production system, enabling both resilience to failure and for reconfiguration of production capabilities. A later addition to the cFS is the Software Bus Network (SBN). Despite being originally intended for use on platforms with relatively simple architectures (for example a single processor handling the majority of vehicle functions), its application has spread to more distributed architectures with multiple processors or even physical platforms; the SBN connects each node in each platform to all others transparently through a publish-subscribe approach. This dynamism is of interest to the design of a flexible in-space production system.

Demonstration of autonomous robotics through the cFS-based architecture has been achieved through ROSPlan (a planning tool from ROS, the Robot Operating System) and the Plan Execution and Interchange Language (PLEXIL), although this has been constrained to the use of "academic" rather than "industrial" equipment and environments (i.e. the use of a Sawyer robot controlled through ROS, rather than an industrial or space-representative robot using a commercial or bespoke controller) [25,26].

The **Space Avionics Open Interface Architecture (SAVOIR)** advisory group was set up to coordinate standardisation activities for ESA's avionics systems (including on-board data systems, attitude and orbit control systems, and flight software) [27]. The primary output of the SAVOIR advisory group is a software reference architecture [28,29] with three main aspects: a component model that describes the component-based software engineering approach; the computational model that defines the computational entities in the system; and the execution platform that provides a set of services each for containers, connectors, components, and abstract components.

The **Consultative Committee for Space Data Systems (CCSDS)** develops a range of international standards for space-related data and communications systems intended to enable lower operating costs and risks, and to ensure interoperability between current and future missions from a range of operators [30,31]. The primary standard managed by the CCSDS is the **Reference Architecture for Space Data Systems (RASDS)** [32,33], which collects all the CCSDS standards together into one coherent approach for describing and architecting complex space data systems and systems of systems. The RASDS provides a number of architectural viewpoints derived from the reference model for open distributed processing [34].

Both the cFS and SAVOIR approaches define a component-based software system integrated over a databus. CCSDS also defines a similar architectural approach – through a service-oriented architecture, which is a higher-level paradigm that often utilises component-based software design – but abstracts away from the specific integration approach required. The intention for all is to use functional separation and software engineering principles to enable software standardisation, reuse, compatibility, and flexibility of the overall system.

While the CCSDS RASDS is only concerned with components of a data system, it is acknowledged in the standard that "a full space system design would include other classes of components and connectors not addressed" [33]. We therefore consider the CCSDS suite of standards to be a good starting point on which to build a conceptual framework and reference architecture for space production systems.

2.3 Space data systems

While each of the viewpoints in the CCSDS RASDS has its own set of entities that it is primarily concerned with, many of those objects will be relevant to another viewpoint. For example, abstract functional objects defined in the functional perspective will have corresponding implementations in hardware or software that are defined in the connectivity viewpoint. The diagram shown in Figure 1 defines how the major ("top-level", or "core") objects relate to each other through the use of relationship arrows and hierarchical classes. It also uses the colouring to show how they are most associated with a given viewpoint:

- Turquoise for enterprise
- Blue for connectivity
- Purple for functional
- Red for information

DOI: 10.13009/EUCASS2022-7352

- Yellow for communications

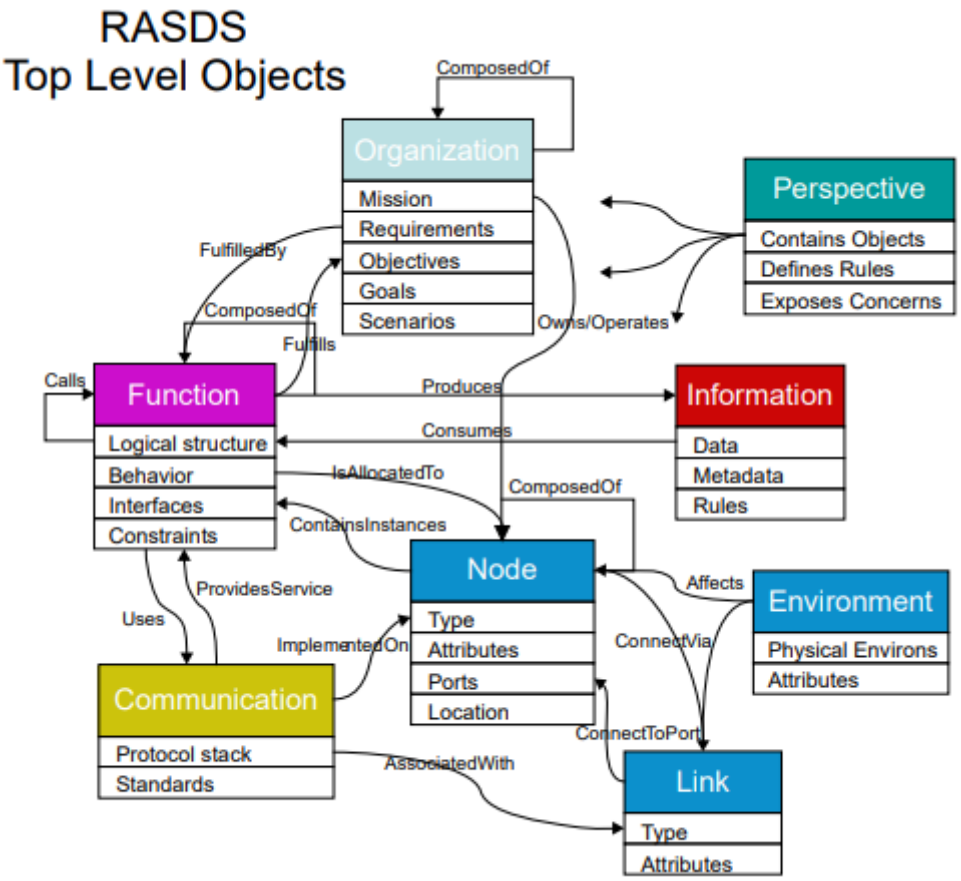

Figure 1. RASDS top-level object relationships (from [33])

We refer to a number of specific standards related to the details of these object groupings:

- The Reference Architecture for Space Data Systems (RASDS) is concerned with the top-level architecture [33].
- The Reference Architecture for Space Information Management (RASIM) is concerned with the information viewpoint [35].
- The Mission Operations standards, including the Services Concept, Reference Model, Message Abstraction Layer, and Common Object Model, are concerned with the functional and connectivity viewpoints [36–39]. This concept for mission operations has also been proposed as the basis for a technology framework on which further specialised application "niches" could be developed [40].
- The Spacecraft On-Board Interface Services (SOIS) is concerned with how the services defined in the remaining standards can be used to fulfil the communications viewpoint [41].

3. Gaps and proposed approach

The objects covered in Figure 1 are necessarily restricted to those that make up a data system. Although modern production systems are increasingly cyber-physical systems with a large data and informatics focus [42–46], simply using a framework for data systems to describe a production system leaves some notable gaps. Examining the basic relationships as described in the RASDS [33]:

- "*Organisations have missions, goals, objectives, and requirements that are fulfilled by the Functions defined within the system. They also own, operate, and develop the Facilities that are engineered as physical Nodes and Links in the system.*" This statement still holds for in-space production systems, though the organisations considered will have to be expanded to include the entire manufacturing supply chain, and what they "own,

DOI: 10.13009/EUCASS2022-7352

operate, and develop" must be expanded to include production equipment, materials, parts, and products. These are all instantiated as physical nodes in the system.

