Space Debris Ontology for ADR Capture Methods Selection*,**

Marko Jankovic*, Mehmed Yüksel, Mohammad Mohammadzadeh Babr

Robotics Innovation Center (RIC)–DFKI GmbH and University of Bremen, Robert-Hooke-Straße 1, 28359 Bremen, Germany

Francesca Letizia, Vitali Braun

IMS Space Consultancy for the European Space Operation Center (ESOC)–ESA, Robert-Bosch-Straße 5, 64293 Darmstadt, Germany

Abstract

Studies have concluded that active debris removal (ADR) of the existing in-orbit mass is necessary. However, the quest for an optimal solution does not have a unique answer and the available data often lacks coherence. To improve this situation, modern knowledge representation techniques, that have been shaping the World Wide Web, medicine and pharmacy, should be employed. Prior efforts in the domain of space debris have only focused onto space situational awareness, neglecting ADR. To bridge this gap we present a domain-ontology of intact derelict objects, i. e. payloads and rocket bodies, for ADR capture methods selection. The ontology is defined on a minimal set of physical, dynamical and statistical parameters of a target object. The practicality and validity of the ontology are demonstrated by applying it onto a database of 30 representative objects, built by combining structured and unstructured data from publicly available sources. The analysis of results proves the ontology capable of inferring the most suited ADR capture methods for considered objects. Furthermore, it confirms its ability to handle the input data from different sources transparently, minimizing user input. The developed ontology provides an initial step towards a more comprehensive knowledge representation framework meant to improve data management and knowledge discovery in the domain of space debris. Furthermore, it provides a tool that should make the initial planning of future ADR missions simpler yet more systematic.

Keywords:

*This document is an accepted version of the manuscript available onlilne at: https://doi.org/10.1016/j.actaastro.2020.03.047.

*Corresponding author.

Email address: marko.jankovic@dfki.de (Marko Jankovic)

space debris, ontology, active debris removal, breakup hazard, protégé, python

1. Introduction

The space infrastructure has become an integral part of the human society. Even the most mundane services, such as navigation, weather forecast, television and banking, are nowadays supported by the on-orbit infrastructure. It is therefore understandable that after more than 60 years of space activities, which resulted in more than 5480 launches and 520 fragmentations [1, 2], there is a growing concern of the society at large over the space debris issue, that places vital services in a constant danger of collision. This situation is being even more exacerbated with the rise of mega-constellations, from commercial companies such as OneWeb, SpaceX and Amazon, that plan to deploy more than 53 000 new satellites in low Earth orbit (LEO). However, dealing with the issue of space debris is at present a challenging task, considering the amount of uncertainty surrounding it [3]. In fact, currently, we can only reliably detect and catalog objects that are larger than 5 cm to 10 cm in LEO and 30 cm to 100 cm in geostationary Earth orbit (GEO) [4]. The population of smaller objects can be estimated only by dedicated models and validated through retrieved surfaces, which were exposed to the space debris environment. In addition, although certain orbital regions, such as LEO, have theoretically exceeded the critical density of objects that would ensure the onset of the Kessler syndrome[1], it is uncertain when and in which measure exactly will this phenomenon appear on-orbit [3]. Furthermore, the currently implemented space debris mitigation guidelines [6] have proven to be necessary but insufficient activities to manage the risk posed by the space debris and therefore maintain a stable space debris environment [3]. The only way to permanently stabilize the current environment and prevent the onset of the Kessler syndrome consists of actively removing the existing large sources of debris, i. e. intact derelict objects (IDOs)[2], via remediation activities. This way future collisions between large objects can be prevented and with them the generation of fragmentation debris that are more difficult to track and, currently, neither practical nor economically feasible to remove actively [3, 7].

Among the currently investigated remediation activities, active debris removal (ADR) is perceived as the only one able to permanently reduce the number of IDOs, at the expense of difficult mission planning, execution and necessity to act over a period of decades. Among all the phases of a generic ADR mission, the capture phase emerges as one of the most challenging ones (along with the close-range rendezvous), since, to best of our knowledge, no spacecraft has ever captured a completely uncooperative target. It consists of actions, performed by a chaser spacecraft, to capture a target, stabilize the compound and

[1] A self-sustaining collision process that would increase the number of on-orbit objects exponentially, due to series of cascading collisions among objects of the existing population [5].

[2] Defined as objects having a size >1 m and >2 m in LEO and GEO, respectively [3].

prepare it for disposal (i. e. de- or re-orbit). Based on the type of "contact" needed to successfully capture a target, the capture approaches currently being researched can be grouped into: contact-based or contactless methods. In fact, while the former include technologies requiring a physical contact with a target, the latter are able to achieve a "capture" by actively controlling the attitude of a target from a stand-off distance. Contact-based methods can be further divided into [8]:

robotics-based systems employing robotic devices (e. g. a manipulator or clamp/tentacle) to capture a target and stabilize the compound

tether-based systems operating a tethered net or harpoon to capture a target from a stand-off distance.

Contactless methods can be likewise further grouped into [8]:

plume impingement-based systems using an ion- or inert gas-based engine to create a plume of particles in front of a target to reduce its momentum and therefore achieve a "capture" within a predefined volume of space, from a stand-off distance

ablation systems applying a concentrated source of electromagnetic radiation (e. g. visible light) to ablate the surface of a target, thus generating a small, but constant thrust opposite to the direction of the applied radiation

electromagnetic-based systems (e. g. eddy brakes, electrostatic tractors) exploiting electromagnetic or electrostatic forces to envelop a target in a magnetic or electric vector field, respectively, and generate necessary dissipative forces to "capture" it.

Every method has his own advantages and disadvantages but there is unfortunately not one that can tackle all possible targets. Moreover, even considering one specific target, it is not guaranteed to be able to easily identify its most suited method(s), as demonstrated by the e.Deorbit study [9]. Furthermore, the available data about cataloged objects often lacks coherence and structure, thus hindering data sharing, collaboration and ultimately decision-making. In fact, it might be argued that the current space debris issue is plagued by the information paradox [10] where we appear to be: "... drowning in information but starved for knowledge" [11].

In this context, we present a knowledge representation framework for the characterization of space debris for ADR capture methods selection in the form of a domain-ontology. The latter is defined in our research as a method to model a field of discourse by explicitly defining the domain concepts, relationships among them, their properties and restrictions [12]. This way, standardized, machine-interpretable vocabulary of characteristics of cataloged objects, and relations among them, can be used by ADR researchers to analyze the domain and infer new knowledge, making the initial ADR mission planning easier yet more systematic. The nature of our results is both theoretical and applied, since not only the methodology used to develop the domain-ontology is described, but also workflows used for its software implementation and usage. With respect to the existing state-of-the-art [13, 14, 15, 16, 17, 18, 19], focused mainly on space situational awareness (SSA), the ontology covers specifically the domain of ADR (or more specifically the domain of ADR capture methods) and establishes the minimal number of parameters needed to identify the most suited capture method(s) for a specific target. Additionally, it describes a method to handle the input of data from an existing database of cataloged objects, an aspect often overlooked by the current state-of-the-art.

The remainder of the article is structured as follows: Section 2 introduces the overall methodology and core terminology of the ontology. In Section 3 its software implementation and development workflows are described. Section 4 presents results of its application onto a database of representative objects. The analysis of those results, along with strengths and weaknesses of the ontology, are discussed in Section 5. Finally, Section 6 is dedicated to the concluding remarks and recommendations for future works.

2. Methodology and Core Terminology

The overall methodology of this work consists of the development of an ontology for ADR capture methods inference, based on physical and dynamical properties of potential targets. In order to do so, firstly the precise domain of interest and scope of the ontology are defined. Then, main classes and their hierarchy are determined. Finally, axioms allowing classification of ADR capture methods, associated with potential targets, are specified, based on the statistical analysis of on-orbit fragmentations and state-of-the-art capture methods characteristics.

2.1. Domain of interest and scope

The methodology used to develop the ontology is primarily a "Simple Knowledge-Engineering Methodology" [12], with some hints taken from the Unified Process for ONtology (UPON) method [20]. It consists of an iterative approach where at first a rough draft of the ontology is developed, starting from the initially defined requirements set and lexicon. Then, the lexicon is transformed into a glossary. Subsequently, the ontology is revised and enriched based on the analysis of the existing draft. After several iterations of the previous step, the ontology is finally formalized and tested using a representative set of data against the initial set of requirements.

The domain of the developed ontology is that of IDOs (i. e. payloads and rocket bodies) and ADR capture methods. Therefore, at least in its current version, it does not cover the domain of de-orbit technologies, although it is formulated with that domain in mind. The scope of the ontology is to create a standardized framework for collection, storage and sharing of characteristics of IDOs for ADR, by leveraging attributes of modern knowledge representation techniques. This way not only a method for efficient storage of complex

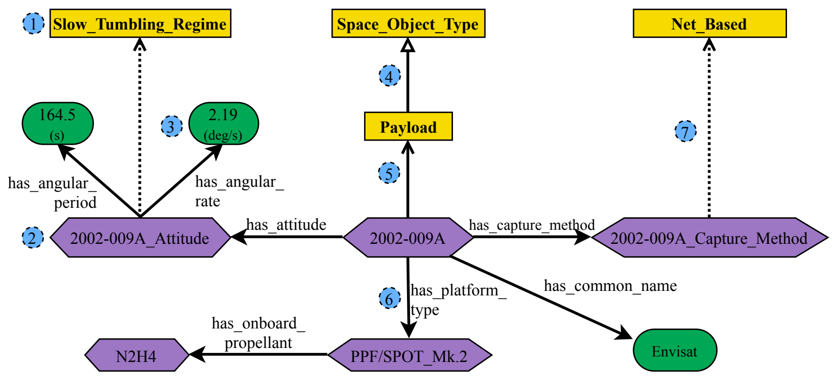

Figure 1: Exemplary diagram of the developed ontology. The conventions used in the diagram are the following: (1) yellow squares represent classes, (2) purple hexagons represent individuals, (3) green rounded squares depict numerical or string values, (4) closed hollow arrows portray subclass or subproperty relations, (5) opened arrows illustrate "type of" relations, (6) semi closed solid arrows depict properties of individuals, (7) dashed arrows portray inferred axioms.

information is provided but also a platform for further analysis of the domain knowledge from the existing data. One such analysis is provided with the ontology and consists of inferring, for a specific object, the most suited ADR capture method(s) and provide a human-readable explanation of the inference.

