3rd European Systems Engineering Conference
Systems Engineering: a focus of European expertise
Pierre Baudis Congress Centre,TOULOUSE, May 21st - 24th, 2002

The Uses of Argument in the Systems Engineering Process: An Application to Automotive Systems Engineering

Patrice Micouin
MICOUIN Consulting for Innovative Systems Engineering
patrice.micouin@ieee.org
www.micouin-consulting.com

Emmanuel Combes
PSA Peugeot Citroën - DRIA
Route de Gizy
F-78943 Velizy-Villacoublay
emmanuel.combes@mpsa.com

Jean Paul Kieffer
Ecole Nationale Supérieure des Arts & Métiers
2, cours des Arts et Métiers
F-13617 Aix-en-Provence Cedex 1
jpk@aix.ensam.fr

Abstract. This article centers around an interest in decision-making processes and the uses of argument, to make decisions in the area of systems engineering.

With that in mind, the paper has five related goals.

Firstly: introduce the engineering processes and knowledge and skills required in order to show the deliberative and bargaining essence of engineering and to justify resorting to argumentation.

Secondly: produce the some results of argumentation theory.

Thirdly: show how well the argumentation process could efficiently drive some engineering processes such as the requirement definition process or the Solution definition process.

Fourthly: show how well the argumentation process could efficiently drive the collaborative working of a development team, during and outside of technical meetings and reviews.

Fifthly: illustrate this explicit argumentation process through examples coming from the automotive domain: the hybrid electric powertrain planned to be embedded inside several carmaker vehicles (HEV).

PROCESSES FOR ENGINEERING A SYSTEM

Systems and end products: According to ANSI/EIA 632, a system consists of both **end products** to be used by an acquirer for an intended purpose, and **enabling products** that enable the creation, realization and use of an end product or an aggregation of end products.

End products are concrete artifacts (physical items or software items) designed to meet stakeholders needs and expectations.

An example, Hybrid Electric Vehicles (HEVs): HEVs combine two or more energy conversion technologies (e.g. heat engines, fuels cells, generators, or motors) with one or more storage technologies (e.g. fuel, batteries, ultracapacitors, or flywheels). The combination of conventional and electric propulsion systems offers the possibility of reducing emissions and fuel consumption.

An obvious way to combine conventional (engine) and electric propulsion systems is to link in series an engine, an alternator, a motor and a transmission. This particular configuration of conventional and electric propulsion systems is called a series hybrid electric powertrain.

In the ANSI/EIA wording, a series hybrid electric powertrain forms an end product.

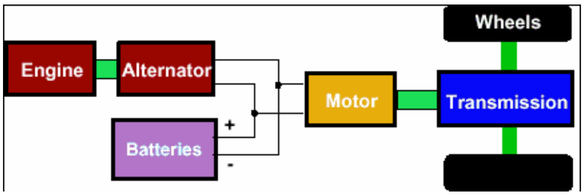

Figure 1: Physical Breakdown of a series hybrid electric drivetrain.

This end product has no use outside a larger end product: here, a series hybrid electric vehicle (series HEV).

The former end product (series hybrid electric powertrain) consists of several subsystems including a heat engine, an alternator, an electrical motor, a transmission subsystem and batteries, cooling system, electrical network, digital control system.

Figure 2: Citroën Saxo series HEV.

The end product reifies a set of requirements and a set of design choices made by the enterprise and the design team that means the end product is the way it is. The observation of the concrete artifact provides some direct information about the aim of the end product (here, supply mechanical power to the driving wheels, reduce fuel consumption, reduce CO_2 emissions), its requirements, the way it operates. But it is more difficult to read why the artifact is the way it is and it is impossible to predict how it meets stakeholder expectations.

Enabling products: Enabling products are used to perform the associated process functions of the system:
a) Develop, produce, test, deploy and support the end product;
b) Train operators and maintenance staff of the end product;
c) Retire or dispose of end products.

Enabling products regularly include (a) the **Requirements Baseline** that sets the stakeholders needs and expectations to be filled by the end products, (b) the Specification Tree and the System Breakdown Structure for implementing the end products and blueprints for building them.

Thus, a part of enabling products reflects the requirements taken into account by the design team and decisions made by the design team to fill these requirements.

Another part of enabling products should contain the designer's thinking and reasoning behind these decisions, that is, the arguments for why the requirements and the end product are the way they are. This part of enabling products, named Design Justification File (DJF) in the ECSS-E-10A standard, is pointed out as rationale, hereafter.

Rationale: The goal of a rationale is to capture the designer's thinking and reasoning behind the decisions of what products are to be like.

Developer's thinking and reasoning is important because a product needs to be understood by a wide variety of people who have to deal with it (designers, users, maintainers.) and these rationales are useful tools in the engineering process from reasoning and reviewing to managing, documenting and communicating.