- "*Functions describe the behaviours in the system, and they are implemented as Engineering Objects (either hardware or software) and allocated to Nodes that may contain instances of several different implemented Functions.*" Although the functions in a space production system will have to be expanded to include production functions as compared to space data systems, the structure and concepts covered by how functions are distributed across several nodes remains the same.
- "*Applications (software Engineering Objects) use Communications protocols to transfer Information among themselves. These protocols are defined by Standards.*" As with functions, the concept of a set of applications communicating with each other can remain the same. We can also use this paradigm to introduce the concept of production processes, discussed below.
- "*Information is produced, transformed, and consumed by Functions.*" Information is still processed by functions; the types of functions under consideration is expanded to production functions, and therefore the types of information will also need to be expanded. We will also need to consider the concepts of digital twins and digital threads of products and production systems [47,48], particularly what happens to a product's lifecycle information when it is complete. This information has to be passed off onto another system in coherent fashion, and may have to be either combined into the digital twin of a larger product (if the original product was a part or sub-assembly), or transformed into a working system digital twin (if the original product was a complete system, for example a satellite).
- "*Nodes are connected via Links that connect to Ports on the Nodes. Communications protocols are associated with these Ports.*" Physical nodes in the system now need to consider more than just the hardware and software engineering objects of a data system. In addition to software components, a production system is composed of a wide variety of hardware components; these range from production equipment, through materials and parts, to finished products. These products could in turn be used as parts for other production processes, and could even be used as production equipment (for example when a space production system is used to produce a robot arm or whole other production system).
- "*The physical Environment affects Nodes and Links alike.*" This still holds, but must be further applied to the production equipment and processes. We can easily imagine production equipment and processes that are affected by the external temperature: the effects of thermal expansion will be much more pronounced in free space than on the ground for example.
- *"Most of these classes of objects may be composed of other subclasses of objects.*" This also holds for production processes; some processes may be composed of sub-processes.

After the changes described above have been accounted for, the addition of production processes to the RASDS top-level objects requires a few more relationships to be described:

- Production resources (hardware or software Engineering Objects) use production processes to transform production materials and parts into products.
- Production materials, parts, and products are produced, transformed, and consumed by processes.

The final diagram and list of relationships for a space production system are summarised in the next section as the first part of a proposed reference architecture for space production systems, along with detailed discussion of each major topic.

4. Developing a reference architecture for space production systems

4.1 Top-level objects of a space production system

A reference concept for space production systems can be developed by extending the reference architecture for space data systems developed by the CCSDS. A summary of the top-level objects of this proposed concept is given by Figure 2, based on the equivalent diagram in [33] (Figure 1).

DOI: 10.13009/EUCASS2022-7352

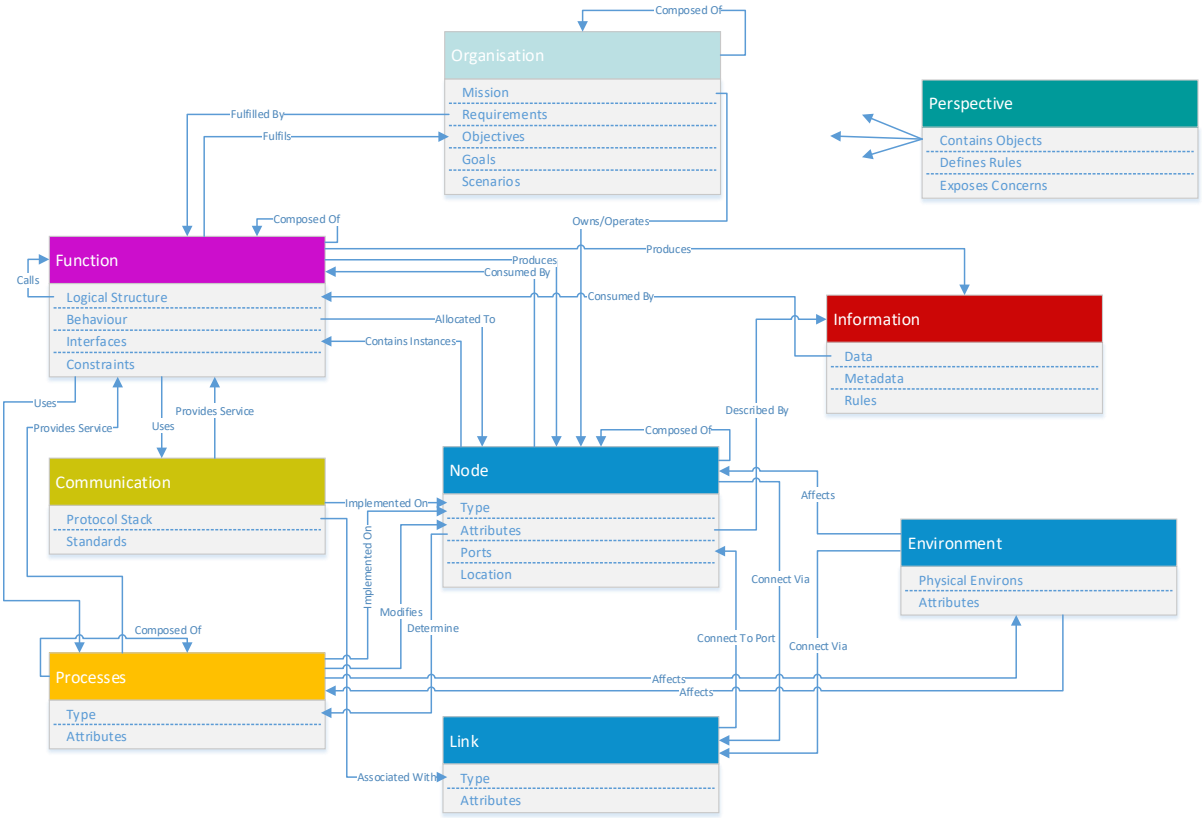

Figure 2. Top-level object relationships for space production systems

Drawing on the descriptions in [33], we can define a set of basic relationships:

- Organizations have missions, goals, objectives, and requirements that are fulfilled by the Functions defined within the system. They also own, operate, and develop the Facilities, Production Resources, Materials, Parts, and Products that are engineered as physical Nodes and Links in the system.
- Functions describe the behaviours in the system, and they are implemented as Engineering Objects (either hardware or software) and allocated to Nodes that may contain instances of several different implemented Functions.
- Applications (software Engineering Objects) use Communications protocols to transfer Information among themselves. These protocols are defined by Standards.
- Production resources (hardware or software Engineering Objects) use production Processes to transform production materials and parts into products.
- Information is produced, transformed, and consumed by Functions.
- Production materials, parts, and products are produced, transformed, and consumed by Processes.
- Nodes are connected via Links that connect to Ports on the Nodes. Communications protocols are associated with these Ports.
- The physical Environment affects Nodes, Links, and Processes alike.
- Most of these classes of objects may be composed of other subclasses of objects.

