Rule-Based System Architecting of Earth Observing Systems: Earth Science Decadal Survey

Citation: Selva, Daniel, Bruce G. Cameron, and Edward F. Crawley. "Rule-Based System Architecting of Earth Observing Systems: Earth Science Decadal Survey." Journal of Spacecraft and Rockets 51, no. 5 (September 2014): 1505–1521.

As Published: http://dx.doi.org/10.2514/1.a32656

Publisher: American Institute of Aeronautics and Astronautics

Persistent URL: http://hdl.handle.net/1721.1/96924

Version: Author's final manuscript: final author's manuscript post peer review, without publisher's formatting or copy editing

Rule-based System Architecting of Earth Observing Systems: The Earth Science Decadal Survey

Daniel Selva[*], Bruce G. Cameron[†] and Edward F. Crawley.[‡]
Massachusetts Institute of Technology, Cambridge, MA, 02139

This paper presents a methodology to explore the architectural trade space of Earth observing satellite systems, and applies it to the Earth Science Decadal Survey. The architecting problem is formulated as a combinatorial optimization problem with three sets of architectural decisions: instrument selection, assignment of instruments to satellites, and mission scheduling. A computational tool was created to automatically synthesize architectures based on valid combinations of options for these three decisions, and evaluate them according to several figures of merit including satisfaction of program requirements, data continuity, affordability, and proxies for fairness, technical and programmatic risk. A population-based heuristic search algorithm is used to search the trade space. The novelty of the tool is that it uses a rule-based expert system to model the knowledge-intensive components of the problem, such as scientific requirements, and to capture the non-linear positive and negative interactions between instruments (synergies and interferences), which drive both requirement satisfaction and cost. The tool is first demonstrated on the past NASA Earth Observing System program and then applied to the Decadal Survey. Results suggest that the Decadal Survey architecture is dominated by other more distributed architectures in which DESDYNI and CLARREO are consistently broken down into individual instruments.

[*] Post-doctoral Associate, Aeronautics and Astronautics, 77 Massachusetts Ave Cambridge, MA 02139, AIAA Member, dselva@mit.edu.

[†] Lecturer, Engineering Systems Division, 77 Massachusetts Ave Cambridge, MA 02139, bcameron@mit.edu.

[‡] Full Professor, Aeronautics and Astronautics and Engineering Systems Division, 77 Massachusetts Ave Cambridge, MA 02139, AIAA Fellow, crawley@mit.edu.

I. Nomenclature

$\mathrm{B}(\cdot)$ = benefit metric (undiscounted)

$\mathrm{B_p}(\mathrm{S})$ = benefit of architecture S to panel p

$\mathrm{B_{p,o}}(\mathrm{S})$ = benefit of architecture S to objective o from panel p

$\mathrm{B_{p,o,r}}(\mathrm{S})$ = benefit of architecture S to requirement r from objective o from panel p

ω_p = relative importance of panel p

$\omega_{p,o}$ = relative importance of objective o from panel p

$\omega_{p,o,r}$ = relative importance of requirement r from objective o from panel p

$\mathrm{Bell}(\mathrm{n})$ = The nth Bell number

$\mathrm{C}(\cdot)$ = cost metric (Small-Is-Better)

$\mathrm{I_i}$ = an instrument

$\mathrm{P_k} = \{S_1, \ldots, S_n\}$ = a partition of the instrument set, i.e., a set of mutually exclusive and exhaustive subsets

$\{\mathrm{P}\}^*$ = the set of non-dominated partitions of instruments in the benefit-cost space

ϕ = the empty set

$\binom{\mathrm{n}}{\mathrm{k}}$ = binomial coefficient with indexes (n, k)

$\mathrm{N_S}$ = random variable defining the number of instruments successfully put into orbit

$\mathrm{R_{LV}}$ = reliability of the launch vehicle

$\mathrm{C_{ops}}$ = operations cost

$\mathrm{C_{S/C}}$ = total spacecraft cost (sum of non-recurring and recurring cost)

RSS = risk of schedule slippage (%)

$PR_{PACK}(\mathrm{P})$ = programmatic risk metric for the instrument-to-spacecraft assignment metric (Small-Is-Better)

$LR(\mathrm{P})$ = launch risk metric for the instrument-to-spacecraft assignment metric (Small-Is-Better)

$\mathrm{TRL_i}$ = technology readiness level of instrument i

$\mathrm{h}(\mathrm{p})$ = entropy of a probability distribution p: $\sum_{\mathrm{j}} \ p_j \log_2(\ \mathrm{p}_j)$

II. Introduction

In 2004, the NASA Office of Earth Science, the National Oceanic and Atmospheric Administration (NOAA) National Environmental Satellite Data and Information Service, and the U.S. Geological Survey (USGS) Geography Division asked the National Research Council (NRC) Space Studies Board to "conduct a decadal survey to generate consensus recommendations from the Earth and environmental science and applications communities regarding a systems approach to space-based and ancillary observations that encompasses the research programs of NASA; the related operational programs of NOAA; and associated programs such as Landsat, a joint initiative of USGS and NASA." [1]

In response to this request, an ad-hoc NRC committee consisting of experts from different disciplines of the Earth sciences produced a report known as the Earth Science Decadal Survey, or simply the "Decadal Survey". The Decadal Survey lays out a reference architecture for an integrated Earth Observing Satellite Systems (EOSS) for the next decade that will fulfill the needs of all the scientific communities in terms of space-based measurements, while also providing essential societal benefits [1].

The Earth Science Decadal Survey is a program that consists of a sequence or campaign of Earth observing missions, i.e., an Earth Observing Satellite System (EOSS). Traditionally, the high-level design of EOSS has been done at the individual mission level. During the mission concept formulation phase, organizations conduct studies on the relative merit of different mission concepts. For example, different constellation designs are compared in terms of coverage, spatial resolution and temporal resolution versus cost. However, the emphasis of this paper is on program architecture rather than on mission architecture, where a program is defined as a sequence or campaign of related missions. Designing a program rather than designing missions individually is advantageous because there are numerous coupling between missions in a program. First, missions in the same program will typically consume resources from the same budget lines. Second, there are synergies between instruments across missions that impose constraints for example on the launch dates of the missions, so as to ensure some overlap between missions for calibration purposes, complete overlap to increase coverage, or to avoid a measurement gap between two similar instruments. Many other examples of couplings exist at all levels of the systems engineering triangle: performance, cost, schedule, and risk. See [2] for a thorough discussion of these issues.

The Decadal reference architecture consists of 15 missions for NASA and 2 missions for NOAA. A total of 39 instruments were assigned to these 17 missions on the basis of a variety of technical, scientific, and programmatic factors, including synergies between instruments, data continuity, orbit compatibility, different instrument maturity levels, and expected yearly budget. For each mission, the report provides a description of the scientific objectives fulfilled by the mission, the physical parameters measured, the instruments used, the orbit required, a rough estimation of the lifecycle mission cost, and the expected mission launch date.

This reference architecture was created based on a series of assumptions that were deemed reasonable at the time, such as mission cost estimates, yearly budget, and precursor missions outside the scope of the Decadal Survey study that were expected to be flying by this decade. However, some of these assumptions are no longer true. First, mission cost estimates according to NASA have increased by 70% on average [3]; note that this number includes missions that have not yet started development and therefore have not yet had the opportunity to suffer any development-related cost overrun. Second, NASA budgets in the last few years show that the budget available for Decadal Survey mission development will be much lower than (about 50% of) the $750M/yr that was assumed in the Decadal Survey report. Finally, some of the precursor missions have failed, or have been severely delayed (e.g., the Orbiting Carbon Observatory mission or OCO was lost at launch; the National Polar-orbiting Operational Environmental Satellite System, or NPOESS, was delayed, reorganized, and descoped; and the Glory mission was also lost at launch). Therefore, the question arises whether the reference is still a good architecture given the current assumptions.

One answer would be to redo the Decadal study, but that would require massive amounts of resources – the cost of the original study was $1.4M according to a personal conversation with NRC staff. Another option is to encode the knowledge that went into the first architecting process in a computational tool and then use the tool to find good architectures under a variety of scenarios. This is the approach adopted in this paper. A discussion of the limitations of the approach, in particular the knowledge elicitation process, is provided in the Conclusion.

Since the publication of the foundational literature of the field of systems architecting [4], the interest in developing tools and methods to improve the process has grown [5], especially in the area of space systems [6], [7], [8], [9], [10], [11]. The motivation is clear: architectural decisions have unique leverage on the ability of the system to deliver value to stakeholders sustainably. The key example is that of the fraction of lifecycle cost that is committed after the system architecture is fixed, which has been proved to be around 70%, whereas only about 10-20% of the lifecycle cost is actually spent during this phase [12].

Computational tools, sometimes borrowed from other fields (e.g., software engineering, network analysis, formal design theory, optimization, heuristic search, decision analysis) have been used in the past to support several aspects of the system architecting process. UML and SysML are used to represent complex architectures using hierarchical models and different views of the architecture [13], [14], [15]. These and other tools such as Petri Nets are used to simulate the emergent behavior of architectures [13], [16]. Combinatorial optimization techniques and search algorithms are used to explore the architectural tradespace [17], [18], [19]. Multi-attribute utility theory and the Analytic Hierarchy Process amongst others are used to evaluate architectures under multiple conflicting objectives [20].

While architecture studies concerning large Earth observing programs of the nature of the Decadal Survey have been conducted in the past at NASA and other organizations, little has been published on the results or process of these studies. A few studies have been published by the NRC that look at architectural aspects in general of Earth Observing programs (see for example [21], [22]), although most of them are general as opposed to specific, descriptive as opposed to prescriptive, and qualitative as opposed to quantitative.

The NASA Earth Observing System (EOS) is a notable exception, and several early program formulation studies have been published. Matossian applied mixed-integer optimization techniques to the design of the EOS constellation [23], [24], [25] for optimal cost, performance, and risk. The NOAA Polar Operational Environmental Satellite Systems (NPOESS) was also the object of architectural studies published by Rasmussen, who analyzed the differences in lifecycle cost and risk between architectures using large, medium, and small satellites [26], [27]. While Matossian analyzed a very large number of architectures with low fidelity, Rasmussen analyzed only a handful of architectures at a higher level of fidelity (although still far from the fidelity required for a point design).

More recently, several architecture studies on the Global Earth Observing System of Systems have been published. Martin presents an architecture tool to assess the societal benefits of Earth observing assets [28]. Rao et al introduce a tool based on Petri nets that can simulate the operation of the GEOSS in a certain situation. However, such studies focus mostly on representing complex architectures or simulating their operation, but rarely on the trade space search process, on which this paper focuses.

When attempting to apply state-of-the-art system architecting tools to the problem of architecting EOSS, two main challenges appear, both of them due to the large amount of expert knowledge needed to solve the problem. The first problem is related to the subjective expert judgment of requirement satisfaction. The second problem is related to the ability to model emergent behavior such as scientific synergies. These two challenges are introduced in the following paragraphs.

Evaluating the ability of an EOSS architecture to meet program requirements necessitates a tremendous amount of expert knowledge. This assessment process is necessarily based on expert judgment, because more objective approaches such as end-to-end simulation [29] or Observing System Simulation Experiments [30] are too computationally expensive to be applied at the architecting level, where a large number of architectures are being considered. Since a program of the size of the Decadal Survey has several measurement requirements from each of many disciplines of the Earth sciences, and since each measurement requirement may itself consist of several requirements at the attribute level (e.g., spatial resolution, temporal resolution, accuracy, or combinations thereof), it follows that a very large number of assessments need to be made by experts. In the worst case, if there are N architectures being considered under M requirements, each requirement needs to be checked against each architecture, yielding a total of NM assessments that need to be made by experts, let alone recomputed under sensitivity analysis.

Regardless of the data structure chosen to represent this knowledge, the resulting computational tool will in general have a large number of these data structures. This threatens the scalability of the tool, because as more knowledge is added to the tool, it becomes more difficult to modify the knowledge base. Furthermore, as the size of the knowledge base grows, it becomes increasingly hard to trace the results provided by such tool back to the reasoning behind expert knowledge judgments. This traceability problem is especially important for public systems developed with tax-payers money, as the organization is accountable for each decision made, and must be capable of explaining the rationale behind it.

The need for increased scalability and traceability calls for the utilization of knowledge-based systems. In particular, rule-based expert systems (a.k.a. production systems) were chosen for this application due to their simplicity, natural flexibility and traceability.

The rule-based approach also provides a good framework to model emergent behavior such as scientific synergies. When interviewing experts on the ability of a certain architecture to meet scientific requirements, the necessity of modeling these synergies became obvious, as they can potentially drive a large fraction of the scientific value of a mission. For example, the ability of a radar altimeter to meet accuracy requirements is not independent of the rest of architecture. Instead, it depends on orbital parameters, and on the presence of other instruments such as microwave radiometers or GNSS receivers, on the same spacecraft, or close enough to enable cross-registration of the datasets. This fact is extremely important because it precludes the utilization of most efficient algorithms (e.g., linear optimization, dynamic programming) which require some degree of additivity in the worth of an architecture, or equivalently, an independence between the worth of different elements in the architecture (e.g., instruments). Rule-based systems provide a scalable way of modeling these synergies with simple logical rules. In other words, one can either efficiently solve a (highly) simplified architecting problem, or attempt to do as good a job as possible in a much more representative problem formulation. This paper claims that rule-based systems can help solve architecting problems using the latter approach.

An expert system is “a computer program designed to model the problem-solving ability of a human expert” [31]. In order to do that, an expert system uses large bodies of heuristic – expert – knowledge. In a rule-based expert system (RBES), expert knowledge is encapsulated in the form of logical rules. In the words of Feigenbaum, considered the major creator of the first expert system, these rules map the knowledge “over from its general form (first principles) to efficient special forms (cookbook recipes)” [32]. This is in opposition to other kinds of expert systems that primarily use different data structures to store expert knowledge, such as Minsky’s frames [33], which are more versatile data structures, intuitively similar to objects in object-oriented programming.