Processes for engineering of a system: According to the IEEE 1220 standard, the systems engineering process is described as a defined set of sub-processes specified in its clause 6.

These sub-processes are named as follows: Requirements Analysis and Requirement Verification (clauses 6.1 and 6.2); Functional Analysis and Functional Verification (clauses 6.3 and 6.4); Synthesis and Design Verification (clauses 6.5 and 6.6); System Analysis (clause 6.7) and Control (clause 6.8).

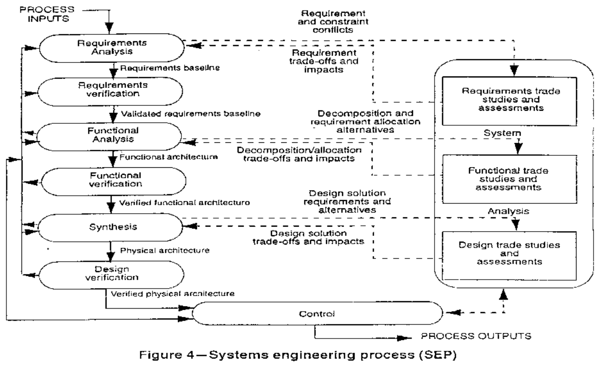

Figure 3: The IEEE 1220 Systems Engineering Process.

Requirement Analysis and Requirements Verification: The goal of Requirements Analysis is to establish what the system shall be capable of accomplishing in the environments in which it operates and the constraints that affect the design. Requirements are to be documented in a **requirement baseline**. The goal of Requirements Verification is to ensure that the baseline represents stakeholder expectations as to whether possible system operation and life cycle support concepts has been addressed.

Requirement Analysis and their verification may raise voids in needs, inconsistencies, needs not properly addressed, but also **requirement conflicts**. In this case, **trade-off analyses** are completed in order to resolve conflicts and finalize a **balanced** requirement baseline. These trade-off analyses are in the scope of Systems Analysis process (clause 6.7)

Functional Analysis and Functional Verification: The goal of functional analysis is to define a set of **alternative** functional decompositions that fulfill system requirements and to select **the best** functional architecture among the alternatives studied. This selection requires trade-off analysis and risk analysis in order to define a balanced set of sub-functions and to assign performance requirements to these sub-functions. These performance allocations may raise trade-off and arbitration issues.

The goal of functional verification is to detect voids or conflicts in order to resolve them through the systems analysis process.

Synthesis and Design Verification: The goal of synthesis is to transform the selected functional architecture in a physical architecture. This process includes the selection of a **preferred solution** among a set of **alternatives** with respect to performances, dependability, schedule, costs and risks criteria. The design verification goal is to detect variance and conflicts in order to fill and resolve them through the systems analysis process.

System Analysis: The goal of system analysis is to resolve all **conflicts** raised by the other processes, to assess the efficiency of functional or physical **alternatives** taken into account, to select **the best** solution, to assess the system efficiency and to manage risk factors throughout the engineering effort.

Required Knowledge and skills: According to the ANSI/EIA-632 standard, the engineering of a system and its related products is accomplished by applying a set of processes to each element of the system hierarchy by a multi disciplinary team of people who have the requisite knowledge and skills.

These requisite knowledge and skills include (a) a knowledge of the domain (for instance, automotive engineering is a domain), (b) technical and scientific knowledge (such as mechanics, thermodynamics, electronics, controls), (c) methodological knowledge and skills to process the engineering of systems such as requirement analysis, functional analysis, economic analysis, safety and dependability analysis, and (d) a capability to make the right decisions and the right choices with respect to a situation.

Two paradigms: As quoted from I Reymen, two dominant scientific belief systems are distinguished in the philosophy of science: Positivism and Constructivism. Two paradigms for describing engineering are based on these two systems: technical rationality or reflective practice.

	Technical Rationality	Reflective Practice
Engineer	Information processor in an objective reality	Person constructing his or her reality
Process	Rational search process	A reflective conversation
Knowledge	Knowledge of engineering procedures and scientific laws	Artistry of engineering: when to apply which procedure or piece of knowledge
Model	Natural sciences	Social sciences

Table 4: Two paradigms for describing engineering (based on Reymen)

Technical Rationality: Technical Rationality is the traditional conception of knowledge of engineers and professionals. According to this model of knowledge, engineer activity consists of instrumental problem solving based on rigorous application of scientific theories and techniques. Here, the usual operating mode requires (a) a mathematical expression of the problem to be solved, applying general physical or chemical laws (differential equations) (b) from this, to deduce problem solutions, thanks to a calculating process.

For instance, the technical rationality program is filled when the engineer is able to express a problem in the shape of an equation defining the analytical transformation of inputs of a processor in its outputs, as following.

$$C_{roue} = (M_{vhl} * \gamma_{vhl} * R_{roue}) + (\frac{1}{2} * \rho * c_d * A * v_{vhl}^2 * R_{roue}) + (f_{roll} * M_{vhl} * g * R_{roue})$$

Figure 5: Mathematical expression of the torque provided by a powertrain to the driving wheels.

