The 5th Joint International Conference on Multibody System Dynamics
June 24 –28, 2018, Lisbon, Portugal

# ESA multibody tool for launchers and spacecrafts: lesson learnt and future challenges

Mario Toso[1], Valerio Rossi[2]

[1]European Space Agency (ATG Europe), Mario.toso@esa.int
[2] ASTOS Solution GmbH, Valerio.rossi@astos.de

ABSTRACT — *Since the early 1980s, the multibody dynamics simulation tool DCAP (Dynamic and Control Analysis Package) has been progressively developed by the European Space Research and Technology Centre (ESTEC) of the European Space Agency (ESA) through several industrial contracts with Thales Alenia Space Italy (TAS-I) in Torino. Since 2014, ASTOS Solution GmbH has taken the lead on the software development and commercialization, with the continued support from the Agency. Future challenges include the demand for a competitive swift deployment of large constellation of satellites with complex deployable appendages: finding attractive solutions to move from a 'single customer – single satellite' proposal toward a versatile single launch of a 'multi-payload – multi-manifest' proposition is key to the success of any space industry. Advanced simulations play a critical role in this landscape). This paper summarizes the major technical milestones, achievements and success stories, and presents the most relevant lesson learnt throughout the DCAP software development. Finally, the envisioned road-map is presented, from the Agency's prospective, aiming to an open discussion with the users community.*

## 1 Introduction

With almost 40 years of space heritage, today DCAP is regarded by the (European) space community as an independently-coded, alternative benchmark for high fidelity multibody simulations and cross-validation of space dynamics problems.

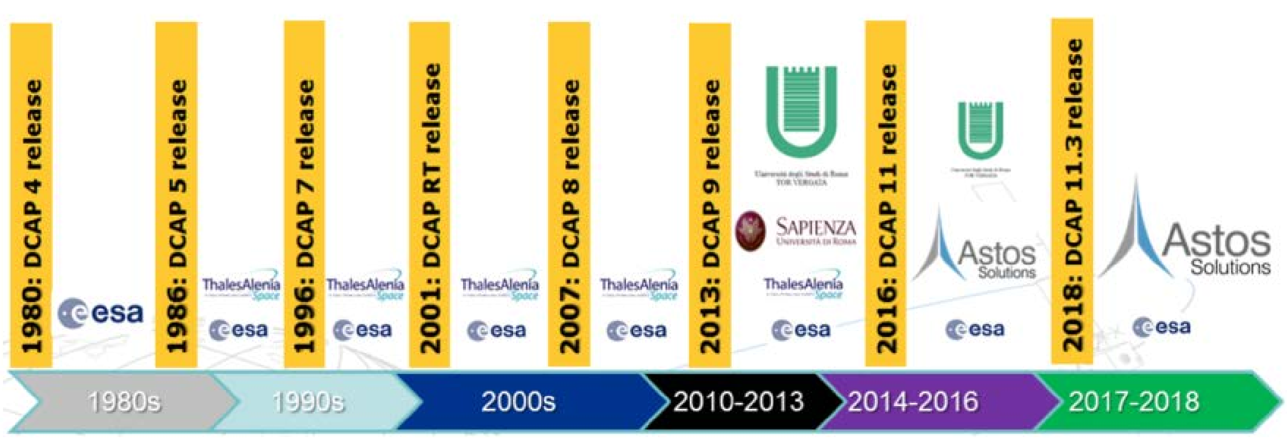


Fig 1: DCAP technical milestones

## 1.1 DCAP heritage

Since the early 1980s, DCAP has been progressively developed by the European Space Research and Technology Centre (ESTEC) through several industrial contracts with Thales Alenia Space Italy (TAS-I) in Torino. Since 2014, ASTOS Solution GmbH has taken the lead on the software development and commercialization, with partial contribution from ESA.

The main technical milestones related to the DCAP software development, as summarized in Fig 1, are:

- 1980: DCAP Release 4.0 included a Lagrange formulation of multibody dynamics, originally developed for the NASA program called DISCOS [3] and oriented towards user-defined subroutines.
- 1986: DCAP Release 5.0 included a set of generic elements to replace the user-defined code. Intermediate versions, under the name of ESA-MIDAS, were produced: ESA-MIDAS Release 6.0, embedding a minimal dimension formulation, was available to ESTEC but not released to other users.
- 1996: DCAP Release 7.0 reflected a fundamental change in the contents of the package [6]. The most significant change was the replacement of the non-linear time history simulation programs with a new simulation program based on a minimum dimension Order(n) algorithm. The equations of motion are symbolically generated by a dedicated symbolic processor in the form of FORTRAN code, which is then compiled and linked to form a highly efficient simulation module. Another major change was the addition of an innovative X/MOTIF-based GUI, that included a graphical full-screen 3D graphics modelling and animation facility, running on Unix / Silicon Graphics workstations. Other changes included a substantial enhancement of the generic elements with the additions of manipulators, brakes, transitions and controller reconfiguration.
- 2001: A complementary software program, DCAP-RT (Real Time simulator), was developed, having the main objective of allowing fast calculations for generic multibody dynamics. The code was also based on a minimal dimension Order(n) formulation and symbolic coding. The running shell was developed in C, and the multibody system could be generated in C or FORTRAN code. DCAP-RT had features very similar to the DCAP-7, while being interfaced to several additional modules describing in detail the detailed functioning of real space sensors and actuators. Interface to MATRIX-X and EUROSIM (real time simulation) code were implemented.
- 2007: DCAP Release 8.0, only available to ESTEC, enhanced DCAP Release 7.0 with the addition of variable mass topologies, sensitivity analysis, interface to MATLAB Simulink, and a renewed GUI based on JAVA and Open GL.
- 2009: An I/O file based interface between DCAP and SIMPACK 8.9 was introduced, in order to take advantage of synergies with SIMPACK pre-processor for interactive graphical model set-up and SIMPACK post-processor for plots and animations.
- 2013: DCAP Release 8.0 was released to the whole users community. An updated DCAP release 9.0beta, with a GUI compatible to MS-Windows machines, was made available and tested at ESTEC for internal use.
- 2014: TAS-I handed over the development, distribution and commercialization of the DCAP release 8.0 software package to Astos Solution GmbH.
- 2015: An intermediate version only available to ESTEC, called DCAP 10beta, incorporated the cumulative technical improvements, developed with the contribution of Università di Roma La Sapienza and Università di Roma Tor Vergata, e.g.: user-centred GUI approach, modal selection based on Modal Assurance Criterion (MAC), dissipative contact, quaternions algorithm, modern variable step and variable order PECE integrator, portable MATLAB-Simulink interface.
- 2016: Integration roadmap of the DCAP inside the ASTOS software was launched, partially supported by the ESA DCAP-Xploitation program (DCAP-X).
- 2018: DCAP-11.3 represents the cornerstone commercial release of the new generation code managed by Astos Solutions and ESA.