In RBES, a logical rule is composed of a set of conditions in its left-hand side (LHS), and a set of actions in its right-hand side (RHS). The actions in the RHS are to be executed if the conditions in the LHS are all satisfied. An example of a logical rule is the following: ”if the car won’t start (LHS); then check the electrical engine (RHS)”.

RBES consist of three major elements: a fact database, a rule database, and an inference engine. The fact database contains relevant pieces of information about the specific problem at hand called facts. Information in facts is organized according to predetermined data structures similar to C structures and Java Beans, with properties and values (e.g. a fact of type car may contain a property make, a property model, and a property price, amongst others). These data structures are called templates – and their properties, slots- in many RBES development tools. Facts can be asserted, modified, and retrieved from the database anytime. The rule database contains a set of logical rules that contain the domain knowledge. The LHS of these rules may match one or more facts in the working memory. The RHS of these rules define the actions to be executed for each of these matches, which typically include asserting new facts, modifying the matching facts, performing calculations, or showing some information to the user.

The inference engine performs three tasks in an infinite loop: a) pattern matching between the facts and the LHS of the rules in the working memory and creation of activation records (also known as the conflict set); b) while there remain activation records, select the next rule to execute, or fire (conflict resolution); c) execute the selected rules' RHS (deduction). Most current rule engines are based on the Rete algorithm, developed by Forgy [34]. The Rete algorithm is faster than other algorithms because it "remembers" prior activation records in a network in memory called the Rete network. The Rete network is very efficient in speeding up the search process because most of the time, the network does not change much between iterations. Note that the improvement in computational time comes at the price of increased use of memory.

The goal of this research is two-fold. The first goal is to develop a methodology and tool to support the architecting process of EOSS. This methodology will use RBES to address the limitations of current tools handling knowledge-intensive problems. More precisely, the methodology must be capable of modeling interactions between instruments (e.g., synergies, interferences), and tracing the entire value delivery loop from stakeholder needs to instrument and mission attributes and data processing algorithms. The resulting tool is to be validated with a past EOSS, namely the NASA Earth Observing System.

The second goal is to conduct an architectural study of the Earth Science Decadal Survey using this tool and method. The value of the reference architecture laid out in the NRC report will be assessed, and recommendations will be provided as to how to improve this architecture.

III. Problem Formulation

This section describes a simple architectural model for an EOSS, and then uses the model to formulate the architecting problem as an optimization problem. Architectural decisions, figures of merit, and constraints are identified.

A. Model Overview

The model consists of two hierarchical decompositions, one mainly in the domain of functions and processes, and one mainly in the domain of form and structure. Both hierarchies are linked at the level of measurements and data products. The functional decomposition of the system assumes that most of the benefit provided by EOSS is in the measurements taken by the system, and the data products developed from these measurements. Stakeholder needs such as "reducing the uncertainty in the water cycle" are at the top level of the hierarchy. These needs are recursively decomposed into lower-level objectives up to the level of measurement requirements (e.g., "measuring soil moisture with an accuracy of 5%, a spatial resolution of 10km, and a temporal resolution of 2-3 days").

The form decomposition of the system is instrument-centric, as opposed to mission-centric, i.e., measurements are produced by instruments as opposed to missions. Instruments are gathered into missions, and the system or campaign is a collection of missions. Synergies between instruments on the same mission are explicitly captured by rules modeling the combination of multiple data products to create new data products.

A pictorial summary of this architectural model is provided in Figure 1. When read from left to right, this diagram illustrates the architecting process; when read from right to left, this diagram suggests a method to assess how a certain architecture satisfies stakeholder needs.

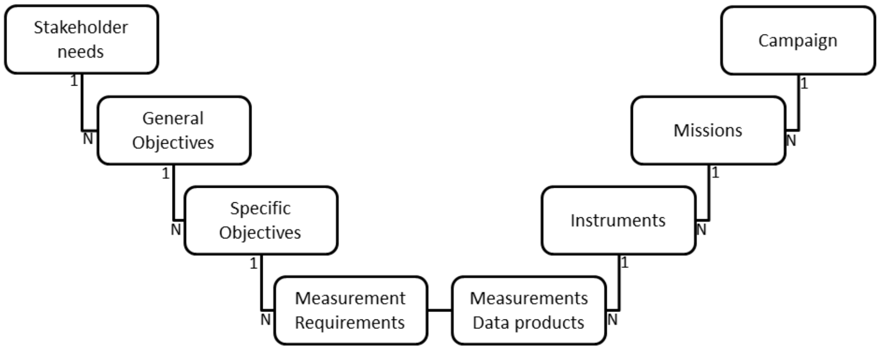

Figure 1: Architectural model for an EOSS

Throughout this paper, it is assumed that the hierarchy in the functional domain, from stakeholder needs to measurement requirements, is available as an input to the architecting process, albeit with a certain degree of fuzziness (e.g., temporal resolution of "2-3 days"). See [35] for a detailed discussion on strategies related to eliciting ambiguous stakeholder needs.

Based on the model presented in Figure 1, three main classes of architectural decisions for EOSS are identified, namely instrument selection (what is the best set of instruments to satisfy the measurement requirements?), instrument-to-spacecraft allocation (a.k.a. instrument packaging, how do we group instruments together in spacecraft?) and mission scheduling (what is the optimal launch sequence for the missions?). The architectural problem is formulated as a series of optimization problems based on these three classes of architectural decisions.

B. Architectural Decisions

The major architectural decisions for an EOSS have already been mentioned, namely instrument selection, instrument-to-spacecraft assignment, and mission scheduling. In the rest of this section, the instrument-to-spacecraft assignment class of decisions, for which results are presented, is more rigorously defined. The reader is referred to [36] for a detailed description of the instrument selection and mission scheduling decisions.

Instrument-to-spacecraft assignment (a.k.a. instrument packaging) decisions deal with the assignment of instruments to spacecraft and orbits. For instance, given a set of m instruments, one option is to fly all the instruments on a large monolithic spacecraft; another option is to fly m dedicated spacecraft, and there are obviously many options in between. More formally, an instrument-to-spacecraft assignment architecture is represented by a partition of the set of m selected instruments into $1 \leq n \leq m$ subsets (spacecraft) such that the n subsets are mutually exclusive and exhaustive:

$$P = \{S_1, S_2, \dots, S_n\}; \bigcup S = I; \bigcap S = \phi \tag{1}$$

The representation chosen for P is an array of m integer variables $\mathrm{P} = [\mathrm{x_1}, \mathrm{x_2}, \dots, \mathrm{x_m}]$ where $1 \leq \mathrm{x_i} \leq 1 + \max_{\mathrm{j}<i} \mathrm{x_j} \leq \mathrm{m}\ \forall \mathrm{i} = 1..\mathrm{m}$, and where $\mathrm{x_i} = \mathrm{k}$ indicates that instrument i is assigned to spacecraft k. This representation ensures by construction that the subsets are non-empty, mutually exclusive, and exhaustive. Thus, architectures generated by mutation or other means are ensured to be valid architectures. The size of this region of the trade space is given by the Bell Numbers $|\{\mathrm{P}\}| = \mathrm{Bell}(\mathrm{m})$, which are sums of Stirling numbers of the second kind $S(m, k)$:

$$S(m,k) = \frac{1}{k!}\sum_{i=0}^{k}(-1)^i \binom{k}{i} (k-i)^m \tag{2}$$

$$Bell(m) = \sum_{k=0}^{m} S(m,k) \tag{3}$$

A variant of this class of decisions is to have a predefined number of notional spacecraft $\mathrm{S_1}, \mathrm{S_2}, \dots, \mathrm{S_m}$, which could for example represent orbits (e.g., a dawn-dusk orbit, a morning and orbit, and an afternoon orbit), and then assign instruments to any number of these spacecraft. In the most general case, the corresponding trade space has a size of 2^{mN}, as each instrument can be assigned to any subset of spacecraft, and there are 2^{m} different subsets including the empty set.

The instrument-to-spacecraft assignment problem can also be viewed as a clustering problem, as the goal is to group the instruments in clusters. Regardless of their representation, there are two main classes of forces driving these decisions: positive interactions between instruments (i.e. attractive forces that tend to create large satellites), and negative interactions between instruments (i.e. repulsive forces that tend to create small satellites). In the case of EOSS, these forces lie on three different domains: the science domain, engineering domain, and the programmatic domain. In the science domain, there are positive interactions through synergies between instruments, and negative interactions through the limitations in spacecraft resources (e.g., power, data rate) that may affect the science output of an instrument. In the engineering domain, most interactions are negative as instruments interfere with each other in a variety of ways (e.g., thermal interactions, mechanical interactions, electromagnetic interactions). A few examples of positive engineering interactions occur when two instruments can share common resources (e.g., two instrument operating on the same band can share a single dish). Finally, in the programmatic domain, most interactions are again negative in the form for instance of schedule slippages of a single instrument that drive the launch dates of many others. Positive programmatic interactions could be defined in terms of instruments that benefit from the same technological investments. A thorough discussion of these issues can be found in [2] or [36] (pp 136-141).

C. Figures of Merit

Seven figures of merit were used to compare EOSS architectures, namely scientific and societal benefit, lifecycle cost, programmatic risk, launch risk, discounted benefit, fairness, and data continuity. From these seven figures of merit, only scientific and societal benefit and lifecycle cost are used as metrics in the instrument-to-spacecraft assignment problem. Launch and programmatic risk are used as constraints to guide the search process. Normalized risk, the normalized average of launch and programmatic risk, is also used as a secondary metric to down-select architectures that have high scientific and societal value and are close to the science-cost Pareto frontier. These four figures of merit are described below. The remaining three figures of merit, namely discounted benefit, fairness, and data continuity are not used in the instrument-to-spacecraft assignment problem and therefore are not described in this paper for the sake of brevity. The interested reader can find a full description of these figures of merit, used in the instrument selection and mission scheduling problems, in [36].

1. *Scientific and societal benefit*

The scientific and societal benefit metric is based on assessing how well requirements from different stakeholders are satisfied. Requirement satisfaction by a given architecture is assessed using the VASSAR methodology. VASSAR is introduced in the next section and described in detail in [37]. The final score provided by VASSAR is an aggregation of the scores of satisfaction for each requirement. In the simplest case, this aggregation is just a composition of weighted averages:

$$B(\mathrm{S}) = \sum_{\mathrm{p}} \omega_p \mathrm{B}_{\mathrm{p}}(\mathrm{S}) = \sum_{\mathrm{p}} \omega_p \sum_{\mathrm{o}} \omega_{p,o} \mathrm{B}_{\mathrm{p,o}}(\mathrm{S}) = \sum_{\mathrm{p}} \omega_p \sum_{\mathrm{o}} \omega_{p,o} \sum_{\mathrm{r}} \omega_{p,o,r} \mathrm{B}_{\mathrm{p,o,r}}(\mathrm{S}) \tag{4}$$

where S is a subset of instruments; the subindex p indicates the panel (e.g., climate, weather), the subindex o indicates an objective within a panel, and the subindex r indicates a requirement within an objective; ω_x are relative weights of the panels (ω_p), objectives ($\omega_{p,o}$), and requirements ($\omega_{p,o,r}$); and $\mathrm{B}_{\mathrm{x}}(\mathrm{S})$ refers to the satisfaction of panels ($\mathrm{B}_{\mathrm{p}}(\mathrm{S})$), objectives ($\mathrm{B}_{\mathrm{p,o}}(\mathrm{S})$), and requirements ($\mathrm{B}_{\mathrm{p,o,r}}(\mathrm{S})$) by S. More sophisticated aggregation schemes are proposed in [37].

It is important to note that the scores for each requirement $\mathrm{B}_{\mathrm{p,o,r}}(\mathrm{S})$ cannot be decomposed in a sum of instrument-specific contributions (i.e., $\mathrm{B}_{\mathrm{p,o,r}}(\mathrm{S}) \neq \sum_{\mathrm{i}} \mathrm{B}_{\mathrm{p,o,r}}(\mathrm{I}_{\mathrm{i}})$), due to synergies and redundancies between instruments.

2. *Lifecycle cost*

The lifecycle cost metric has the following components: payload cost; bus cost; integration, assembly and testing cost; program overhead; launch cost; and operations cost. Some of these components are further divided in recurring and non-recurring cost. It is important to note that the goal of this metric is not to provide accurate absolute cost estimates, but rather to provide a metric that has enough sensitivity across different architectures.

For non-recurring and recurring bus, integration and testing, and program overhead costs, cost estimation relationships are taken from [38]. Since cost estimating relationships (CERs) for bus cost are mass-based, an estimation of the mass budget of the satellite is required. This mass budget is obtained using a simplified iterative spacecraft design algorithm that utilizes physics-based equations and parametric relationships to design the satellite's subsystems. The algorithm is introduced in the next section. See [39] or [36] (pp 151-159) for a full description of the algorithm.

The payload cost model is based on the NASA Instrument Cost Model [40], which uses mass, power, and data rate as parameters. The launch cost model is based on the assumption that each spacecraft is launched individually, and that it falls into one of the launch vehicle categories defined in **Table 4**. The assignment of each spacecraft to one of these launch vehicle categories is done using two main criteria, namely launch vehicle performance and launch vehicle fairing dimensions. For operations cost, the NASA Mission Operations Cost Model was used[§]. This model provides a rough order of magnitude estimate of the yearly cost of orbital operations for a spacecraft of a certain type and development cost. The cost estimating relationship used in this model for Earth observation satellites is provided below. The standard error and range of applicability of this CER are unknown.

$$\mathrm{C_{ops}(FY00\$k)} = 0.035308 \cdot \mathrm{C_{S/C}(FY00\$k)}^{0.928} \cdot \mathrm{T(yrs)} \tag{5}$$

An expected cost overrun is systematically added to the total initial lifecycle cost because it is noted that the probability of schedule slippage and cost overrun is larger in larger missions. Some studies have suggested that the probability of schedule slippage is correlated with the maturity of the technology used in the mission, as described by the initial mission TRL [41]. The expected cost overrun is computed in two steps. First, given the initial TRLs of all the instruments in a given spacecraft, the expected schedule slippage (RSS) in % of total development time is computed as a function of the lowest instrument TRL in the spacecraft, using the findings by Saleh et al [41]:

$$\mathrm{RSS\ (\%)} = 829 \cdot \mathrm{e}^{-0.56 \cdot \mathrm{min_{TRL}}} \tag{6}$$

This regression has an R^2 =0.94 for 28 data points. While some recent articles have discussed the validity of using TRL as a quantitative metric for this purpose [42], the idea behind the study, namely that less mature technologies are more likely to suffer from schedule slippage, remains a fair assumption.