Simulations, used to predict the dynamic behavior of systems, are completely based on this turn of mind. Simulators use mechanical, thermal, electrical equations and measured component performance to model existing or conceptual systems

So, the below figure shows a dynamic behavior simulation MMI of a series hybrid electric vehicle (a series HEV).

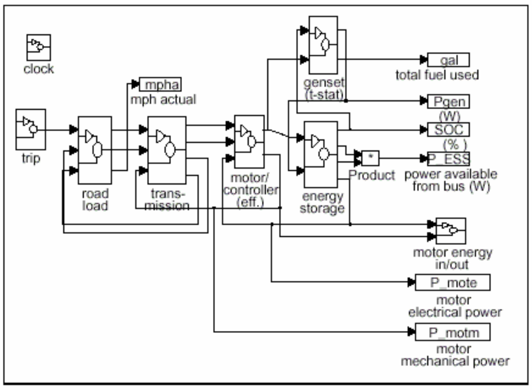

Figure 6: Matlab/Simulink series HEV Simulation.

Obviously, technical rationality is not questioned in this article insofar as no artifact works if offending physical laws reflected by technical rationality. Moreover, technical rationality provides a wide range of rational arguments in engineering decision-making processes.

Again, several extensions of this paradigm have been studied to help decision making, such as, Decision Theory due to Von Neuman, the Constraint-Satisfaction Problem or Multi-Criteria Decision-Making. However, in the everyday engineer practice, there is a gap between the promised potential of these “hard” scientific methods and their actual performance or use in the operational projects.

Thus, we were engaged in researching a way to make decisions, as reasonable as possible, explicitly justified with facts and reasoning, in the time and the cost limits consistent with the system development. These justifications should be approved and borne by decision-makers and examined, agreed with or challenged by anyone.

Knowing-in-action: In "*The Reflective Practitioner*", Donald Schön defined knowing-in-action as a very common way to make efficient decisions when the smooth flow of action is not interrupted by surprise. In this case, the knowing is built in the action and revealed by the performance of the action. We do not have to think about it prior to or during its performance and we are usually unable to describe the knowing which our action reveals. A lot of good practices of competent practitioners are Knowing-in-action practices. This kind of decision based on knowing-in-action has no other rationale than "the experience shows that it is the good decision".

Reflection-in-action: Donald Schön described reflective practice as an alternative to the technical rationality and defined it as a "reflective **conversation** with the materials of a situation". Previously, Horst Rittel viewed design as a process of **negotiating** and **deliberation**, fundamentally dealing with **uncertainty** and **conflict**, and recommended emphasizing investigations into the understanding of designing as an argumentative process.

What are the materials of a design situation? It could include uncertainty about needs and expectations of stakeholders. For instance, during the requirement definition process (in 2001) of a new product offered into a mass market (from 2005), a manufacturer tries to define as accurately as possible, customers expectations, enterprise and project constraints and external constraints. Nevertheless, nothing guarantees that the product will gain the customer's approval, at the same level as the manufacturer expectations. The manufacturer has only a presumption, confirmed or not in actual facts, later. Obviously, who is able today to predict with assurance what will be as of 2005, the political, economical, social and ecological environment of potential users? What will be the customer expectations in that context? What will be the state of regulations? etc. In other words, the manufacturer makes a bet about the future, essentially uncertain and risky.

It could also include conflicts between schedule and workload, costs and market prices, performances and costs. Requirement conflicts need arbitration or compromise among several solutions, some of them « better » or « worse » depending on one's standpoint. For instance, during the design and implementation of the new product, the manufacturer will be confronted with several architectural or technological alternatives. One solution may be better in face of a certain subset of requirements (performances, dependability), than an other solution alternative, but worse compared with another subset of requirements (costs, just on time).

Nevertheless, an unambiguous choice among solutions under consideration is required depending on guessed expectations of customers, unless a way of compromise is to search a new option.

Again, the manufacturer takes out an option on the future, which is without guarantee just at that moment.

According to the Chaim Perelman and Lucie Olbrechts-Tyteca standpoint: "The domain of argumentation is that of the credible, the plausible, the probable, to the degree that the latter eludes the certainty of calculations" (*La Nouvelle Rhétorique: Traité de l'Argumentation*, 1958).

ARGUMENTATION THEORY

The argumentation theory such as it has been expressed by S. Toulmin in his main work: *The uses of Argument*, then expanded upon by authors such as S. Newman and made use of by other authors such as H. Rittel (IBIS), J. Lee & K-Y. Lai (DRL), A. Mac Lean (QOC) and R McCall (PHI) is presented as an illustration of our second aim.