The competency questions (CQs) the ontology should provide answers to are identified as the following:

CQ1 How could a domain knowledge about IDOs be captured in a standardized, formal, machine-interpretable way useful to ADR?

CQ2 What are the minimum parameters needed to characterize an IDO for an ADR capture phase?

CQ3 How can the degree of hazard of an IDO to an ADR capture phase be represented?

CQ4 How could the most suited ADR capture method be inferred?

CQ5 How could the input of data into the ontology be simplified and made compatible with an existing space debris catalog, such as the Database and Information System Characterising Objects in Space (DISCOS) of the European Space Agency (ESA) [1]?

The intended users of the ontology are space debris domain experts, ADR mission planners and decision makers, that should have at their disposal a standardized way for data collection, storage and access of complex domain knowledge, such as that of IDOs and ADR. By using such a framework the parameter space of each object can be kept hidden from the user, as much as possible, and usage of a semantic reasoner allows for queries and knowledge inference from the existing data.

2.2. Main classes and class hierarchy

Among the existing knowledge representation languages, this work considers the Web Ontology Language (OWL 2), developed by the World Wide Web Consortium (W3C) for the Semantic Web. Ontologies developed with OWL 2 store the information about the domain of interest into Semantic Web documents, capable of representing classes, properties, individuals and relationship among them (see Fig. 1) [21].

Classes are the main building blocks of an OWL 2 ontology and their identification generally involves analyzing and extracting terms from the existing documentation, technical manuals, standards and/or similar ontologies [12, 20]. This was done in our work by extracting the most important terms from the preceding taxonomy of LEO space debris [8, 22], whose purpose was similar, albeit more limited, due to the inherent constraints of the used knowledge representation method (i. e. a taxonomy).

Classes identified as sufficient to characterize an IDO, in an unambiguous manner for a capture maneuver, and included within the developed ontology are: Attitude_Regime, Onboard_Propellant, Space_Object_Type, Breakup_Criticality and ADR_Capture_Method. Additional classes, i. e. Launch_Vehicle, Orbital_Regime, Spacecraft_Platform and Stage_Type, are introduced for completeness, however are deemed nonessential for the purpose of this work. Inclusion of further classes although possible, was excluded in order to keep the ontology simple and clutter free.

The hierarchy of those classes consists, as illustrated in Fig. 2, of three levels, each specifying a domain concept into more detail. Starting from the top of Fig. 2, the Attitude_Regime class characterizes instances describing the attitude state of an object. It contains two additional layers specifying the exact type of attitude state of an object, which in our study is confined to either being stable or tumbling (i. e. Stable_ or Tumbling_Regime). The former is defined as a state where the angular velocity of a target object is equal to zero $\deg\,\mathrm{s}^{-1}$ i. e. $\boldsymbol{\omega}_t = 0\deg\,\mathrm{s}^{-1}$. The latter is instead further detailed into fast, medium and slow tumbling regimes (i. e. Fast_, Medium_ and Slow_Tumbling_Regime subclasses), each defined as the angular rate of an object being: $0 < \boldsymbol{\omega}_t < 5\deg\,\mathrm{s}^{-1}$, $5 \leq \boldsymbol{\omega}_t < 18\deg\,\mathrm{s}^{-1}$, $18 \leq \boldsymbol{\omega}_t < \infty\deg\,\mathrm{s}^{-1}$, respectively [8].

The Onboard_Propellant class defines the propellant type of the main propulsion system of an IDO, or better of its platform (i. e. bus or propulsion unit, in case of a payload or rocket body, respectively). Based on the state of matter of the oxidizer and fuel used by the propulsion system, a distinction is made between Liquid_, Solid_ and Hybrid_Propellant [23]. Two additional classes are added for completeness, i. e. No_Propellant and Other_Propellant, in case of spacecraft having no propulsion system or having an onboard propellant that cannot be classified within one of the previous classes (e. g. butane, xenon, etc.). Lastly, the Liquid_Propellant class, is further subdivided into Cryogenic_, Hypergolic_ and Petroleum_Propellant subclasses, based on the specific oxidizer-fuel mixture used by the propulsion system, as defined in [23].

The type of IDO is specified with the Space_Object_Type class, which is

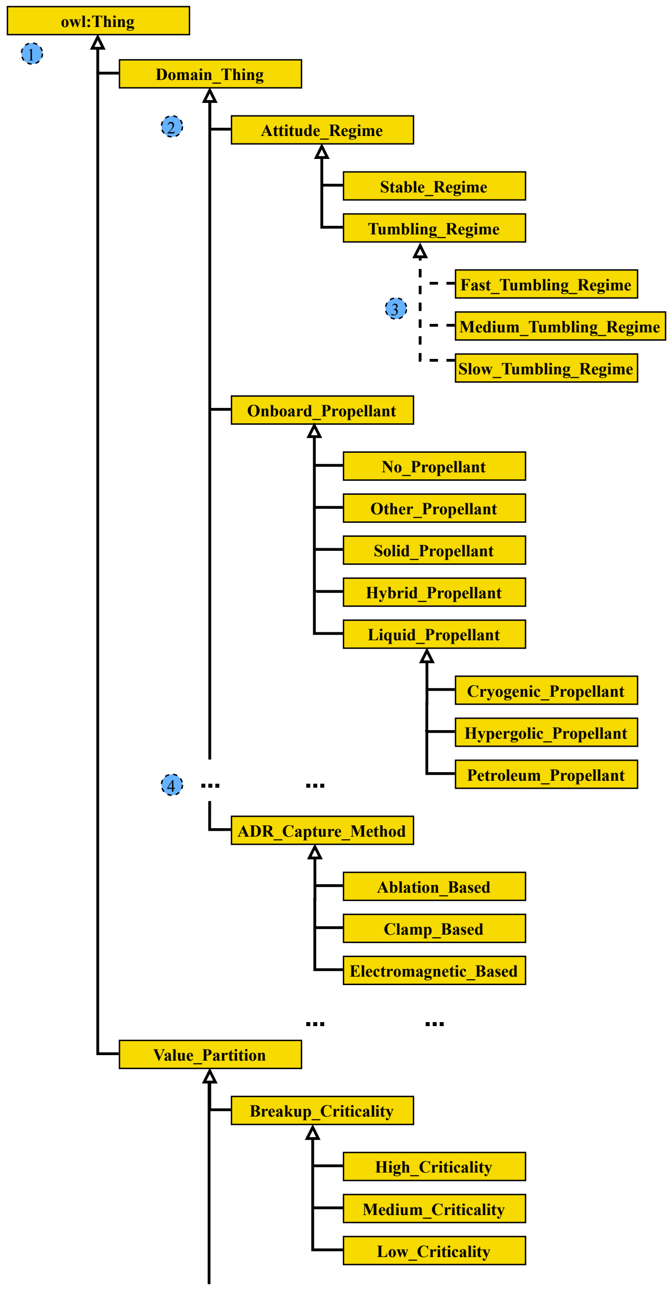

Figure 2: Extract of the class hierarchy from the developed ontology. The conventions used in the diagram are the following: (1) yellow squares represent classes, (2) closed hollow arrows portray subclass relations, (3) dashed arrows portray inferred axioms, (4) three dots indicate an omission of the existing structure.

Table 1: Criticality matrix (adapted from Table 5-3 of [24]).

Severity Level	Severity Number (SN)	Probability Limits			
		$\leq 10^{-4}$	$\leq 10^{-2}$	$\leq 10^{-1}$	$>10^{-1}$
		Probability Number (PN)			
		1	**2**	**3**	**4**
Catastrophic	**4**	4	8	12	16
Critical	**3**	3	6	9	12
Major	**2**	2	4	6	8
Negligible	**1**	1	2	3	4

further subdivided in the two most relevant object types identified in this work, i. e. Payload and Rocket_Body, defined as in [2].

The remaining main classes, i. e. Breakup_Criticality and ADR_Capture_Method, represent the core of the developed ontology, as they are instrumental in the ADR capture method inference.

The Breakup_Criticality class establishes the breakup hazard of an object, due to its inherent probability of breakup, as a consequence of variety of causes, as defined in [2].

The ADR_Capture_Method class instead defines the suitability of an ADR capture technology, described in Section 1, to capture an object based on its breakup criticality and level of uncooperativeness. The latter is defined as the degree of difficulty to capture a particular object due to its dynamical and physical properties [8].

More details about both of these concepts and how they influence the classification of capture methods are outlined in the next two subsections.

