



Assessing component impairing at mission level

Romain Kervarc, Ariane Piel, Stephanie Lala, Jean Bourrely

► To cite this version:

Romain Kervarc, Ariane Piel, Stephanie Lala, Jean Bourrely. Assessing component impairing at mission level. *Mathematics and Computers in Simulation*, 2018, 147, pp.140-151. 10.1016/j.matcom.2017.05.006 . hal-01849116

HAL Id: hal-01849116

<https://hal.science/hal-01849116>

Submitted on 17 Aug 2018

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Assessing component impairing at mission level

Romain Kervarc, Ariane Piel, Stéphanie Lala, Jean Bourrely

*ONERA – the French Aerospace Lab
91123 Palaiseau, France
e-mail: [firstname].[lastname]@onera.fr*

Abstract

Whilst the amount of debris around the Earth steadily increases, the danger of collisions between debris and operating man-made space systems becomes a major issue. Whilst on the one hand fine physical simulation allows predicting accurately failure occurrences or manoeuvre consequences, and on the other hand fine behavioural models allow apprehending the services expected from a given system as an organisation of functions performed by various agents, it is difficult to have both approaches work together. This paper proposes a bridge between both approaches by developing a model describing in a unified way dependencies, redundancies, and interactions between physical components concurring to ensure elementary functions of a system, and shows how this approach was applied to the assessment of remedial measures to space debris issues for space assets.

Keywords: Space systems, System of systems, Performance assessment, Vulnerability assessment, Simulation exploitation

1. Introduction

Space debris, i.e. non-operating man-made objects found in orbit around the Earth, have increased since the beginning of space exploitation to a critical point where the space environment has become unstable and debris population grows by itself as the result of debris collision [18]. Figure 1 shows a monthly average number of objects seen in Earth orbit, sorted by object type: active spacecrafts are in blue, mission or launching related debris in green and orange, and on top of that the purple curve shows fragmentation debris. The figure evidences the phenomenon of exponential growth

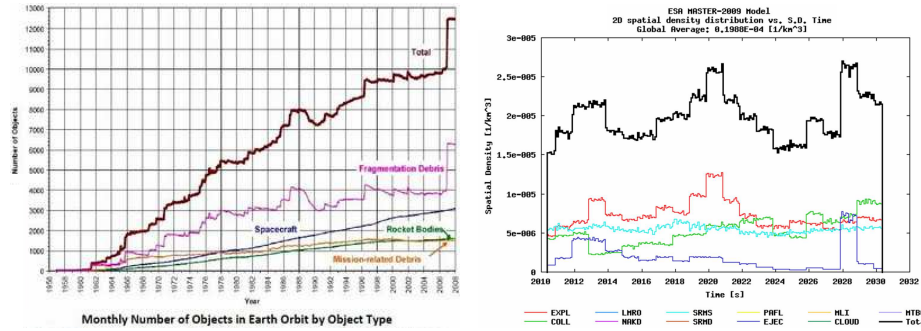


Figure 1: Space debris: monthly number by type [20] (left) and density (right)

of fragmentation debris, which make up the most part of objects in orbit. Since two
catastrophic events in the late 2000s, awareness has increased about this major threat
for operating man-made space systems, and several remedial measures to be enforced
are been considered by major space actors so as to reduce space mission risk and en-
sure sustainable space activities. These measures rely on models of long-term debris
environment evolution [26] coupled together with risk assessment tools.

Projections using software MASTER [8], a debris flux simulation tool developed
on behalf of ESA¹ and experimentally checked using data from ISS² maintenance,
show that the steady growth of space debris in the past years is a tendency that will be
kept in the future. Figure 1 shows the monthly number of objects in orbit around Earth,
with calculated previsions until 2030 (the total curve is particularly relevant here). The
situation is particularly dire in orbits closest to Earth, where debris population has been
increasing for more than 50 years of space activity. Recent simulations indicate that
the in-orbit population, wracked by the Kessler syndrome, is instable and will grow for
the next centuries [18]. Even with a 90 % implementation of the commonly adopted
mitigation measures, based on the initial population of 2009 provided by ESA, the
debris population in orbits closest to the Earth is expected to increase by an average of
30 % in the next 200 years [19].

The physics of the phenomenon is well-understood and allows producing fine sim-

¹European Space Agency

²International Space Station

ulations and credible projections confirmed by feedback information coming from servicing missions on existing space systems (e.g. returned parts of Hubble’s solar panels
30 after space repair [12]) or from observed effects on the satellite or on its trajectory —
e.g. impact on Jason-1 deduced from attitude and electrical current disturbances [21]
— or, for larger debris, from Earth observation and debris cataloguing. This leads to
fine simulations such as MASTER 2009, which provide detailed information on debris
fluxes (and is the source for the data on Figure 1), which can in turn be used as an input
35 to physical simulations assessing the effects of debris.