5. Reference Architecture for Space Production Systems

In alignment with the viewpoints used by the CCSDS for the RASDS, we will now describe an initial proposal for a reference architecture for space production systems (RASPS) from the functional, information, connectivity, and communications viewpoints. We will not be discussing the enterprise viewpoint in detail as this was out of scope for the SMARTER project.

DOI: 10.13009/EUCASS2022-7352

5.1 Functional Viewpoint

The RASPS takes a service-oriented architectural approach to the production systems; as such the system is decomposed into services that provide independent functions to the system. These functions may either provide a capability directly to the system (as "user applications") or may form part of the underlying infrastructure of the system (as "infrastructure services"). The functions relevant to the space data systems are divided across the ground and space segments. The ground segment side of the system is mostly as expected for a space system, with the potential addition of the "production ground segment" supply chain and supporting infrastructure[1]. Conversely, the space segment has pulled many of the traditional ground segment aspects up to the spacecraft in order to support increased autonomy, and also split into the platform control and production control aspects. An overall view is presented in Figure 3, and a detailed example breakdown of the spacecraft functions in Figure 4.

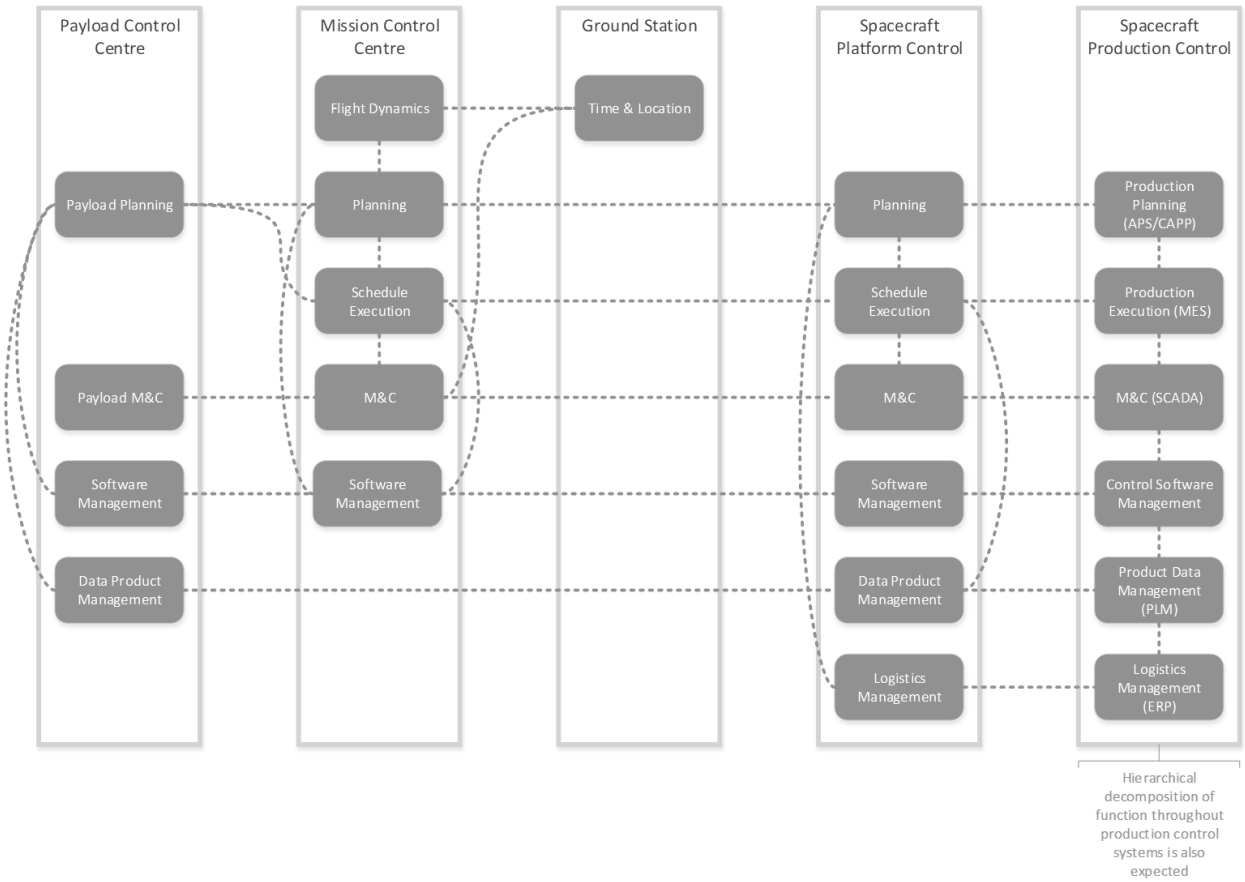

Figure 3. Distribution of mission operation functions for a space production system

[1] This is out of scope for the SMARTER project, and therefore also out of scope for this paper.