Second, this expected schedule slippage is transformed into a cost overrun by applying the relationship found by Weigel and Hastings [43]:

$$\text{cost overrun (\%)} = 0.24 \cdot \mathrm{RSS(\%)} + 1.7 \tag{7}$$

[§] The NASA Mission Operations Cost Model was retrieved from http://cost.jsc.nasa.gov/MOCM.html (last accessed Jan 24th 2012).

This regression has a much lower R^2 =0.56 for 15 data points. This relationship was chosen because of its simplicity, even though it neglects other important sources of cost overrun.

Finally, since different contributions to the overall lifecycle cost estimation come from different sources and are expressed in different years' dollars, they are all corrected for inflation and translated into $FY00M.

3. *Programmatic risk*

In the context of this paper, programmatic risk refers to risk of schedule slippage and cost overrun. More precisely, it only refers to the components of this programmatic risk that are architecturally distinguishing. Hence, causes of programmatic risk that do not vary across architectures are not considered.

Even though some programmatic risk is captured in the lifecycle cost metric, some users may find it clearer and preferable to have this information in a separate metric. Hence, two different metrics were developed for programmatic risk: one that is more appropriate for selecting instruments, and one that is more appropriate for assigning instruments to spacecraft. The programmatic risk metric used in the instrument-to-spacecraft assignment problem is described below. The reader can find a description of programmatic risk metric used in the instrument selection problem in [36].

In the instrument-to-spacecraft assignment problem, the following definition of the programmatic risk metric is used:

$$PR_{PACK}(\mathrm{P}) = \frac{1}{n}\sum_{\mathrm{S}_j \in \mathrm{P}} \Delta_{\mathrm{PACK,j}} \tag{10}$$

$$\Delta_{\mathrm{PACK,j}} = \begin{cases} 1 \text{ if } \mathrm{median}_{\mathrm{i}\in \mathrm{S}_j} \mathrm{TRL_i} - \mathrm{min}_{\mathrm{i}\in \mathrm{S}_j} \mathrm{TRL_i} > 1 \\ 0 \text{ otherwise} \end{cases} \tag{11}$$

The idea behind this programmatic risk metric is to avoid situations in which a high risk instrument could delay the deployment of important mature instruments. A high value of $\mathrm{PR_{PACK}}$ indicates that there is an imbalance in the TRLs of the instruments, which could be improved if instruments with similar levels of maturity were binned together.

4. *Launch risk*

Distributed architectures are perceived as more desirable than monolithic architectures in terms of launch risk, because they are more robust to a single launch failure. Although "real" launch risk measured as the average number of instruments successfully put into orbit is independent of the assignment of instruments to satellites assuming identical launch vehicle reliabilities, perceived launch risk varies across architectures due to risk aversion [2]. The concept of entropy from information theory is used to model this risk aversion. Let p be an array containing the number of instruments in each satellite $\mathrm{p_j} = \left|S_j\right|$, and $\hat{\mathrm{p}}$ be the convex normalization of p (i.e. $\widehat{\mathrm{p_j}} = \frac{\mathrm{p_j}}{\sum_{\mathrm{i}} \mathrm{p_i}}$). $\widehat{\mathrm{p_j}}$ can be interpreted as the probability of a random instrument to belong to satellite j. The risk metric can now be defined as one minus the entropy of this probability distribution:

$$\mathrm{LR(P)} = 1 - \mathrm{h}(\,\hat{\mathrm{p}}\,) = 1 + \sum_{\mathrm{j \in P}} \widehat{\mathrm{p_j}} \log_2\!\left(\widehat{\mathrm{p_j}}\right) \tag{12}$$

The entropy of a probability distribution of length N is a real value between 0 and $\log_2 \mathrm{N}$. The value of 0 is achieved by a delta probability distribution with all its weight in a single value. This is the equivalent of the monolithic architecture with only one satellite in the instrument packaging problem. The value of $\log_2 \mathrm{NS}$ where NS is the number of spacecraft in the architecture is achieved by a completely distributed architecture with as many instruments as satellites. Thus, LR(P) is a Small-Is-Better metric, as should intuitively correspond to a risk metric.

D. Formulation as an Optimization Problem

Given the three classes of architectural decisions defined previously, a straightforward problem formulation is to decompose the overall architecting problem in a sequence of three sub-problems: instrument selection problem, instrument-to-spacecraft assignment problem, and mission scheduling problem.

For example, the instrument-to-spacecraft assignment problem that is highlighted in this paper is defined as the problem of finding the set of non-dominated partitions $\{P\}^*$ that optimize the trade-off between benefit (i.e. requirement satisfaction) and cost, with controlled launch and programmatic risk.

$$\{P\}^* = \underset{\mathrm{S}}{\operatorname{argmin}} \left[-\mathrm{B(P)} \quad \mathrm{C(P)}\right] \tag{19}$$

subject to:

$$LR(\mathrm{P}) \leq \mathrm{LR}_{max}$$

$$PR_{PACK}(\mathrm{P}) \leq \mathrm{PR}_{max}$$

The summary of metrics and constraints utilized in each problem is shown in Table 1.

Table 1: Summary of the metrics utilized to architect EOSS

Figure of Merit	Description	Selection	Assignment	Scheduling
Science	Undiscounted aggregated benefit to all panels with synergies between instruments.	Metric	Metric	
Lifecycle cost	Payload, bus, integration, overhead, launch, and operations cost. Bus cost based on mass budget which incorporates interferences between instruments.	Metric	Metric	
Programmatic risk (SEL)	% of instruments in program that require significant development (TRL < 5)	Constraint		
Programmatic risk (PACK)	Median over all spacecraft of the offset between min and median TRL over all instruments on the spacecraft		Constraint	
Launch risk	Entropy of a pseudo-probability distribution given by the normalized number of instruments on each spacecraft.		Constraint	
Fairness (SEL)	Minimum panel score	Constraint		
Fairness (SCHED)	Worst case deviation between curves of benefit delivery to panels over time			Constraint
Data continuity	Weighted increase of observations of critical measurements due to an architecture			Metric
Discounted benefit	Time-discounted aggregated benefit to all panels			Metric

We perform the three tasks in the order selection-assignment-scheduling because it is essential to know the instruments selected in order to think about the packaging architecture, and it is essential to know what the missions are before scheduling them. However, the three sub-problems are also coupled in other ways: one cannot, as it has been pointed out, make an optimal decision in the instrument selection problem without thinking about the instrument-to-spacecraft assignment. Both the benefit and cost of a subset of instruments depend on how these instruments are distributed into missions (e.g., synergies, packaging efficiencies of buses and launch vehicles). Similarly, it is hard to make an optimal instrument-to-spacecraft assignment decision without thinking about the scheduling of the missions, since one may want to fly an instrument that closes an important data gap in the spacecraft that is to be launched first. These and other couplings are illustrated in the N^2 diagram of Figure 2.

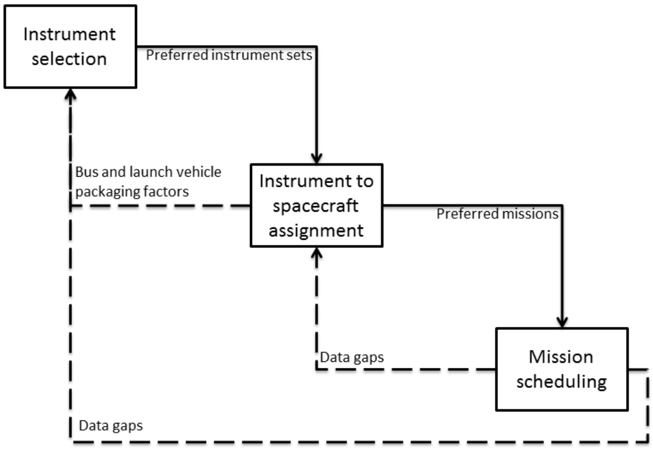
Instrument selection
Preferred instrument sets
Bus and launch vehicle packaging factors
Instrument to spacecraft assignment
Preferred missions
Data gaps
Mission scheduling
Data gaps

Figure 2: N^2 diagram showing coupling between instrument selection, instrument-to-spacecraft assignment, and mission scheduling problems

As a consequence of this coupling, simply solving the individual sub-problems and combining the results will in general lead to suboptimal solutions, just as designing the aerodynamics of an airplane without coupling it to the structures dynamics leads to suboptimal designs.

This problem can be treated in three different ways. The first option is to solve the global problem without decomposition. This would imply having a single optimization problem with an architectural vector containing the three types of decisions. However, in reality, this problem is intractable for relatively small numbers of instruments. Furthermore, this poses additional implementation problems, as it would require dealing with mixed types of variables and variable chromosome/design vector lengths. A second strategy consists in iteration. Sub-problems can be solved individually in a loop until a set of termination criteria are met. The problem with this approach is that there is no guarantee of convergence in general. Furthermore, iteration can also be extremely resource-consuming. Finally, heuristic rules can be used to guide the search in a sub-problem towards solutions that are likely to be good in the other sub-problems. For instance, one can include a constraint so that architectures that make it impossible to find a scheduling that covers a certain set of key data gaps are eliminated from further consideration in the selection or assignment problems. In the context of this study, the chosen strategy was a hybrid of iteration and heuristic rules capturing the couplings between the problems.

IV. Methods

This section describes the tools, methods, and algorithms used to solve the optimization problems formulated in the previous section. In particular, the VASSAR methodology - at the core of the scientific benefit metric -, the spacecraft design algorithm - the basis of the cost model -, and the rule-based heuristic search algorithm used to explore the trade space are succinctly described.

A. Science assessment using VASSAR

The VASSAR methodology was used to assess the relative benefit of different EOSS architectures. This methodology uses a rule-based system for increased traceability and scalability of the science valuation process. The input to VASSAR is an EOSS architecture as defined in the previous section, and the output is two-fold: one or more fuzzy numbers representing the level of satisfaction of different panels or stakeholders, and a set of explanations that show how these numbers are traced back to specific requirements, instruments, and data processing algorithms. The main steps of this methodology are illustrated in Figure 3.

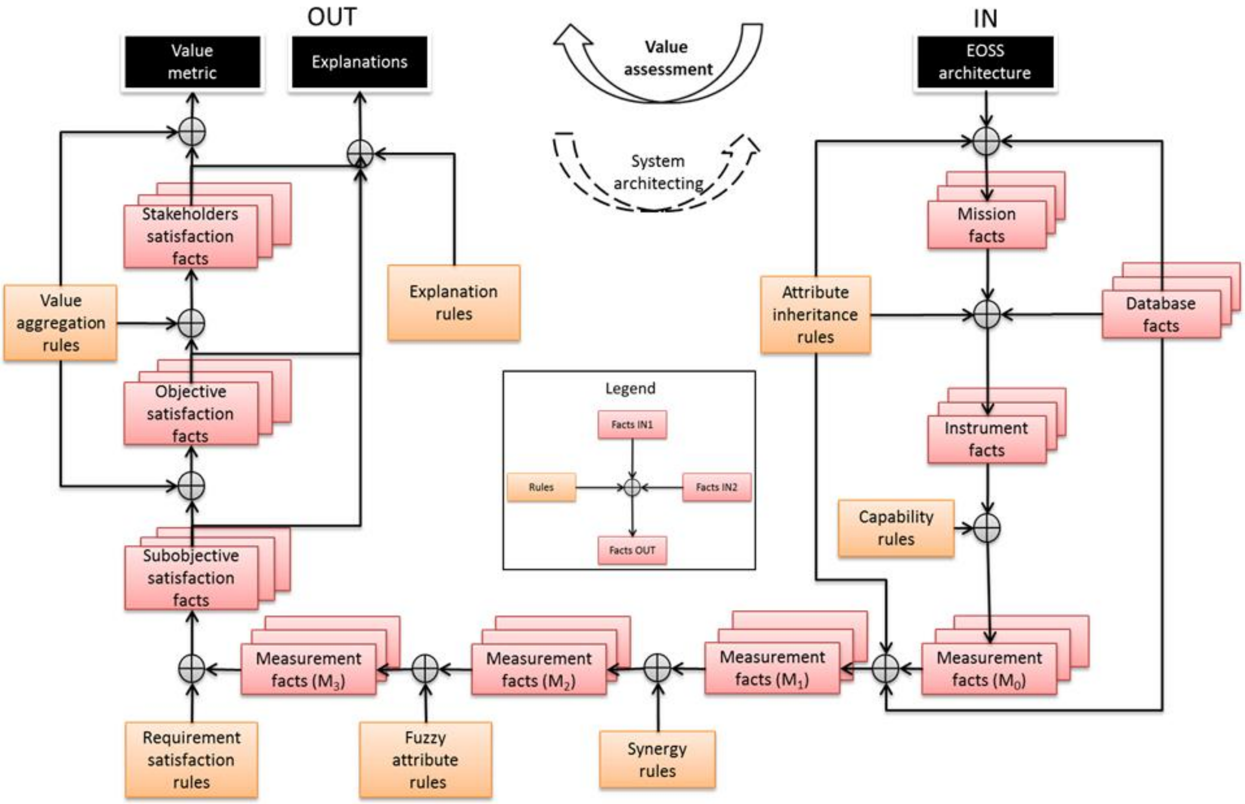

Figure 3: Application of the VASSAR methodology for assessing the scientific value of EOSS

If read from left to right, Figure 3 loosely illustrates the process of architecting an EOSS, starting with the decomposition of stakeholder needs in a set of requirements, and continuing with the selection of a set of instruments and spacecraft to meet these requirements. If read from right to left, Figure 3 shows the process that

VASSAR follows to assess the value of an architecture, starting with the missions and instruments, assessing how well each requirement is satisfied, and aggregating requirements into objectives and stakeholders. This process is described in more detail below.