However, this evaluation is to be performed at several levels, with different methods each implementing a different physics and extremely varied effect models. Hence, obviously, the exploitation of this knowledge is not easy at all, especially with the purpose of providing recommendations about the future effects of the various remedial
40 measures that can be considered. In order to address this issue, an important aspect to
be taken into account is the mission of the system, i.e. more specifically the services
provided by the system to its users or customers on the ground. Indeed, the availability
of these services throughout time is the only criterion that can be used to compare
objectively the effects of various debris issues and various remedies.

45 Such methods have been discussed in previous works [13, 14, 15, 3], where the
ATLAS³ approach is presented. It relies on a modelling of the behaviour of the system
leading to the service being provided to the ground by a temporal logic of the interval
family [27, 9, 10] based on Allen’s operators [1, 2], leading to a logic tree describing
the organisation of the different elementary functions of the system, over which an
50 availability metric is computed, allowing the computation of a vulnerability index [11,
4], which defines a risk of mission performance degradation due to effects of both
trackable and untrackable space debris on a complex space system.

The purpose of this research is to connect these two different fields of expertise: on
the one hand, there exist fine physical simulations providing a good knowledge of the
55 way components evolve in their environment and information such as life expectancy
(i.e. component time-evolved failure rates); on the other hand, there also exist evolved

³Analysis by Temporal Logic of Architectures of Systems

multi-agent representation that allow expressing mission fulfilment as a combination of elementary function fulfilment and reason in terms of fulfilling user needs. Indeed, physical simulation relies on concrete physical components of the system, whilst multi-agent mission assessment rely on functions. Each representation is pertinent for its own field and inadequate for the other, and a bridge is needed between both.

This paper provides a theoretical framework which may be used as a common representation for spacecraft design knowledge and where it is possible to compute inputs for the ATLAS method from fine physical simulations results. It is organised as follows. Section 2 presents the physical knowledge about debris that we want to exploit, as well as the ATLAS approach that will be used as an integration framework. Section 3 presents the theoretical notions which are used to transfer information from physical simulation to ATLAS mission assessment. Section 4 shows a case study. Section 5 gives some conclusions and perspectives about this work.

2. The issue of debris effect integration

2.1. Space debris effect knowledge

As pointed out in the introduction, there exist fine simulations such as MASTER 2009 by ESA, which provide detailed information on debris fluxes. This statistical information can in turn be used as an input to simulations implementing debris effect models. These simulations are of two different types, since the debris population may be separated in two distinct populations.

“Large” debris are objects trackable from the ground (i.e. roughly of a size larger than a certain size (ranging from 10 cm in LEO⁴ to 50-100 cm in GEO⁴. with intermediate sizes in MEO⁴. which may be listed in a catalogue (about 16,000 such objects are known today). Collisions with such objects are always catastrophic for the spacecraft, but are predictable, which allows manoeuvring to avoid them. However, collision avoidance manoeuvres have several damagefull consequences, among which that some functionalities of the spacecraft (such as its payload) are generally unavailable during

⁴LEO: Low Earth Orbit, GEO: Geostationary Earth Orbit, MEO: Medium Earth Orbit

manoeuvre; thus, each manoeuvre temporarily prevents the system to fulfil its mission;
85 and that manoeuvres consume propellant, which is also required for station keeping;
thus, more frequent manoeuvres reduce the lifetime expectancy of the spacecraft.

“Small” debris are untrackable objects (the number of which is evaluated to range
in billions) for which the only available knowledge on the ground is a statistical repar-
tition model. Collision with such object is not predictable, but may be less severe if
90 the spacecraft is suitably protected and the size of the object is small enough (typically
less than 1 cm). Still, too frequent impacts also have dire consequences for the space-
craft: impacts may accelerate the ageing of spacecraft components and increase their
failure rate, reducing their performance as time goes by; and, if the impacting object is
large enough, or if the impacted component is vital enough, may cause the loss of the
95 complete system or at least of one important subfunction.

An intermediate category of untrackable debris which are large enough to cause a
catastrophic collision is sometimes distinguished as a specific subcategory.

Hence, there are two possible studies:

- The study of untrackable debris relies on penetration probabilities base on bal-
100 istic models, assuming that a penetrated component fails [24]. This leads to
average lifetimes for spacecraft components, but these studies are limited to a
physical failure of the system, which does not necessarily imply the inability of
the system to perform at least partially its mission, and hence tends to overesti-
mate the effect of small debris, especially in densely populated orbits.
- The study of trackable debris relies on collision prediction and avoidance ma-
105 noeuvre computation. This allows two different computations: a probability
to be manoeuvring at a given time to address short-term effects, as well as an
expectation of the remaining amount of propellant to address long-term effects.