2.3. Breakup criticality definition

In this work, the breakup criticality of an object is formalized in alignment with ESA's standard on failure modes, effects (and criticality) analysis (FMEA/FMECA) [24]. Therefore, it is defined as a combination of a severity and probability of occurrence of a fragmentation event (i. e. the associated severity and probability numbers, identified hereafter with symbols SN and PN, respectively), having the worst possible consequences for a capture maneuver (see Table 1). Consequently, a distinction is made not only between different object types, e. g. rocket bodies and payloads, but also between non-passivated and passivated objects [8].

An object is characterized with a high criticality level (i. e. high criticality number (CN)) and is considered as critical for capture when one of the following conditions is fulfilled [8]:

- the severity of its worst fragmentation event is defined as catastrophic, i. e. its SN = 4

- the criticality number of its worst fragmentation event is greater or equal to eight, i. e. its CN ≥ 8 (see Table 1).

In these cases, any close contact with a target is to be avoided and only methods capable of achieving a capture from a significant stand-off distance (e. g. $\geq$50 m) should be considered. Moreover, in these cases special care should be exerted during the capture and stabilization maneuvers to avoid shocks and sources of sparks that might trigger a catastrophic breakup [8].

The remaining criticality levels are defined in this work as:

medium if the criticality number of its worst fragmentation event is equal to six, i. e. its CN $= 6$

low if the severity level of its worst fragmentation event is considered negligible, or its criticality number is lower or equal to four, i. e. its CN ≤ 4.

In these cases, several capture methods can be employed, depending not only on the CN but also on physical and dynamical properties of an object, such as its type, attitude regime, etc., as it will be described in the next subsection.

The worst possible breakup event of rocket bodies was identified as either an anomalous or propulsion event, depending on whether the spacecraft were passivated or not, based on a statistical analysis of related fragmentations extracted from DISCOS. The severity of those events, distinguished by the onboard propellant type, is listed in Table 2. The "No. of fragments" column details the observed median number of fragments per breakup event. The threshold of 1.05 years in the "Parent objects" column indicates the orbit age of an object (from launch) after which a propulsion related event can be expected to be statistically less severe, possibly due to a depletion/venting of most of the stored on-board propellant.

The probability of occurrence of those events was estimated using the survival analysis that was successfully employed to estimate the reliability of spacecraft from incomplete data [25]. For this purpose, the data was collected and processed from DISCOS, containing information about all large rocket bodies (not related to any manned mission) from October 1957 till July 2019, resulting in a total of 5185 cataloged objects and 15 555 observed (breakup and censored) events. Censoring occurs either because: a) an object has reentered, b) the reentry date of an object is beyond the observational window (in our case July 2019), c) the reentry date of an object is unknown. Furthermore, during the analysis of a particular breakup, all other breakups form an additional source of censoring that needs to be accounted for [25]. The survival function $S(t)$ of the population of objects within the database, and therefore the probability function from which it is derived, $P(t) = 1 - S(t)$, was estimated using the non-parametric Kaplan-Meier estimator [26]. While other methods exist, this one was chosen for its accuracy, as it does not fit any pre-defined distribution and is based on the actual data [25].

In case of payloads, the worst possible breakup event of non-passivated spacecraft was found to be a combination of anomalous and electrical events. The

Table 2: Severity numbers of the worst fragmentation events of rocket bodies in terms of the medium number of fragments (RBs = rocket bodies; N/A = not applicable).

SN	Breakup class	Propellant type	No. of fragments	Parent objects
4	Propulsion	All	$\geq$200	Non-passivated RBs with orbit age $\leq$1.05 years
3	Propulsion	Hypergolic	79.5	Non-passivated RBs with orbit age >1.05 years
2	Propulsion	Cryogenic	7.5	
1	Propulsion	Petroleum & Solid	2.5	
1	Anomalous	N/A	1	Passivated RBs

combination of anomalous, collision and unknown events was instead taken into account in case of passivated spacecraft. The severities of those events were both assigned the major severity level, i. e. SN = 2, due to their similar median number of generated fragments. The probability of occurrence of mentioned events was estimated, as in case of rocket bodies, using the Kaplan-Meier estimator applied on data consisting of 5002 payloads and 20 008 observed (breakup and censored) events. However, since in both cases the dominant probability function was that of anomalous events (see Fig. 3), no significant difference was found between the derived probabilities.

2.4. ADR capture method selection

In this work, the selection of the most suited ADR capture method(s) for a specific target is determined via the evaluation of its previously defined breakup hazard and degree of uncooperativeness. This way, not only the safety of the capture maneuver is considered during the evaluation, but also its degree of difficulty, which is manifested by the physical and dynamical properties of an object.

The characteristics of an object used to define its degree of uncooperativeness are: 1) the angular rate, 2) existence of a dedicated grapple feature, 3) type of material of the capture interface, 4) mechanical clearance of the capture interface. The definition of those characteristics is summarized in Table 3 and detailed hereafter based on the perceived capabilities of the current state-of-the-art capture methods. Therefore, the threshold values used to define them in this article are provided as default ones. Users are encouraged to modify them or even introduce new ones using software tools presented in the next section.

The **angular rate** is expressed in the ontology as a data property and describes the attitude state of an object. The thresholds defining the tumbling states, visible in Table 3, are chosen based on: a) the maximum value of the relative angular rate that a state-of-the-art robotic manipulator should be able cope with (i. e. $5\,\mathrm{deg\,s^{-1}}$) [27, 28], b) the value of the relative angular rate

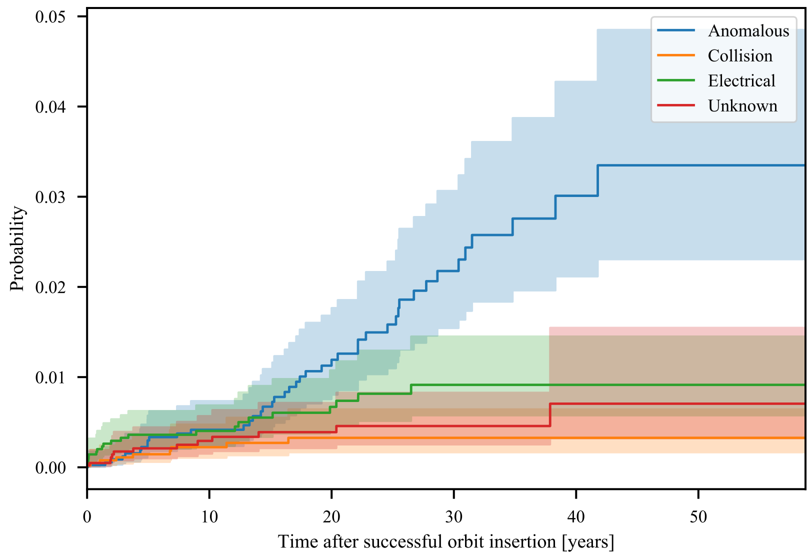

Figure 3: Breakup probabilities (with 95 % confidence intervals) of large, unmanned payloads due to anomalous, collision, electrical and unknown causes calculated using the Kaplan-Meier estimator.

Table 3: Summary of characteristics of a target used to determine its degree of uncooperativeness (ω_t = angular velocity of a target; A = area centered on the capture interface of a target) (adapted from Table 6 of [22]).

Characteristic	Definition
Angular rate	Stable: $\boldsymbol{\omega}_t = 0\,\mathrm{deg\,s^{-1}}$ Slow tumbling: $0 < \boldsymbol{\omega}_t < 5\,\mathrm{deg\,s^{-1}}$ Medium tumbling: $5 \leq \boldsymbol{\omega}_t < 18\,\mathrm{deg\,s^{-1}}$ Fast tumbling: $18 \leq \boldsymbol{\omega}_t < \infty\,\mathrm{deg\,s^{-1}}$
Grapple feature existence	True: *"dedicated" grapple feature exists* False: *"dedicated" grapple feature does not exist*
Capture interface material	Isotropic: *e. g. metal, ceramics, polymer* Anisotropic: *e. g. composite materials*
Capture interface clearance	Narrow: $A < 0.28\,\mathrm{m}^2$ Broad: $A \geq 0.28\,\mathrm{m}^2$

above which any synchronization effort would be considered very difficult (i. e. $18\,\mathrm{deg\,s^{-1}}$) [29]. Consequently, objects having angular rates greater or equal to $18\,\mathrm{deg\,s^{-1}}$ are assigned a high degree of uncooperativeness and should be captured only with contactless "capture" methods [8].

The **grapple feature existence** is defined as a Boolean data type property. A "dedicated" grapple features, is identified in this work as a surface feature, with a regular enough geometry (e. g. launcher adapter ring (LAR), common on many spacecraft), that can be easily grappled. Otherwise, a capture has necessarily to be performed on some other feature, not envisioned to be grappled, or even a surface [8]. Therefore, the existence of a grapple feature is considered advantageous for the capture maneuver since it can be approximated with a more common berthing operation, commonly used in the context of loading/unloading of cargo from the International Space Station.

The **capture interface material** is defined as a string data type property and reflects the versatility (and reliability) of a capture method. Considering the directional dependent mechanical properties of an isotropic type of material, it is associated with capture methods capable of distributing applied contact forces (such as clamp or net-based methods). An anisotropic material is, on the other hand, expected to be able to withstand concentrated loads, independent of their direction, which is why it is instead correlated with capture methods that are anticipated to exert such forces (such as manipulators or harpoon-based methods) [8].