First, from the EOSS architecture in the top right corner, the corresponding mission and instrument facts are asserted, and their characteristics are inherited from information contained in upper levels in the hierarchy (e.g. instrument orbit is inherited from mission orbit) or from a database (e.g. instrument field of view is inherited from an instrument database). Second, instrument capability rules assert an initial set of measurement facts (M_0 in Figure 3) associated to each of the manifested instruments. Specific examples of instrument capability rules are given in the results section. Attribute inheritance rules are used again to regulate how attributes are inherited between missions, instruments, and measurements: measurement attributes are inherited from their instrument, their mission, or they are derived from combinations of other mission and instrument attributes, yielding a modified set of measurements (M_1 in Figure 3). For instance, the ground spatial resolution of an instrument is computed from its orbital parameters, angular resolution, and viewing geometry. Third, the initial set of measurement capabilities is substantially changed with new and modified measurements through the synergy rules (M_2 in Figure 3). Synergy rules are really at the core of this methodology, as full satisfaction of requirements usually comes from data products that originate from combinations of other data products from different instruments. A few examples of synergy rules are shown in **Table 2**. Synergy rules are applied only to instruments that are either on the same spacecraft or on different spacecraft in a train configuration, where satellites have very similar orbital parameters with small differences in mean anomaly (e.g., the NASA A-Train). Hence, synergies between spacecraft on different orbits are not taken into account.

Table 2: A few representative examples of synergy rules

Synergy rule	**IF (original data products)**	**THEN (new data products)**
Soil moisture disaggregation data processing algorithm	Soil moisture (high accuracy, low spatial resolution) from L-band radiometer Soil moisture (low accuracy, high spatial resolution) from L-band radar	Soil moisture (high accuracy, medium spatial resolution) from combination
Spatial resampling	Any parameter (high spatial resolution, high variable error)	Any parameter (lower spatial resolution, lower variable error)
Altimetry error budget	Sea level height (high tropospheric error, high orbit determination error, poor overall accuracy) from radar altimeter only Atmospheric sounding from microwave	Sea level height (low tropospheric error, low orbit determination error, good overall accuracy)

	radiometer Orbit determination from GNSS receiver	
Cloud mask	Many parameters (error due to presence of clouds) Cloud images	Many parameters (low error due to presence of clouds)

Fourth, a set of fuzzy attribute rules transform back and forth between fuzzy numbers (e.g., high spatial resolution) and crisp numbers (e.g., 0.5m) as required. Fuzzy attribute rules are based on Zadeh's fuzzy sets and use simple triangular membership functions. Fifth, this new set of measurement capabilities is compared against measurement requirements defined in the requirement rules, which assert requirement satisfaction facts (full or partial satisfaction). Sixth, requirement satisfaction facts are logically combined to produce objective satisfaction facts, and objective satisfaction facts are combined to produce panel satisfaction metrics. These two steps occur through the value aggregation rules, which in the simplest case are weighted averages, but which may include any logical or arithmetic operator (e.g., an objective may be fully satisfied if at least 2 out of 3 requirements are fully satisfied). Finally, panel satisfaction metrics are weighted to produce an overall EOSS score. For a more detailed explanation of each step in the methodology, the reader is referred to [37] and [36] (pp 161-174).

B. Spacecraft Design Algorithm

The spacecraft design algorithm takes the payload characteristics as input and uses equations and rules-of-thumb to provide several outputs, including the optimal orbit in terms of the science and cost trade-off, a mass budget, power and volume estimates, and the less costly compatible launch vehicle. These data are fed to the lifecycle cost model to estimate different parts of the overall lifecycle cost.

1. Orbit Selection using a Fuzzy Rule-based Classifier

Orbit parameters affect both scientific performance and cost. Science changes with orbit through coverage, spatial resolution, swath, temporal resolution, illumination conditions, sensitivity to specific phenomena with local time, signal to noise ratio, and so forth. Cost is affected through varying illumination conditions (e.g., frequency and duration of eclipses), or disturbance torques (e.g. atmospheric drag). Furthermore, different instruments may have different preferred orbits. For example, high energy instruments such as lidars or synthetic aperture radars are typically placed in lower orbits in order to minimize power requirements, while passive optical imaging instruments are placed in higher orbits to maximize coverage. Thus, in general, the optimal orbit for a multi-instrument mission will result from a compromise between all these individual considerations, with some consideration to relative importance of instruments. If the orbit of a spacecraft is not provided by the user, an algorithm is needed to compute the optimal orbit for a given set of instruments. Such algorithm is presented in this section, and it takes the form of a fuzzy rule-based classifier, as opposed to other classification algorithms (e.g. k-nearest neighbors, Bayes classifier, linear classifier, and neural networks).

The set of allowed orbits considered by the model is provided in **Table 3**. Note that only LEO circular orbits are considered. Eccentricity and argument of the perigee are thus irrelevant. Mean anomaly is only used in the definition of constellations with more than one satellite per plane. The algorithm uses a rule-based classifier to assign a set of instruments to an orbit **Table 3**. An example of fuzzy rule would be the following: “IF there is a lidar, then it is very likely that the altitude of the orbit is 400km”. Rules may assert membership or non-membership to an orbit with various degrees of certainty. After all rules have been executed, the orbit with the highest degree of certainty is chosen. Conflicts between rules are thus broken by degrees of certainty assigned to rules and set by experts and/or model training. It is also possible to declare primary and secondary instruments. An orbit by default (viz. a 600km SSO with an AM local time of the descending node) is assigned to cover an extremely unlikely case in which no rules were executed. While this approach cannot guarantee that the optimal orbit is assigned to each instrument set, its performance in practice is good enough for architecting purposes, as shown in **Table 12**. An exhaustive study of different orbit classifiers with a larger dataset is left for future work.

Table 3: The set of orbits considered in the model

Orbit parameter	Allowed values
Altitude	Altitude = {275km, 400km, 600km, 800km, 1300km}
Inclination	Inclination = {SSO, polar, near-polar, near-tropical, equatorial}
Local time of ascending node	LTAN (for SSO) = {Dawn-dusk, AM, Noon, PM}

2. Mass Budget

The masses of the spacecraft and subsystems are important because they are used by the cost model to estimate non-recurring and recurring bus costs. The tool uses a bottom-up approach to estimate the mass of the power subsystem based on payload power requirements, as described in [44]. For the rest of subsystems, low-fidelity models typically use rules of thumb in the form of constant payload-to-subsystem mass ratios α_{subs} to estimate the mass of the subsystem m_{subs} given payload mass m_{payl} [45]:

$$m_{subs} = \alpha_{subs} m_{payl} \tag{21}$$

This approach captures one typical driving requirement in subsystem design, namely payload mass. However, other important requirements (e.g., payload data rate for the communications subsystem), are left out of consideration. Furthermore, a linear model does not capture non-additive interactions between instruments that are often the source of subsystem complexity and cost. The model presented here solves these two limitations by introducing complexity penalties that capture non-linear interactions between instruments. These complexity penalties are Boolean variables computed using logical rules in the rule-based framework. For example, a complexity penalty C_{DR} is defined for cases in which the cumulative instrument data rate $\sum_i R_{b,i}$ exceeds a given threshold $R_{b,THR}$:

$$C_{DR} = \begin{cases} 1, & \text{if} \sum_i R_{b,i} \geq R_{b,THR} \\ 0, & \text{otherwise} \end{cases} \tag{22}$$

In addition to C_{DR}, other complexity penalties are defined to model instrument interactions, namely electromagnetic interference between instruments (C_{EMC}), the presence of instruments requiring cryocooling (C_{CRY}), the presence of large scanning instruments in platforms carrying instruments with stringent attitude control requirements (C_{ADC}), and the need for mechanisms to deploy instruments, solar panels, or other structures (C_{DEP}). While this may not be a complete list of non-linear interactions between instruments, seven interviews with senior system engineers at ESTEC did not identify any additional major source of complexity.

These complexity penalties are used in two ways in the model. First, payload-to-subsystem mass ratios depend on complexity penalties. For example, the data rate complexity penalty C_{DR} is used in the design of the communications subsystem:

$$\alpha_{subs} = \begin{cases} \alpha_{HI}, & C_{DR} = 1 \\ \alpha_{LO}, & C_{DR} = 0 \end{cases} \tag{23}$$

This models the fact that after a certain data rate threshold, regardless of payload mass, a higher performance communications system (higher frequency, with directive antennas) will be required, with an impact on the mass and cost of the subsystem.

Second, complexity penalties may be used to model mass penalties that are independent of payload mass. For instance, the mass of the structure subsystem will incur a fix relative mass penalty if there are active and passive instruments working in the same spectral region (e.g., 5%). This models the extra mass (e.g. long boom) or design effort (e.g., configuration) required to solve this problem.

$$m_{struc} = \alpha_{struc} m_{payl} \cdot (1 + C_{EMC} \cdot \Delta m_{EMC}) \tag{24}$$

Complexity penalties are a simple way of capturing non-linear interaction between instruments in a quantitative way. Their main advantage is that they provide a basis for rigorous comparison between architectures of similar payload mass but different complexities. However, this comes in some cases at the expense of absolute accuracy in the estimation of spacecraft mass. Therefore, this approach should only be used to compare the cost of different architectures in a relative basis, and never as a means of forecasting spacecraft mass. Just for illustrative purposes, the absolute accuracy of the mass budget algorithm with a database of 17 satellites is provided in Figure 10. See [36] (pp 151-159) for a more thorough description of this model.

3. Launch Vehicle Selection

An important fraction of mission lifecycle cost for most Earth observing missions is taken by launch cost. Moreover, differences in instrument-to-spacecraft assignment architectures are sometimes driven by launch cost. Therefore, obtaining an accurate launch vehicle classification is important. The launch vehicle selection algorithm selects the less costly launch vehicle from a database of available launch vehicles that is compatible with the mission requirements (mass, orbit, volume). Several assumptions are used to check whether a launch vehicle is compatible with a spacecraft. First, the launch vehicle needs to have enough performance to put the spacecraft wet mass into the required orbit. Second, the maximum dimension of the spacecraft when folded needs to fit in the fairing height. Finally, the nadir area of the spacecraft when folded needs to fit in the launch vehicle fairing area, computed as fairing height times the diameter. Note that the last two rules require having an idea of the dimensions of the spacecraft when folded. These dimensions are calculated using very rough rules-of-thumb. For instance, the maximum dimension of the spacecraft is driven by the maximum dimension of the instrument, and the nadir area of the spacecraft is directly proportional to the sum of the instrument nadir areas. While these rules-of-thumb can only provide approximations of the real dimensions of the spacecraft, they are sufficient in practice to correctly assign launch vehicle classes (e.g., Atlas 5/Delta IV heavy/Ariane 5 versus Delta II/Soyuz versus Rockot/Vega/Taurus) to most spacecraft. The exact launch vehicle chosen within a class (e.g., Ariane 5 versus Atlas5 or Delta IV) has lower cost impacts, and is typically more driven by availability or political issues. The development of more sophisticated bus configuration models is left for future work.

C. Trade Space Search using a Multi-Level Meta-heuristic Optimization Algorithm

A population-based heuristic search algorithm is used to search the architectural tradespace. The algorithm is shown in Figure 4. First, an initial population is randomly created. Then, the algorithm evaluates the population of architectures, selects a subset of architectures using typical selection operators, and generates a new population by applying a set of operators to the selected architectures, including some elitism and fitness-based selection. These operators are encoded in the form of logical rules and include domain-independent genetic operators such as crossover and mutation [46], as well as domain-specific improvement or repair operators (e.g., improvement operators based on closing data gaps, or capturing synergies between instruments). The algorithm continues iteratively until a set of termination criteria (e.g., convergence in average distance between non-dominated solutions, or maximum number of iterations or simulation time), are met.

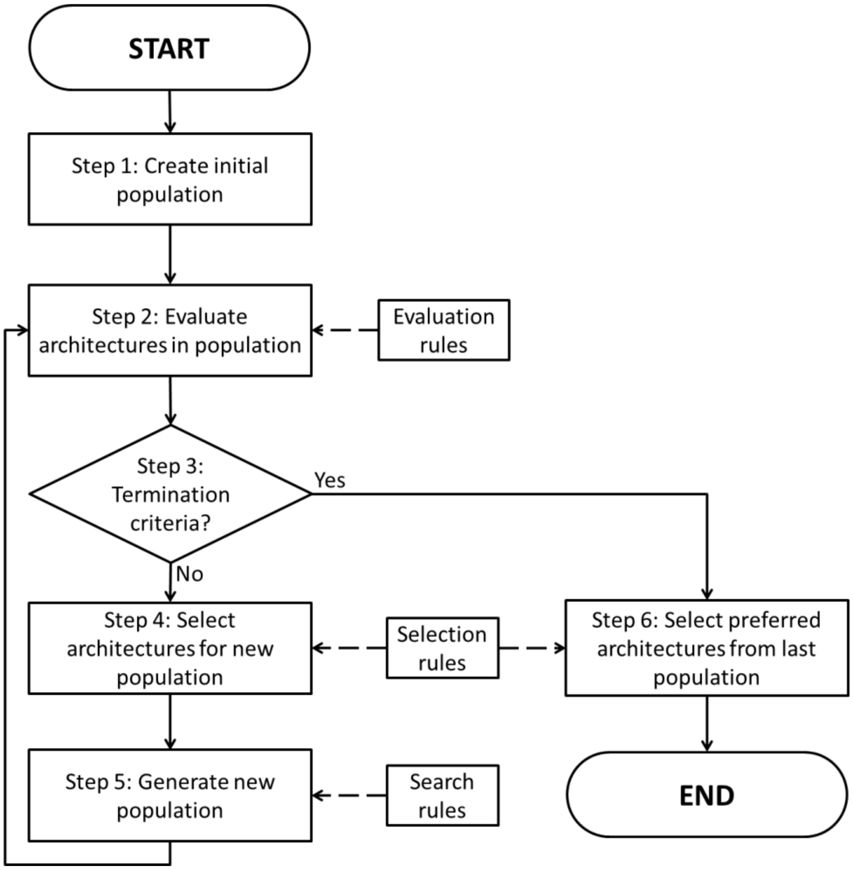
START
Step 1: Create initial population
Step 2: Evaluate architectures in population
Evaluation rules
Step 3: Termination criteria?
Yes
No
Step 4: Select architectures for new population
Selection rules
Step 6: Select preferred architectures from last population
Step 5: Generate new population
Search rules
END

Figure 4: Population-based search algorithm developed for the framework

Like all heuristics search algorithms, this algorithm cannot guarantee that the global optima are found, nor can it prove any bound with respect to the global optima; the goal is rather to find “good” architectures that meet the requirements at reasonable cost and risk. The particularity of this algorithm is that the scalability of rule-based systems is leveraged in the search process, to use a large and variable number of heuristic operators simultaneously. The computational burden of wrapping the operators in search rules is negligible, as the computational complexity of the algorithm is completely dominated by the time required to evaluate architectures (a few seconds).