This presentation is carried out in order to distinguish three levels in a practical decision process: firstly, the argumentation schema; secondly, the decision structure; thirdly, the decision tree.

Argumentation schema: According to Stephen Toulmin, the simplest model of an argumentation has the following aspect:

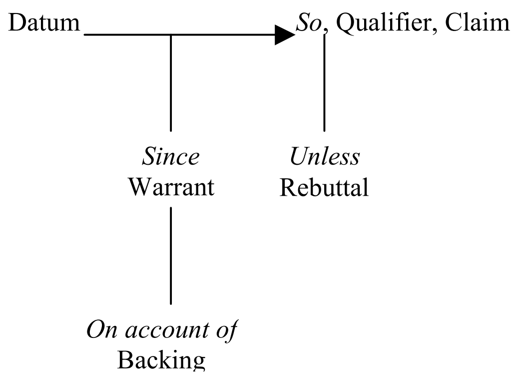

Figure 7: Toulmin's micro-argument structure.

and includes no more than six elements:

The claim: The first step of an argumentation is the expression of an assertion (claim) by an arguer. The justification of this claim is made by the arguer.

For instance, under highway or mixed driving cycle conditions, series hybrid electric vehicles do not provide fuel economy gains in comparison with conventional gasoline or diesel fueled vehicle.

A Datum: A datum is a fact that enforces the confidence about the claim and that the arguer has to produce, if required.

For instance about the previous claim, a simulation based on the Matlab/Simulink model produced above shows no fuel economy

The Warrant: The warrant is an inference rule that allows the move from the datum to the claim.

For instance about the previous claim, the confidence granted to the Matlab/Simulink simulation constitutes a warrant of the inference.

A Backing: A backing may be put forward if the warrant is not directly agreed and requires reinforcement.

For instance about the previous claim, the arguer could remark that this confidence was established by means of a survey such as the Randall Donn Senger Thesis.

A Rebuttal: A rebuttal states exceptions, that question, datum, warrant or backing.

For instance about the previous claim, the challenger could remark that the fact and the warrant referred to by the defender model, do not take into account that it is possible to use grid energy (charging at standstill) and this later option (grid dependent HEV) allows fuel consumption gains.

The Qualifier: The optional qualifier determines the level of confidence assigned to the claim and ranges from possibly to certainly.

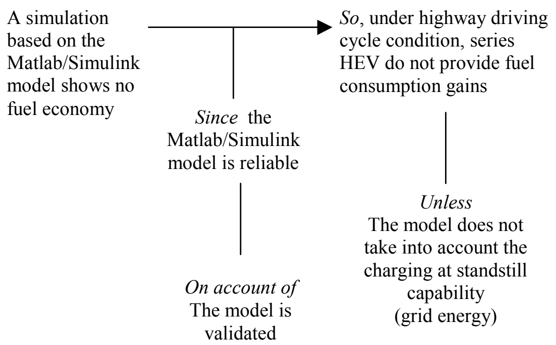

Figure 8: Toulmin-like representation of argumentation.

Complex Argumentation Schemas: As suggested by S. Newman, this basic structure of argumentation may be combined, allowing more complicated arguments. As an illustration, we show a claim supported by two independent argumentative lines.

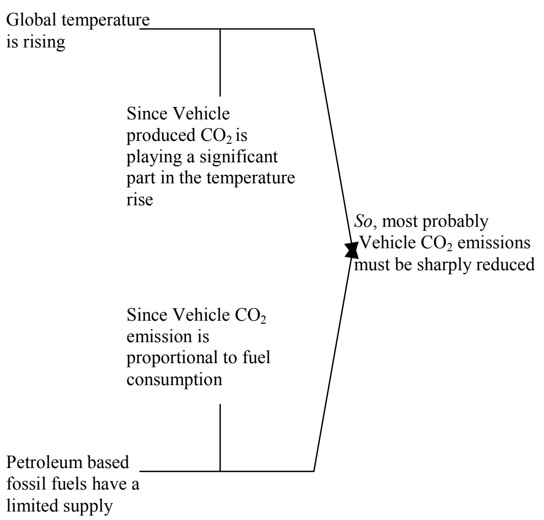

Figure 9 : Convergence of arguments.

Decision Structure A basic decision structure includes the following elements: an issue, options, evaluation criteria, an option assessment facing each criterion based on an argumentation, an overall weighing of arguments and a decision.

Issue: The first element is a relevant issue or a controversial question raised during an engineering process.

An example, parallel HEVs: a parallel HEV is configured with two power paths, so that either the engine or the electrical propulsion system – or both – can be used to produce the motive power to turn the wheels.