In the 60's, spacecrafts' designers were conceiving and implementing innovative controllers on spacecrafts faster than project support engineers could derive, code and debug their analysis programs aimed at validating the controllers' stability and performance [1][2]. NASA researchers and project support engineers started investing considerable effort to improve spacecrafts' attitude control validation capabilities while drastically reducing the analysis cost and lead time: as a result, in the 70s' a number of in-house proprietary codes with impressive capabilities were born (e.g.: SADII, NBOD, DISCOS,…). Few years later, DISCOS [3] was selected by the European Space Agency to become the backbone of its future Dynamic and Control Analysis Package (DCAP).

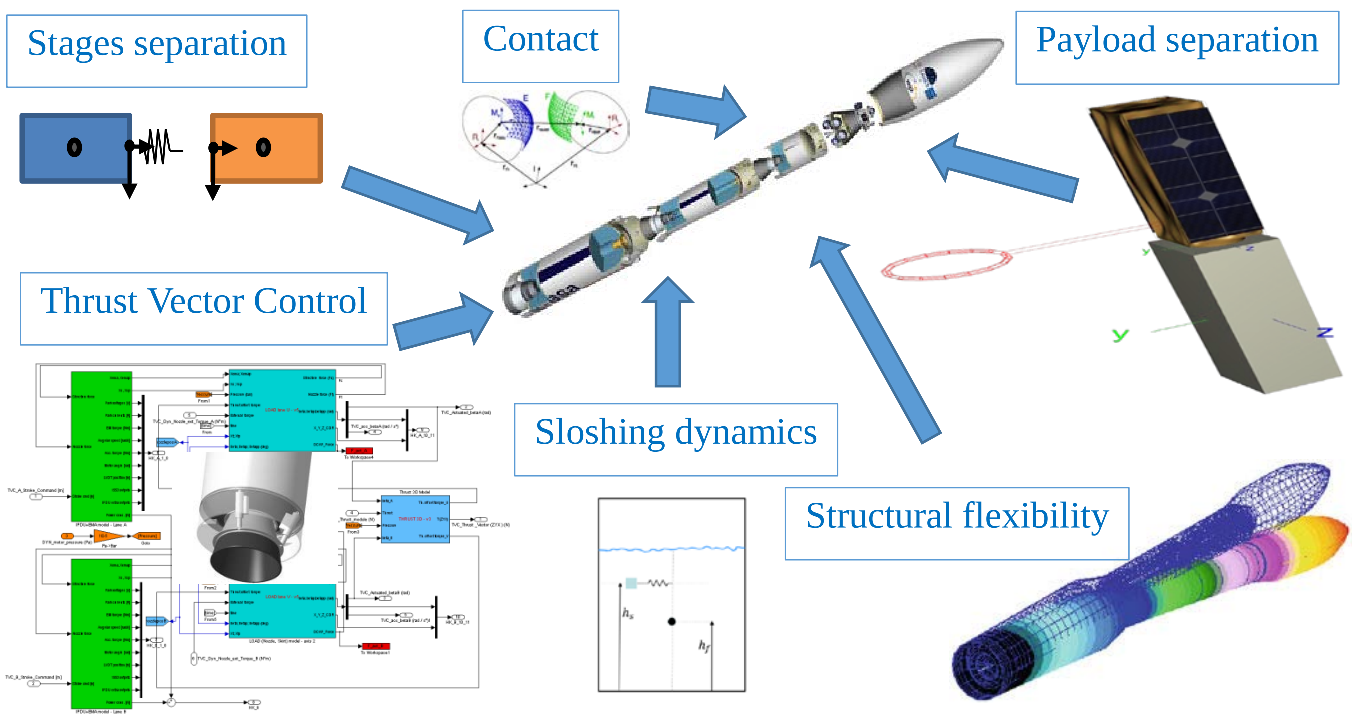


Fig. 2: DCAP launcher simulator features

After 40 years, DCAP still remains a suite of fast, effective computer programs that provides the user with capabilities to model, simulate and analyse the dynamics and control performances of coupled rigid and flexible structural systems subjected to possibly time varying structural characteristics and space environmental loads [5]. By means of dedicated interfaces to other specialised software, it enables reproducing most of the key subsystems and disciplines (such as mechanical configurations, structures, mechanisms, aerodynamics, propulsion, GNC, trajectory, scenarios,...) of the launcher in a seamless simulation environment [11]. The simulator has been also tuned for tackling specific complex events [5][13], such as multi-payload separation dynamics, thrust vector control subsystem studies, lift-off analysis, general loads, as shown in Fig. 2

## 1.2 Technical showcase

The most recent evolutions have focused on the multidisciplinary unique capabilities, particularly relevant to the design and development of rockets and space transportation systems. Four major milestones have been achieved within the last years of developments:

- Sub-Modelling;
- Static Variables;

- Detailed flexible analytic beam model;
- 3D contact dynamics.