D. Limitations

A discussion of the limitations is appropriate for models of this complexity. Some of these limitations are related to the modeling of Earth observing systems, while others are inherent to the utilization of rule-based systems. On the modeling side, the most obvious limitation is the uncertainty in the instrument characteristics and scientific requirements. Moreover, the mass budget and cost estimation relationships are mostly based on regression models that are based on relatively few points. This is a problem that is common to most parametric models used in space systems engineering. Some models in the tool use rules-of-thumb instead of empirical data. These rules of thumb are particularly simple for the spacecraft configuration and launch vehicle selection algorithms and they should be improved if the goal were to apply them for specific launch vehicle selection, as opposed to launch vehicle class selection. On the side of rule-based systems for this application, the most salient limitations are related to the knowledge elicitation process. This process can be long and resource-consuming, and in the best case, there might be doubts about the quality of the knowledge base. Ensuring the completeness and consistency of such systems can be difficult. All these limitations require carefully following a verification and validation plan and/or the inclusion of dedicated layers of rules for quality control, with the subsequent penalty in computational performance.

V. Validation with the NASA Earth Observing System

During the 1990s, the EOS was de-scoped from 43 instruments to 25 instruments, and the architecture of EOS went from two extra-large Shuttle-class platforms to three mid-size satellites (Terra, Aqua, Aura) and several smaller satellites. The goal of this “validation case study” is to try to replicate these architectural decisions. While replication of these results does not imply validity of the model, it does provide some level of confidence to apply it to other EOSS.

A. EOS Program Requirements: Aggregation and Requirement Rules

Scientific objectives were identified and ranked in importance for each of seven panels (water and energy cycles (WAE), ocean circulation and productivity (OCE), tropospheric chemistry and greenhouse gases (GHG), land ecosystems and hydrology (ECO), cryospheric systems (ICE), ozone and stratospheric chemistry (OZO), and volcanoes and climate effects of aerosols (SOL)), based on the information available in [47], [48], [49], [50]. This information was contrasted with several experts that were involved in the early development of the EOS program. For example, eight objectives were identified for the WAE panel. They are listed in **Table 6**. These eight objectives were ranked in three groups of importance. Cloud radiative feedback and radiation budget were assigned higher priority (3/16) because of their very close relationship to key climate variables. Ice and snow and land surface water were assigned the lowest priority (1/16) because these objectives are primary objectives of other panels, namely the cryosphere and ecosystem panels. The other objectives were assigned medium priority (2/16). A similar process was followed for the other six panels, yielding a total of 4 objectives. Details about the decomposition of stakeholder needs for the other six panels are provided in [36] (p 384).

Each of the 43 objectives was decomposed into as many measurements requirements as needed, which resulted in 121 requirements. An example of this decomposition is shown in **Table 7** for the radiation budget objective of the WAE panel, which has 4 measurement requirements associated to it. In this case, short-wave and long-wave radiation measurements are assigned a higher priority than the total solar irradiance and albedo measurements (35% versus 15%), because they contain more information. Indeed, total solar irradiance and albedo can be inferred from the SW+LW outgoing flux assuming equilibrium, but the opposite is not true. A similar process was conducted for the other 42 objectives. See [36] (pp 385-391) for the details.

For each of the 121 subobjectives, requirement rules were created that express the measurement requirements for full satisfaction, and several cases of degraded satisfaction **Table 8** shows the full satisfaction rule for subobjective WAE1-1, as well as several partial satisfaction rules. Note that this table only includes a subset of the attributes used in the rule. The actual rules include additional requirements in terms of cloud mask, spectral sampling, and accuracy amongst others. Additional information about the EOS requirements rules is provided in [36] (pp 392).

B. EOS Instrument Characterization: Instrument Capability Rules

Forty-three instruments were considered in this case study. For each instrument, a rule was created that asserts the measurements that the instrument can take. For instance, it was assumed that the TES instrument it is capable -

by itself - of providing measurements of atmospheric ozone, water vapor, methane, carbon monoxide, land surface and atmospheric temperature, and mononitrogen oxides. References [51], [52], [53], [54], [55], [56], [57], [58], [59], [60], [61], [62], [63], [64], [65], [66], [67], [68] describe the characteristics of these instruments that are relevant for the calculation of their capabilities, including mass, power, data rate, dimensions, spectral region, angular resolution, and so forth. More on information on the capability rules for the NASA EOS case study is provided in [36] (pages 201-202, 392).

C. Results

The three problems (instrument selection, instrument-to-spacecraft assignment, and mission scheduling) were explored for the NASA EOS case study. Only results concerning the instrument-to-spacecraft assignment problem are presented in detail in this paper. Results for the instrument selection and mission scheduling cases are briefly summarized at the end of this subsection.

Sixteen instruments were considered for the NASA EOS instrument-to-spacecraft assignment problem, namely those that were ultimately flown on the Terra, Aqua, and Aura spacecraft. The tradespace after 30 generations, including the reference architecture, is shown in Figure 5. One obvious feature of the tradespace is the existence of clusters of architectures that achieve the same scientific scores. These are different ways of capturing the same synergies between instruments. The chart on the left on Figure 6 contains the preliminary results, which assumed that a dedicated bus, designed with the spacecraft design algorithm presented earlier, was built for each spacecraft. These results identified a number of architectures that achieved the same scientific score as the reference architecture at a slightly lower cost, generally using a more distributed solution than the reference architecture - smaller spacecraft were used to leverage favorable relative prices of smaller buses and smaller launch vehicles.

In reality however, only a few commercial buses were truly considered for the NASA EOS program, namely the T-330 (up to 1mt of payload), the BCP2000 (up to 300kg of payload), and the Pegastar (up to 70kg of payload). When the constraint to use one of these three buses was enforced, the shape of the entire tradespace changed as illustrated in the right side of Figure 5. While the average cost of architectures was reduced on average thanks to the use of a commercial bus instead of developing a dedicated bus, the relative cost of architectures changed. Some architectures that were the least costly in the dedicated case became noticeably more expensive because of this restriction, and the best architectures became much more similar to the reference architecture. The best architecture in this case is shown on Figure 6 (right), together with the reference architecture on the left. Small blue boxes on

Figure 6 represent instruments. Spacecraft are marked with the larger black boxes. In addition to several instruments, spacecraft contain more information, namely: the orbit, dry mass of the spacecraft and the launch vehicle on top of the instruments, and information concerning the complexity penalties used in the cost model for the spacecraft below the instruments. In particular, six complexity penalties are shown: the presence of mechanisms (MEC), the presence of instruments with active cryocooling (THE), the presence of high data rate instruments (DAT), the presence of instruments with high attitude control requirements (ADC), the presence of scanning instruments (SCA), and the presence of active and passive instruments using the same frequency band (EMC). These penalties can be either active (red) or inactive (green). Active penalties increase the complexity and the cost of the spacecraft.

The best architecture is thus a 3-satellite architecture, like the reference architecture, and the major difference with respect to the reference concerns MOPITT, which is assigned to the equivalent of the Aura spacecraft instead of the Terra spacecraft. This choice enables capturing potential synergies with other chemistry instruments, and the resulting reduction in the mass of Terra enables the use of a Delta-2 launcher instead of the more expensive Atlas 5. This incorrect assignment is explained because of a hard constraint by the Canadian partners behind MOPITT to fly their instrument on the first spacecraft to be launched, regardless of which one it was. Other minor differences in assignments between the two PM spacecraft are observed, due to the assumption that synergies between spacecraft in the same orbit are fully captured. This assumption makes the two PM spacecraft basically the same from the perspective of science synergies: it is effectively the A-train.

This validation case shows that the model was able to accurately reproduce the logic behind the NASA EOS decision, choosing from millions of possible architectures. This end-state was not an input to the model; it is an emergent output resulting from synergies and interferences between instruments affecting science and cost.

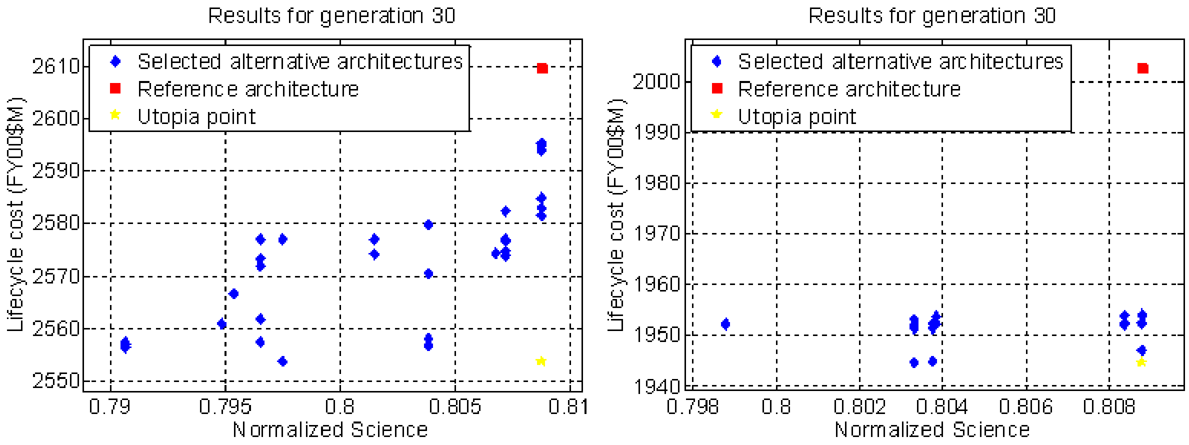

Figure 5: Results for the NASA EOS instrument-to-spacecraft assignment problem, with dedicated buses (left) and commercial buses only (right).

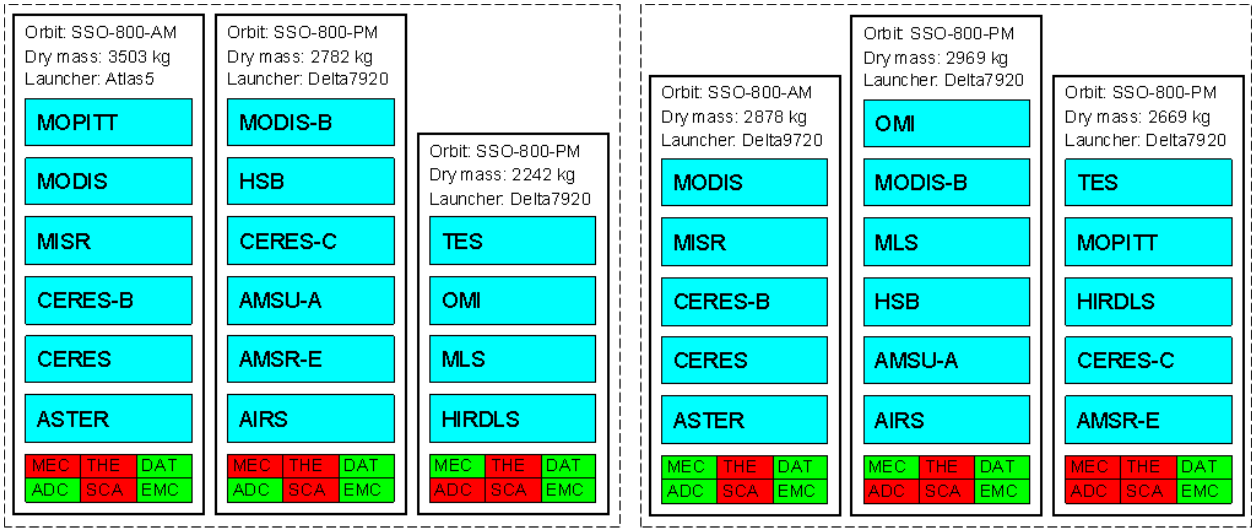

Figure 6: EOS reference architecture (left) versus a good architecture found by tool in the standard bus case (right).

Results concerning the instrument selection and mission scheduling problem are not presented in this paper for the sake of brevity. However, the tool did successfully replicate the EOS instrument selection decisions, with a few caveats that were identified and explained, mostly related to the EOS polarimeter (EOSP), a small, high performance instrument that was unique in its capabilities due to the multiple polarization measurements, which were very valuable to the cloud and aerosol communities, beyond the multi-angular measurements of MISR. Indeed, most non-dominated architectures in the model carried this key instrument which was cancelled in the real program, arguably due to its low TRL. The results for the EOS mission scheduling problem were the least satisfactory. Model assumptions concerning the investment profile over time that have were successful in previous work were deemed inappropriate for the EOS case study, which led to a few clear differences between the reference architecture and the best architectures identified by the model. These differences have their origin in the uncertainty and ambiguity in the cost and budget information available. Overall, the results obtained were deemed satisfactory to proceed with the Decadal case study.

