EngOpt 2008 - International Conference on Engineering Optimization
Rio de Janeiro, Brazil, 01 - 05 June 2008.

A formal approach for optimized concurrent System Engineering

Yann Pollet

CNAM/CEDRIC, Paris, France. pollet@cnam.fr

Olfa Chourabi

CNAM/CEDRIC, Paris, France. Olfa.chourabi@voila.fr

1. Abstract

System Engineering (SE) is a multidisciplinary activity in witch a system engineer team aims at building a global technical solution in response to a problem stated as a set of requirements. SE may be seen as process of optimizing some system quality factors under constraints of required functions, cost and delay. Complex products and systems, involve a large number of components which are arranged under constraints/relationships in engineering space. This may lead to uncontrolled and unpredictable backtracks in the engineering processes. Effectively managing such process is a major factor witch can increase product development capability and quality and reduce the development cycle time and cost. In this paper, we propose a formal approach for optimizing component allocation and parameter configuration in SE process. We introduce the concept of Alternative Quantity (AQ) to enable manipulation and reasoning on partial engineering choices. The main contribution is to offer efficient evaluation of global systems proprieties although particular technical choices have not been done yet.

2. Keywords: System Engineering Processes, Concurrent Engineering, Optimization, Product configuration.

3. Introduction

System engineering (SE) is an interdisciplinary approach to enable the realization of successful systems. It is defined as an iterative problem solving process aiming at transforming user's requirements into a solution satisfying the constraints of: functionality, cost, time and quality [1]. This process is usually composed of the following seven tasks: State the problem, Investigate alternatives, Model the system, Integrate, Launch the system, Assess performance, and Re-evaluate. These functions can be summarized with the acronym SIMILAR: State, Investigate Model, Integrate, Launch, Assess and Re-evaluate. [2][3].This process is shown in figure 1.
It is important to note that the Systems Engineering Process is not sequential. The tasks are performed in a parallel and iterative manner. At each step a comprehensive set of possible engineering models arises witch are progressively combined and refined to define the target system.
Many engineering domains involve an intricate interplay of conceptual synthesis of alternative requirements and design configurations, preliminary impact analysis of these alternatives using complex simulations tools, and human decision-making. [4]
This may be seen as a process of optimising some system quality factor under constraints of required functionality, cost and delay. In current practices, SE activity is a decomposition of an initial problem into a hierarchy of sub problems, followed by design decisions taken for each elementary problem. A decision often consists in the choice of a given solution among n possible ones (e.g. the allocation of a required function to a given off-the-shelf component). As sub problems are not necessary independent, this leads in general to suboptimal solutions issued from putting together local decisions. In addition, multiple iterations may lead to uncontrolled and unpredictable backtracks in the engineering process. Concurrent Engineering shows situations in which all aspects of the system lifecycle are expected to be optimised together.
In this paper, we focus on decisions related to component allocation choices and parameter configuration, which may be formalised as optimisation problems under constraints. To fully evaluate the system and ensure its compliance to constraints, the system needs to be tested in a large number of crash scenarios. System performance is usually assessed using simulations. Therefore, computer modeling and simulation are used as one of the primary tools in the process. The number of potential combinations of alternative solutions is almost countless and the items to be considered are very closely related.
Effectively managing such process is a major factor witch can increase product development capability and quality and reduce the development cycle time and cost. We aim to present an original method to reduce the complexity of the system evaluation process. We propose an approach based on the well formalised concept of Alternative Quantity (AQ). An AQ offers a structure that enable the evaluation of global system properties while accurate design choices have not been done yet. The benefit is to reduce the number of simulation and to help to border the solution space.
Formally, an AQ represents a particular set of possible values from a D domain, e.g. a continuous domain such as the set of reals numbers or a discrete domain such as a set of states of a sub system, each of one being associated to a specific hypothesis. AQ are distributions of a confidence measure over a D domain, where the values belong to a Boolean algebra of hypothesis, instead of being a real number between 0 and 1. In the case of the real number domain, it is possible to build algebra of AQ having calculation properties similar to those of classical reals numbers. This leads to an easy transposing of classical calculus on AQ, enabling the evaluation of global systems properties although particular technical choices have not been done yet.