When it comes to design and simulate a multi-payload separation scenario with a complex configuration of moving parts, maximizing user-friendliness becomes a mandatory key aspect in selecting a software tool. Lesson learnt from previous programs identified the ability for different experts to separately collaborate inside a consistent model framework and the possibility to parametrize the system's critical design features, as the most important missing features in DCAP.

The capability to import a slave model into a master scenario is called sub-modelling. This feature introduces a new concept of designing a multibody system. A detailed self-standing model of a mechanism can thus be designed once, and used in several master projects by importing it as a slave model, as shown in Fig. 3. Once the sub-model feature is activated in the DCAP GUI, the slave model definitions are copied in the DCAP master model files. The link is indeed completely static which ensure that the resulting model does not relay on any external dependency.

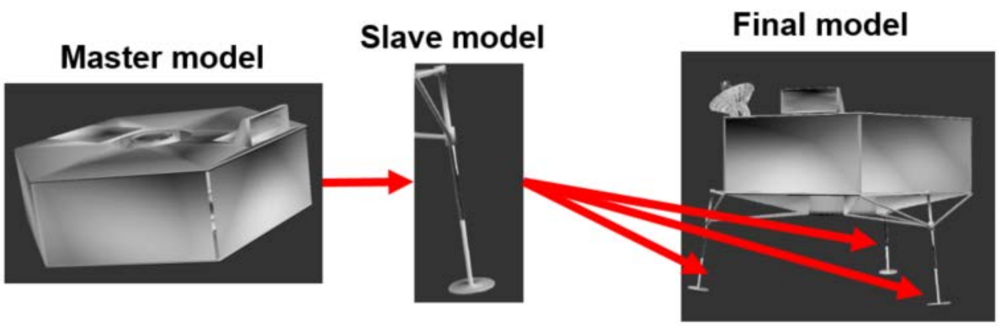


Fig. 3: Example of Master and Slave models

Since the release of DCAP 11.3, the user is allowed to create and link variables in the GUI. This property is called static variables and allows to associate several input fields of different model properties to the same numerical value. In such a way by changing only one parameter, the GUI automatically spreads the modification to any feature which makes use of that static variable. By using the main static variables panel, the user can easily manage the linked properties and the actual numerical values. This feature drastically reduces redundant inputs and collects the most important number in an easy-to-access summary panel. For example the user defines a stiffness value as a static variable and then links it to several spring elements as shown in Fig. 4.

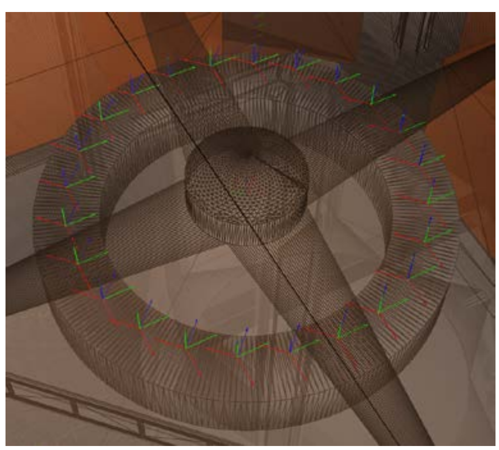

Fig. 4: The stiffness value of each clamp band spring element can be linked to a single static variable

DCAP is also able to account for the single component flexibility. Even if Finite Element Models (FEM) produce high accurate results, they are time consuming to build and to customize. A fast and simple solution is usually the best way to go, especially in the first project phases. DCAP embeds a linear Euler-Bernoulli flexible beam model which can be used without the need of any external FEM software. Since DCAP release 11.3, bending, axial and torsional flexible modes are available.

Finally the design of particular spacecraft scenarios often involves friction by contact. The perimeter of the traditional point-to-surface algorithms, already implemented in DCAP, have been extended to different types of geometric primitives (i.e.: finite plane, sphere, cylinder) and includes dissipation. Moreover, a trade-off was performed on more advanced 3D contact models, taking advantage of the extensive academic work performed in the field of 3D elasto-plastic contact dynamics. The module incorporated in DCAP is entirely based on the Polygonal Contact Method (PCM) algorithm [16], with an optimised interface to the internal source code. The PCM module allows to reproduce the complex contact mechanism between two or more surfaces, see Fig. 5. Each surface is represented by a mesh geometry (wavefront obj file) attached to a specific body in the mechanical system. The algorithm has been validated against analytical computation and benchmarked with respect to the already existing simple point-to-point contact feature in DCAP.

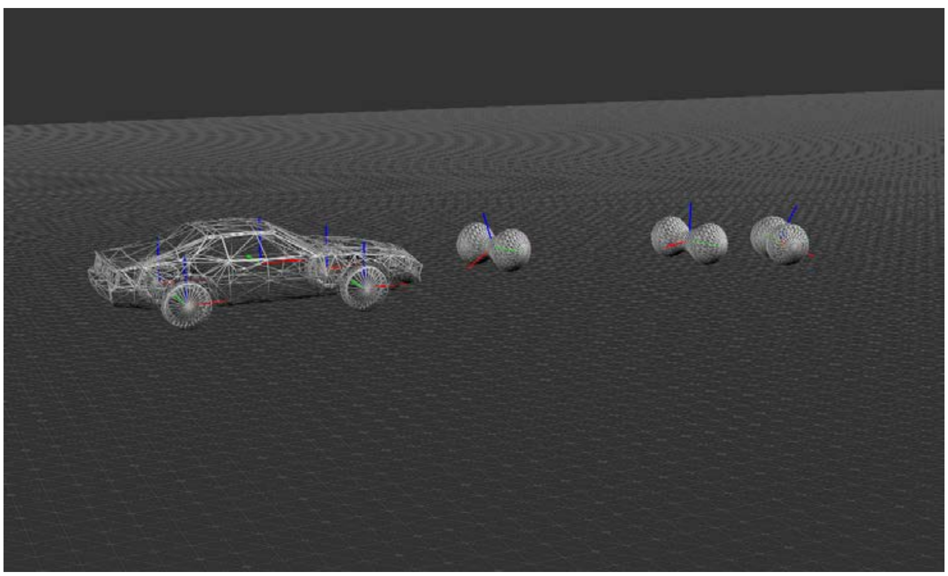

Fig. 5: 3D contact dynamics showcase: Polygonal Contact Method

## 2 Applications heritage

DCAP has extensive heritage in supporting ESA projects [6]. Regarding payload separation and satellites orbit insertion prediction, the simulations for the SWARM and GALILEO Projects included the long term trajectory propagation in order to verify the risk of collision before commissioning.