VI. Application to the Earth Science Decadal Survey

A. Decadal Program Requirements: Aggregation and Requirement Rules

Measurement requirements were categorized in six panels for the Decadal case study, according to the Decadal Survey report [1]: climate (CLI), weather (WEA), water (WAT), land and ecosystems (ECO), human health (HEA), and solid Earth (SOL). These six panels were weighted in importance according to the findings of Sutherland's thorough stakeholder analysis [69], which predicts the following relative panel importance: $CLI \approx WEA > WAT > ECO \approx HEA$. Scientific objectives were identified and ranked in importance for each of the six panels, based on the information available in the Decadal Survey report, and leveraging from previous work by Seher [3]. This information was also contrasted with several experts that were involved in the Decadal survey committees. For example, seven objectives were identified for the weather panel, as shown in **Table 10**. These seven objectives were ranked in four groups of importance and assigned scores according to Seher's findings [3]. Atmospheric winds and GPS radio occultation were top priorities of the panel, followed by air pollution and tropospheric ozone measurements. All-weather atmospheric sounding and aerosol-cloud properties are tied third in importance, leaving tropospheric aerosols as the least important objective for the panel. For the details of how these scores were calculated, the reader is referred to [3] or [70]. Details about the first level of decomposition of stakeholder needs for the other six panels are provided in [36] (pp 393-396), yielding a total of 35 objectives.

Each of the 35 objectives was decomposed into several measurement requirements, for a total of 188 requirements. An example of this decomposition is shown in **Table 11** for the atmospheric winds objective of the WEA panel. In this case, the capability to perform direct measurements of 3D atmospheric wind fields is given a higher priority than simpler indirect measurements of winds based for example on water vapor transport.

For each of the 188 measurement requirements, requirement rules were created that express the measurement requirements for full satisfaction, and several cases of degraded satisfaction Details about the 2nd level decomposition of the rest of Decadal objectives are provided in [36] (pp 242, 397).

B. Instrument Characterization

Thirty-nine instruments were considered in this case study. Information about instruments comes from the Decadal Survey report and individual instrument or mission publications (e.g., [71], [72], [73], [74], [75], [76]), and was partially verified by experts at NASA.

C. Results

The three problems (instrument selection, instrument-to-spacecraft assignment, and mission scheduling) were explored for the Decadal Survey case study. Only results concerning the instrument-to-spacecraft assignment problem are presented in this paper. The reader is referred to [36] (pp 244-276) for the detailed results on the two other problems.

For the instrument-to-spacecraft assignment problem, all instruments of the following missions are considered: SMAP, ICESAT-II, CLARREO, and DESDYNI (all Tier I missions), and ASCENDS, HYSPIRI (Tier-II missions). This represents a total of 13 instruments, or 27.6 million architectures ($Bell(13) \approx 2\dot{7}.6 \cdot 10^6$). Other Tier-II missions were left out of the analysis for several reasons, such as the existence of very particular orbit requirements (e.g., GEO-CAPE and PATH are GEO missions and therefore are very unlikely to be combined with the other missions). The results after 30 generations, including the reference architecture, are shown in Figure 7.

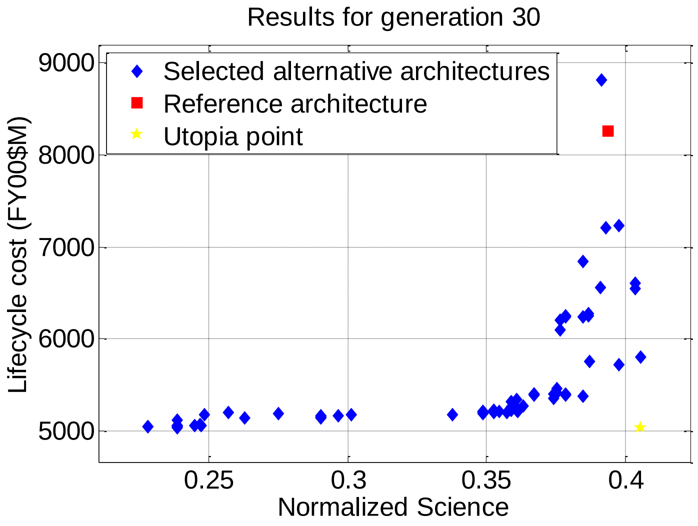

Figure 7: Fuzzy Pareto frontier of Decadal packaging architecture after 30 generations

Figure 7 shows only the architecture on the fuzzy Pareto frontier. There are multiple definitions for a fuzzy Pareto frontier. In this case, we use the formulation in terms of Pareto ranking: architectures on the true Pareto front are assigned a Pareto ranking of 1 and removed. A new Pareto front is computed, and architectures on this second Pareto front are assigned a Pareto ranking of 2 and removed. The procedure is repeated recursively until all architectures have been assigned a Pareto ranking, or a predetermined maximum Pareto ranking has been reached. Under this definition, the fuzzy Pareto front consists of the architectures with Pareto ranking below a certain threshold (which in this case was empirically set to 4).

Three alternative architectures were identified on Figure 7 that achieve a higher scientific score than the reference architecture at a lower cost. These architectures are highlighted on the fuzzy Pareto frontier. Figure 8 provides a pictorial representation of the reference architecture and the three alternative architectures. For each satellite, mass, orbit, lifecycle cost, and launch vehicle class are shown, in addition to the complexity penalties previously defined.

In terms of science, alternative architecture 1 (top left on Figure 8) achieves a higher science score than the reference architecture because by separating the SAR and lidar portions of DESDYNI, the science output of both instruments increases. In the reference architecture, the SAR and the lidar fly together in a compromise dawn-dusk SSO at 600km. In alternative architecture 1, the SAR improves coverage by flying at a higher altitude (800km), and the lidar improves spatial sampling by flying at a lower orbit. The explanation facility shows that the improvement in coverage of the SAR provides benefits in terms of snow cover, hydrocarbon monitoring, and surface deformation data products. Flying the lidar lower has benefits in terms of vegetation data products.

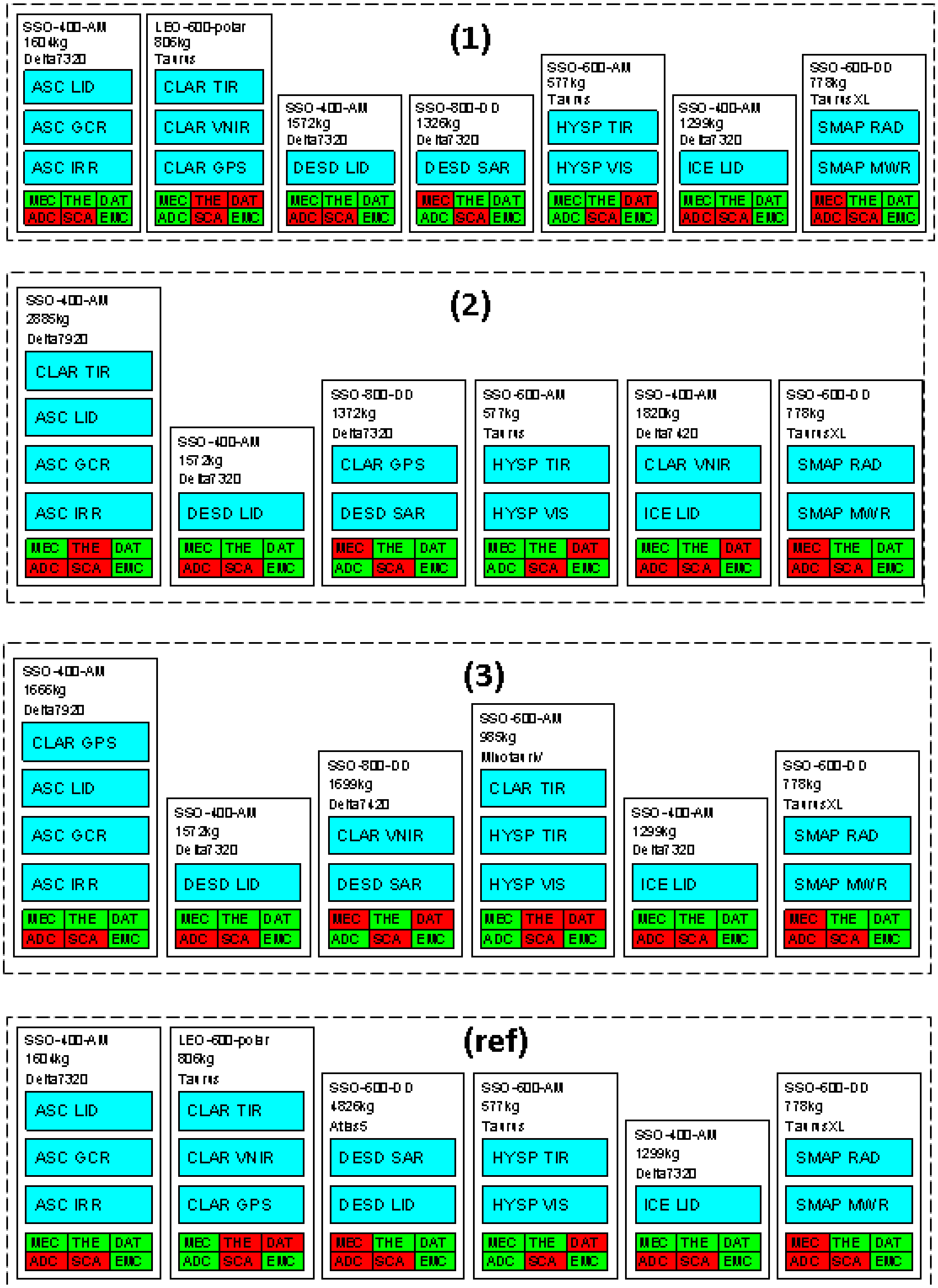

Figure 8: Three Decadal architectures and the reference architecture (bottom)

In terms of cost, differences between the architectures on Figure 8 are driven by the DESDYNI lidar orbit altitude (600km versus 400km). Flying the lidar at 600km is much more costly due to the increase in power requirements to maintain SNR. The need for an Atlas-5 launch is also a driving factor. The reference architecture is the only one using the Atlas-5 launch vehicle, which results in higher launch costs than the alternative architectures by 7.5% ($20-$25M). Another factor concerns the use of large satellites at low orbits. Satellites flying at 400km suffer from high atmospheric drag that may drive the design of the ADCS. The larger and more massive the satellite is, the more stringent the requirements on the ADCS. In other words, the model suggests minimizing the mass that flies at this orbit altitude to the few instruments that have strong reasons to fly there (e.g., lidars), and distributing it in smaller spacecraft with favorable drag coefficients (i.e., smaller areas). Finally, the ability of the current NASA ground stations to downlink the required data volume is identified as a driving issue. The model penalizes architectures that require downlink data rates higher than 500Mbps because that is beyond the current capabilities of the current NASA Space Communication and Navigation Network. This is a strong penalty, as it increases the lifecycle cost of the architecture to include the cost of deploying a new ground asset to meet the demand. While the model may be too pessimistic – these costs could be assigned to multiple missions, and would come from a different part of the budget – this is a very real limitation often driving the architecture of Earth observation missions.

The positions of the three architectures from Figure 9 on the fuzzy cost-risk iso-science Pareto frontier are shown. One can conclude that at this stage, alternative architecture #1 seems to be the most promising one, as it achieves the highest science score in the most efficient way, and it is also non-dominated in the cost-risk space.

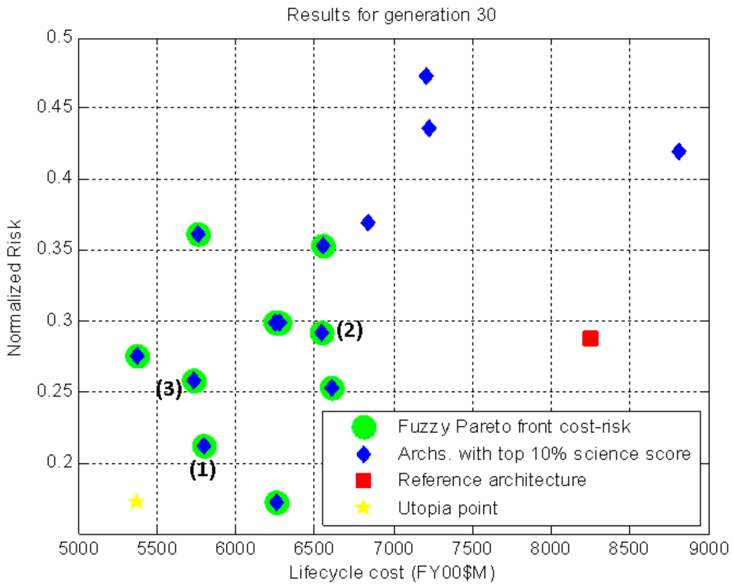

Figure 9: Iso-science, cost-risk space in the Decadal packaging problem

VII. Conclusion

This paper has presented a methodology to explore the architectural trade space of Earth observation satellite systems such as the NASA Earth Observing System or the Earth Science Decadal Survey. The methodology addresses limitations of current architecting tools for tackling real-life architecting problems in terms of the scalability and traceability of the tool when a large body of expert knowledge is required. The methodology is an implementation in the domain of Earth Observing Systems of a generic framework developed by the authors, named VASSAR, which uses a rule-based engine to provide a scalable and traceable framework for tackling knowledge-intensive problems in system architecting.

The problem of architecting EOSS was formulated as a set of coupled combinatorial optimization problems: an instrument selection problem, an instrument-to-spacecraft assignment problem, and a mission scheduling problem. Architectural variables and figures of merit were defined for these problems, and their coupling was discussed.

The framework used to solve these optimization problems was described, focusing on the heuristic search algorithm, the spacecraft design algorithm, and a rule-based scientific benefit evaluator. The heuristic search algorithm uses multiple heuristics (mutation, crossover, improvement) to identify good architectures in large trade spaces using a mix of domain-specific and domain-independent information. The spacecraft design algorithm consists of a heuristic orbit classifier, a complexity-corrected mass budget model, and a simple launch vehicle selection model. The science evaluation component of the methodology consists of three major steps: computing the measurements and data products that the system architecture as a whole can provide; comparing measurements and data products with stakeholder measurement requirements; aggregating requirement satisfaction into stakeholder satisfaction.