The **capture interface clearance** is expressed as a string data type. It reflects the overall complexity of the approach and capture operations and is defined as an area, A, enclosed by a circle centered on the capture interface. The threshold value is defined using the combination of ESA's recommendations on the mechanical clearance of mechanisms [30] and the value of a maximum achievable precision of a typical guidance, navigation & control (GNC) system, in all three axis, during a berthing maneuver (i. e. $0.1\,\mathrm{m}$), using the following formula $A = \pi(3 \times 0.1)^2$ [8, 31]. The smaller the interface clearance, the more precise the capture maneuver needs to be. Therefore, the smaller the interface clearance of a target, the higher its degree of uncooperativeness is.

Using the defined traits, the most cooperative targets are identified as those having: stable to low tumbling attitude regimes and a "dedicated" grapple feature. The most uncooperative targets are defined instead as those having fast tumbling attitude regimes, irrespective of other features.

The ADR capture method(s) selection for a particular target, i. e. the classification of the associated ADR capture method(s) instance(s), is performed by applying the class axioms detailed in Table 4 to the ADR_Capture_Method subclasses.

As in case of the definition of characteristics of target objects, these axioms are outlined based on the perceived capabilities of the current state-of-the-art capture methods. Thus, they are provided as a template within the developed ontology. Users are encouraged to modify them or even introduce new ones using software tools presented in the next section.

From the outlined axioms it can be deduced that **manipulator-based**

Table 4: Class axioms used for the classification of ADR capture methods (N/A = not applicable; RB = rocket body; PL = payload).

Capture method subclasses	Object properties						
	Object type	Breakup criticality	Passiv. state	Attitude regime	Grapple feature	Capture interface	
						Material	Clearance
Manipulator_Based	N/A	Low	N/A	Stable-Medium tumbling	True	N/A	N/A
Clamp_Based	RB	Low	N/A	Stable-Medium tumbling	False	N/A	Broad
Net_Based	N/A	Low-Medium	N/A	Stable-Medium tumbling	N/A	N/A	N/A
Harpoon_Based	PL	N/A	True	Stable-Slow tumbling	False	Isotropic	N/A
Plume_Impingement	PL	Low-Medium	N/A	Fast tumbling	N/A	N/A	N/A
Electromagnetic_Based	RB	Low-Medium	N/A	Fast tumbling	N/A	N/A	N/A
Ablation_Based	PL	N/A	True	Fast tumbling	N/A	N/A	N/A
No_Solution	N/A	High	False	Fast tumbling	N/A	N/A	N/A

methods are designated as a preferred solution for capturing objects (irrespective of their type) having: a low breakup criticality, stable to medium tumbling attitude and "dedicated" grappling feature. A non-existing grappling feature would make any usage of a manipulator-based capture method more complex and less safe. Therefore, in these cases the most suited capture methods are either **clamp- or tether-based**, selected on the basis of object type, capture interface material and required clearance.

Clamp-based solutions are deemed suitable only for rocket bodies, due to their expected lack of appendages that would otherwise complicate capture maneuvers.

Harpoon-based solutions, in contrast, are associated with payloads, due to the expected higher efficacy of these methods on flat surfaces rather than on the curved ones, commonly found on rocket bodies.

Finally, contactless solutions are coupled exclusively with objects having high levels of uncooperativeness, since any capture effort using the previously mentioned methods would be considered very difficult and expensive (in terms of fuel). The selection between different contactless methods is made on the basis of the estimated breakup criticality of objects, i. e. on the "required" stand-off distance that each method requires to achieve a successful "capture" maneuver. Furthermore, **plume impingement** and **ablation-based methods** are deemed more suitable for payloads, considering that the efficacy of both is maximized on flat surfaces rather than on the curved ones [32]. However, payloads usually present a small percentage of conductive material, with respect to their overall mass, making them challenging targets for electromagnetic-based methods [32]. As a consequence, **electromagnetic-based methods** are bound, in this work, to rocket bodies, usually containing a large percentage of conducting material with respect to their overall mass.

No solution has been found suitable for targets having: a high breakup criticality, fast tumbling attitude regime and non-passivated state. The reason behind this result arises from the current unavailability of ADR capture methods that could safely tackle targets having those characteristics.

3. Ontology Implementation

The domain-ontology developed within this research is implemented within the onTology foR ACtive dEbris Removal (TRACER) repository hosted on Zenodo[3] and GitHub[4] platforms. At the moment, the accessibility to the repository on both hosting platforms is restrained. However, our near future goal is to provide open access to it, under the new BSD license, in order to simplify and encourage further development of the library.

The development workflow of TRACER is divided into two processes: the ontology database generation and ontology implementation. The former in-

[3] `https://zenodo.org/`
[4] `https://github.com/`

cludes activities allowing collection and pre-processing of raw structured and unstructured data, using the Python programming language. The latter consists instead of activities leveraging the output of previous tasks to implement the methodology, described in Section 2, employing the ontology editor Protégé. Moreover, the process includes tasks to import the desired data into TRACER and perform unitary tests to assure ontology consistency.

3.1. Data sources and pre-processing

The data sources considered in TRACER are both, structured and unstructured, due to the current nonexistence, to best of our knowledge, of a single source of data containing all the information required for ADR capture methods selection, defined in Section 2. ESA's DISCOS database comes very close with respect to the required amount of data and provides a machine-to-machine interface: the DISCOSweb application programming interface (API)[5], through which retrieval of structured object data is possible. For this reason, TRACER is developed to "interface" with the output of the DISCOSweb API, although a manual query to the maintainers of the database may be required due to the existing access restrictions and limited default information available through the API itself.

The structured data expected by TRACER consists of: the identification properties of potential ADR targets, their physical dimensions, orbital properties, launch and reentry dates (if any), activity status, onboard propellant type, launcher name and country of origin.

Additional information needs to be provided manually using unstructured sources that might consist of: web resources, such as Encyclopedia Astronautica [33], Gunter's Space Page [34], Earth Observation Portal [35], RussianSpaceWeb.com [36]; user's manuals/guides of launchers, such as Ariane 5 [37], Atlas V [38], etc.; academic publications in the fields of space debris monitoring and modeling [39, 40, 41, 42].

The unstructured data expected by TRACER consists of: the attitude states of potential targets, their passivation states, onboard fuels, buses or propulsion unit types, grappling feature existence and potential capture interface properties.

Once the required data is extracted from the chosen sources, the pre-processing step of the workflow involves integrating that data and creating a database of the ontology individuals compatible with TRACER. The integration of data is done via a custom script written in Python 3 (version 3.6.8) programming language. The script leverages the Python's `pandas` and `numpy` libraries (versions 0.24.2 and 1.16.4, respectively) to import, merge, format and manipulate the extracted data accordingly. For the computation of breakup probabilities, the Python library `lifelines` [43] (versions 0.21.0) is used within the mentioned script. The `lifelines` module, built on top of the `pandas` library, implements

[5] https://discosweb-api.sdo.esoc.esa.int/

in Python the survival analysis and provides an API to effectively estimate the probability of an event based on historical data.

3.2. Data import using `Owlready2` Python library

The next step in the implementation consists of importing data into the domain-ontology, while minimizing the required user input and therefore the possibility of human error. For this purpose, the experience accumulated within the Robotics Innovation Center (RIC)–DFKI GmbH, in the field of knowledge representation of the robotics domain [44, 45], is harnessed and a method for the manipulation of semantic models, currently being developed within the D-Rock[6] and Q-Rock[7] projects of the institute, is employed. The method allows transparent manipulation of ontologies using the Python 3 programming language and an ontology programming interfaces, `Owlready2` (version 0.18) [46]. `Owlready2` module allows to load OWL 2 ontologies as Python objects, manipulate them, perform reasoning via `HermiT` or `Pellet` semantic reasoners, and save them. It uses similarities between object models and ontologies to enable high level access to OWL 2 ontologies via Python notation, therefore allowing an easy-to-use and highly concise syntax. However, `Owlready2` currently supports a limited functionality when it comes to knowledge inference compared to other tools, such as the Protégé editor (introduced in the next subsection). For this reason, it is not considered in TRACER for classification purposes.

3.3. Ontology implementation with Protégé desktop

Protégé is an open-source framework providing users with a suite of tools to develop, edit and manage domain models. It fully supports the Semantic Web standards of the W3C, such as the Resource Description Framework (RDF) and OWL 2 [21, 47]. Furthermore, it allows a connection to description logic reasoners (i. e. semantic reasoners) that might be used to probe an ontology for inconsistencies as well as to infer new knowledge. Finally, the program provides an option to display human-readable inference explanations, useful to verify justifications of inferred classifications [48]. In view of these considerations, the Protégé Desktop (version 5.5.0) is selected in this research as the tool of choice for the implementation, testing and visualization of results of TRACER.

The particularities of TRACER implementation consist of: a) a minimal hierarchy of classes, b) an ontology design pattern (ODP) usage, c) a differentiation between space debris objects and their key characteristics, in terms of individuals.

The minimal classes hierarchy is intentional in order to reduce the user input and leverage instead the capability of a semantic reasoner to classify the ontology autonomously (i. e. to compute the class hierarchy from the imposed axioms).

The ODP used in TRACER is the Value Partition. It is considered as a "good practice" in ontology development since its usage allows more robust,

[6] https://robotik.dfki-bremen.de/en/research/projects/d-rock.html

[7] https://robotik.dfki-bremen.de/en/research/projects/q-rock.html

cleaner and easier to maintain ontologies [49]. The pattern addresses the problem of representing a descriptive feature of an object with a constrained set of possible values (also known as feature space) [50, 49]. An example in our case can be the "breakup criticality" of a space debris object, which can be regarded as a feature to be represented and its values "high", "medium" and "low" as its feature space. This way attributes are separated from elements being described, enforcing an easier to maintain modeling [49].