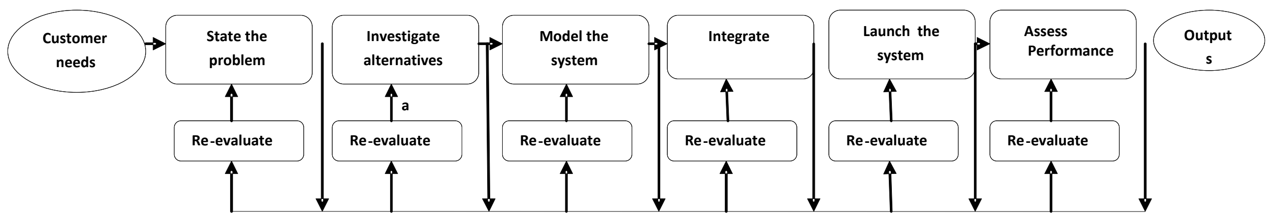

Fig1. System Engineering Process [2]

The paper is structured as follows. Section 2, discusses key background information about System Engineering processes and describes a motivating research example. Section 3, presents an overview of some related approaches for concurrent system engineering processes optimization. Section 4, defines Alternative Quantity (AQ) formalism with its underlying reasoning mechanism. In section 5, we evaluate our approach through an example of an automated transport sub system. The paper concludes with an overview of our contribution and future work directions.

4. Background and motivation

In this section, we focus on decisions related to component allocation choices and parameter configuration phase in SE process. In this setting, the main objective is to find configurations of parts that implement a particular function. Practically, system engineer team must consider various constraints simultaneously, the constituent part of a system are subject to different restrictions, imposed by technical, performance, assembling and financial considerations, to name just a few. Combinations of those items are almost countless, and the items to be considered are very closely related. Figure 2, shows the intricate interplay of constraints in a system engineering process.

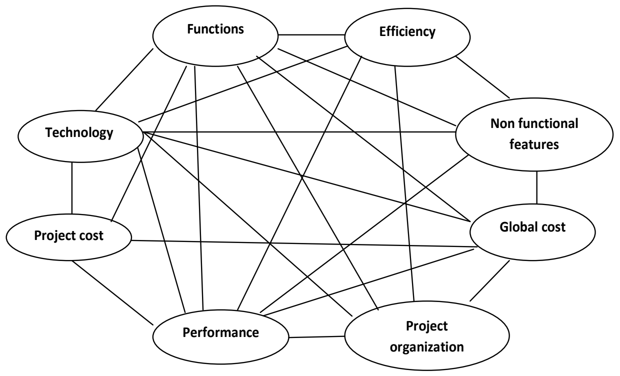

Figure 2: System Engineering process as a multi-objective optimization problems

As an example, we consider a typical component allocation process of a transportation sub system: an automated wagon.
We assume that the sub system's functional view is represented by the data flux diagram depicted in figure 3. The main functions considered are: F1: capture speed, F2: capture position, F3: control movement, F4 propel, F5: break, F6: contain travelers.
The system functions should be mapped to the physical components. Functions mapping to physical components can be one to one or many to one. This functional view was derived from the initial functional requirement: to transport travellers from one point to another. In addition Physical solution is constrained with non functional requirements (or soft goals) such as: system performance with attributes of travel duration, facility, acceleration limitation, comfort, reliability etc.
A possible physical solution for this system need to implement functions set, under constraints of non functional requirements and constraint of cost and leads time. Concurrently, models should be constructed and evaluated, simulation data should be derived, and prototypes should be built and measured.
In addition, it is important to emphasize that non functional requirements are reflected on allocated component in multiple ways. We could distinguish direct and indirect (or composite) requirement allocation. In our example, the attributes space and quality are directly assigned to the component passenger's cells, whereas, system weight attribute should be shared out among the system constituents. Others constraints impacting the global system and needs to be checked by simulation, such as efficiency of transport duration in different scenarios, system reliability etc.
Furthermore, each physical constituent choice raises a set of possible engineering alternatives. The global requirements are traded-off to find the preferred alternatives. An intricate interplay usually exists among alternatives. For example, the functions speed capture and position estimation choosing *inertial station* that delivers the speed as well as the position, for implementing the function *speed capture* would restrict the engineering choices to exclude specific transducers. Likewise, interplay may concern a common parameter. We could consider as a sample the dependencies between efficiency time parameter and the aggregate system weight parameter. We should note here that, in general, a subset of requirement is "rigid" i.e imposed and other is more flexible.