DOI: 10.13009/EUCASS2022-7352

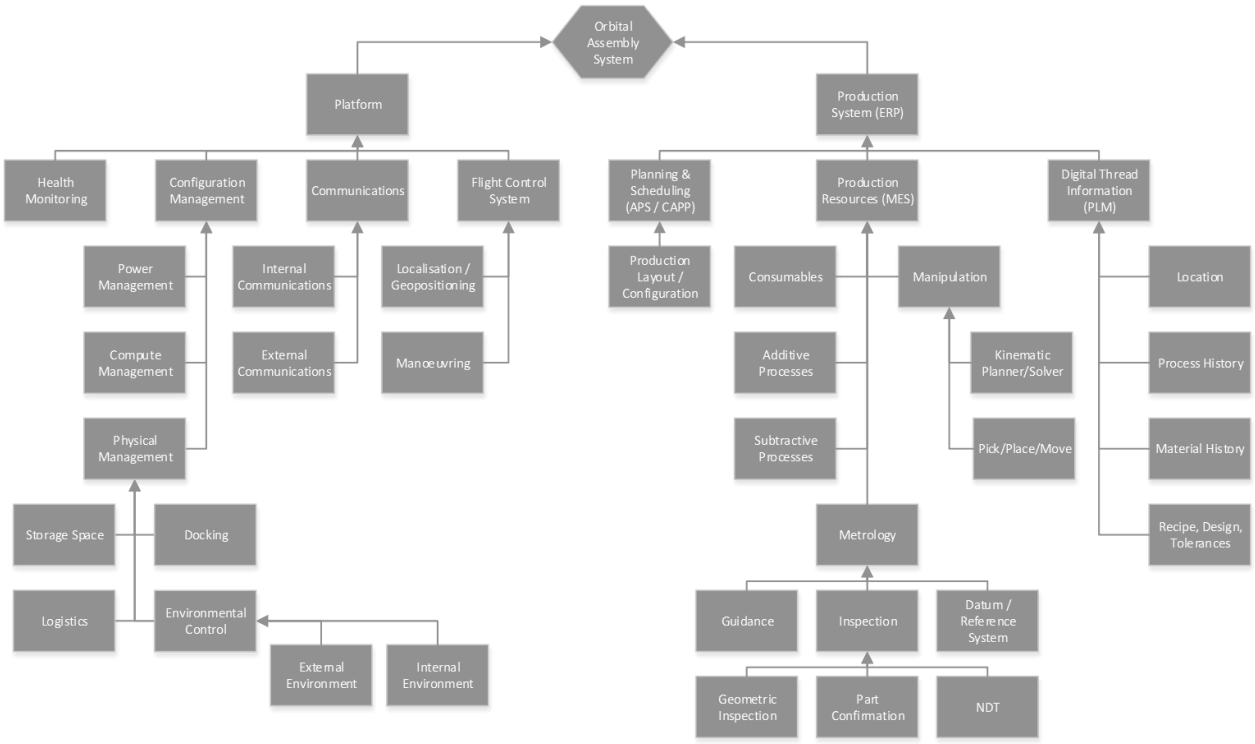

Figure 4. Service-oriented architectural framework for space production systems; the left half of the tree is dedicated to the operation of the platform as a spacecraft, and the right half is dedicated to the operation of the platform as a production system

5.2 Information Viewpoint

The information viewpoint describes in terms of a layered information architecture how information is generated by production resources and used to control production resources. These information objects can then be manipulated by software components which interface with each other (and a set of infrastructure services that they contribute to) over a middleware. When composed together, these components and infrastructure services jointly constitute the service-oriented architecture that fulfils the functions of the system. This is shown in Figure 5, an extension of the RASIM/RASDS that includes an additional layer representing the virtualised physical resources described in the next section.

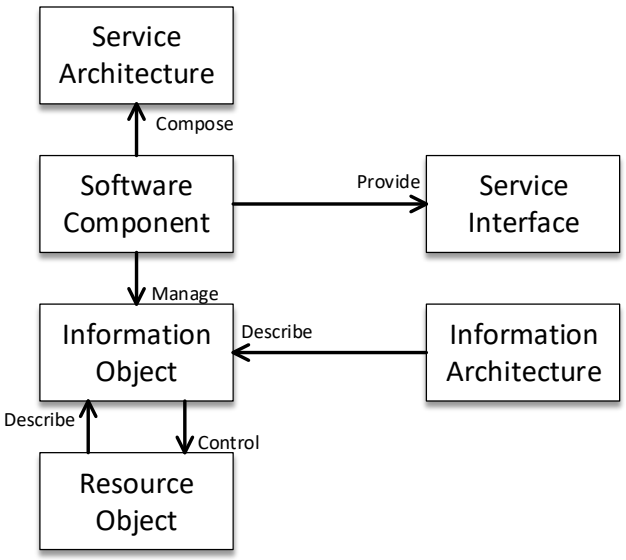

Figure 5. Information viewpoint on the RASPS

DOI: 10.13009/EUCASS2022-7352

5.3 Connectivity Viewpoint

The connectivity viewpoint is concerned with the physical aspects of the system, particularly which nodes and links exist in the system and how the system functions are assigned to them. To address this aspect of the system, we rely more heavily on production system concepts than on space system concepts.
The EPSRC Evolvable Assembly Systems (EAS) project [49] was a research programme that investigated the concept of adaptive and reconfigurable assembly systems based on a complex collective adaptive systems approach [50] that is similar to the well-known Industrie 4.0 paradigm [51]. The EAS approach to resource management is to use intelligent agents [52,53] to virtualise physical production resource equipment into a set of production capabilities. To summarise the information from a number of papers [54–56] into a single diagram, an EAS resource object can be described as shown in Figure 6. A piece of production hardware is controlled by a controller (which may be hardware or software, but is in either case logically separate from the resource hardware). This controller runs some software that performs the low-level control operations for the hardware; converting user-designed commands (move, pick, cut, etc) into lower-level code and eventually electrical signals for the hardware actuators. This code can be interacted with by an embedded computer running an intelligent agent. This agent performs high-level control of the resource and allows multiple resources to be connected together into a multi-agent control system which can perform higher-level processing or decision-making, thereby imbuing the production system with a level of intelligence and/or autonomy. The agents are connected together with a data distribution service (DDS, [57]), a type of data-oriented bus middleware. Each agent virtualises its resource hardware into a set of capabilities that can be advertised to the system as being available for use. In this way, the EAS approach can fulfil the service-orientation, virtualisation, and servitisation required by the RASPS as it follows the guidance laid out in the RASDS.

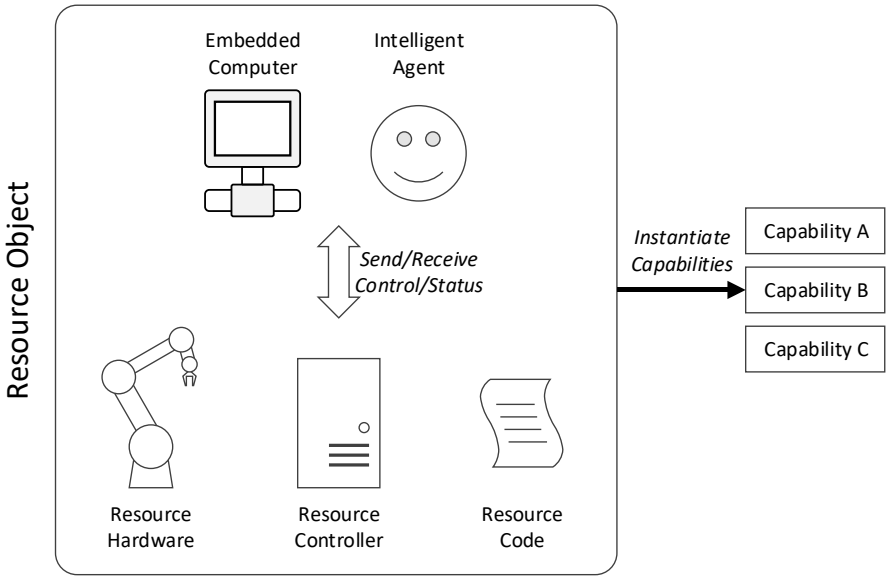

Figure 6. Resource object for intelligent production systems

It can also be noted that the EAS project includes a function-behaviour-structure approach to systems design and integration [58], where the function of the system is used to design desired behaviours, and then the expressed behaviours of physical structures are tested to determine if the desired and actual behaviours match. This ties in closely with the concept of a service-oriented architecture being designed through a functional decomposition to ensure that the provision of capabilities is abstracted away from the physical concerns of the system.

5.4 Communications Viewpoint

Closely related to the connectivity viewpoint is the communications viewpoint. In the RASPS we have already determined to use an approach similar to the EAS architecture, which uses a DDS to connect each resource object. The DDS acts as both the connectivity link, deals with the communications stack, and also functions as a distributed data storage.