The issue that arises here is the following: how is it possible to integrate all these
110 effects into a same performance model?

Operator	Symbol	Sequencing	Degenerate cases

Table 1: ATLAS operators

2.2. The ATLAS approach

The ATLAS approach is documented in various previous publications [13, 14], as well as its application to space system performance assessment [15, 3], and the notion of vulnerability index that can be added to it to compare space debris scenarios [11, 4].

115 Let us just recall that the method relies on a decomposition of the mission into elementary functions that can be linked to system components, according to the operators described on Table 1

Once this decomposition has been performed, a performance measure is computed over the tree. This measure is an *availability* notion, defined as follows: $\pi^F(s, t)$ is the
120 (conditional) probability that a function F succeeds at time t provided it was requested at time s . Availability π is provided as an input for elementary functions, and computed recursively for higher level functions according to the semantics associated to the various operators, as shown on Figure 2.

Once this availability notion has been computed for the root function (which represents the mission of the system), it is possible to compare it for various configurations
125

of the system and therefore assess a loss or gain of service which may be compared to user needs. For further detail, the reader may refer e.g. to [3].

2.3. The gap between physical and logical models

The issue is to provide input for elementary functions described above. Consider the workflow for space system performance assessment on Figure 2: the “debris knowledge” level represents physical simulation. The “vulnerability data” level represents the information needed to integrate the various effects of space debris, thanks to the “mission architecture” level: these two levels represent the ATLAS approach.

Here, we see that trackable debris effect may directly be integrated into mission availability measures, since the computed effects (probability for the payload to be unavailable, or probability that the propellant reaches a level where the spacecraft must cease to be operated) are directly interpretable in terms of mission.

However, untrackable debris effect is more difficult to integrate: it may be translated in term of physical component liveness, but this requires an additional level of knowledge (called “spacecraft design” here) in order to express these effects in terms of system mission, which will be done in the next section.

An important point to take into consideration is the fact that the form of this spacecraft design knowledge is something which may vary considerably in form in industry: each company, if not each team, will have its own way to represent this knowledge. Therefore, it is particularly important, since the issue address here is not limited to a specific spacecraft platform, to have a common way to describe this knowledge so as to allow the use of a consistent methodology over very varied space assets.

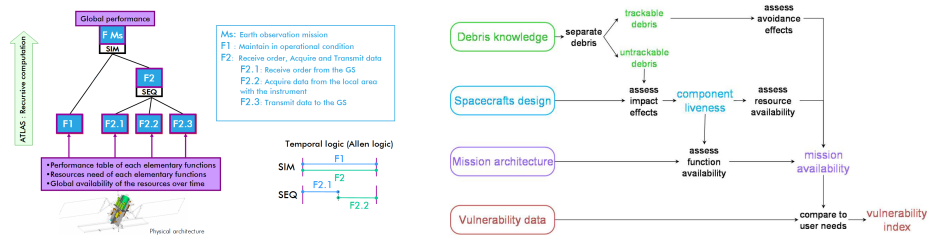


Figure 2: ATLAS method overview (left) and Space system performance assessment workflow (right)

3. The CRIOS model

In order to address the issue above, a model is developed, which is called an *architecture association*, and aims at making the link between components and functions. Moreover, this model includes a notion of resource, which allows taking into account transversal infrastructures upon which the system relies without having dependency issues on functions, since the ATLAS computations rely on an independency hypothesis.

3.1. Resource

Resource modelling allows tackling dependency issues in the functional model: whilst it is clear that most components will e.g. require that electrical supply components work, it is not possible to link these components to all functions. Indeed, the elementary functions would then be associated to dependent probabilities, which would make probability computation throughout the tree extremely problematic.

In this modelling, each component, or group of components, is associated to one or several resources. A resource defines a physical quantity that is *produced* by a component, or *required* by it to work properly. For a given instant, if the availability in a given resource drops below the need, defined by the quantity required by the active physical components, then the system cannot fulfill its mission anymore. The probability of success of the mission shall drop to zero.

This notion may be illustrated on two different examples, both pertaining to space system study: electrical power, which is essential to most components and is locally produced by solar arrays, and propellant, for which a given amount is available at the beginning and is consumed without replacement throughout the life of the spacecraft.

Two measures are associated to each resource R_i :

- 1) the *needed* amount of resource is associated to each function F : $r_i^F(s, t)$ is the amount of resource R_i needed by F when it is requested at time s and succeeds at time t .
- 2) the *available* amount of resource is defined at the level of the whole system: $M^{R_i}(s)$ is the amount of resource R_i available at time s .

The first measure, r is very similar to π , and is also computed recursively throughout the functional tree according to a semantics given in terms of operators. On the

other hand, the second measure, M , strongly depends on the kind of resource and on the information available in the scenario.