Lastly, the design choice was made to differentiate between space debris object themselves and their key attributes, in terms of individuals, similarly to what was done in [15, 16, 18]. The relationship with a parent object is guaranteed via object properties. In this manner, six additional individuals/instances are created for each space debris object describing its: 1) attitude, 2) orbit, 3) breakup criticality, 4) probability, 5) severity, 6) capture method. Two further individuals/instances are used to represent the platform type (i. e. bus or propulsion unit, in case of a payload or rocket body, respectively) and onboard propellant of an object, allowing their re-use, in case of objects having the same platform and/or onboard propellant. Consequently, a total of eight individuals/instances are used in TRACER to compliment the data properties expressed within the space debris object instance.

The advantages of such an implementation, especially when used in combination with Protégé, comprise of an easy to use and maintain ontology in which the characteristics of an object are represented as data properties while the contextual information and relationship with other individuals as object properties. Furthermore, the deductive reasoning capabilities of an ontology can be extended, if required, via the Semantic Web Rule Language (SWRL) [51], for which Protégé provides a development environment. This way, the `orbit_age` of an object, for example, can be deducted by a reasoner, based on the defined SWRL rules, instead of being derived in the pre-processing step. However, the provided SWRL development environment was found to be very limited in terms of debugging capabilities and was therefore dismissed in favor of a more familiar Python environment, as illustrated in the previous subsections.

The disadvantages of the described implementation are mainly tied to "limitations" of the Protégé editor itself which requires a quite pedantic and repetitive insertion of class restrictions, individuals and properties, without the possibility to automatize the process within the editor itself. For this reason, the `Owlready2` is used to populate TRACER with the required data and might be used in the future to manage the developed ontology in more depth, using a more familiar Python environment.

4. Ontology Application

The effectiveness of the developed ontology was tested by applying it onto a database of representative objects. The employed application workflow is illustrated in Fig. 4 and consists of two main processes: the data input and ontology query. The former encompasses the collection, pre-processing and input of desired data into TRACER, while the latter involves the knowledge

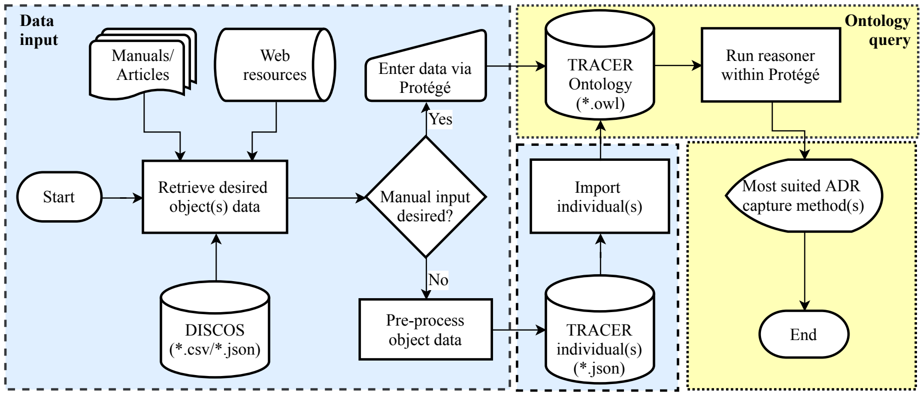

Figure 4: Usage workflow of TRACER (see ISO 5807:1985 for symbols definition)

inference from TRACER, via a software reasoner and query of results via a GUI.

4.1. TRACER example database

The example database created for the evaluation of TRACER, consists of 210 ontology individuals, related to 30 large intact cataloged objects (i. e. 19 payloads and 11 rocket bodies), for which we were able to obtain information about their attitude states, while avoiding the inclusion of objects having identical properties (i. e. bus/propulsion platform and angular rate).

The database was assembled using the available data from DISCOS and the unstructured data sources mentioned in Section 3.1. Only one of the considered objects has at the time of writing[8] an active status, i. e. having the international designator (COSPAR ID) 2014-037A. The rest, either have an unknown status or can be considered inactive, being deactivated on purpose or having experienced a catastrophic failure which forced their early end-of-life disposal. The characteristics of the first 10 objects of the database are illustrated in Table 5.

The most represented platforms within the database are the: Uragan Block IIv bus (with five related objects), H10 propulsion unit (with two related objects), Étage à Propergols Stockables (i. e. Storable Propellant Stage) (EPS) L9 propulsion unit (with two related objects) and "ADEOS" bus (with two related objects).

The distribution of objects within the orbital classes, as defined in DISCOS, is the following: LEO: 15 objects, medium Earth orbit (MEO): eight objects, GEO transfer orbit (GTO): four objects, highly eccentric Earth orbit (HEO): two objects and GEO: one object.

The median in-orbit age (since launch) of objects was found to be 22.89 and 19.3 years (with interpercentile ranges being 13.7 to 25.51 and 17.55 to

[8] July 2019.

Table 5: Characteristics of first 10 objects of the database (RB = rocket body; PL = payload; LEO = low Earth orbit; MEO = medium Earth orbit; GTO = geostationary Earth transfer orbit).

COSPAR ID	Type	Orbit	CN	Passiv. state	Angular rate (deg/s)	Orbit age (years)
1978-018B	RB	LEO	3	False	67.8	41.96
1978-121A	PL	LEO	6	False	2	41.11
1989-001B	PL	MEO	6	False	38.88	31.06
1990-005H	RB	LEO	3	True	0	30.03
1990-045A	PL	MEO	6	False	38.96	29.71
1991-084C	RB	GTO	6	False	1.74	28.13
1992-052A	PL	LEO	6	True	32.1	27.48
1993-061A	PL	LEO	6	False	2	26.36
1994-021A	PL	MEO	6	False	3.03	25.88
1994-021B	PL	MEO	6	False	8.41	25.88

25.89 years) for payloads and rocket bodies, respectively. Therefore, their median breakup probability values were derived to be equal to 3.05×10^{-2} and 2.55×10^{-2} (with interpercentile ranges being 1.35×10^{-2} to 3.61×10^{-2} and 9.86×10^{-3} to 3.15×10^{-2}), respectively.

The assumptions made during the consolidation of the database were the following:

- the passivated state of an object was assumed as True only where the documentation supporting such a state was found, e. g. existence of a deactivation date, flight numbers of Ariane launches, etc.
- the grapple feature existence was assumed as True and its material isotropic for all objects within the database, considering that they should all be fitted with either a LAR or a launch vehicle adapter (LVA) (or in some cases even with an Ariane structure for auxiliary payloads (ASAP)). Those interfaces while might be overall made of a composite material (especially true for LVAs), should all present metallic junction surfaces, considered to be suitable as potential grapple features.
- the failure date and type information recorded within the database refers to a catastrophic type of failure of an object which would force its premature shutdown, if at all possible.

4.2. TRACER classification results

The results of the inferred classification of ADR capture methods, related to the 210 individuals of the example database, are illustrated in Table 6 as well as Figs. 5 to 7.

Table 6: Results of the ADR capture methods classification with all numeric values being median a part from those within the "No. of individuals" column (CN = criticality number; PN = probability number; SN = severity number).

ADR capture methods class	No. of individuals	CN	PN	SN	Angular rate (deg/s)	Orbit age (years)
Ablation_Based	1	6	3	2	32.1	27.48
Electromagnetic_Based	1	3	3	1	67.8	41.96
Manipulator_Based	9	3	2	1	2	18.32
Net_Based	26	6	3	2	2	19.15
Plume_Impingement	3	6	3	2	38.88	29.71

The classification was performed using the `Pellet` semantic reasoner within the Protégé Desktop, on a 64-bit PC platform equipped with an Intel® Core™ i7-3630QM CPU, clocked at 2.40 GHz, and 16 GiB of RAM. The required average time for a classification was found to be 13.5 s. With lower numbers of individuals, i. e. 142 and 72 individuals (corresponding to 20 and 10 cataloged objects, respectively), the average inference computation time was found to be 3.51 s and 1.25 s, respectively.

Overall ten capture methods were classified simultaneously in two classes, suggesting that, under the considered assumptions, more than one capture method was found to be suitable for those targets. The two capture method pairs involved in the simultaneous classification were the following: (Manipulator_Based, Net_Based) and (Ablation_Based, Plume_Impingement). The number of occurrence of these results was nine times and one time, respectively.

Fig. 5 portrays the overall number of capture methods being classified within one or more ADR_Capture_Method subclasses of the ontology, along with their distribution over the two considered classes of target objects, i. e. payloads and rocket bodies. More specifically, 26 capture methods have been classified within the Net_Based class, nine within the Manipulator_Based class, three within the Plume_Impingement class and one within the Electromagnetic_Based and Ablation_Based classes, respectively.

The distribution of inferred classification over the breakup criticality numbers and attitude regimes of related targets are illustrated in Figs. 6 and 7, respectively.

5. Analysis and Discussion

The following section analyses the meaning of the results presented in the previous one and discusses strengths and weaknesses of the developed ontology, in light of the imposed competency questions defined in Section 2.1.