DOI: 10.13009/EUCASS2022-7352

Figure 7 shows how the DDS stack relates to the SOIS stack[2]; the real-time publish subscribe wire protocol handles a part of the subnetwork services at the low level, the DDS API is provided to the user to develop their own applications built on the services provided by the middleware, which both bridges the gap and provides some of the services found at both the application support layer and the subnetwork layer in SOIS.

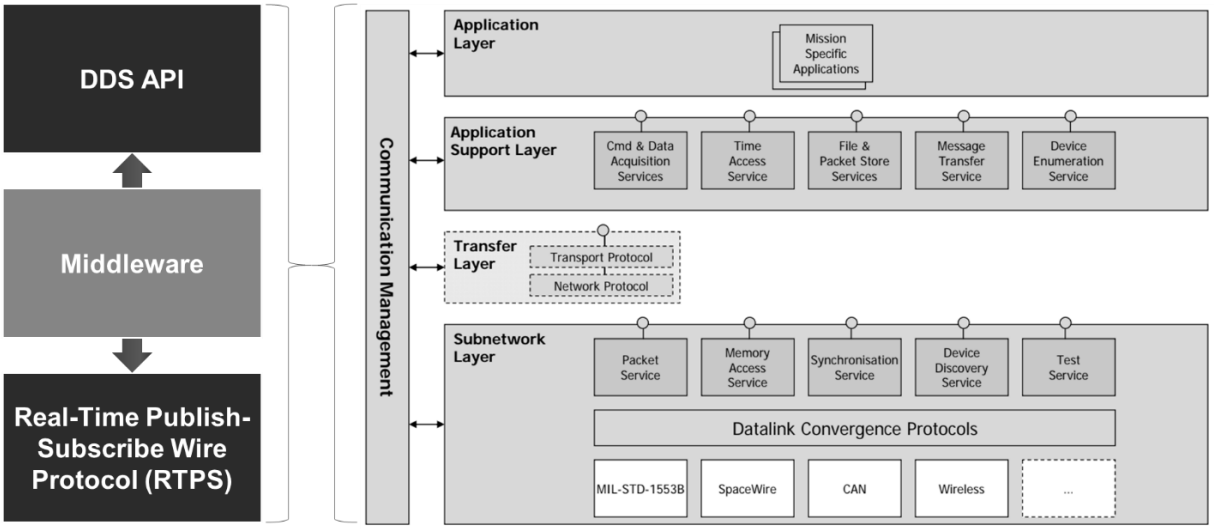

Figure 7. Alignment between DDS and SOIS (right hand side of the diagram from [41])

As mentioned above, the DDS acts as a distributed data store as well as a communications link. The alignment between DDS and RASIM can be seen in Figure 8; while the DDS API can be used to develop the user application, the middleware and RTPS jointly provide the infrastructure services, store the data objects, and connect them to the applications through a service bus.

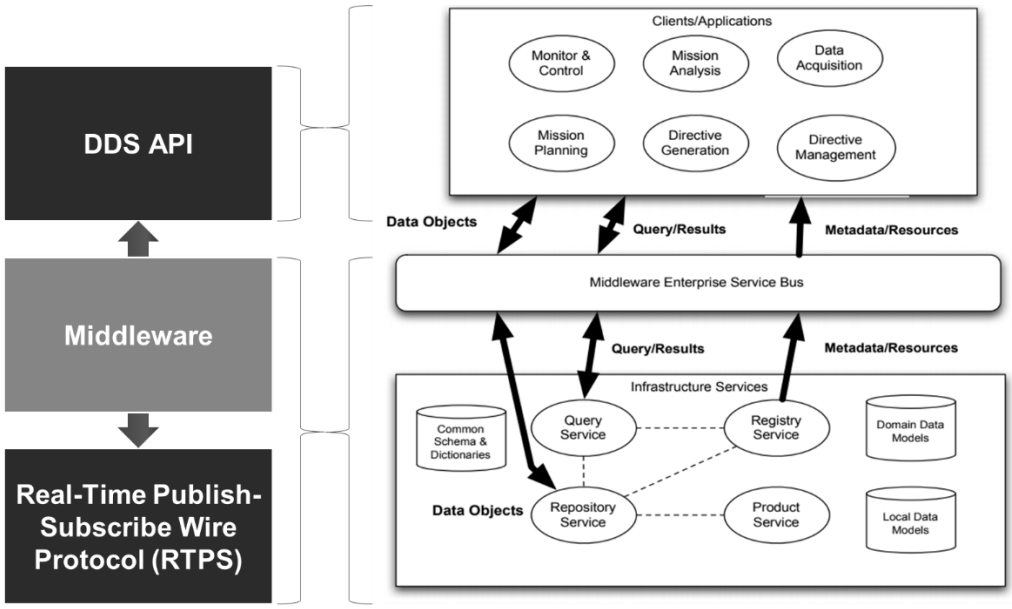

Figure 8. Alignment between DDS and RASIM (right hand side of the diagram from [35])

5.5 General design for a space production system

The overall design for a general space production system is given in Figure 9, and will be used as the basis of a physical demonstrator to validate the RASPS. In line with the proposed architecture, all capabilities in the system are accessed through a service-oriented manner, according to a functional distribution. There are two main object types:

[2] This should not be taken to be a technical comparison, but rather a functional one

DOI: 10.13009/EUCASS2022-7352

generic software services, which are common to the RASDS and virtualise service devices or software; and production resource objects, which are new to the RASPS and virtualise production hardware or software. In addition to this, there are a number of "executives" which are each a special object that allows access to a service from a top level. In the case of production systems this could be the manipulation or health monitoring services for example. The local device or service will collect and process local information and pass it up to a global executive that can collate the information at a global level in order to reduce the transmission load and enable more efficient processing. Finally, there is a link between the production control side and the platform control side of the whole space system. We have not discussed this link in any detail as it will be highly dependent on the design of the platform control system and was out of scope for the SMARTER project. All of these objects are linked together with a DDS that provides integration, communication, and servitises the underlying data storage devices.

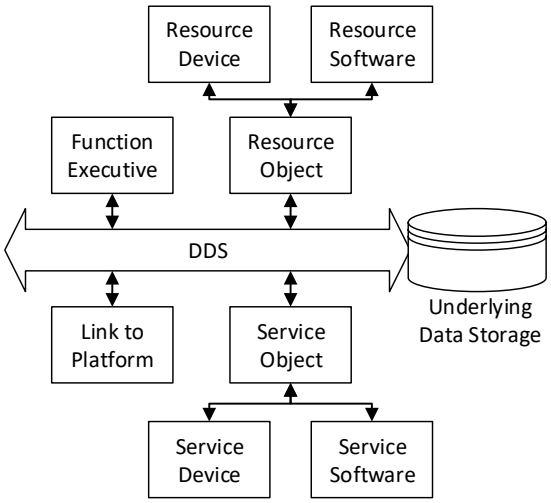

Figure 9. Overview of integration plan for space production system

6. Demonstrator

6.1 Context

Based on existing approaches in the literature [5,8,13], the concept of hexagonal panel assembly was chosen as the context for the physical demonstrator developed in the SMARTER project. The chosen use case was for the demonstrator to be able to autonomously assemble two different product types from the hexagonal panels: one representing an antenna structure (that required overall assembly tolerances to be met); and one where each panel around a central core provided a different functionality (requiring correct ordering and placement of each panel). This matches known use cases from the terrestrial manufacturing industry, where companies are increasingly looking to develop flexible and adaptive "multi-product" production systems, able to cope with disruption in the form of changing requirements and demands [59]. Figure 10 shows the concept developed in the SMARTER project for an orbital assembly platform; the left-hand side of the image shows an orbital assembly platform stowed inside a fairing along with the parts required for the mirror assembly (hexagonal mirrors shown in blue, trusses stored in the yellow section, and nodes stored in the red section).