Resource measures are used to correct availability π . Indeed, π is computed assuming infinite resource availability to avoid dependency issues. It is therefore post-processed to take resource into account:

$$\bar{\pi}^F(s, t) = \begin{cases} \pi^F(s, t) & \text{if } \forall i (r_i^F(s, t) \leq M^{R_i}(s)) \\ 0 & \text{otherwise} \end{cases}$$

The computation of π and r for elementary functions will be addressed by CRIOS together, whilst that of M will be addressed separately.

3.2. Architecture association

As stated above, it is necessary to describe a link between the elementary functions of the functional tree and the physical architecture, taking into account interactions and redundancies between components, i.e. the fact that several physical solutions may satisfy the generic elementary functions of the tree.

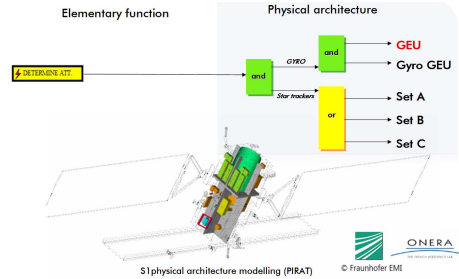


Figure 3: Association of a physical architecture to function “Determine Attitude”

In order to do this, information from spacecraft design knowledge is used to write a mapping between elementary functions and components. This mapping is called an *architecture association*. Figure 3 represents an example of architecture association (functional – physical). The physical architecture is described by the means of classical logic that define all the possible configurations of the system to fulfill the function. To satisfy the “Determine Attitude” function of a spacecraft, two types of alive gyroscopes and at least one among three set of alive star trackers are required. In nominal mode,

regardless the debris threat, the performance of each function is given by the ability of the system in each configuration to satisfy the function.

195 The idea here is to be able to assess thanks to fine physical simulation the way in which components degrade over time, thus providing probabilities to be in each configuration, so as to be able to assess the effects of this degradation on elementary function performance, and, through the logical model, its consequences on overall performance.

The architecture association includes the notion of resource by defining, for each
200 component and each element, how much resource is needed for the fulfilment of the element and, for each element, how much resource it produces.

More precisely, an architecture association contains the following information:

- 1) for each function, a list of elements pertaining to the function (the fact that E pertains to F is denoted by $E \in F$), each containing: a list of n components; a number k of alive
205 components needed among these n to ensure the element; an order of priority for the use of components; for each component C and each resource i , an amount $r_i^C(s, t)$ of this resource needed by C to contribute to ensure the element between s and t ; the element; note that the fact that a component C contributes to ensure an element E is denoted by $C \in E$;
- 2) for each function, a list of configurations, where each configuration X contains: a
210 list of ensured elements for the function; an elementary temporal performance table on sample S , i.e. a function $\pi^X(s, t)$ defined for each configuration X and each pair $(s, t) \in S$;
- 3) for each resource R_i , a list $CL(R_i)$ of components producing this resource, as well
215 as a production table for each component C , i.e. an instantaneous amount $m_C^{R_i}(t)$ of produced resource; Note that elementary measures π , r , and m are the outcome of industrial know-how about spacecraft design.

3.3. The CRIOS model

CRIOS⁵ is able to use the information above to convert component liveness into
220 the three kinds of input needed by ATLAS, namely, for each elementary function, its

⁵Component Redundancy and Interaction in Operational Systems

availability and resource need throughout time, and, for the overall system, available resource. These three features are respectively called the π -converter, the r-converter, and the M-converter. The following notation is introduced to represent the combinatorics underlying these converters: for each element E of the function, a *partial permutation* is a vector $A \in \{0, 1\}^n$ stating, for each of the n components contributing to ensure the element, whether it is alive (1) or not (0); the *cardinal* of each such partial permutation is the number of alive components, i.e.: $\#A = \sum_{C \in E} A(C)$ where $A(C)$ denotes the value (1/0) indicating whether component C is alive or not.

Assuming the existence of the following elements:

- 1) a time sample S , made of instant pairs (s, t) , which is not explicited;
- 2) for each component C of the system, a probability $F^C(t)$ that the component has failed at time t — this probability is the output of physical simulation: it must be assessed homogeneously over all components of the system and, if it is associated to an uncertainty, this information may be used to compute several cases from best to worst and determine sensitivities;
- 3) an architecture association linking the physical system to the elementary functions of a functional tree;

it is possible to compute the various inputs needed by the ATLAS method. Indeed, it is possible to compute the probability that a given partial permutation A occurs:

$$p^A(t) = \prod_{A \text{ such that } A(C)=1} (1 - F^C(t)) \times \prod_{A \text{ such that } A(C)=0} F^C(t)$$

and, therefore, the probability that element E is ensured:

$$p^E(t) = \sum_{A \text{ such that } \#A \geq k} p^A(t)$$

. and from this, the probability to be at a given instant t in configuration X :