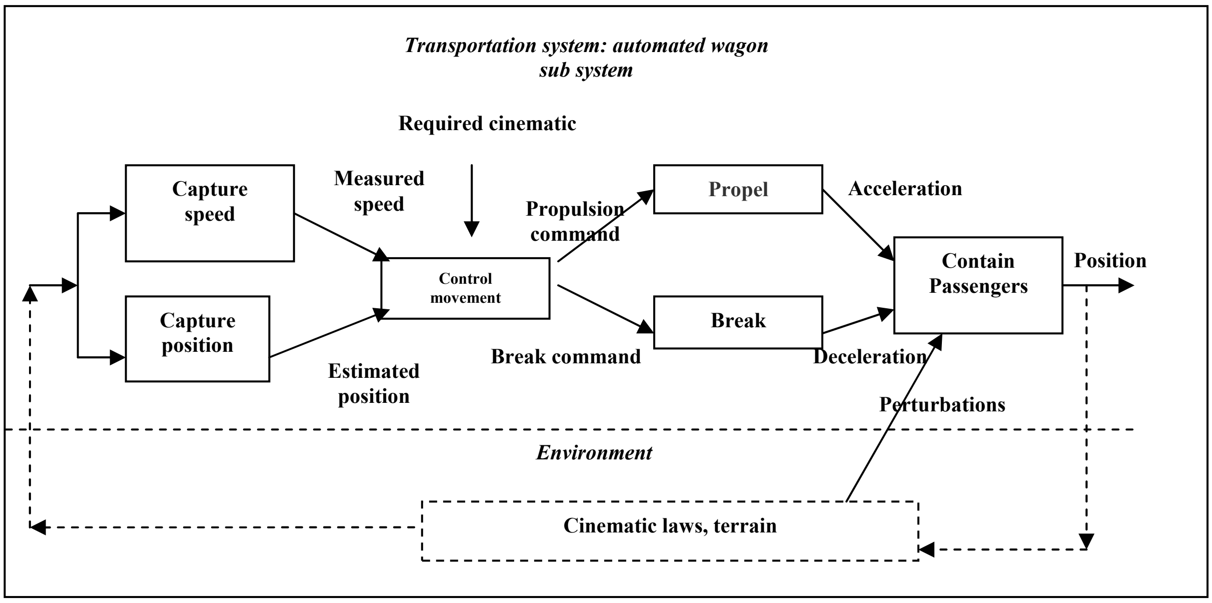

Fig3. Data flow diagram

The automated wagon example highlights multiple dependencies between technical solutions and mutual influences and trade off that arises in a quasi simplified use case. In system engineering practices, we face more complex interplays among local choices relating to sub system and global engineering decisions relating to the whole system.
Broadly speaking, the Re-evaluate loops depicted in figure.1 are arguably the most time consuming activities in System Engineering processes. Engineers have used feedback to help control systems and improve performance. It is one of the most fundamental engineering tools. Re-evaluation should be a continual process with many parallel loops. Re-evaluate means observing outputs and using this information to modify the system, the inputs, the product or the process.
It would clearly be useful to have some way of managing the multiple dependencies and exclusions between hypothetic alternative engineering choices, while saving traceability links and justifications.

5. Related work

In this section we examine some approaches to support decision making in system engineering processes. We discuss truth maintenance systems and constraint satisfaction problems approaches and we explain their inadequacy in our context. Then we explain our proposition of alternative quantities (AQ) informally, by the means of some examples. These examples are representative for the new structure we are putting forward, depicting different possible practical engineering use of AQ.

-*Truth Maintenance Systems (TMS) / Assumption truth maintenance Systems (ATMS):* Truth maintenance (also called belief revision or reason maintenance) is an area of AI concerned with revising sets of beliefs and maintaining the truth in the system when new information contradicts existing information. It is a subsystem that manages the use of assumptions in the reasoning process of a problem solver. Doyle's [8] original motivation for creating a truth maintenance system was to augment a reasoning system with a control strategy for activities concerning its non monotonic state of beliefs. These systems can be viewed as constraint propagation mechanisms. Given a disjunctive set of sets of premises and a set of (monotonic) deductive constraints, de Kleer's ATMS [9][10][11] tells a client problem solving system what things it is currently obliged to believe, assuming one or another of the sets of premises. Doyle's and Goodwin's TMS's, on the other hand, tell the client problem solving system what things it is currently obliged to believe, given a single set of premises under deductive constraints, some of which may be non monotonic in nature.
Typically, complex engineering artifacts, involves successive choices and multiple iterations with backtracks in case of facing an unsatisfied constraint or requirement. But this process, of managing a complex network of assumptions is very time consuming and difficult to apply in practice. In addition, the demands on the development of systems are steadily increasing: shorter time to market, better quality, and better productivity are critical for the competitiveness of today's SE organizations. For this reason, we argue that TMS/ATMS are not adequate in our context.