Based on the orbital assembly platform concept, a ground-based physical demonstrator was designed, as shown on the right-hand side of Figure 10. The demonstrator was designed to showcase: a flexible assembly process; a production process for hexagonal panels with integrated active components; a design for panel connectors; a machine learning approach for system health monitoring and diagnosis; and the RASPS integration approach. For clarity, only the overall demonstrator structure and the architectural integration are described here. Two example products (panel assemblies) can be seen in the figure on the low-friction surface used as the assembly bed. Two robot arms are mounted at one end of the assembly bed; between them are a panel storage rack and mounting arm for the depth cameras by the assembly process. Housed underneath the assembly bed (not visible) are the robot controllers and computing equipment used for the dashboard, health monitor, and other software.

DOI: 10.13009/EUCASS2022-7352

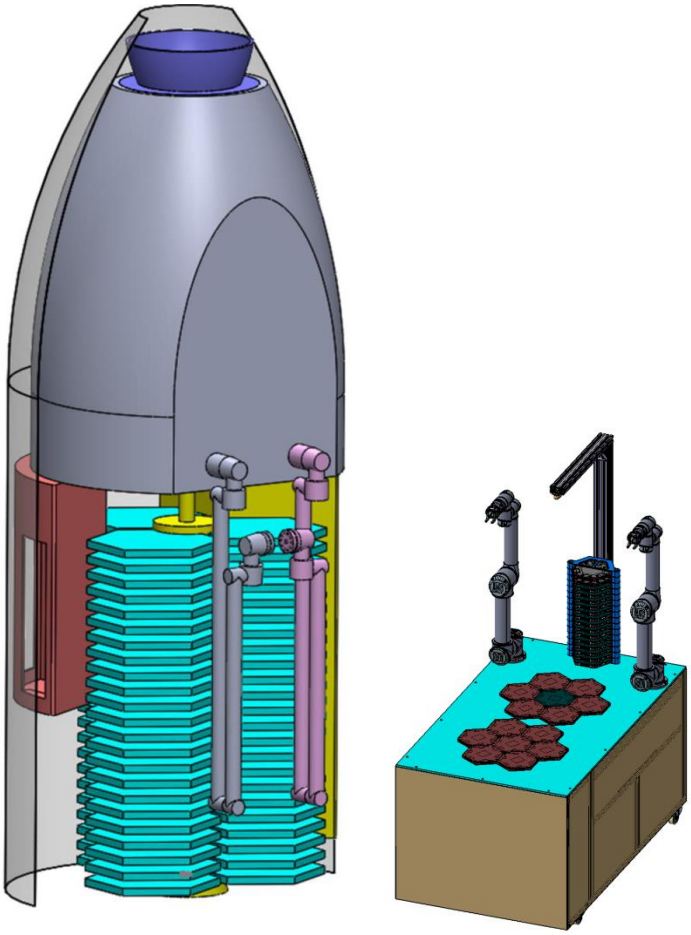

Figure 10. SMARTER concept for an orbital assembly platform (left) and physical demonstrator (right)

6.2 Implementation

The software implementation of the demonstrator was based on the RASPS as validation of the approach. All nodes in the system connect to a common distributed data service bus: the processors, actuators, and sensors in the product panels; sensors mounted on the frame; the robotic assembly system (arms and cameras); the dashboard display; and the AI health monitor are all shown in Figure 11. The modular service approach enabled a mixture of bespoke software written in C#, Python, etc, to be connected seamlessly with existing off-the-shelf software. RTI Connext DDS [60] was used as the bus in this case, which removed the requirement for explicit data format conversion, and for separate data storage[3]. Clear definition of the data model was used to coordinate the different project partners before implementing their connection to the bus.

Figure 12 shows the completed physical demonstrator during commissioning. The demonstrator successfully assembled a range of product configurations. During the assembly process, the demonstrator monitored the status of the production system and the products, and was able to identify potential issues, and – where possible – respond to these disruptions.

[3] Due to space constraints, implementation details for the DDS, including the data model and participants/nodes, are available from the authors on request.