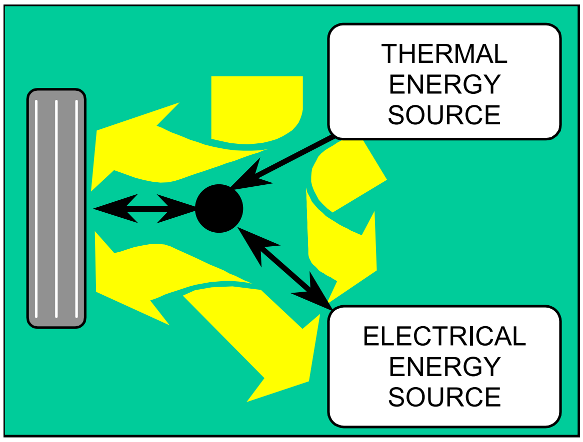

Figure 10: parallel hybrid electric drivetrain concept.

In the context of a parallel HEV design, a relevant issue is: "Where are the two power paths added?"

Options: The second set of entities is the set of options or alternatives, which are then the conceivable solutions to the specified issue.

For instance, about the previous issue, two options are conceivable.

The first option, called simple shaft, consists in adding the torque of thermal origin and the torque of electrical origin on the primary shaft of the transmission gear. The following figure shows this first option

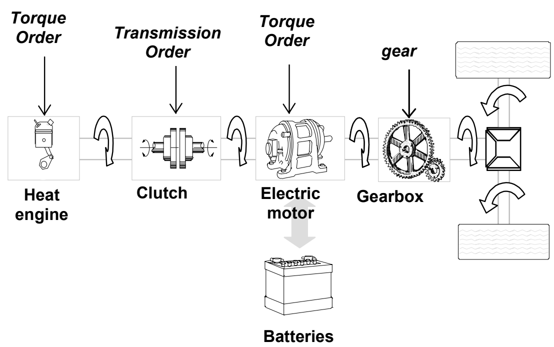

Figure 11: Simple Shaft Option.

The second option, called double shaft, consists in adding the torque of thermal origin and the torque of electrical origin on the output shaft of the transmission gear. The following figure shows this second option:

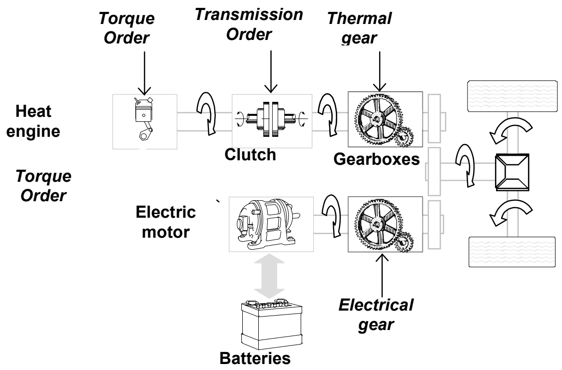

Figure 12: Double Shaft Option.

Criteria: The third set of entities is the set of criteria, i.e. the relevant standpoints from which it is possible to argue for or against each option.

For instance, about the previous options, one criterion may be risks about development costs and deadlines. Another criteria may be the functional efficiency, the acquisition cost, the dependability, the geometrical dimensions of each option,..

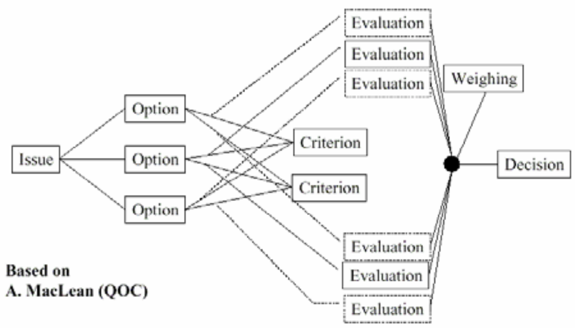

Figure 13: Decision structure.

Evaluation: The fourth set of entities is the set of evaluations. An evaluation is a claim about an option facing a criterion. This evaluation is shaped as a Toulmin's argument and may refer to data, warrants, backing and rebuttal. This evaluation is context dependant and may be qualitative (such as high, medium and low about a risk) or quantitative (an extra-cost, for example).

For instance, about the previous options, someone could argue that Simple shaft option involves a medium level of risk (first claim) while Double shaft option involve a lot of risks (second claim). Datum put forward about this second claim may be: enterprise does not hold double shaft gearbox on the shelf.

Weighing: Weighing arguments is also a claim about overall evaluations. This claim is also shaped as a Toulmin's argument and may refer to data, warrants, backing and rebuttal (including qualitative appreciation or computations). It is context dependent and is intended to express a preference about discussed options.

For instance, about the previous options, someone could express that the Simple shaft option is the preferred solution and argue about it (weak functional advantages of double shaft option and costs, deadlines, technological risks, dimensions disadvantages).

Decision: The last entity is the decision which makes arrangements to act.

For instance, about the previous options, Simple shaft option is selected and investigations about double shaft option are thrown out (in contrast, decision may be to pursue investigations about the two options).