$$p^X(t) = \prod_{E \notin X} (1 - p^E(t)) \times \prod_{E \in X} p^E(t).$$

and this allows computing the overall availability of the function by:

$$\pi^F(s, t) = \sum_X p^X(t) \pi^X(s, t).$$

The resource need may also be computed for each partial permutation where E is ensured, i.e. each partial permutation A with $\#A \geq k$:

$$r_i^A(s, t) = \sum_{\substack{k \text{ first components } C \\ \text{such that } A(C)=1}} r_i^C(s, t)$$

where “first components” is to be understood in the sens of the priority order available in the input. Moreover, for each partial permutation A where E is not ensured, the associated need in any resource i is: $r_i^A(s, t) = 0$.

With this notion, also using the probability $p^A(t)$ to be in a given partial permutation at a given instant t defined above, it is possible to compute the expectation of the resource needed to ensure element E :

$$r_i^E(s, t) = \sum_A p^A(t) r_i^A(s, t)$$

and, consequently, the resource need for each function F , defined by:

$$r_i^F(s, t) = \sum_{E \in F} r_i^E(s, t)$$

It is worth noting that the notion of configuration defined for the π -converter is not used here: this is due to the fact that the probability associated to each partial permutation is rather included in the expectation for the element.

Finally, the available amount of resource may be computed. It depends however on the type of resource considered:

- for a *cumulative* resource: $M^{R_i}(t) = \sum_{C \in CL(R_i)} F^C(t) m_C^{R_i}(t)$
- for a *concurrent* resource: $M^{R_i}(t) = \max_{C \in CL(R_i)} F^C(t) m_C^{R_i}(t)$

4. Case study

4.1. Deorbitation of retired satellite ENVISAT

The case study presented here serves as an illustration of the ATLAS performance analysis method and has no purpose of realism concerning the data used. It shows what could be achieved with the intervention of technical experts to determine the data associated to each elementary functionality.

245 Let us consider the task of completing the deorbitation of retired satellite EN-
 VISAT [5]. ENVISAT (“Environmental Satellite”) is an inoperative Earth-observing
 satellite still in orbit. It was launched in 2002, into a Sun synchronous polar orbit at an
 altitude of 790 km, and the task ended in 2012. It is now considered as large debris:
 260 26 m × 10 m × 5 m and 8,200 kg. For this reason it a candidate for a space debris
 removal task. The aim is to study different possible ways of removing ENVISAT with
 a chaser. The general task may be decomposed as follows: 1) launching the chaser; 2)
 reaching ENVISAT’s orbit; 3) placing the chaser; 4) capturing ENVISAT; 5) initiating
 deorbitation.

We choose to compare the following different possibilities: (i) the choice of the
 255 launcher between Vega and Soyuz; (ii) the choice of the chaser (Fig. 4): with two nets
 or a robotic arm; (iii) the choice of the deorbitation technique: with a deorbitation kit
 (propulsive element appended to the debris) or by dragging the debris.

4.2. Functionality description

The first step is to detail each elementary functionality of the mission and specify
 260 temporal constraints between these elementary functionalities, success probabilities as-
 sociated to them, links to system components, and evolution throughout time of these
 components. As described above, the task is decomposed in five elementary function-
 alities: Launching, Reaching ENVISAT orbit, Placing, Capturing ENVISAT, Initiating
 deorbitation. These functionalities have to exactly follow each other, so the appro-
 265 priate operator to be used between all functionalities is operator “meets” detailed in
 Table 1. Each elementary functionality must now be precisely detailed and linked to
 components to assess its associated success probabilities through CRIOS.

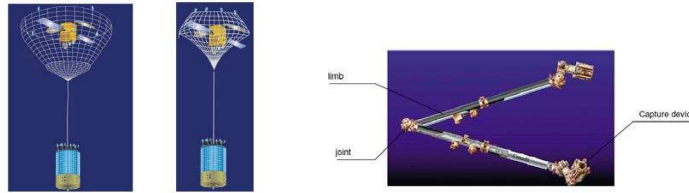


Figure 4: Different possible equipments for the chaser [6]: ROGER net system (left) and Canadarm-2 ISS operating arm principle (right)

Functionality 1 The launcher launches the chaser in the orbital plane of the debris. The launching may not be operated all the time and is only possible during a 2h period each day of the first trimester of 2020. These time slots are considered as system components from the CRIOS point of view. If the task manager decides anyhow to launch out of such a time slot, the launch is operated at the beginning of the next slot. If the order is initiated inside a nominal slot, the chaser is immediately launched. Weather and other external conditions are not considered here since they would have the same impact on the task for all studied configurations. For similar reasons, it is considered that, if the launch vehicle takes off, the probability of success of this functionality is 1.