DOI: 10.13009/EUCASS2022-7352

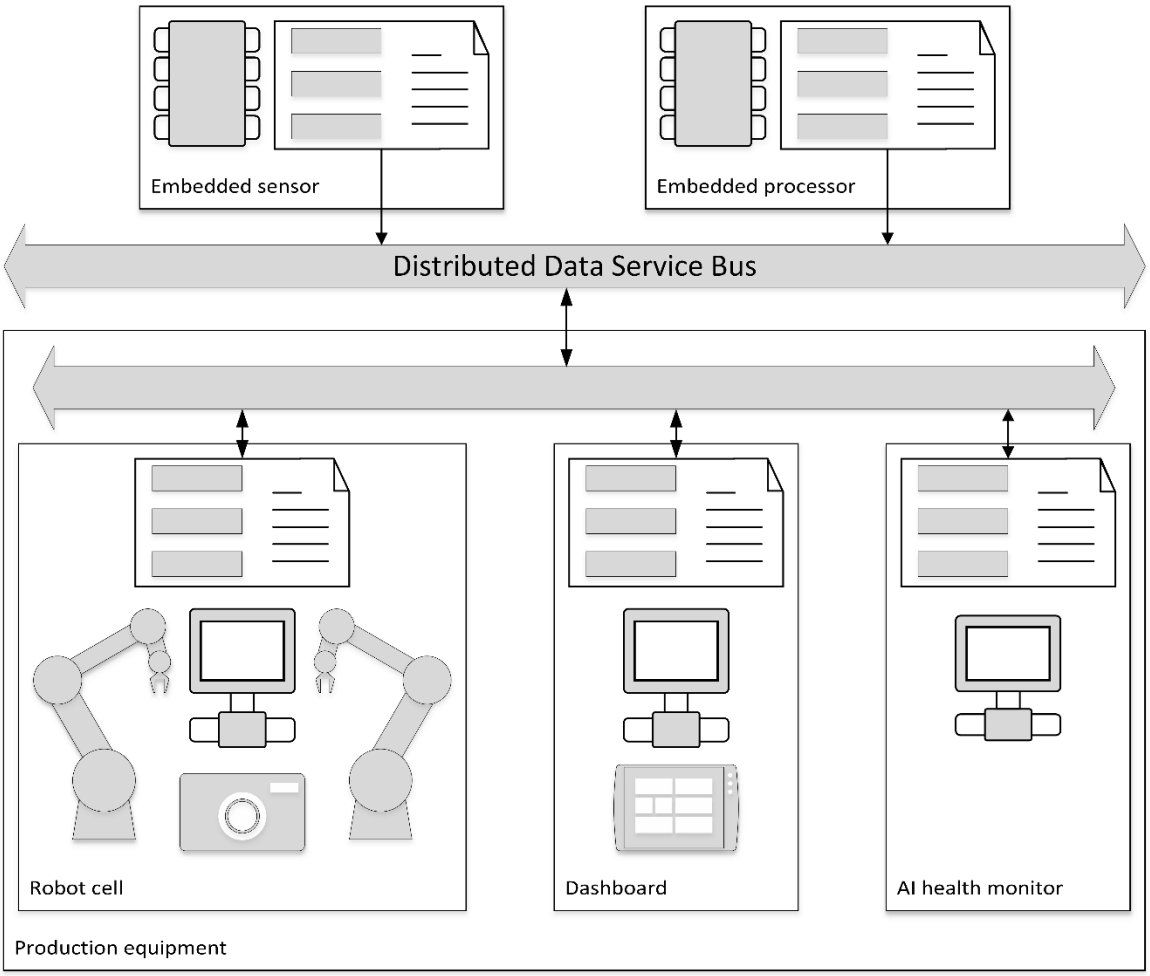

Figure 11. Demonstrator software block diagram

Figure 12. Physical demonstrator

DOI: 10.13009/EUCASS2022-7352

7. Conclusions

A reference architecture for space production systems (RASPS) has been defined through the functional, information, connectivity, and communications viewpoints. The RASPS extends the CCSDS suite of standards to meet the requirements of space production systems, using a component-based modular design that connects servitised functions across a data distribution service. An initial design and integration approach were presented with reference to common requirements for advanced production systems. This design has been validated through implementation in a self-contained ground-based robotic demonstration cell that can perform a flexible assembly task. The demonstrator cell has been successfully used to assemble a variety of hexagonal panels in a number of different designs, where each panel had varying integrated technologies.

8. Acknowledgements

The authors gratefully acknowledge the support of Innovate UK through the project "SMARTER - Space Manufacturing, Assembly and Repair Technology Exploration and Realisation" (reference 104057) and EPSRC through the project “Made Smarter Innovation – Research Centre for Connected Factories” (reference EP/V062123/1). The authors would also like to thank the consortium of SMARTER project partners (additionally including Lena Space, Magna Parva, Printed Electronics Ltd, Reaction Engines, and the Satellite Applications Catapult) for their work in the project and on the aspects of the demonstrator not described in this paper.

9. References

[1] SMARTER: Space Manufacturing, Assembly and Repair Technology Exploration and Realisation Project Grant (Reference 104057). https://gtr.ukri.org/projects?ref=104057. Accessed Jun. 11, 2022.

[2] Shayler, D. J. *Assembling and Supplying the ISS*. Springer International Publishing, Cham, 2017.

[3] Mueller, R. P., Sibille, L., Hintze, P. E., Lippitt, T. C., Mantovani, J. G., Nugent, M. W., and Townsend, I. I. Additive Construction Using Basalt Regolith Fines. 2015.

[4] Bowman, L. NASA Langley Space Technology In-Space Assembly Portfolio: Leveraging Our Heritage to Enable the Future. 2017.

[5] Doggett, W. Robotic Assembly of Truss Structures for Space Systems and Future Research Plans. No. 7, 2002, pp. 3589–3598.

[6] Whittaker, W., Urmson, C., Staritz, P., Kennedy, B., and Ambrose, R. Robotics for Assembly, Inspection, and Maintenance of Space Macrofacilities. 2000.

[7] Staritz, P. J., Skaff, S., Urmson, C., and Whittaker, W. Skyworker: A Robot for Assembly, Inspection and Maintenance of Large Scale Orbital Facilities. No. 4, 2001, pp. 4180–4185.

[8] Dorsey, J. T., Doggett, W. R., Hafley, R. A., Komendera, E., Correll, N., and King, B. An Efficient and Versatile Means for Assembly and Manufacturing Systems in Space. 2012.

[9] Komendera, E. E., and Dorsey, J. T. Initial Validation of Robotic Operations for In-Space Assembly of a Large Solar Electric Propulsion Transport Vehicle. 2017.

[10] Mukherkee, R. Robotic Assembly of Space Assets: Architectures and Technologies. 2018.

[11] Boyd, I. D., Buenconsejo, R. S., Piskorz, D., Lal, B., Crane, K. W., and De La Rosa Blanco, E. *On-Orbit Manufacturing and Assembly of Spacecraft*. 2017.

[12] Patane, S., Joyce, E. R., Snyder, M. P., and Shestople, P. Archinaut: In-Space Manufacturing and Assembly for Next-Generation Space Habitats. 2017.

[13] Doggett, W. R. *A Guidance Scheme for Automated Tetrahedral Truss Structure Assembly Based on Machine Vision*. 1996.

[14] Azimi, S., Zemler, E., and Morris, R. A. Autonomous Robotics Manipulation for In-Space Intra-Vehicle Activity. 2019.

[15] Allen, C. L. *An Expert System Executive for Automated Assembly of Large Space Truss Structures*. 1993.

[16] Herstrom, C. L., Grantham, C., Allen, C. L., Doggett, W. R., and Will, R. W. *Software Design for Automated Assembly of Truss Structures*. 1992.

[17] Brat, G., Denney, E., Giannakopoulou, D., Frank, J., and Jonsson, A. Verification of Autonomous Systems for Space Applications. No. 2006, 2006, pp. 1–11.

[18] Sanders, G. B., and Larson, W. E. “Integration of In-Situ Resource Utilization into Lunar/Mars Exploration through Field Analogs.” *Advances in Space Research*, Vol. 47, No. 1, 2011, pp. 20–29. https://doi.org/10.1016/J.ASR.2010.08.020.