The framework was first demonstrated on the NASA Earth Observing System for validation purposes. It was shown how decisions made concerning the allocation of instruments into three large spacecraft can be replicated if the constraints in terms of launch vehicles and buses available are correctly taken into account. While this does not validate the model in the strictest sense, it does provide some confidence about the ability of the model to produce useful results. Validation of complex models such as the one presented in this paper is very challenging, mostly because of the lack of a “truth” dataset with which the model results can be compared. This effort can thus be continued by obtaining a richer dataset on an increasing number of retrospective case studies, including programs from different sizes and organizations. Then, the framework was used to provide insight into architectures of the

Earth Science Decadal Survey that are potentially better than the architecture outlined in the NRC report in terms of scientific and societal value, affordability, and programmatic risk. In particular, it is shown how splitting the DESDYNI mission in its lidar and radar component, and flying the CLARREO instruments as opportunity payloads in other missions have the effect of increasing scientific and societal value while lowering lifecycle cost and programmatic risk.

The tool presented in this paper has the potential to radically improve the performance and lower the cost of large program architecting processes, such as the Decadal Survey NRC review. However, several components of the framework can be improved. More sophisticated heuristics, both domain-specific and domain-independent, can be added to the heuristic search algorithm in order to improve its efficiency. Better spacecraft design algorithms and cost models would increase the fidelity of the lifecycle cost metric. This new knowledge can be incorporated into the tool without changing the rest of the code, which is physically separated from the rule database. The bottleneck in this process continues to be the knowledge elicitation process. Modeling expert knowledge is itself an extremely time-consuming task. However, we argue that modeling expert knowledge is already an integral part of the systems engineering process. Most large space organizations have a knowledge management component, for example in the form of requirement databases, lessons learned databases, or engineering model databases, often managed by a Knowledge Management team. Such roles can proactively manage knowledge in such a way to contribute to future robust architecture decision-making systems. For instance, these databases could be directly used by knowledge-intensive architecting tools like the one presented in this paper. This requires however some effort to migrate these databases into data structures that are easier to interface. Finally, incorporating a machine learning layer on top of the rule-based system could exploit the datasets obtained as the model evaluates architectures to improve the performance of the search process.

VIII. Appendix

Data concerning different components of the framework is presented in this Appendix.

Table 4: Launch vehicles considered in the model, with their assumed performance and characteristics

Launch vehicle	Performance to LEO polar 400km (kg)	Performance to SSO 600km (kg)	Performance to SSO 800km (kg)	Performance to GTO (kg)	diameter(m)	length(m)	cost ($M)
Atlas-5	20000	15000	10000	10000	4.57	7.63	110
Delta-7920	3642	3400	3200	500	2.69	7.53	65
Delta-7420	2269	2123	1989	300	2.69	7.16	55

Delta-7320	1982	1740	1620	250	2.51	6.82	45
Minotaur-IV	1225	1110	1050	0	2	5.44	35
Taurus XL	1015	961.5	870	0	1.98	5.71	30
Taurus	1015	961.5	870	0	1.4	2.67	20
Pegasus XL	300	240	190	0	1.18	2.13	15

Table 5: Scientific panels for the NASA Earth Observing System

Panel	Id	Description	Weight
Clouds and radiation	WAE	Cloud formation, dissipation, and radiative properties, which influence response of the atmosphere to greenhouse forcing	2/13
Oceans	OCE	Exchange of energy, water, and chemicals between the ocean and atmosphere, and between the upper layers of the ocean and deep ocean	2/13
Greenhouse Gases	GHG	Chemistry of the troposphere and lower stratosphere	2/13
Land & Ecosystems	ECO	Land hydrology and ecosystem processes	2/13
Glaciers and Polar Ice Sheets	ICE	Glaciers, Sea Ice, and Ice Sheets	2/13
Ozone and Stratospheric Chemistry	OZO	Ozone and Chemistry of the upper stratosphere	1/13
Solid Earth	SOL	Volcanoes and Climate Effects of Aerosols	2/13

Table 6: 1st level of decomposition of stakeholder needs for the WAE panel in the EOS case study

Objective	Description	Weight
WAE1	Atmospheric circulation	13%
WAE2	Cloud radiative feedback	19%
WAE3	Precipitation patterns	13%
WAE4	Water vapor	13%
WAE5	Aerosols	13%
WAE6	Radiation budget	19%
WAE7	Ice and snow	6%
WAE8	Land Surface Water	6%

Table 7: 2nd level of decomposition of needs for one objective of the WAE panel in the EOS case study

Requirement	Description	Weight
WAE6-1	Total solar irradiance	15%
WAE6-2	Short-wave radiation (solar reflected)	35%
WAE6-3	Long-wave radiation (thermal emission)	35%
WAE6-4	Albedo and reflectance	15%

Table 8: Requirement satisfaction rule for atmospheric temperature fields for the EOS case study

Attribute	**Target**	**Threshold**	**Justification**
Temporal Resolution	Highest(~12h)	High(~24h)	Operational weather
Horizontal Spatial Resolution	Low(10-30 km)	Very-Low(30-50 km)	Grid size of NWP models

Vertical Spatial Resolution	Medium (~1km)	Low (~2km)	Vertical transport
Absolute Accuracy	High (~0.5K)	Medium (~1K)	NWP models

Table 9: Scientific panels for the Decadal case study

Panel	Id	Description	Weight
Weather	WEA	Weather (including space weather and chemical weather)	21%
Climate	CLI	Climate variability and change	21%
Land	ECO	Land-use change, ecosystems dynamics, and biodiversity	21%
Water	WAT	Water resources and the global hydrological cycle	16%
Health	HEA	Human health and security	11%
Solid Earth	SOL	Solid Earth hazards, resources, and dynamics	11%

Table 10: 1st level of decomposition of stakeholder needs for the Decadal case study: WEA panel objectives

Objective	Description	Weight
WEA1	Atmospheric winds	19%
WEA2	High temporal resolution air pollution	15%
WEA3	All-weather temperature and humidity profiles	12%
WEA4	Comprehensive global tropospheric aerosol characterization	8%
WEA5	Radio Occultation	19%
WEA6	Comprehensive global tropospheric O3 measurements	15%
WEA7	Aerosol-cloud discovery	12%

Table 11: 2nd level of decomposition of needs for the Decadal case study: atmospheric winds subobjectives

Requirement	Description	Weight
WEA1-1	Atmospheric wind speed	20%
WEA1-2	Atmospheric wind direction	20%
WEA1-3	Ocean surface wind speed	30%
WEA1-4	Ocean surface wind direction	20%
WEA1-5	Water vapor transport winds	10%

Table 12: Validation of orbit selection rules with a set of real multi-instrument missions

Mission	**Actual orbit class**[**]	**Best orbit class according to classifier**	**Justification for difference**
ACRIMSAT	SSO-600-SSO-AM	LEO-600-polar-NA	Radiation budget wants true polar
AQUA	SSO-800-SSO-PM	SSO-800-SSO-PM	OK
AURA	SSO-800-SSO-PM	SSO-800-SSO-PM	OK
ICESAT	LEO-600-polar-NA	LEO-400-polar-NA	Lidar wants to fly low
JASON-1	LEO-1300-near-polar-NA	LEO-1300-near-polar-NA	OK
SEAWIFS	SSO-800-SSO-AM	SSO-800-SSO-AM	OK
QUIKSCAT	SSO-800-SSO-DD	SSO-800-SSO-DD	OK
SORCE	LEO-600-equat-NA	LEO-800-equat-NA	Passive imager wants to fly high
TERRA	SSO-800-SSO-AM	SSO-800-SSO-AM	OK
LANDSAT-7	SSO-800-SSO-AM	SSO-800-SSO-AM	OK
SMAP	SSO-600-SSO-DD	SSO-800-SSO-DD	Passive imager wants to fly high
ICESAT-2	LEO-400-polar-NA	LEO-400-polar-NA	OK
DESDYNI-LID	SSO-400-SSO-DD	SSO-400-SSO-DD	OK

[**] The true orbit class is the orbit from the set of available orbits that is closest to the true orbit

DESDYNI-SAR	SSO-800-SSO-DD	SSO-800-SSO-DD	OK
ASCENDS	SSO-400-SSO-AM	SSO-400-SSO-AM	OK
ACE	SSO-400-SSO-PM	SSO-400-SSO-DD	DD minimizes cost (more power)
HYSPIRI	SSO-800-SSO-AM	SSO-800-SSO-AM	OK
GRACE	LEO-500-polar-NA	LEO-275-polar-NA	Max sensitivity to gravity
GPSRO	LEO-800-polar-NA	LEO-800-polar-NA	OK
LIST	SSO-400-SSO-DD	SSO-400-SSO-DD	OK
SCLP	LEO-800-polar-NA	LEO-800-polar-NA	OK
XOVWM	SSO-800-SSO-DD	SSO-800-SSO-DD	OK
3DWINDS	SSO-400-SSO-DD	SSO-400-SSO-DD	OK
GACM	SSO-800-SSO-AM	SSO-800-SSO-AM	OK

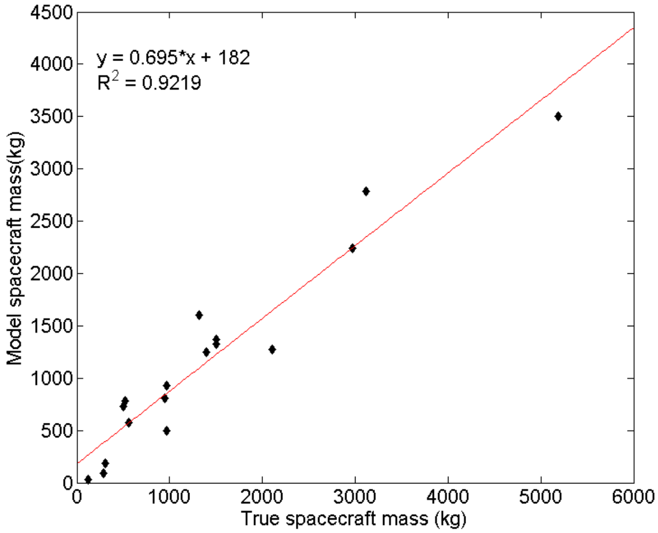

Figure 10: Statistical performance of the mass budget model for illustrative purposes

IX. Acknowledgments

The authors would like to thank several organizations for partially funding this work, namely the “La Caixa” Foundation, Draper Laboratory, and the NASA Goddard Space Flight Center. The authors also acknowledge the contributions of several senior systems engineers in these organizations as well as the European Space Agency and the Jet Propulsion Laboratory during the interviews to gather data for the knowledge-based system.

X. References

[1] National Research Council, “Earth Science and Applications from Space: National Imperatives for the Next Decade and Beyond,” The National Academies Press, Washington DC, 2007.

[2] D. Selva and E. Crawley, “Integrated assessment of packaging architectures in Earth observing programs,” in *Aerospace Conference, 2010 IEEE*, 2010.

[3] T. Seher, “Campaign-level Science Traceability for Earth Observation System Architecting,” MS Thesis, Dept. of Aeronautics and Astronautics, 2009.

[4] M. W. Maier and E. Rechtin, *The Art of Systems Architecting*. New York: CRC press, 2000.

[5] E. Crawley, O. De Weck, S. Eppinger, C. Magee, J. Moses, W. Seering, J. Schindall, D. Wallace, and D. Whitney, “The influence of architecture in engineering systems - Engineering Systems Monograph,” 2004.

[6] H. Mcmanus, D. Hastings, and J. Warmkessel, “New Methods for Rapid Architecture Selection and Conceptual Design,” *Journal of Spacecraft and Rockets*, vol. 41, no. 1, pp. 10–19, 2004.

[7] W. K. Hofstetter, P. D. Wooster, and E. F. Crawley, “Analysis of Human Lunar Outpost Strategies and Architectures,” *Journal of Spacecraft and Rockets*, vol. 46, no. 2, pp. 419–429, Mar. 2009.

[8] A. M. Ross, D. E. Hastings, J. M. Warmkessel, and N. P. Diller, “Multi-attribute Tradespace Exploration as Front End for Effective Space System Design,” *Journal of Spacecraft and Rockets*, vol. 41, no. 1, pp. 20–28, 2004.

[9] H. McManus and J. Warmkessel, “Creating Advanced Architecture for Space Systems: Emergent Lessons from New Processes,” *Journal of spacecraft and rockets*, vol. 41, no. 1, pp. 69–74, 2004.

[10] M. A. Walton and D. Hastings, “Applications of Uncertainty Analysis Applied to Architecture Selection of Satellite Systems,” *Journal of Spacecraft and Rockets*, vol. 41, no. 1, pp. 75–84, Jan. 2004.

[11] B. Donahue and M. Cupples, “Comparative analysis of current NASA human Mars mission architectures,” *Journal of Spacecraft and Rockets*, vol. 38, no. 5, pp. 745–751, 2001.

[12] C. Haskins, “INCOSE Systems engineering handbook - A guide for system life cycle processes and activities,” International Council on Systems Engineering, INCOSE-TP-2003-002-03, 2006.

[13] M. Rao, S. Ramakrishnan, and C. Dagli, “Modeling and Simulation of Net Centric System of Systems Using Systems Modeling Language and Colored Petri-nets : A Demonstration Using the Global Earth Observation System of Systems,” *Systems Engineering*, vol. 11, no. 3, pp. 203–220, 2008.

[14] T. Weilkiens, *Systems engineering with SysML/UML: modeling, analysis, design*. Heidelberg, Germany: The Morgan Kaufmann/OMG Press, 2006.

[15] D. Dori, *Object-Process Methodology: A holistic paradigm*. Berlin, Heidelberg: Springer, 2002.

[16] K. B. Lassen and S. Tjell, “Model-based requirements analysis for reactive systems with UML sequence diagrams and coloured petri nets,” *Innovations in Systems and Software Engineering*, vol. 4, no. 3, pp. 233–240, Aug. 2008.