Functionality 2 The goal of this functionality is to manoeuvre the chaser to phase it with the debris. Placing the chaser in the right orbit (ENVISAT's orbit) depends on the choice of the launcher. Vega places it on an orbit of around 300km in about 800s [22]. One must wait several minutes to a few hours to compensate the possible phasing difference between ENVISAT and the chaser. It is assumed that the launching slots have been judiciously chosen so that the phase difference be minimised. To simplify this example that has a strictly illustrative vocation, the 2h time slot component is divided into three overlapping time slot elements (0 to 40 min, 40 to 80 min, and 80 to 120 min), that are considered successively alive in the sense of CRIOS, and that are associated to three different configurations, where the time necessary to reach the orbit is respectively of 10 min, 20 min, and 30 min. The success probability of this function is 90% (respectively 40%, 30%, and 20% according to launching time). The situation is simpler for the Soyuz launcher since it places the chaser directly in the correct orbit (790km) in about 60 min [23].

Functionality 3 This functionality varies depending on the equipment of the chaser. In the case of a chaser equipped with nets:

F3.1 This corresponds to the observation and debris evaluation phase and may take more or less time to succeed. A delay of one or two minutes is considered.

F3.2 The chaser manoeuvres to move closer to the debris (to reach a distance of about 100m) and positions itself so as to be able to observe the debris with a dedicated

sensor. Depending also on time slots above, several elements are considered, associated to configurations where the duration of this functionality is either 5 min, 7 min or 10 min.

300 In the case of a chaser equipped with a robotic arm:

F3.1 Greater precision is required for the operation. If necessary three attempts may be made, attempts being also modelled in CRIOS as elements, with one out of three having to success and a priority given according to attempt order. They are associated to configurations with delays respectively of 5 min, 10 min, and 15 min, and each with a success probability of 50%.

305 F3.2 The chaser manoeuvres to reach a distance of about 1m close to the debris and positions itself. In link with the above elements, It is assumed that the duration of this functionality is either 30 min, 40 min or 50 min.

Functionality 4 This functionality aims at establishing the mechanical contact with the debris. It also varies depending on the equipment used.

In the case of nets, it is considered that two nets are on-board. If the first net fails, a second net is launched 5 min later. It is assumed that both nets have the same success probability which is estimated at 60%. The capture time is constant (about 1 min). These two nets are also modelled in CRIOS as different elements for which one at least must be ensured.

315 In the case of a robotic arm, the functionality has three chances to succeed in hitching the debris, after 5 min, 6 min, and 7 min, each of which has a probability of success or 30%. These attempts are also modelled as CRIOS elements concurring to a hitching component, and another component is taken into account to represent the de-tumbling that must be operated by the robotic arm and takes about 30min and has a high success probability of 95%, invariant over the mission.

Functionality 5 Two options are considered here. If a deorbitation kit is chosen, it has to be attached to the debris so as to change its trajectory. It is assumed that this functionality would take about 1 min and would have a success probability of 80%. If the debris is dragged, the success probability is evaluated to 90%. For practical reasons,

the deorbitation kit is only considered when using a robotic arm since the latter would be necessary to append it to the debris.

4.3. ATLAS results

The results presented here do not have the vocation to provide an actual answer to the studied issue since the data used is approximative. The aim is to give an idea of the types of result the CRIOS-ATLAS method may offer.

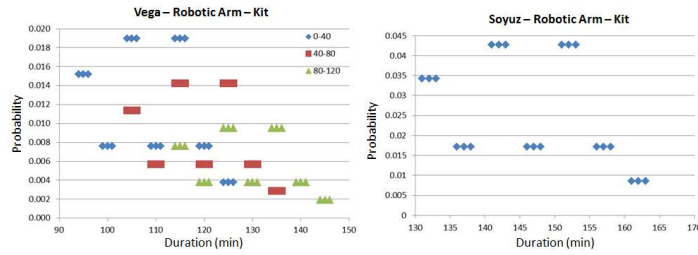


Figure 5: Graphical representation of the ATLAS results

If M is the whole task, the CRIOS-ATLAS chain provides the probability $\Pi_M(s, t)$ that the deorbitation be successful depending on the chosen configuration, at a given instant t , knowing the task has started at instant s . Different post-treatments may then be applied to the resulting data to interpret it.

For instance, since the launching 2h time-slot has been divided into three with the Vega launcher (cf. Functionality 2), it may be interesting to consider the overall success probability in each time-slot, i.e. for the first slot: $\sum_t \sum_{0 \leq s \leq 40} \Pi_M(s, t)$. These results are presented in Table 2. If the duration of the task is crucial, it is also possible to take into account a time limit. For instance, if the duration should be limited to 100min because of a limited quantity of propellant (so that the chaser doesn't become a debris to be deorbitised), the success probabilities are $\sum_{t, t-s \leq 100} \sum_{0 \leq s \leq 40} \Pi_M(s, t)$ presented in the last column of Table 2.