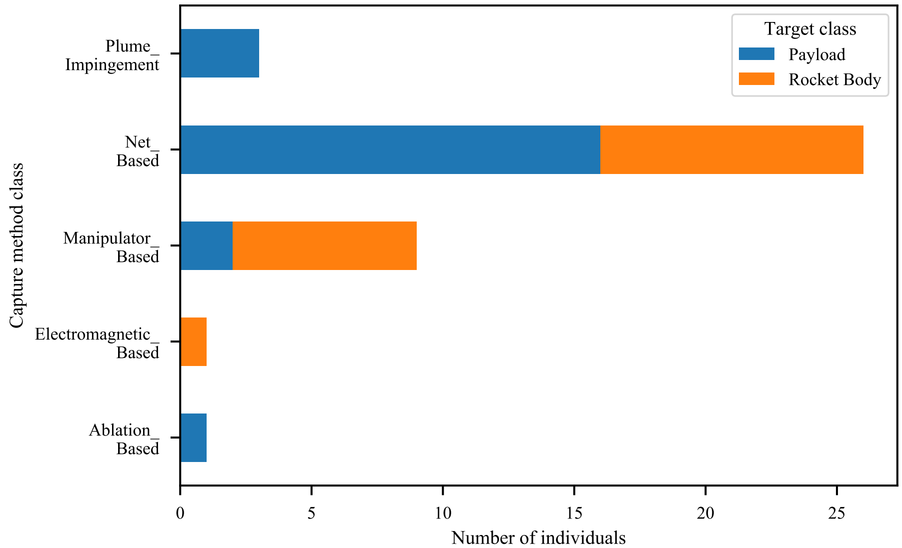

Figure 5: ADR capture methods classification results per target object class.

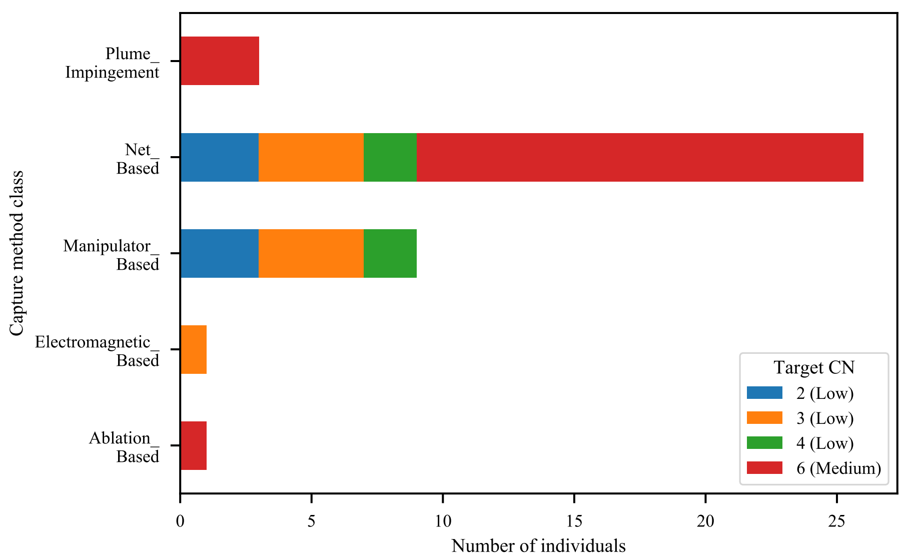

Figure 6: ADR capture methods classification results per target object criticality number (CN = criticality number).

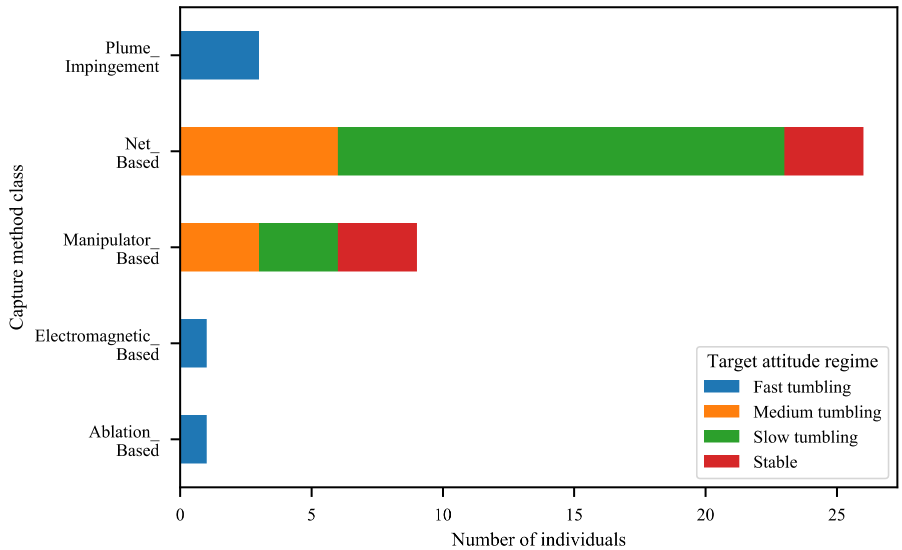

Figure 7: ADR capture methods classification results per target object attitude regime (as defined in Section 2.2).

5.1. Results analysis

The overall classification results (illustrated in Table 6) point toward the Net_Based class as the most numerous one, holding ≈87 % of all individuals of capture methods. The second most numerous class is the Manipulator_Based, holding a total of 30 % of individuals, while the remaining three classes are found to collect, in total, only 16 %. These figures were expected and are to be attributed to: the assumptions made during the creation of the example database (defined in Section 4.1) and restrictions imposed to the ADR_Capture_Method subclasses, (defined in Table 4). Indeed, the database is dominated by slow/medium tumbling objects, having low/medium breakup criticalities, which makes them ideal targets for contact-based capture methods, such as nets or manipulators. Furthermore, all targets were assumed to have a grapple feature that, once again, made them suitable for manipulator-based methods, especially in case of objects having low to medium tumbling rates and low breakup criticalities. Additionally, certain targets were associated with both net- and manipulator-based methods, as their capture methods satisfied more than one class membership constraint. This result is illustrated in Fig. 6, i. e. in the mirrored classification distribution of capture methods associated with targets having low CNs (i. e. $2 \leq \mathrm{CN} \leq 4$).

The classification distribution over target object classes, as depicted in Fig. 5, was also an anticipated result that can be justified by the overall higher number of payloads within the example database. In fact, the number of payloads within the database is almost double with respect to that of rocket bodies and this is reflected in the overall higher number of payloads being associated with the most generic capture method defined, i. e. the net-based method. The manipulator-based method is instead associated with a higher number of target rocket bodies, due to the higher number of rocket bodies (i. e. seven) with an overall lower median CN with respect to that of payloads.

The association of the remaining classes (i. e. Plume_Impingement, Electromagnetic_Based and Ablation_Based) with one target class or the other was also an awaited outcome that is a direct consequence of class restrictions specified in Table 4.

The influence of the class restrictions specified in Table 4 is particularly evident in Fig. 6 and 7. Indeed, the overall characteristics of target objects associated with net-based capture methods were found to be: low to medium criticality and stable to medium attitude regime, in line with the imposed restrictions. In case of objects correlated with manipulator-based methods, those characteristics were identified with: low criticality, stable to medium tumbling attitude regime and the existence of a grapple feature, once again matching the imposed restrictions of the axioms defined in Table 4. Finally, objects associated with contactless methods were characterized by low to medium breakup criticalities and high angular rates, once again as expected.

The total absence of individuals within the Clamp_Based or Harpoon_Based classes reflects the very nature of the example database, which does not include rocket bodies or payloads with missing grapple features.

5.2. Discussion

The overall scope of this work was twofold, i. e. to: a) create a framework for data collection, storage and sharing of characteristics of intact derelict objects, using a modern knowledge representation method, b) enable a discovery of new domain knowledge, such as the inference of the most suited ADR capture methods of objects. The presented results confirm the capability of the developed framework to fulfill the mentioned objectives, within the constraints defined in Section 2. In fact, using the developed framework a user can capture, in a standardized, formal and machine-interpretable way, the domain knowledge of IDOs (in particular of payloads and rocket bodies) useful to ADR. Furthermore, the framework provides a way to represent the degree of hazard of an IDO to an ADR capture phase, due to its fragmentation potential and inability to support a capture maneuver. Additionally, the framework displays how a knowledge representation method can be used to infer the most suited ADR capture methods, based on the existing data. Finally, the framework provides tools necessary to transparently handle the input data from an existing database of space debris into an ontology, a detail often overlooked, or at least not sufficiently documented, by the existing state-of-the-art methods. However, the required inference time points towards a framework with an exponential complexity, as demonstrated by the exponentially increasing computation time with the number of individuals. This suggests the limitation of the current implementation to deal with a classification of large number of space objects in one run. Indeed, in case of 100 space objects (which would roughly translate into 700 ontology individuals), the expected computation time[9] would require around 90 h. Nevertheless, the computation time can be drastically reduced by simply halving the number of objects to be classified in one run. In fact, considering 50 objects, or circa 360 individuals, to be classified within TRACER in one run, the expected computation time should be around 4 min. Hence, this limitation is not currently seen as a serious impediment of the framework, especially knowing that the developed repository includes automation tools adequate to classify large number of IDOs in batches of 50 objects at a time, for example. Moreover, in case of large groups of objects to be analyzed, consisting mainly of spacecraft of the same platform/bus (such as in case of planned mega-constellations), objects might be further grouped by attitude regimes (as defined in Section 2.4) and classification performed only on representative objects of each group, while extrapolating results for the rest. In this manner, the entire group can be analyzed without the need to process each single individual.