[17] B. Bairstow, “Multiobjective Optimization of Two-Stage Rockets for Earth-To-Orbit Launch,” in *47th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, 2006, no. May, pp. 1–16.

[18] C. Jilla and D. Miller, “Assessing the performance of a heuristic simulated annealing algorithm for the design of distributed satellite systems,” *Acta Astronautica*, vol. 48, no. 5–12, pp. 529–543, Mar. 2001.

[19] D. Bayley and R. Hartfield, “Design optimization of a space launch vehicle using a genetic algorithm,” *Journal of Spacecraft and Rockets*, vol. 45, no. 4, pp. 733–740, 2008.

[20] P. Biltgen and D. Mavris, “Technique for concept selection using interactive probabilistic multiple attribute decision making,” *Journal of Aerospace Computing, Information, and Communication*, vol. 6, no. January, pp. 51–67, 2009.

[21] National Research Council, “Assessment of mission size trade-offs for NASA’s Earth and Space Science Missions,” The National Academies Press, Washington DC, 2000.

[22] National Research Council, “The Role of Small Satellites in NASA and NOAA Earth Observation Programs,” The National Academies Press, Washington DC, 2000.

[23] M. G. Matossian, “Design for Success - Optimizing the Earth Observing System for Performance and Cost With Managed Risk,” in *Proceedings of the 46th International Astronautical Congress*, 1995.

[24] M. MATOSSIAN, “Earth Observing System constellation design optimization through mixed integer programming,” *Efficient Frontier Astronautics, Inc, The Sixth Alumni Conference of the International Space University*, pp. 158–171, 1997.

[25] M. G. Matossian, “Earth Observing System mission design - Constrained optimization of the EOS constellation configuration design,” in *Proceedings of the 46th International Astronautical Congress*, 1995.

[26] A. Rasmussen, “Cost models for large versus small spacecraft,” *SPIE 3rd Earth Observing Systems Conference*, vol. 3439, no. July, pp. 14–22, 1998.

[27] A. Rasmussen and R. Tsugawa, “Cost-effective applications of constellation architectures of large, medium and small satellites,” AIAA, AIAA Paper 97-3750, 1997.

[28] J. Martin, “Using Architecture Modeling to Assess the Societal Benefits of the Global Earth Observation System of Systems (GEOSS),” *IEEE Systems Journal*, vol. 2, no. 3, pp. 304–311, Apr. 2008.

[29] G. Catastini, S. Cesare, S. de Sanctis, E. Detoma, R. Dumontel, M. Floberghagen, M. Parisch, G. Sechi, and A. Anselmi, “The GOCE end-to-end system simulator,” in *Proceedings of the 2003 EGS - AGU - EUG Joint Assembly*, 2003.

[30] A. Stoffelen, G. J. Marseille, F. Bouttier, D. Vasiljevic, S. de Haan, and C. Cardinali, “ADM-Aeolus Doppler wind lidar Observing System Simulation Experiment,” *Quarterly Journal of the Royal Meteorological Society*, vol. 132, no. 619, pp. 1927–1947, Jul. 2006.

[31] J. Durkin, *Expert Systems: Design and Development*. New York: Macmillan Publishing Company, 1994.

[32] E. A. Feigenbaum, B. G. Buchanan, and J. Lederberg, “On generality and problem solving: a case study using the DENDRAL program,” in in *Machine Intelligence 6*, B. Meltzer and D. Michie, Eds. Edinburgh, Scotland, 1971, pp. 165–190.

[33] M. Minsky, “A framework for representing knowledge,” in in *The Psychology of Computer Vision*, P. H. Whinston, Ed. New York, NY: McGraw-Hill Book, 1975.

[34] C. C. L. Forgy, “Rete: A fast algorithm for the many pattern/many object pattern match problem,” *Artificial intelligence*, vol. 19, no. 3597, pp. 17–37, Sep. 1982.

[35] A. Aliakbargolkar, “A Framework for Space Systems Architecting under Stakeholder Objectives Ambiguity,” PhD dissertation, Massachusetts Institute of Technology, ProQuest/UMI, Ann Arbor, 2012.

[36] D. Selva, “Rule-based system architecting of Earth observation satellite systems,” PhD dissertation, Massachusetts Institute of Technology, ProQuest/UMI, Ann Arbor, 2012.

[37] D. Selva and E. Crawley, “VASSAR: Value Assessment of System Architectures using Rules,” in *Aerospace Conference, 2013 IEEE*, 2013.

[38] H. Apgar, D. Bearden, and R. Wong, “Cost Modeling,” in in *Space Mission Analysis and Design*, 3rd Editio., W. J. Larson and J. R. Wertz, Eds. Hawthorne, CA: Microcosm, 1999, pp. 783–820.

[39] M. Sanchez, D. Selva, B. Cameron, E. Crawley, A. Seas, and B. Seery, “Exploring the Architectural Trade Space of NASA’s Space Communication and Navigation Program,” in *Aerospace Conference, 2013 IEEE*, 2013.

[40] H. H. Agahi, G. Ball, and G. Fox, “NICM Schedule & Cost Rules of Thumb,” in *AIAA Space Conference 2009*, 2009, no. September, pp. 6512–6512.

[41] G. Dubos, J. Saleh, R. Braun, and L. Meshkat, “Technology readiness level, schedule risk, and slippage in spacecraft design,” *Journal of Spacecraft and Rockets*, vol. 45, no. 4, pp. 837–842, 2008.

[42] E. Conrow, “Technology Readiness Levels and Space Program Schedule Change,” *Journal of Spacecraft and Rockets*, vol. 48, no. 6, pp. 1068–1071, 2011.

[43] A. Weigel and D. Hastings, "Measuring the Value of Designing for Uncertain Future Downward Budget Instabilities," *Journal of Spacecraft and Rockets*, vol. 41, no. 1, pp. 111–119, Jan. 2004.

[44] R. S. Bokulic, C. C. DeBoy, S. W. Enger, J. P. Schneider, and J. K. McDermott, "Spacecraft Subsystems IV — Communications and Power," in in *Space Mission Engineering: The new SMAD*, Hawthorne, CA: Microcosm, 2011.

[45] E. I. Reeves, "Spacecraft Design and Sizing," in in *Space Mission Analysis and Design*, 3rd Editio., W. J. Larson and J. R. Wertz, Eds. Hawthorne, CA: Microcosm, 1999, pp. 301–352.

[46] J. H. Holland, *Adaptation in natural and artificial systems*, 2nd Editio. Cambridge, MA: MIT Press, 1992.

[47] M. D. King and R. Greenstone, "1999 EOS Reference Handbook: A Guide to NASA's Earth Science Enterprise and the Earth Observing System," Greenbelt, MD, 1999.

[48] G. Asrar and J. Dozier, *EOS: Science Strategy for the Earth Observing System*. Woodbury, NY: American Institute of Physics, 1994.

[49] D. M. Butler, R. E. Hartle, M. Abbott, S. Ackley, R. Arvidson, R. Chase, C. C. Delwiche, J. Gille, P. Hays, E. Kanemasu, C. Leovy, L. McGoldrick, J. Melack, V. Mohnen, B. Moore III, R. Phillips, A. Rango, G. deQ Robin, V. Suorni, and P. Zinke, "Earth Observing System Science and Mission Requirements Working Group Report," vol. 1, 1984.

[50] A. J. Tuyahov, "The Earth Observing System for the 1990 and 2000 decades," *Science of the Total Environment*, vol. 56, no. 1986, pp. 3–15, 1986.

[51] A. F. H. Goetz and M. Herring, "The high resolution imaging spectrometer (HIRIS) for Eos," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 27, no. 2, pp. 136–144, Mar. 1989.

[52] W. P. Chu, "Calibration for the SAGE III/EOS instruments," in *Proceedings of SPIE*, 1991, vol. 1491, p. 243.

[53] G. Rottman, “Solar UV irradiance measurements: The UARS and EOS SOLSTICE,” *Physics and Chemistry of the Earth, Part C: Solar, Terrestrial & Planetary Science*, vol. 25, no. 5–6, pp. 401–404, 2000.

[54] T. N. Woods, “Overview of the EOS SORCE mission,” *Proceedings of SPIE*, vol. 4135, pp. 192–203, 2000.

[55] D. J. Diner, C. J. Bruegge, J. V. Martonchik, T. P. Ackerman, R. Davies, S. a. W. Gerstl, H. R. Gordon, P. J. Sellers, J. Clark, J. a. Daniels, and others, “MISR: A multiangle imaging spectroradiometer for geophysical and climatological research from EOS,” *Geoscience and Remote Sensing, IEEE Transactions on*, vol. 27, no. 2, pp. 200–214, Mar. 1989.

[56] P. F. Levelt, G. H. J. van den Oord, M. R. Dobber, A. Malkki, H. Visser, J. de Vries, P. Stammes, J. O. V. Lundell, and H. Saari, “The ozone monitoring instrument,” *Geoscience and Remote Sensing, IEEE Transactions on*, vol. 44, no. 5, pp. 1093–1101, 2006.

[57] W.-T. Tsai, M. Spencer, C. Wu, C. Winn, and K. Kellogg, “SeaWinds on QuikSCAT: sensor description and mission overview,” *Proceedings of the 2000 International Geoscience and Remote Sensing Symposium*, vol. 3, pp. 1021–1023, 2000.

[58] R. Willson, “The ACRIMSAT/ACRIM III Experiment-Extending the Precision, Long-term Total Solar Irradiance Climate Database,” *The Earth Observer*, vol. 13, no. 3, pp. 14–17, 2001.

[59] M. P. McCormick, W. P. Chu, J. M. Zawodny, L. E. Mauldin III, and L. R. McMaster, “Stratospheric aerosol and gas experiment III: aerosol and trace gas measurements for the Earth Observing System,” in *Proceedings of SPIE*, 1991, vol. 1491, no. Sage Iii, p. 125.

[60] J. C. Gille, J. J. Barnett, J. G. Whitney, M. A. Dials, D. Woodard, W. P. Rudolf, A. Lambert, and W. Mankin, “The High-Resolution Dynamics Limb Sounder (HIRDLS) experiment on AURA,” in *Proceedings of SPIE*, 2003, vol. 5152, p. 162.

[61] B. A. Wielicki, B. R. Barkstrom, and E. F. Harrison, “Clouds and the Earths radiant energy system (CERES): An Earth observing system experiment,” *Bulletin of the American Meteorological Society*, vol. 77, no. 5, 1996.

[62] F. M. Naderi, M. H. Freilich, and D. G. Long, “Spaceborne Radar Measurement of Wind Velocity Over the Ocean-An Overview of the NSCAT Scatterometer System,” *Proceedings of the IEEE*, vol. 79, no. 6, 1991.

[63] M. Luo, C. P. Rinsland, C. D. Rodgers, J. a. Logan, H. Worden, S. Kulawik, A. Eldering, A. Goldman, M. W. Shephard, M. Gunson, and M. Lampel, “Comparison of carbon monoxide measurements by TES and MOPITT: Influence of a priori data and instrument characteristics on nadir atmospheric species retrievals,” *Journal of Geophysical Research*, vol. 112, no. D9, pp. 1–13, May 2007.

[64] B. H. Lambrigtsen and R. V. Calheiros, “The humidity sounder for Brazil-An international partnership,” *Geoscience and Remote Sensing, IEEE Transactions on*, vol. 41, no. 2, pp. 352–361, Feb. 2003.

[65] J. B. Abshire, J. C. Smith, and B. E. Schutz, “The Geoscience Laser Altimeter System (GLAS),” *AIP Conference Proceedings*, vol. 420, no. 1, pp. 33–37, 1998.

[66] Y. Yamaguchi, A. B. Kahle, H. Tsu, T. Kawakami, and M. Pniel, “Overview of Advanced Spaceborne Thermal Emission and Reflection Radiometer (ASTER),” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 36, no. 4, pp. 1062–1071, Jul. 1998.

[67] H. H. Aumann and C. R. Miller, “Atmospheric infrared sounder (AIRS) on the Earth Observing System,” in *Proceedings of SPIE*, 1995, vol. 2583, p. 332.

[68] G. Bianchini, U. Cortesi, L. Palchetti, and E. Pascale, “SAFIRE-A (spectroscopy of the atmosphere by far-infrared emission-airborne): optimized instrument configuration and new assessment of improved performance,” *Applied optics*, vol. 43, no. 14, pp. 2962–77, May 2004.

[69] T. A. Sutherland, “Stakeholder Value Network Analysis for Space-Based Earth Observations,” MS Thesis, Dept. of Aeronautics and Astronautics, 2009.

[70] B. Suarez, “Integrating Spacecraft and Aircraft in Earth Observation System Architectures,” MS Thesis, Dept. of Aeronautics and Astronautics, 2011.

[71] B. Lambrigtsen, S. Brown, T. Gaier, P. Kangaslahti, and A. Tanner, “A Baseline for the Decadal-Survey PATH Mission,” in *Proceedings of the 2008 IEEE International Geoscience and Remote Sensing Symposium*, 2008, pp. III – 338–III – 341.

[72] T. Gaier, B. Lambrigtsen, B. Lim, P. Kangaslahti, A. Tanner, I. O. Dwyer, C. Ruf, M. Flynn, B. Block, and R. Miller, “GeoSTAR-II Technology Risk Reduction for the NASA Decadal Survey PATH Mission,” in *Proceedings of the 2010 NASA Earth Science Technology Conference*, 2010.

[73] NASA Earth Science and Technology Office, “Aerosol-Cloud-Ecosystems (ACE) Decadal Survey Mission,” in *2009 NASA Earth Science Technology Conference*, 2009, no. August.

[74] Jet Propulsion Laboratory, “Soil Moisture Active/Passive (SMAP) Mission NASA Workshop Report,” 2007.

[75] A. Donnellan, P. Rosen, and J. Graf, “Deformation, ecosystem structure, and dynamics of ice (DESDynI),” in *Aerospace Conference, 2008 IEEE*, 2008, no. 3.

[76] D. W. Cline, R. E. Davis, and S. H. Yueh, “Cold-land Processes Pathfinder Mission Concept,” 2005.