It may also be interesting to consider the success probability of the task with regard to the task duration, i.e. for the first time-slot $\sum_{0 \leq s \leq 40} \Pi_M(s, s + d)$ where d is the task's length. These results are presented as graphs in Fig. 5

Beyond the straightforward conclusion that the Soyuz launcher with a robotic arm and the dragging option is the best configuration, ATLAS details the precise delays

linked to the successful realisation of the overall task. Post-treatments such as pre-
 350 sented here may thus be applied to the result data to bring to light different aspects. In
 addition, changes in configuration may easily be compared, which allows to study the
 sensitivity of the system.

Table 2: Success probabilities and task length of ENVISAT deorbitation

			S_M	Duration d (min)	Success	Success with $d \leq 100$
Vega	Net	Drag	0-40	30-43	29.0%	29.0%
			40-80	40-53	21.8%	21.8%
			80-120	50-63	14.6%	14.6%
			0-120	30-63	65.4%	65.4%
	Robotic Arm	Drag	0-40	93-125	26.9%	7.7%
			40-80	103-135	20.2%	0%
			80-120	113-145	13.5%	0%
			0-120	93-145	60.6%	7.7%
		Kit	0-40	94-126	23.9%	6.1%
			40-80	104-136	18.0%	0%
			80-120	114-146	12.0%	0%
			0-120	94-146	53.9%	6.1%
Soyuz	Net	Drag	0-120	67-80	72.6%	72.6%
	Robotic Arm	Drag	0-120	130-162	60.6%	0%
		Kit	0-120	131-163	53.9%	0%

5. Conclusions and perspectives

CRIOS is a generic representation allowing to make a correspondance between
 355 high-level system functions and mission logical architectures on the one hand, and sys-
 tem components and physical architectures on the other hand. It allows incorporating
 industrial knowledge using a common representation in order to be able to translate in-
 formation about component evolution in terms of functional consequences. It is hence a
 useful common language that can be shared between various industrial, physical simu-
 360 lation specialists, and system performance experts. It allowed general result integration
 in project P²ROTECT in a highly innovative multifold approach:

- 1) The tool works at mission level (i.e. the services offered to Users) instead of spacecraft level only, thus allowing the analysis and meaningful comparison of different architecture solutions (spacecraft redundancy, spacecraft factorization. . .), as well as the
365 evaluation of global trade-off between different remedial measures (e.g. comparison between Spacecraft Protection, Space Surveillance Tracking, Active Debris Removal).
- 2) At space segment level, it takes into account threats coming from both trackable (i.e. inducing avoidance maneuvers) and untrackable debris (i.e. inducing collisions with the spacecraft from catastrophic to minor levels) in a commensurable way.
- 370 3) This allows showing effects (or sensitivities) of multi-scale protection solutions (i.e. at mission level, at space segment level, a at component level) as well as combined external solutions (i.e. Space Surveillance, Mitigation, and Active Removal).

Further work will focus on several axes. Firstly, resource limitations are taken into account in a rather simplistic way. Indeed, lack of resource availability is considered in
375 a binary way in the update of π . We plan to develop a finer model in order to account for smoother degradations of performance, thus leading potentially to less conservative results. Secondly, the architecture association could also be improved: indeed, redundancies are treated using a priority list, whilst more elaborate reconfiguration strategies could be taken into account. We plan to extend this model in this direction,
380 thus allowing FDIR (Fault Detection, Identification, and Reconfiguration) strategies to be implemented so as to compare their efficiencies under various debris conditions.

6. References

- [1] J. F. Allen. An interval-based representation of temporal knowledge. In *Proc. 7th International Joint Conference on Artificial Intelligence*, p. 221–226, 1981.
- 385 [2] J. F. Allen. Maintaining knowledge about temporal intervals. *Communications of the ACM*, p. 832–843, 1983.
- [3] S. Bertrand, S. Lala Prudhomme, S. Merit, C. Jolly, R. Kervarc, & T. Donath. Space system vulnerability assessment to space debris: a methodology and a program, in *Proc. IEEE Aerospace Conference*, 2012.
- 390 [4] S. Bertrand, S. Lala Prudhomme, R. Kervarc, C. Jolly, Th. Lang, & T. Donath, A temporal logical methodology for probabilistic vulnerability analysis of space missions: Application

to vulnerability analysis of an Earth observation mission due to catalogued space debris, in *Proc. 62nd International Astronautical Congress*, 2011.