Swarm is a ESA mission launched in 2013, with the aim to study the Earth’s magnetic field. The Swarm constellation consists of three satellites, placed in different polar orbits, two flying side by side at an altitude of 450 km and a third at an altitude of 530 km, see Fig. 6. The SWARM deployment mechanism is rather complex, involving pyronuts and push-and-roll hinges. DCAP has been used to support the design, to predict the motorisation margins [9] and to simulate the far field trajectories of the three satellites after the separation in order to avoid excessive proximity potentially causing a catastrophic collision.

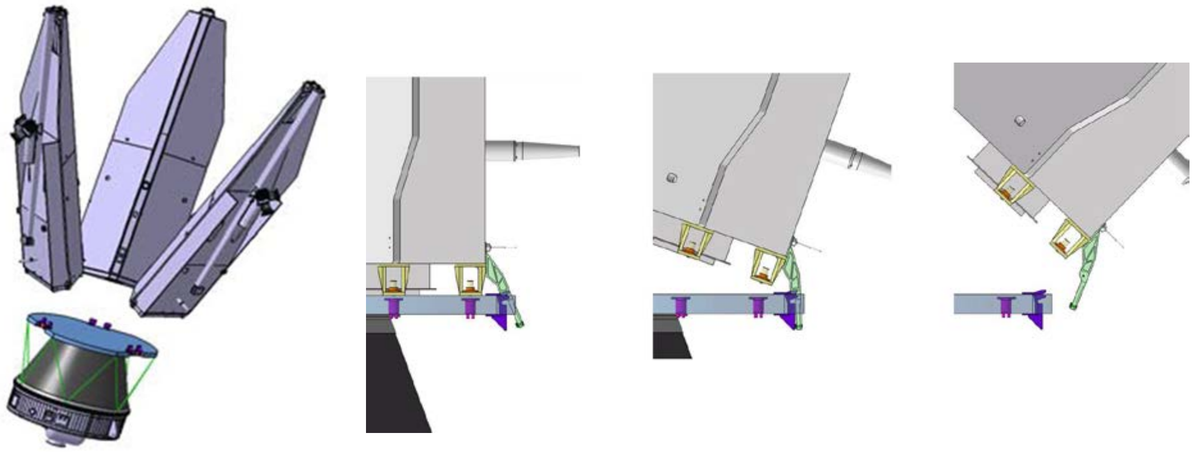

Fig. 6: SWARM satellites separation system

The new versatile Small Satellites Mission Service (SSMS) dispenser allows VEGA launcher to deploy multiple light satellites [1]. It is composed by a lower module suited to accommodate 6 Smallsats up to 70 Kg and/or Cubesat deployers, typically 12 units able to carry 12U Cubesats each and a versatile upper part (composed by a platform and 3 or 4 lateral towers and a central column) available in several modular configurations.

DCAP is employed to perform a clearance analysis during satellite deployment. The separation of each satellite affects the overall system attitude and generates centrifugal forces. This investigation has the objective to check for undesired contacts between the satellites and the dispenser body. For analysis purposes, four satellites are considered as payload of the SSMS. The dispenser consist of a lower module, fixed directly to the Payload Adaptor (PA), and 4 tower modules. The VEGA upper stage AVUM is also part of the model due to its inertial influence during the separation. The satellites are attached to the lower module among the towers via light clamp band devices [10].

The new DCAP 11.3 features allow a fast system modelling:

- global stiffness and damping values are defined as static variables for all the springs in every clamp band;
- since the clamp band device is a self-standing model, the user can model it separately as slave model, and then import it as many times as needed in the final SSMS assembly scenarios.

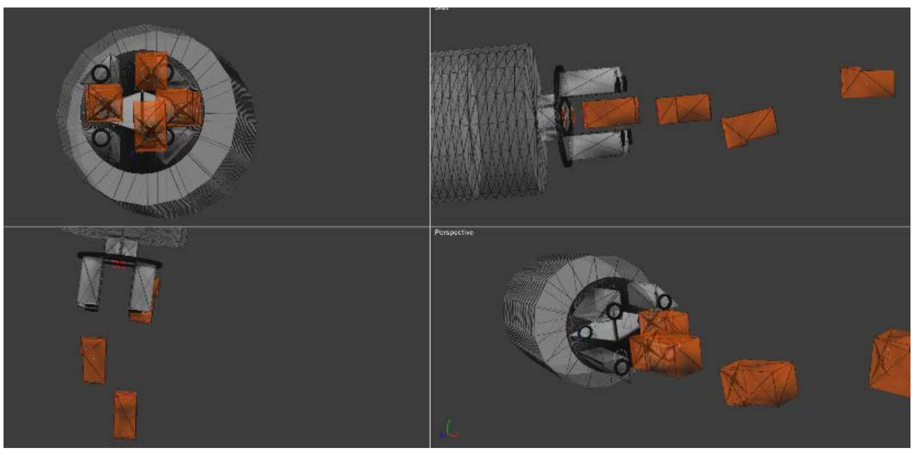

Fig. 7: SSMS separation analysis

The separation timing and the number of springs for each satellite clamp band are chosen in order to minimize the movement of the dispenser and to increase the clearance between the satellites and the SSMS structure during the separation. In order to check that no collision occurs during the separation, 8 sensors are defined to measure the distance between the SSMS dispenser body and the satellite envelope lower corners. The final multibody model consists of 7 bodies (AVUM, PA, SSMS and 4 satellites), 58 elastic elements and 4 transition time logic triggers to release the payload. Fig. 7 shows the DCAP post-processing animation of the satellites separation.