Decision Tree A complex decision-making process including a lot of interdependent decisions such as encountered throughout systems engineering processes. According to R.J. McCall, it is possible to represent this interrelated network of decisions as a Procedural Hierarchy of Issues or Decision Tree.

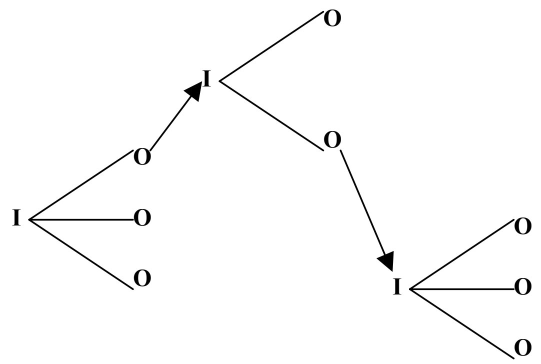

Figure 14: Decision Tree.

For instance, about the electric propulsion component of the parallel HEV powertrain, someone could shows the following decision tree:

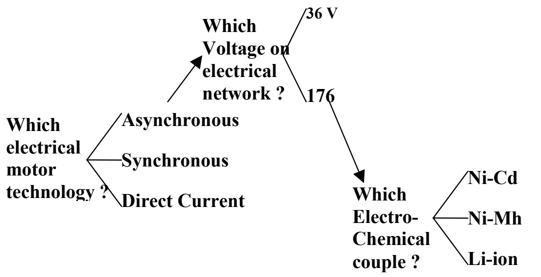

Figure 15: Decision Tree.

Firstly, note that the decision tree is process-oriented in the sense that it records chronologically, questions issued during work in progress. In contrast, decision schema is structure-oriented in the sense that it is concerned with the structure of the space of all alternatives, which may be constructed.

Secondly, for every step of its growth, the decision tree provides then a snapshot of the design situation. The emergence of a new issue may be tacitly handled (knowing-in-action) or may trigger an explicit decision-making process (reflection-in-action) involving options identification, criteria definition, etc..

APPLICATION TO SYSTEMS ENGINEERING

Argumentation skills may be regarded as mainly useful in some engineering processes such as acquisition and supply processes or technical management processes. Nevertheless, our subject is to show its usefulness in technical processes such as requirement definition, solution definition, (verification and validation processes are not addressed).

Balanced set of requirements and the uses of argument: The establishment and evolution of a complete and consistent set of requirements that enables delivery of feasible and cost effective system solutions is the resultant of three force fields.

Firstly, the stakeholder expectations captured through the Requirements Definition Process.

Secondly, the disclosure of inconsistencies and voids concealed in the expectation formulations through the Requirement Validation Process.

Thirdly, the feasibility of solutions provided by technologies and the market identified through the Solution Definition Process.

These force fields are antagonistic and disclose uncertainties and conflicts. Uncertainties require risk recognition and risk lowering actions. Conflicts require arbitration and compromises. These activities which are in the scope of the Systems Analysis Process appear as Schön said: a "reflective **conversation** with the materials of a situation".

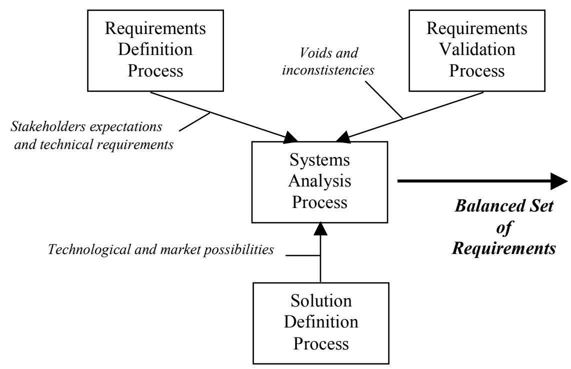

Figure 16: Balanced requirements as result of a deliberation process.

For instance, about HEV, since its intended goal is to reduce emissions and fuel consumption, an interesting capability could be to run as a Zero Emission Vehicle (ZEV), if the engine is switched off. This ZEV mode of operation could notably be interesting in city areas and provide a strong environmental benefit. So, someone could expect ZEV mode of operation as a strong requirement and define the following performance related to ZEV mode: speed range: [0, 50 km/h], autonomy: 25 km.

Unfortunately, this requirement has a negative impact on batteries system mass, dimensions, dependability and cost (feasibility feedback from solution definition process) and conflicts with other requirements such as target price, availability, trunk roominess.

This conflict opens the way to a deliberation about the hybridization rate for parallel HEVs.

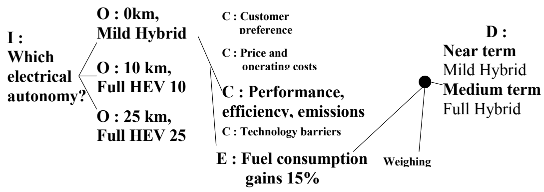

Figure 17: Requirement arbitration.