- [5] Ch. Bonnal, J.-M. Ruault, M.-Ch. Desjean, Active debris removal: Recent progress and
395 current trends, in *Acta Astronautica* 85:51–60, 2013.
- [6] X. Clerc & I. Retat, Astrium vision on space debris removal, in *Proc. 63rd International Astronautical Congress*, vol. 15, 2012.
- [7] A. Elie, R. Kervarc, P. Carle, S. Lala Prudhomme, S. Kempf, F. Schäfer, & T. Donath. A method and a tool for assessing in-orbit debris-related risk for space systems, in *Proc. 6th*
400 *European Conference on Space Debris*, 2013.
- [8] S. Flegel, J. Gelhaus, C. Wiedemann, P. Vorsmann, M. Oswald, S. Stabroth, & H. Krag. The MASTER-2009 Space Debris Environment Model, in *Proc. 5th European Conference on Space Debris*, 2009.
- [9] M. R. Hansen and C. Zhou. Semantics and completeness of the duration calculus, in *Real*
405 *Time: Theory in Practice*, LNCS vol. 600, p. 209–225, Springer, 1992.
- [10] M. R. Hansen and C. Zhou. Duration calculus: logical foundations. *Formal Aspects of Computing*, 9(3):283–330, 1997.
- [11] C. Jolly, Th. Lang, T. Donath, D. Grenier, & R. Kervarc, Satellite protection: a system’s approach through the 3R concept, in *Proc. AIAA Space Conference*, 2009.
- [12] A. T. Kearsley, G. Drolshagen, J. A. M. McDonnell, J.-C. Mandeville, & A. Moussi. Im-
410 pacts on Hubble space telescope solar arrays: discrimination between natural and man-Made particles. In *Advances in Space Research* 35(7):1254–1262, 2005.
- [13] R. Kervarc, J. Bourrely, and C. Quillien. A generic logical-temporal performance analysis method for complex systems. *Mathematics and Computers in simulation* 81(3): 717–730,
415 Elsevier, 2010.
- [14] R. Kervarc, C. Louyot, S. Merit, Th. Dubot, S. Bertrand, & J. Bourrely, Performance evaluation based on temporal logic, in *Proc. 9th IMACS/ISGG Meetings on Applied Scientific Computing and Tools*, IMACS Series on Computational and Applied Mathematics, 2009.
- [15] R. Kervarc, S. Bertrand, & S. Lala Prudhomme. Evaluation of space mission vulnerability
420 to space debris, in *Proc. 11th IMACS/ISGG Meeting on Applied Scientific Computing and Tools*, IMACS Series in Computational and Applied Mathematics, 2011.
- [16] Th. Lang, R. Destefanis, L. Evans, L. Grassi, S. Kempf, F. Schäfer, & T. Donath. Assessing debris mitigation efficiency using risk-oriented criteria: application to LEO European mission, in *Proc. 6th European Conference on Space Debris*, 2013.

- 425 [17] Th. Lang, R. Kervarc, S. Bertrand, P. Carle, T. Donath, R. Destefanis, L. Grassi, F. Tiboldo, F. Schäfer, S. Kempf, & J. Gelhaus. Assessing debris risk reduction measure short and long term efficiency: application to a European LEO mission, *Advances in Space Research* 55(1):282–296, Elsevier, 2015.
- [18] J.-C. Liou & N.L. Johnson. Instability of the present LEO satellite populations, *Advances in Space Research* 41(7): 1046–1053, Elsevier, 2008.
- 430 [19] J.-C. Liou, A. Rossi, H. Krag, M. Xavier James Raj, A. K. Anikumar, T. Hanada, & H. Lewis. *Stability of the Future Leo Environment*, Inter-Agency space Debris Coordination committee (IADC) report No. IADC-12-08, 2013.
- [20] NASA, *Orbital Debris Quarterly News* 12(1), 2008.
- 435 [21] NASA. *Orbital Debris Quarterly News* 15(3), 2011.
- [22] E. Perez, Vega user’s manual. Arianespace, 154(3), 2012.
- [23] E. Perez, Soyuz from the Guiana space centre user’s manual. Tech. rep. Arianespace, 2012.
- [24] R. Putzar, F. Schäfer, & M. Lambert. Vulnerability of spacecraft harnesses to hypervelocity impacts, *International Journal of Impact Engineering* 35(12): 1728–1734, 2008.
- 440 [25] M. Shan, J. Guo, & E. Gill, Review and comparison of active space debris capturing and removal methods, in *Progress in Aerospace Sciences* 80:18–32, 2016.
- [26] R. Walker, C.E. Martin, P.H. Stokes, J.E. Wilkinson, & H. Klinkrad. Analysis of the effectiveness of space debris mitigation measures using the DELTA model, *Advances in Space Research* 28(9): 1437–1445, Elsevier, 2001.
- 445 [27] C. Zhou, C. Hoare, and A. P. Ravn. A calculus of durations, *Information Processing Letters*, 40(5):269–276, 1991.