# 3 Lesson learnt

One of the main goals of the Agency in developing special software packages, such as DCAP, is to supplement the internal experts with the necessary state-of-the-art technical tools, share the cumulated knowledge with the European space industry and eventually encourage a commercial spin-off of the tool, facilitating a self-sustained market driven evolution.

In the long fragmented evolution of DCAP, many painful lessons have been learnt. This chapter intends to summarise the most valuable recommendations.

## 3.1 Strict coding policies

In the long term, enforcing strict rigorous policies in maintaining the software structure sound pays off. The contribution of engineering and software experts tends to bring the most effective results. For commercial software such as DCAP, it is also very important to have knowledge of and trace the origin and originality of each portion of the software source code, in order to guarantee it abides to the law, including IPR restrictions. This typically needs to be declared in a so called ‘Software Reuse File’ (SRF): for example, in Space business, the reference standard is the ECSS [12]. Reusing existing software is of course possible, and often preferable or encouraged, but the necessary precautions shall be taken.

## 3.2 Lean and organized functions and file structure

With time, software source code, especially in the engineering field, tend to be messy and redundant, although functionally sound. In-house developers tend to have good knowledge of the entire source code and its embedded functions, and therefore they tend to preserve and re-use the original files systems and perform only minimal changes and complimentary additions. In case of outsourced developments, or in case of parallel in-house implementation of new features, the software tends to be self-contained with minimal interaction to the main source and the other parts.

Technically intervening on heterogeneous code is lengthy and costly: re-structuring of the source code is a long term investment, but it is routinely performed by all successful commercial tools. Often, the importance of this aspect is underestimated, because it is believed that most benefits of a clean structured code remain within the internal project maintenance engineers and developers, rather than with the end-user. At the same time, paying customers pretend swift effective root cause identification and bug fix implementation, which greatly benefit from a rational setup. As many old FORTRAN program, all DCAP files were stored in a single source folder, where several assisting subroutines were often packaged inside the same file of a particular feature, but then also called by other completely unrelated functions, with no trace.

A massive restructuring of the DCAP file structure was performed by ASTOS Solutions in 2014, in order to drastically improve the software maintainability and cross-function traceability. The user-area, including the demonstration and user models repository, is neatly separated from the software operative core, where the source code itself has been re-organised according to individual functional modules.

### 3.3 Systematic version tracking

Depending on specific project support incumbent needs, particular functions were introduced and packaged into a dedicated simulator. Very quickly, different streams are born for different specific needs and become incompatible to each other. Also systematic propagation of the bug fix among the various flavours of the source code quickly become unmanageable. This has created a multitude of subversions which in time tend not to merge together properly and creates many duplications of the same files. Even worse, an important development in a specific version might be completely lost if not properly merged.

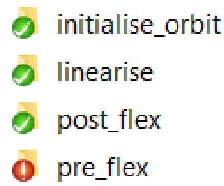


Fig. 8: SVN flags higlight the local modifications with respect to the committed version

Since 2016, a subversion client tool (SVN) has been used to track and merge all the different versions of the DCAP source code. It is one of the most convenient way to effectively develop and especially maintain a software. Several people of the development team can work simultaneously on the some code. A SVN tool improve the code development by tracking every actions and by merging multiple modifications of the same file by different people.

It allows also to branch the code and create different versions of the same software. The main advantage is indeed the history of all the actions performed on the source code. The developer could easily re-create the status of the source code at any date in the past.

### 3.4 Check and cross validate each modification

Debugging a new feature is extremely time consuming. For this reason, it is preferred to introduce every new function as much as possible in a self-contained manner. Only in a second step, when the function is well understood, the process of merging it with the other already existing functions can start. Often, the effort to cross validate the new functionality with the existing ones is underestimated and it may prove surprisingly challenging. For example, a sudden contact may trigger the undesired activation of characteristic flags, which may conflict with some advanced integrator convergence algorithms.

For each new feature, as well as for each new model, validating the results against an independent method is key.

### 3.5 Choice of framework

Every piece of software is bound to work with a number of other third parties' software and hardware, the evolution and sustainability of which is completely independent and mostly unpredictable.

In 1996 DCAP was available with a state of the art X/MOTIF-based graphic user interface, which was specifically coded for UNIX (Sun, Silicon Graphics) work stations, based on IRIX (e.g. SGI Indigo R4000). While at the time, this was strongly innovative, the incumbent exponential progress made by personal computers, their software operating systems (Windows and Linux primarily) and new functional object-oriented programming languages, abruptly reduced the need for expensive work stations for the level of computing power needed by multibody tools. Therefore a large part of the developed source code was obsolete: only five years later, in 2001, a tradeoff showed that its adaptation to the other frameworks was impractical and anti-economic.

A new branch of DCAP GUI was created as shown in Fig. 9, taking advantage of the new technologies (i.e.: JAVA, C++), but the effort of rebuilding from scratch the entire complexity of the tool was in the end massively underestimated.