Requirements baseline rationale and argumentation: The requirement baseline is the result of a complex process where hypothesis, practical reasoning, trade-off as reasonable as possible are done to limit uncertainties, to arbitrate conflicts, and solve inconsistencies and to fill voids.

So, the requirement baseline appears as an abstract artifact that represents the engineers' decisions of what the system is to be like.

Hypothesis, practical reasoning, and trade-off are facts, warrants and backing behind decisions to balance conflicting requirements. These Hypothesis, practical reasoning, and trade-off should be documented in a Requirement baseline rationale in order to capitalize organization knowledge and facilitate future evolutions of the baseline.

Indeed baseline evolutions involve new arbitration that requires remembering engineer's thinking and reasoning behind previous compromises.

For instance, an HEV requirement baseline rationale should capture hypothesis, reasoning that make the justification of what the requirements baseline is to be like

Figure 18: Citroën Xsara full hybrid.

Balanced design solution, design rationale and argumentation: It is possible to argue about the suitability of argumentation-based design and design rationale in the same way as balancing or justifying requirements. However, we prefer to devote our attention to different kind of issues, criteria and arguments encountered during Solution Definition Process.

Design issues: Upon our practice, we encountered several types of issues that we classify in the following way:

Architecture issues deal with components (subsystem, set, equipment, assembly, part) and their linkages. For instance, "how to connect the heat engine part and the electrical motor part to the transmission subsystem of a parallel hybrid electric powertrain?" is an architectural issue.

Assignation issues deal with how to allocate functions or performances to physical component or how to split them. For instance, about a gearbox, "how to allocate the shafts synchronization function during a gear change?" is an assignation issue.

Sizing issues deal with how to size functional performances or physical characteristics of a component. For instance, about a gearbox, "how many gears are required?" is a sizing issue.

Technological issues deal with the technological base used to implement a functional set. For instance, about a gearbox, "What kind of gearbox choice? (mechanical or automatic)" is a technological issue.

Control issues deal with how to coordinate the various subsystem operations in order for the whole to function correctly. For instance, about a gearbox, in the hybrid context, "when is a gear change required?" and "How to perform this change efficiently?" are control issues.

Design criteria: As an indication, design criteria include:

Functional criteria: Includes the ability of the option to (a) perform intended effects throughout a required duration in the environments where the artifact has to operate; (b) produce undesirable effects below acceptable thresholds; (c) create unacceptable hazard.

Physical criteria: Includes the ability of the option to be implanted into (a) a constrained geometrical environment; (b) inside fields of mechanical, thermal, electromagnetic constraints.

Cost criterion: Ability of the option to be acquired, operated, maintained and disposed for acceptable costs.

Just on time criterion: Ability of the option to be available just on time.

Risk criterion: Ability of the option to generate solely controllable risks.

Design arguments: We classify arguments in the following way:

Surveys provide a first set of arguments. A survey contains facts that could be used to support or challenge a design decision. The survey itself warrants these facts and is backed by the authority of its issuer.

Computations and **simulations** provide a second set of arguments. A computation or a simulation provides results that are data. The computation or the simulation itself warrants these data and is backed on physical or chemical laws.

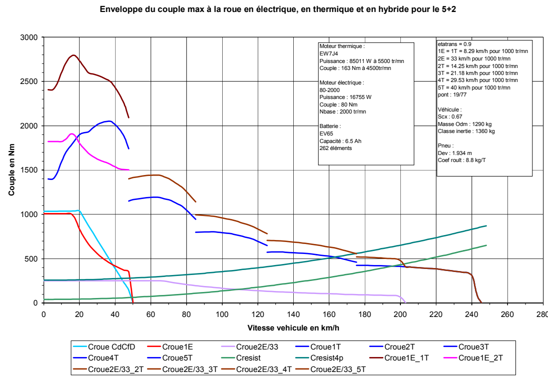

Figure 19: Number and stages of gears rationale based on simulation

Tests provide a third set of arguments. A test provides data that are facts. The test process itself warrants these facts and is backed on the test condition and test hypothesis.

Practical experience provides a fourth set of arguments. The practical experience provides facts that are warranted by the practitioner authority.

Note that these arguments are rational or quasi rational and take into account advice of experienced practitioners (knowing-in-action).

ARGUMENTATION AND COLLABORATIVE WORKING

Aristotle distinguished arguments according the purpose they are intended to serve, arguments designed to (a) achieve absolutely certain knowledge are called apodictic arguments; (b) to lead to generally acceptable opinions are called dialectical arguments; (c) convince a particular audience of the correctness of a standpoint are called rhetorical arguments.