Additional limitations of the current version of TRACER consists of: a) its inability to classify all possible types of orbital regimes of parent objects, e. g. Extended Geostationary Orbit, b) the omission of shape, size and mass properties from characteristics used to determine the degree of uncooperativeness of a target (see Table 3), c) the nonexistence of a cost parameter within the class axioms used for ADR capture methods classification detailed in Table 4.

Further constraint of the developed framework is its dependence from the unstructured data that are not always easily retrievable, e. g. attitude states of cataloged objects, their onboard propellant types, passivation states, etc. Therefore, further research in these areas is considered of paramount importance, towards a goal of either building a comprehensive database of objects and their properties, or developing a machine learning algorithm that could infer the required data from the existing ones.

Finally, TRACER in its current form does provide an answer only to one phase of an ADR mission, i. e. the capture phase. Other phases, such as the close-range rendezvous or disposal (e. g. de-orbit) phases, have not been considered. Therefore, for TRACER to represent a comprehensive ADR planning solution, something that is out of the scope of the current article, all possible mission phases of a generic ADR mission should be included, e. g. as separate ontologies.

6. Conclusions

The ever-increasing population of space debris has long been recognized by the scientific community as a critical issue that needs to be addressed with urgency. However, choosing the right way to address this issue is currently a difficult task mainly due to the information paradox characterizing the space debris domain. Past studies have addressed this problem, in the context of SSA, with modern knowledge representation techniques, such as taxonomies,

[9] On the PC platform mentioned in Section 4.2.

ontologies or knowledge graphs. Nevertheless, none has explicitly addressed the domain of ADR and most of them appear to overlook the handling of the input data from existing databases of cataloged objects. In this work, we bridged this gap by developing a framework, in the form of a domain-ontology, for data collection, storage and sharing of characteristics of intact derelict spacecraft useful to ADR. The framework defines the minimal set of physical and dynamical parameters of an object deemed sufficient to infer, via a semantic reasoner, its most suited ADR capture method(s), safety wise. This way, not only the management, but also the discovery of the new knowledge is facilitated. At the same time, the framework is equipped with tools to transparently handle the input of data from an existing space debris catalog, i. e. DISCOS, thus reducing user input and consequently possibility of a human error. The practicality and validity of the developed framework were demonstrated by applying it onto a database of representative objects, for which we were able to obtain attitude states from publicly available resources. The overall classification results pointed toward net-based capture methods as the most frequently associated with target objects, followed by manipulator-based methods, as it was expected, considering the nature of objects within the example database and restrictions imposed onto ADR capture methods classes.

As future work, we expect to further develop the presented ontology to: a) address some of its current limitations, such as its inability to classify objects based their shape, size and mass, b) add more features, such as its ability to cope with missing data, c) extend its domain beyond ADR capture methods, in order to provide the community with a comprehensive tool for ADR missions planning.

Acknowledgments

The research presented in this article was supported in part by the EU Marie Curie Initial Training Network Stardust, FP7-PEOPLE-2012-ITN [grant agreement 317185]. Therefore, the authors would like to thank the European Commission and the Research Executive Agency for their support and funding. Moreover, this work was conducted using the Protégé resource, which is supported by grant GM10331601 from the National Institute of General Medical Sciences of the United States National Institutes of Health.

References

[1] ESA Space Debris Office, DISCOS - Database and Information System Characterising Objects in Space, https://discosweb.esoc.esa.int/, (accessed 7-July-2019) (2019).

[2] ESA Space Debris Office, ESA's Annual Space Environment Report, Tech. rep., ESA, Darmstadt, Germany, https://bit.ly/2LFgLxs (accessed 03-September-2019) (2019).

[3] V. Agapov, F. Alby, C. Bonnal, C. Cazaux, E. Christiansen, J.-C. Dolado-Perez, D. Finkleman, S. Kibe, H. Klinkrad, H. Krag, P. Krisko, J.-C. Liou, D. McKnight, T. Masson-Zwaan, C. Mathieu, M. Metz, D. Oltrogge, F. Schaefer, T. Schlidknecht, M. Sorge, IAA Situation Report on Space Debris - 2016, Tech. rep., International Academy of Astronautics (IAA), Paris, France, https://bit.ly/2o71m26 (accessed 03-September-2019) (2017).

[4] ESA, About Space Debris, https://bit.ly/2pFoHsb, (accessed 7-July-2019) (Feb. 2018).

[5] D. J. Kessler, B. G. Cour-Palais, Collision frequency of artificial satellites: The creation of a debris belt, J. Geophys. Res. 83 (A6) (1978) 2637–2646. doi:10.1029/JA083iA06p02637.

[6] Steering Group and Working Group 4, IADC Space Debris Mitigation Guidelines, Tech. rep., Inter-Agency Space Debris Coordination Committee (IADC), Vienna, Austria, https://go.nasa.gov/2o9ZntV (accessed 03-September-2019) (2007).

[7] M. Kaplan, B. Boone, R. Brown, T. Criss, E. Tunstel, Engineering Issues for All Major Modes of In Situ Space Debris Capture, in: AIAA SPACE 2010 Conference & Exposition, Space Department Applied Physics Laboratory, AIAA, Anaheim, California, USA, 2010, pp. 1–20. doi:10.2514/6.2010-8863.

[8] M. Jankovic, F. Kirchner, Taxonomy of LEO Space Debris Population for ADR Selection, in: 67th International Astronautical Congress (IAC), International Astronautical Federation (IAF), Guadalajara, Mexico, 2016, pp. 1–15.

[9] R. Biesbroek, T. Soares, J. Hüsing, L. Innocenti, The e.Deorbit CDF Study: A Design Study for the Safe Removal of a Large Space Debris, in: 6th European Conference on Space Debris, ESA Space Debris Office, Darmstadt, Germany, 2013, pp. 1–8.

[10] L. Orman, Information Paradox : Drowning in Information, Starving for Knowledge, https://bit.ly/2oKFSIe, (accessed 7-July-2019) (Jun. 2017).

[11] J. Naisbitt, Megatrends: Ten New Directions Transforming Our Lives, Warner Books, 1982.

[12] N. F. Noy, D. L. Mcguinness, Ontology Development 101: A Guide to Creating Your First Ontology, Tech. rep., Stanford Knowledge Systems Laboratory, https://stanford.io/2zxjaVz (accessed 03-September-2019) (2001).

[13] M. P. Wilkins, A. Pfeffer, P. W. Schumacher, M. K. Jah, Towards an Artificial Space Object Taxonomy, in: 2013 AMOS Conference, 2013, pp. 1–18.

[14] C. Frueh, M. Jah, E. Valdez, P. Kervin, T. Kelecy, Taxonomy and Classification Scheme for Artificial Space Objects, in: 2013 AMOS Conference, 2013, pp. 1–13.

[15] A. P. Cox, C. K. Nebelecky, R. Rudnicki, W. A. Tagliaferri, J. L. Crassidis, B. Smith, The Space Object Ontology, in: 19th International Conference on Information Fusion (FUSION), IEEE, Heidelberg, Germany, 2016, pp. 146–153.

[16] R. J. Rovetto, An ontological architecture for orbital debris data, Earth Sci. Inf. 9 (1) (2016) 67–82. doi:10.1007/s12145-015-0233-3.

[17] R. Furfaro, R. Linares, D. Gaylor, M. Jah, R. Walls, Resident Space Object Characterization and Behavior Understanding via Machine Learning and Ontology-based Bayesian Networks, in: 2016 AMOS Conference, 2016, pp. 1–14.

[18] B. Liu, L. Yao, D. Han, Harnessing ontology and machine learning for RSO classification, SpringerPlus 5 (1) (2016) 1655. doi:10.1186/s40064-016-3258-2.

[19] S. Le May, B. Carter, S. Gehly, S. Flegel, Leveraging Web data and graph structures to support rapid space object identification, in: 69th International Astronautical Congress (IAC), International Astronautical Federation (IAF), Bremen, Germany, 2018, pp. 1–10.

[20] A. De Nicola, M. Missikoff, R. Navigli, A software engineering approach to ontology building, Information Systems 34 (2) (2009) 258–275. doi:10.1016/j.is.2008.07.002.

[21] W3C OWL Working Group, OWL 2 Web Ontology Language Overview, https://bit.ly/2nWSb4n, (accessed 16-July-2019) (Dec. 2012).

[22] M. Jankovic, F. Kirchner, Taxonomy of LEO Space Debris Population for ADR Capture Methods Selection, in: M. Vasile, E. Minisci, L. Summerer, P. McGinty (Eds.), Stardust Final Conference, Advances in Asteroids and Space Debris Engineering and Science, 1st Edition, Springer, Cham, 2018, pp. 129–144. doi:10.1007/978-3-319-69956-1_8.

[23] R. A. Braeunig, Rocket propellants, https://bit.ly/2pDMV63, (accessed 23-July-2019) (2008).

[24] ECSS Secretariat, Space product assurance: Failure modes, effects (and criticality) analysis (FMEA/FMECA), Standard ECSS-Q-ST-30-02C, ESA-ESTEC, Noordwijk, Netherlands, https://bit.ly/2pFzhPV (accessed 03-September-2019) (Mar. 2009).

[25] J. H. Saleh, J.-F. Castet, Spacecraft reliability and multi-state failures: a statistical approach, 1st Edition, John Wiley & Sons, Ltd., The Atrium, Southern Gate, Chichester, West Sussex, PO19 8SQ, United Kingdom, 2011.