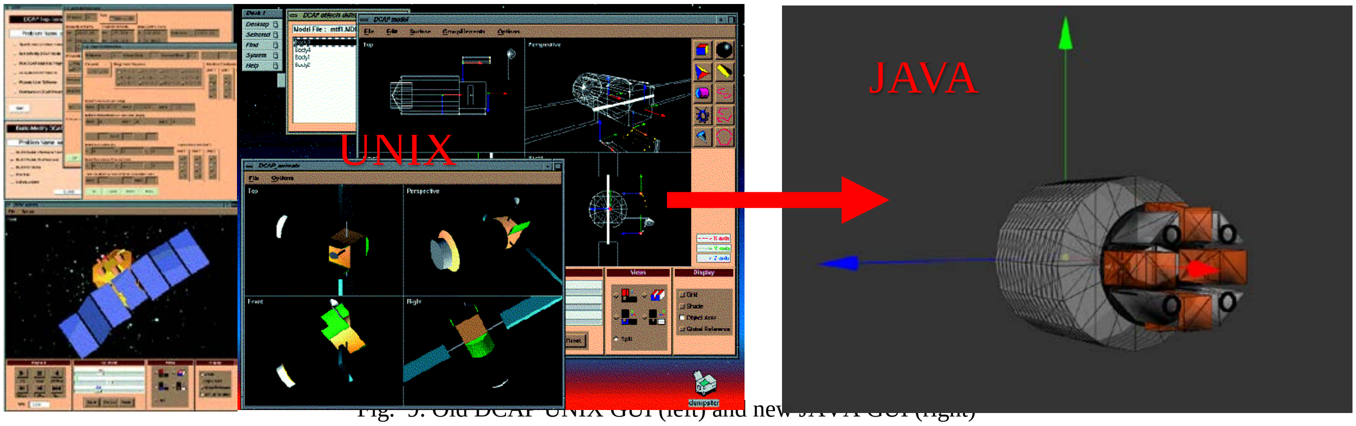

Fig. 9: Old DCAP UNIX GUI (left) and new JAVA GUI (right)

### 3.6 User-friendliness

In academia as well as in engineering technically driven institutions, user-friendliness typically remains a nice to have wish. One of the top three reasons why the Agency's attempts to spin off DCAP into a commercial product failed is linked to the lack of attention to the practical aspects that the commercial end user was expecting.

From 2011, a new user-centred approach was introduced in the software development plan, aimed at closing the technology gap in this domain.

Based on users' feedback, three main actions led to a major improvement of the usability and customer appreciation of the software:

- GUI centred approach. The user interacts mainly through the graphical user interface without the need to work directly with the model definition files;
- An almost free-coding experience for the user. Only specific settings of advanced features require a minimum amount of coding;
- Increased attention to maintain an up-to-date documentation data package and User Manual. Unnecessary and obsolete references have been removed and unclear options have been further described with examples;
- Fully implemented demonstration models are included, with a short technical reference in the documentation.

### 3.7 Documentation and user support

The adequacy and correctness of the supporting technical documentation, such as the user manual and the theory manual, is often overlooked. It is a costly and time consuming exercise, but providing technical detailed clear information about how the software function operates is key to the end user.

More modern complementary alternatives are:

- community forum, mostly available for open-source codes;
- dedicated technical helpdesk, typical of commercial tools.

### 3.8 Modular development

A structured code is designed in such a way that each module is self-standing and can be easily attached or detached to the main core. This approach also allows to interface with external module without messing the software core and the other functions.

For example, the 3D contact dynamics has been implemented in DCAP with a lean interface to the external Polygonal Contact Method module [17]. The development did not need any core modification since the code was already compartmentalized and debugged.

### 3.9 Model validation and benchmarking

Due to the usual absence of hardware test data against which validating the DCAP predictions, an initial benchmark case is employed. The same model is thus built in DCAP as well as in another independent commercial software. Such kind of validation gives confidence about the feature of software implementation state, but unfortunately does not give a clear indication about the accuracy of the simulation with respect to the real phenomenon.

For example, the SMSS clearance analysis has been compared with the results coming from the SIMPACK software. In this case, it was decided to simulate the highest payload capacity case, where the Smallsats rigid bodies are accommodated on two decks (Fig 10) and sequentially deployed.

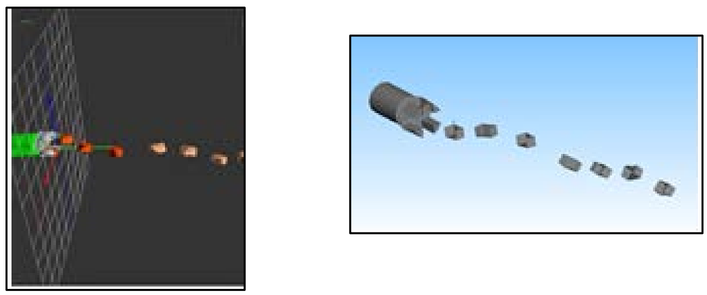

Fig. 10: Multi-satellite benchmark model: DCAP (left) and SIMPACK (right)

The trajectories of each body are registered in the global inertial frame and compared between the two software tools. As expected, the DCAP results well match the SIMPACK predictions (Fig. 11).

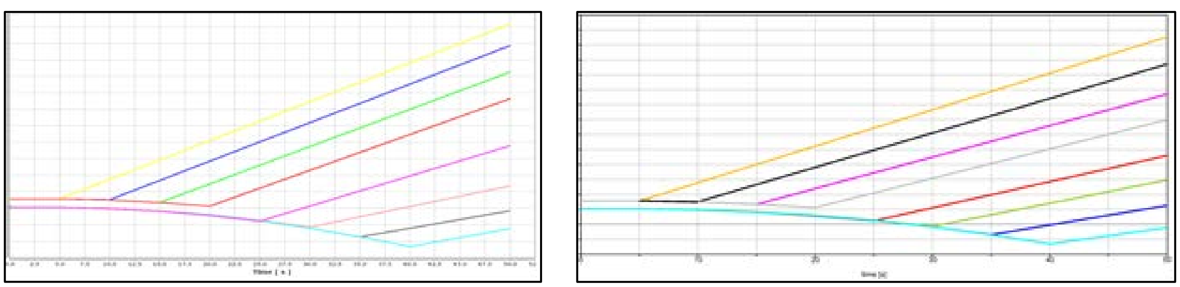

Fig. 11: Multi-satellite benchmark results: DCAP (left) and SIMPACK (right)
(Numerical values are ESA confidential and cannot be publicly disclosed)

The only main difference is detected in the trajectories of the lower deck satellites. After further investigation, the root cause was identified in a small difference in the coding of the release spring forces: DCAP force vector is coded to push perpendicular to clamp band plane, while the SIMPACK standard spring acts as a point-to-point force. When the mass ratio between the dispenser and the released payload reduces, the relative rotation experienced by the dispenser as result of the acting separation force increases, emphasising the trajectory discrepancy. This root cause was finally confirmed by creating a simple 2 bodies model connected by point-to-point spring forces and the results were matched with the analytical solution.