The use that we make of argumentation for engineering systems (arguing for a claim or rebutting it, evaluating an option compared with a criterion, weighing arguments to decide) is obviously (a) dialectical insofar as it attempts to reach the viewpoints considered as the best by the design team, the supplying enterprise (b) rhetorical insofar as it attempts to convince other stakeholders.

Argumentation and audience are inseparably interwoven since arguments are (a) addressed to several audiences: engineer him (or her)-self, design team, management, quality assurance engineers, economic analysts, manufacturers, repairmen, salesmen, acquirers, .. (b) dialectically worked out through discussions and debate involving these audiences.

The theoretical foundation of our approach of the collaborative working is based upon the audience concept as defined by Perelman and Olbrechts-Titeca.

Perelman, introduced three types of audience: (a) self-audience, the arguer him (her)self; (b) particular audiences consisting of particular groups; (c) universal audience consisting of all reasonable human beings. In the same way, we could introduce three types of audience in the systems engineering process: (a) self-audience, the designer him (her)self; (b) particular audiences consisting of stakeholder groups; (c) universal audience consisting of all targeted acquirers. And the engineering process has to convince these audiences.

In this perspective, the development process of a system should be grasped as concurrent phases of investigation of the system definition punctuated with deliberation and synthesis stages: technical meetings and reviews.

The development process is firstly stressed by milestones, which are the main events of collaborative working: requirements and specification reviews, design and design critical reviews, qualification and acceptance reviews, …

As Rittel pointed it out, because of nobody among stakeholders has a guarantee that his knowledge is superior to any stakeholder's knowledge with regard to the problem at hand, the process should be organized as an argument.

In the name of this principle of "symmetry of ignorance", the engine designer's knowledge is not superior to those of electrical engineer or transmission engineer, etc. with regard to the system development what imposes the way of deliberation on them. Each meeting gathers a Perelman's particular audience and is a place for debate where the stakeholders have to defend their standpoints, to make the most of their arguments, to raise their objections, to weigh and assess them, in order to reach agreements.

These agreements based on arguments, as explicit, reasonable and complete as possible, address a set of decisions related to choices (specification, functional, allocation and physical design, verification choices) or about additional investigations.

We enhance how well an explicit argumentation process, as described here above, sets up an efficient canvas to lead technical meetings and why beyond the meetings, this process forms a framework in favor of technical activities placed between milestones.

In the same way that the efficiency of engineering of a system depends on its success with the marketplace for which it is intended, Perelman consider that the soundness of argumentation depends on its success with the audience for whom it is intended.

CONCLUSION

As conclusion, we underline immediate benefits that this process procures for concrete development projects. Limit concerns the lack of a tool integrated to system engineering environments. We end with prospective improvements for this process. These improvements deal with fallacies and biased reasoning in the systems engineering domain and the necessary limitation of the argumentation process at the minimum required.

REFERENCES

ANSI/EIA-632-1998, *Processes for Engineering a System Process*. EIA Standard, jan 7,1999.

ECSS-E-00A and ECSS-E-10A, *Space Engineering, Policy and Principles , System Engineering*. European Cooperation for Space Standardization, 1996.

IEEE Std 1220-1998, *IEEE Standard for Application and Management of the Systems Engineering Process*. IEEE Computer Society, 1998.

Reymen, Isabelle, *Improving Design Processes through Structured Reflection*. SAI Report 2001/1 Eindhoven, October 2001.

Schön, Donald A.: *The Reflective Practitioner, How Professionals Thing in Action*, Basic Books, 1983.

Rittel, Horst, *Second-Generation Design Methods*, in Developments in Design Methodology, Cross Nigel, Wiley, 1984.

Rittel, Horst, *Planning Problems are Wicked Problems*, in Developments in Design Methodology, Cross Nigel, Wiley, 1984.

Perelman, Chaïm and Oblrechts-Titeca, Lucie *Traité de l'argumentation*. Editions de l'Université de Bruxelles, 5ième edition, 2000.

Toulmin, Stephen, *The uses of argument*. Cambridge University Press (CUP), 1958.

Newman, S, *Pushing Toulmin Too Far*. Xerox Palo Alto Research Center, 19XX.

Lee, Jintae and Lai, Kum-Yew *What's in Design Rationale ?* , in Design Rationale, LEA, 1996.

Mac Lean, A, and alii, *Questions, Options and Criteria : Elements of Design Space Analysis,* in Design Rationale, LEA, 1996.

McCall, Raymond and alii, *Making Argumentation Serve Design,* in Design Rationale, LEA, 1996.

Eemeren van, Frans and alii , *Fundamentals of argumentation theory*. L Erlbaum Associates (LEA), 1996.

Bosh, Robert, *Automotive Handbook*, R. Bosch GmbH, 2000.

Graham, R, *Comparing the Benefits and Impacts of HEV Options*, EPRI Final Report, July 2001.

http// www.citroen.com/site/htm/fr/technologies/demain.