[26] E. L. Kaplan, P. Meier, Nonparametric Estimation from Incomplete Observations, J. Am. Stat. Assoc. 53 (282) (1958) 457–481. doi:10.2307/2281868.

[27] M. Castronuovo, Active space debris removal-A preliminary mission analysis and design, Acta Astronaut. 69 (9-10) (2011) 848–859. doi:10.1016/j.actaastro.2011.04.017.

[28] C. Bonnal, J.-M. Ruault, M.-C. Desjean, Active debris removal: Recent progress and current trends, Acta Astronaut. 85 (2013) 51–60. doi:10.1016/j.actaastro.2012.11.009.

[29] S. Matsumoto, Y. Ohkami, Y. Wakabayashi, M. Oda, H. Ueno, Satellite capturing strategy using agile Orbital Servicing Vehicle, Hyper-OSV, in: 2002 IEEE International Conference on Robotics and Automation (Cat. No.02CH37292), Vol. 3, IEEE, 2002, pp. 2309–2314. doi:10.1109/ROBOT.2002.1013576.

[30] ECSS Secretariat, Space Enginering: Mechanisms, Standard ECSS-E-ST-33-01C Rev.2, ESA-ESTEC, Noordwijk, Netherlands, https://bit.ly/2n9BJ00 (accessed 03-September-2019) (Mar. 2019).

[31] W. Fehse, Sensors for rendezvous and navigation, in: M. J. Rycroft, W. Shyy (Eds.), Automated Rendezvous and Docking of Spacecraft, 2008th Edition, Cambridge University Press, New York, USA, 2003, Ch. 7, p. 226.

[32] N. O. Gomez, Eddy currents applied to space debris objects, Ph.D. thesis, University of Southampton, https://bit.ly/2oa0J7Z (accessed 03-September-2019) (September 2017).

[33] M. Wade, Encyclopedia Astronautica, https://bit.ly/2pBCa45, (accessed 7-July-2019) (2019).

[34] G. D. Krebs, Gunter's Space Page, https://bit.ly/2oKqprF, (accessed 7-July-2019) (Jul. 2019).

[35] ESA, eoPortal, https://bit.ly/2pDGTlX, (accessed 7-July-2019) (2019).

[36] A. Zak, RussianSpaceWeb.com, https://bit.ly/2o56DqM, (accessed 7-July-2019) (Jul. 2019).

[37] Arianespace, Boulevard de l'Europe, BP 177 91006, Evry-Courcouronnes Cedex -France, Ariane 5 User's Manual, Issue 5, Revision 2, https://bit.ly/2pz0eVe (accessed 3-September-2019) (Oct. 2016).

[38] United Launch Alliance, LLC., 9100 East Mineral Circle, Centennial, CO 80112, Atlas V User's Guide, Revision 11, https://bit.ly/2oKM6I6 (accessed 3-September-2019) (Mar. 2010).

[39] J. Linder, Esther Silha, T. Schildknecht, M. Hager, Extraction of spin periods of space debris from optical light curves, in: 66th International Astronautical Congress (IAC), International Astronautical Federation (IAF), Jerusalem, Israel, 2015, pp. 1–9. doi:10.7892/boris.73954.

[40] T. Schildknecht, H. Krag, T. Flohrer, Determining, Monitoring and Modelling the Attitude Motion of Potential ADR Targets, https://bit.ly/2oQpnKD, proceedings of 2016 Clean Space Industrial Days (accessed 3-August-2019) (May 2016).

[41] T. Schildknecht, J. Silha, J.-N. Pittet, A. Rachman, Attitude states of space debris determined from optical light curve observations, in: 1st IAA Conference on Space Situational Awareness (ICSSA), Bern Open Repository and Information System (BORIS), Orlando, Florida, USA, 2017, pp. 1–4. doi:10.7892/boris.106946.

[42] J. Silha, T. Schildknecht, J.-N. Pittet, G. Kirchner, M. Steindorfer, D. Kucharski, D. Cerutti-Maori, J. Rosebrock, L. Leushacke, S. Sommer, P. Kärräng, R. Kanzler, H. Krag, Debris Attitude Motion Measurements and Modelling by Combining Different Observation Techniques, in: T. Flohrer, F. Schmitz (Eds.), 7th European Conference on Space Debris, Vol. 7, ESA Space Debris Office, ESA Space Debris Office, Darmstadt, Germany, 2017, pp. 1–12.

[43] C. Davidson-Pilon, J. Kalderstam, P. Zivich, B. Kuhn, A. Fiore-Gartland, L. Moneda, Gabriel, D. WIlson, A. Parij, K. Stark, S. Anton, L. Besson, Jona, H. Gadgil, D. Golland, S. Hussey, R. Kumar, J. Noorbakhsh, A. Klintberg, E. Martin, E. Ochoa, D. Albrecht, dhuynh, D. Medvinsky, D. Zgonjanin, D. Chen, C. Ahern, C. Fournier, Arturo, A. F. Rendeiro, CamDavidsonPilon/lifelines: v0.22.0, https://bit.ly/2oK7Nbr, (accessed 10-July-2019) (Jul. 2019). doi:10.5281/zenodo.3267531.

[44] M. Yüksel, L. M. V. Benitez, D. Zardykhan, F. Kirchner, Mechatronical design and analysis of a modular developed exoskeleton for rehabilitation purposes, in: 10th International Conference on Electrical and Electronics Engineering (ELECO), 2017, pp. 711–716.

[45] M. Eich, M. Goldhoorn, F. Kirchner, Semantic Labeling: Classification of 3D Entities Based on Spatial Feature Descriptors, in: IEEE International Conference on Robotics and Automation (ICRA), IEEE, Anchorage, Alaska, 2010, pp. 1–7.

[46] J. B. Lamy, Owlready: Ontology-oriented programming in Python with automatic classification and high level constructs for biomedical ontologies, Artif. Intell. Med. 80 (7) (2017) 11–28. doi:10.1016/j.artmed.2017.07.002.

[47] M. A. Musen, The Protégé project: a look back and a look forward, AI Matters 1 (4) (2015) 4–12. doi:10.1145/2757001.2757003.

[48] The Board of Trustees of the Leland Stanford Junior University, Protégé products, https://stanford.io/2oIrRuG, (accessed 11-August-2019) (2019).

[49] University of Manchester, Ontology Design Patterns (ODPs) public catalog, https://bit.ly/2nWZaKE, (accessed 11-August-2019) (2009).

[50] B. Rodriguez-Castro, M. Ge, M. Hepp, Alignment of Ontology Design Patterns: Class As Property Value, Value Partition and Normalisation, in: On the Move to Meaningful Internet Systems: OTM 2012, Springer Berlin Heidelberg, 2012, pp. 682–699. doi:10.1007/978-3-642-33615-7_16.

[51] I. Horrocks, P. F. Patel-Schneider, H. Boley, S. Tabet, B. Grosof, M. Dean, SWRL: A Semantic Web Rule Language Combining OWL and RuleML, https://bit.ly/2o5Q3Hq, (accessed 2-August-2019) (May 2004).

Vitae

Marko Jankovic is a space robotics researcher and project leader of the Stardust Reloaded project at the Robotics Innovation Center (RIC) of the German Research Center for Artificial Intelligence (DFKI) GmbH. His research mainly focuses on the development of solutions for Active Debris Removal and On-orbit Servicing. He was a visiting researcher at the Astronautics Research Group of the University of Southampton and ESA/ESTEC's Advanced Concepts Team. Furthermore, he was a research assistant at the DLR's Institute of Robotics and Mechatronics. His academic background consists of an ongoing Ph.D. in Engineering, M.Sc. in Astronautical Engineering and B.Sc. in Aerospace Engineering.

Mehmed Yüksel studied Systems Engineering, earning the bachelor and master degrees at the University of Bremen, Germany. He has also earned technician degrees at the High School for Technical Science of the University of Marmara, in the field of Telecomnunications, and at the Technical College Haydarpaşa, in the field of Control Systems in Istanbul, Turkey. He has majors in mechatronics, robotics and system design. He started working at the DFKI GmbH in 2008 in the field of search and rescue robotics. Since 2010, he has been working as a researcher at the DFKI GmbH in the fields of mobility and autonomous driving as well as robotic rehabilitation.

Mohammad Mohammadzadeh Babr is an electrical engineer with a specialization in machine learning, and programming. He is a research assistant at the DFKI GmbH since 2018 and his academic background consists of an M.Sc in Automation and Information Engineering, and a B.Sc in Electrical Engineering.

Francesca Letizia is a space debris engineer at IMS Space Consultancy for the European Space Agency (ESA) Space Debris Office, where her main tasks include the assessment of general compliance to space debris mitigation guidelines and of the contribution of a mission to the space debris environment. Previously, she was a mission analysis engineer at OHB Systems, where she worked on trajectory design and collision avoidance strategies for satellites with electric propulsion. Francesca holds a Ph.D. from the University of Southampton, where she graduated with a thesis on space debris clouds evolution and worked on the assessment of the compliance with planetary protection guidelines.

Vitali Braun is a space debris engineer at IMS Space Consultancy for the European Space Agency (ESA) Space Debris Office, where his main tasks include the maintenance of ESA's MASTER model and the DRAMA software suits, which are used in risk assessments and the debris mitigation compliance verification. Vitali holds a Ph.D. from the University of Technology in Braunschweig, where he worked on topics in the space surveillance area and developed methods to provide orbit solutions with predetermined bounded accuracy.