Because the DCAP clamp band representation corresponds exactly to the physical scenario, no modifications to the model is introduced before progressing with the final simulation campaign. The DCAP multibody simulation has proved that no collision occurs during the satellites separation and a safe clearance of minimum 10 cm is guaranteed between the satellite envelopes and the SSMS dispenser body.

Fig. 12 shows the distance between the two lower satellite envelope corners and the corresponding rod connector nodes on the dispenser side. Each colour in the graph corresponds to a different satellite. After the separation, the payload and the dispenser gets closer but within a safe clearance. When the entire satellite envelope overcomes the dispenser tower module, there is no possibility for any other collision.

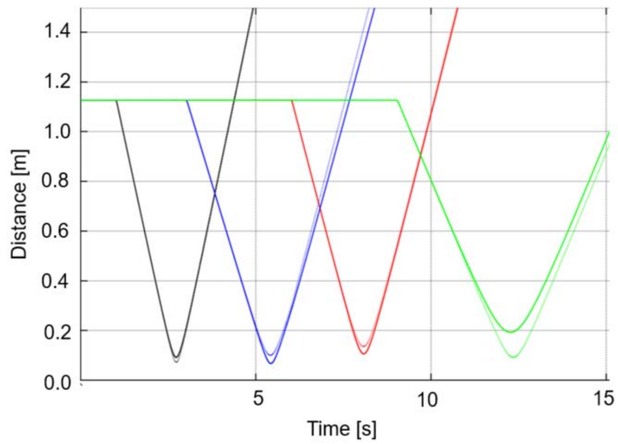


Fig. 12: Clearance between the rod connectors and the lower corners of every satellite

In order to validate the torsional and axial flexible modes, a comparison against analytical reference and external FEM software has been done. The test case consists of a simply supported beam and NASTRAN NX has been used as reference tool.

Fig. 13 reports the benchmark comparison on the axial and torsional mode frequencies computed by DCAP and by NASTRAN NX [15].

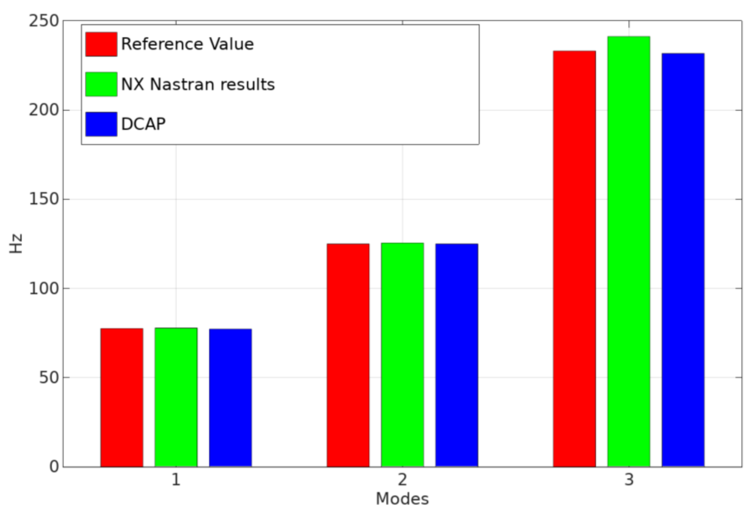


Fig. 13: Comparison of torsional and axial modes with benchmark analytical reference and NASTRA NX v10

# 4 Future challenges

Modern simulations tools are becoming complex multi-disciplinary interacting ecosystems. The capability to technically create an organic multibody dynamics simulation environment appears to be the common industry trend for future software: merging on one hand the need for seamlessly interact with non-linear elastoplastic characteristics, hyper-speed impact dynamics, actively controlled multifunctional smart materials and stochastic-based input, while, on the other hand, enhancing intuitiveness and user-friendliness by means of effectively smart graphical user interfaces based on augmented/virtual reality is one of the new biggest challenges.

## 4.1 DCAP-ASTOS integration framework

The development and design of launch vehicles, especially the flight dynamics simulation, is nowadays heavily relaying on numerical simulations in order to reduce hardware test and to increase the design confidence. The industry trend, in the numerical simulation field, is to combine the features of different tools in one organic and homogenous software.

The ESA LauMBS (Launcher Multibody Simulator) Project had the aim to enhance the DCAP capabilities by linking it to a trajectory optimisation tool in order to build an advanced multi-disciplinary launcher design software. The project objective tool is to perform mechanical design and simulation of the launcher vehicle reproducing all the complex phenomena and environment which the rocket experiences during the flight and payload separation. The software framework provides the building blocks to simulate a complete launcher scenario considering vehicle flexibility, sloshing effects, stages separation, engine pressure oscillations and complex aerodynamic loads distribution.

The trajectory optimization tool used in the LauMBS Project is ASTOS (Analysis, Simulation and Trajectory Optimization Software for Space Applications), a multi-purpose tool for space applications [18], which has been originally designed for trajectory optimization, provides now modules for a variety of analysis, simulation and design capabilities for the whole project life-cycle.