[19] Sanders, G. B., Simon, T. M., Rosenbaum, B. J., Larson, W. E., Quinn, J. W., Captain, J. E., and Boucher, D. S. Overcoming the Hurdles of Incorporating In Situ Resource Utilization (IRSU) into Human Lunar Exploration. 2010.
[20] Hofstetter, W. K., Wooster, P. D., and Crawley, E. F. "Analysis of Human Lunar Outpost Strategies and Architectures." *Journal of Spacecraft and Rockets*, Vol. 46, No. 2, 2009, pp. 419–429. https://doi.org/10.2514/1.36574.
[21] Drake, B. G., Hoffman, S. J., and Beaty, D. W. Human Exploration of Mars, Design Reference Architecture 5.0. 2010.
[22] Stanley, D., Cook, S., Connolly, J., and Hanley, J. Exploration Systems Architecture Study: Overview of Architecture and Mission Operations Approach. 2006.
[23] Cooke, D. "Implementing the Vision: Exploration Strategy and Architecture." *2nd Space Exploration Conference*, 2006.
[24] NASA. Core Flight System. https://cfs.gsfc.nasa.gov/. Accessed Oct. 10, 2019.
[25] Levinson, R., Frank, J. D., Iatauro, M., Sweet, A., Aaseng, G. B., Scott, M., Ossenfort, J., Soeder, J., Ngo, T., Greenwood, Z., Csank, J., Carrejo, D., and Loveless, A. T. Development and Testing of a Vehicle Management System for Autonomous Spacecraft Habitat Operations. 2018.
[26] Frank, J. D. Artificial Intelligence: Powering Human Exploration of the Moon and Mars.
[27] ESA. SAVOIR. http://savoir.estec.esa.int/. Accessed Oct. 11, 2019.
[28] Jung, A., Panunzio, M., and Terraillon, J.-L. *SAVOIR-FAIRE - On-Board Software Reference Architecture (Issue 1 Revision 0)*. 2010.
[29] Alaña, E., Herrero, J., Urueña, S., Macioszek, K., and Silveira, D. A Reference Architecture for Space Systems. 2018.
[30] CCSDS Standards | NASA Technical Standards System (NTSS). https://standards.nasa.gov/ccsds-standards. Accessed Oct. 10, 2019.
[31] CCSDS.Org - The Consultative Committee for Space Data Systems (CCSDS). https://public.ccsds.org/default.aspx. Accessed Oct. 11, 2019.
[32] Shames, P. Reference Architecture for Space Data Systems. 2004.
[33] CCSDS. *CCSDS 311.0-M-1 Reference Architecture for Space Data Systems. Magenta Book. Issue 1*. 2008.
[34] Raymond, K. Reference Model of Open Distributed Processing (RM-ODP): Introduction. In *Open Distributed Processing*, Springer US, Boston, MA, 1995, pp. 3–14.
[35] CCSDS. *CCSDS 312.0-G-1 Reference Architecture for Space Information Management. Green Book. Issue 1*. 2013.
[36] CCSDS. *CCSDS 520.0-G-3 Mission Operations Services Concept. Green Book. Issue 3*. 2010.
[37] CCSDS. *CCSDS 520.1-M-1 Mission Operations Reference Model. Magenta Book. Issue 1*. 2010.
[38] CCSDS. *CCSDS 521.0-B-2 Mission Operations Message Abstraction Layer. Blue Book. Issue 2*. 2013.
[39] CCSDS. *CCSDS 521.1-B-1 Mission Operations Common Object Model. Blue Book. Issue 1*. 2014.
[40] Coelho, C. *A Software Framework for Nanosatellites Based on CCSDS Mission Operations Services with Reference Implementation for ESA's OPS-SAT Mission*. Graz University of Technology, 2017.
[41] CCSDS. *CCSDS 850.0-G-2 Spacecraft Onboard Interface Services. Green Book. ISsue 2*. 2013.
[42] Lu, Y. Industry 4.0: A Survey on Technologies, Applications and Open Research Issues. *Journal of Industrial Information Integration*. Volume 6.
[43] Zhong, R. Y., Xu, X., Klotz, E., and Newman, S. T. "Intelligent Manufacturing in the Context of Industry 4.0: A Review." *Engineering*, Vol. 3, No. 5, 2017. https://doi.org/10.1016/J.ENG.2017.05.015.
[44] Thoben, K.-D., Wiesner, S., and Wuest, T. "'Industrie 4.0' and Smart Manufacturing – A Review of Research Issues and Application Examples." *International Journal of Automation Technology*, Vol. 11, No. 1, 2017, pp. 4–16. https://doi.org/10.20965/ijat.2017.p0004.
[45] Monostori, L. "Cyber-Physical Production Systems: Roots, Expectations and R{&}D Challenges." *Procedia CIRP*, Vol. 17, 2014, pp. 9–13. https://doi.org/10.1016/j.procir.2014.03.115.
[46] Monostori, L., Csáji, B. Cs., Kádár, B., Pfeiffer, A., Ilie-Zudor, E., Kemény, Zs., and Szathmári, M. "Towards Adaptive and Digital Manufacturing." *Annual Reviews in Control*, Vol. 34, No. 1, 2010, pp. 118–128. https://doi.org/10.1016/j.arcontrol.2010.02.007.
[47] Negri, E., Fumagalli, L., and Macchi, M. "A Review of the Roles of Digital Twin in CPS-Based Production Systems." *Procedia Manufacturing*, Vol. 11, 2017, pp. 939–948.
[48] Kinard, D. Digital Thread and Industry 4.0. 2018.
[49] Ratchev, S. Evolvable Assembly Systems - Towards Open, Adaptable, and Context-Aware Equipment and Systems (Grant Reference EP/K018205/1).
[50] Sanderson, D., Antzoulatos, N., Chaplin, J. C., Busquets, D., Pitt, J., German, C., Norbury, A., Kelly, E., and Ratchev, S. Advanced Manufacturing: An Industrial Application for Collective Adaptive Systems. 2015.

DOI: 10.13009/EUCASS2022-7352

[51] Bundesministerium für Bildung und Forschung. *Industrie 4.0 - Innovationen Für Die Produktion von Morgen*. 2014.
[52] Wooldridge, M., and Jennings, N. R. "Intelligent Agents: Theory and Practice." *Knowledge engineering review*, Vol. 10, No. 2, 1995, pp. 115–152.
[53] Wooldridge, M., and Jennings, N. R. R. "Agent Theories, Architectures, and Languages: A Survey." *Lecture Notes in Computer Science*, Vol. 890, 1995, pp. 1–21.
[54] Chaplin, J. C., Bakker, O. J., de Silva, L., Sanderson, D., Kelly, E., Logan, B., and Ratchev, S. M. "Evolvable Assembly Systems: A Distributed Architecture for Intelligent Manufacturing." *IFAC-PapersOnLine*, Vol. 48, No. 3, 2015, pp. 2065–2070. https://doi.org/10.1016/j.ifacol.2015.06.393.
[55] Chaplin, J. C., and Ratchev, S. M. Deployment of a Distributed Multi-Agent Architecture for Transformable Assembly. 2018.
[56] Sanderson, D., Turner, A., Shires, E., Chaplin, J., and Ratchev, S. "Demonstration of Transformable Manufacturing Systems through the Evolvable Assembly Systems Project." *SAE Technical Papers*, 2019. https://doi.org/10.4271/2019-01-1363.
[57] Object Management Group. Data Distribution Service Specification (Current). http://www.omg.org/spec/DDS/Current.
[58] Sanderson, D., Chaplin, J. C., and Ratchev, S. "A Function-Behaviour-Structure Design Methodology for Adaptive Production Systems." *International Journal of Advanced Manufacturing Technology*, 2019, pp. 1–12. https://doi.org/10.1007/s00170-019-03823-x.
[59] de Silva, L., Felli, P., Chaplin, J. C., Logan, B., Sanderson, D., and Ratchev, S. "Realisability of Production Recipes." *Frontiers in Artificial Intelligence and Applications*, Vol. 285, 2016, pp. 1449–1457. https://doi.org/10.3233/978-1-61499-672-9-1449.
[60] RTI. RTI Connext DDS Professional. https://www.rti.com/products/connext-dds-professional. Accessed Sep. 16, 2019.