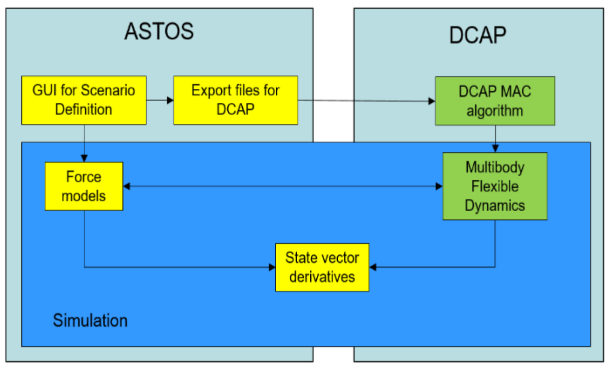


Fig. 14: DCAP and ASTOS interaction framework

In this framework, DCAP is used as a slave tool and it is called at run time by ASTOS, the master software, for the computation of the multibody system dynamics, see Fig. 14. The vehicle model and the launcher scenario are thus set up through the ASTOS GUI. During the simulation, ASTOS exports the needed DCAP files for the definition of the multibody model. DCAP then evaluates the vehicle flexible properties (mode shapes and frequencies). The simulation finally starts and the three actions are performed at each time step:

- ASTOS computes and passes to DCAP the external forces (aerodynamic loads, gravity accelerations and actuator output);
- DCAP evaluates the system dynamics and the state derivatives;
- ASTOS integrates the state derivatives.

Such kind of application is the start of a new life for DCAP, where its technical strengths are incorporated into a modern multifunctional engineering tool. The future challenge of DCAP is to be even more integrated in ASTOS and increase the mutual co-simulation capabilities.

# 5 Conclusions

DCAP is a multibody software specifically designed to tackle space applications such as ascent launcher scenarios, payload separations and space mechanisms design, with extensive heritage in supporting ESA Projects.

With this paper, the Agency and the authors wished to share with the multibody community the most important lessons learnt during DCAP long and fragmented development history, hoping it will help mitigating the impact of falling in the same pitfalls.

In particular, focusing on improving the tool's user-friendliness has been key to the renewed commercial appeal of DCAP release 11.3. At the same time, the increased complexity and multi-disciplinary nature of the next generation simulation ecosystem creates unprecedented challenges and makes even more mandatory to always technically validate the results against an independent source. DCAP confirms to be an efficient and practical technical support tool for the simulation of complex system dynamics problems as well as in the preliminary phases of space mechanisms design

# References


[1] R.H. Schuett, B.A. Appleby, J.D. Martin, "Dynamic loads analysis of space vehicle systems: launch and exit phase", *NASA technical report GDC-DDE66-012*, Jet Propulsion Laboratory, California, USA, 1966

[2] C.S. Bodley, A.C. Park, "Influence of structural flexibility on the dynamic response of spinning spacecraft", *AIAA Paper 72-348*, San Antonio, Texas, USA, 1972.

[3] C.S. Bodley, A.D. Devers, A.C. Park, H.P. Frisch, "A Digital Computer Program for the Dynamic Interaction Simulation of Controls and Structure (DISCOS), Volume I-II" *NASA Technical Paper TP-1219*, NASA Goddard Space Flight Centre, May 1978.

[4] G. Kehrl, "Mission Design Challenges for Multi-Manifest Missions," in *10th Annual CubeSat Developers Workshop*, San Luis Obispo, California, USA, 2013.

[5] B.N. Suresh and K. Sivan, *Integrated Design for Space Transportation System*, Springer, 2015.

[6] R. Franco, M.L. Dumontel, S. Portigliotti, R. Venugopal, "The Dynamics and Control Analysis Package (DCAP) - A versatile tool for satellite control", *ESA Bulletin Nr.87*, August 1996

[7] G.. Baldesi and M. Toso, "European Space Agency's launcher multibody dynamics simulator used for system and subsystem level analyses," *CEAS Space Journal*, vol. 3, no. 1–2, pp. 27–48, 2012.

[8] N. Shah, O. Hsu, et al, "A Multibody Slosh Analysis for the Lunar Reconnaissance Orbiter," in *Proceedings of the 20th International Symposium on Space Flight Dynamics*, Annapolis, USA, 2007.

[9] "Flight separation mechanisms", *NASA technical report Sp-8056*, 1970.

[10] M. Such Taboada, "High Fidelity FE Model of Clamp Band System with Catcher mechanism,"in *NAFEMS European Conference: Coupled MBS-FE Applications: A New Trend in Simulation*, Frankfurt, Germany, 2013.

[11] G. Baldesi, T. Voirin, A. Martinez Barrio, and M. Claeys, *Methodologies to perform GNC design and analyses for complex dynamic systems using multibody software*, Advances in Aerospace Guidance, Navigation and Control, pp. 431–450. Springer Berlin Heidelberg, 2011.

[12] F. Mastroddi, G. Goppotelli, et al, "Vibro-acoustic response analysis to pressure oscillations in a solid rocket motor – comparisons with the experimental fire-test data," in *XIX Congresso Nazionale AIDAA*, 2007, Forlì, Italy, 2007.

[13] "ECSS-E-ST-40C, Space Engineering – Software", Annex N, p.156, *ECSS-E-ST-40C*, European Cooperation for Space Standardization, ESA-ESTEC, 2009.

[14] “ECSS-E-HB-40-01A, Space Engineering – Agile software development handbook”, *ECSS-E-HB-40-01A*, European Cooperation for Space Standardization, ESA-ESTEC, 2009

[15] M. Toso, E. Pennestrì and V. Rossi, “ESA multibody simulator for spacecrafts' ascent and landing in a microgravity environment,” *CEAS Space Journal*, vol. 7, no. 3, pp. 335–346, 2015.

[16] NX Nastran 10 Verification Manual," Siemens Product Lifecycle Management Software, 2014.

[17] G. Hippmann, “An algorithm for compliant contact between complexly shaped surfaces in multibody dynamics,” in *ECCOMAS Thematic Conference Multibody*, Lisbon, Portugal, 2003.

[18] F. Cremaschi, S. Winter, V. Rossi, A. Wiegand, Launch vehicle design and GNC sizing with ASTOS. *CEAS Space Journal*, vol. 10, no.1, pp. 51–62, 2018