-*Constraints satisfaction problems (CSP):* A general introduction to constraints is found in [5]. Constraint-satisfaction problems can be stated as follows: We are given a set of variables, a finite and discrete domain for each variable, and a set of constraints. Each constraint is defined over some subset of the original set of variables and limits the combinations of values that the variables in this subset can take. The goal is to find one assignment to the variables such that the assignment satisfies all the constraints. In some problems, the goal is to find all such assignments. [6][7]
The major advantage of this paradigm is its high declarativity, domain independence and simplicity of use. Nevertheless, this formalism is not powerful enough to capture or to take advantage of essential aspects of system engineering practices, such as the unknown a priori number of constituent parts of a system, the unknown choice of a particular physical component and so on. All variables must be specified in advance, making it awkward to represent problems witch requires simulation results as inputs in order

to keep or eliminate alternative solutions.
In such setting, we need to manipulate parameters that have not been totally defined. To address this modeling need, we introduce the concept of Alternative Quantity (AQ) [12]. AQ may be considered as structure encapsulating a set of possible values, enabling reasoning and calculation facilities with sound results, exactly as the case of well defined parameter.
Let's consider for example the case of the automated wagon. Let's assume that we haven't yet chosen the *propel mode* while we need the *constituent weight* in order to conduct simulations. Or, let's assume that we haven't yet fixed the *transducer component* while we need to have preliminary results to validate an engineering choice. In current practices, system engineers must simulate the global solution in order to assess system performance. As simulations require specific parameters this leads to multiple trial and error process before exhibiting an optimal solution.

With Alternative Quantities we are able to express that some variables can be assigned to a sub problem, rather than a simple value. In this way, components can be selected at a higher level, before being specified in terms of subassemblies. An AQ allows to encapsulate sets of possible values and to combine them soundly with the remaining system elements such as physical relations. Possible AQ values could be numeric or discrete, e.g. system state, as well.

6. Alternative Quantity: Formal definition

In order to express that a quantity is able to take multiple possible values in an engineering process, on develop an alike theory to a distribution of possibilities over a domain of values, called fuzzy quantity [13][14]. Possibility theory is a mathematical theory for dealing with certain types of uncertainty and is an alternative to probability theory. Zadeh [13] introduced possibility theory as an extension of his theory of fuzzy sets and fuzzy logic. D. Dubois and H. Prade [15] further contributed to its development.

6.1 Formal definitions

Distribution of possibilities (reminder)

Let Q be a domain of values discrete or continuous. A distribution of possibilities over Q is defined by a function μ mapping each element of Q into a closed unit interval [0, 1]. [15][16]

$\forall q \in Q: q \rightarrow \mu(q) \in [0,1]$
$\exists q0; \mu(q0) = 1$

A fuzzy quantity may be noted: $Q = \mu_1/q_1 + \mu_2/q_2 + \ldots + \mu_n/q_n$. Where *qi* denotes the possible values and *μi* denotes the values of corresponding possibilities.
For instance: the quantity Q= 1/1.2 + 0, 5/1.3 embody a real number having two possible values 1.2 and 1.3.
Certain operations of ordinary algebra, in particular *addition (x+y)*, multiplication (*xy)*, and construction of inverse (*-x)*, could be extended and applied to fuzzy quantities. But we notice that basic proprieties of these operations are lost, e.g. associativity, inverse and distributivity.
For example, let q1, q2, q3 be fuzzy reals. We have: $(q1+q2)*q3 \neq q1*q3 + q2*q3$. The associatively propriety is not verified.
E.g. if q1, q2, q3 embody respectively two possibilities, then (q1+q2)*q3 is evaluated to 8 possibilities where as q1*q3 +q2*q3 is evaluated to 16 possibilities. In other term, the result depends on the operand order.
This example highlights a significant inadequacy to the applicability of this theory to our approach. A fuzzy quantity doesn't keep trace of the rationale underlying its calculation process. Thus, we can't have the same result as the sum of real number. This leads to limited applicability in a simulation context because we would like to be able to reason over possible values.

We introduce the concept of Alternative Quantity (AQ) as an extended fuzzy quantity that satisfies the following requirements:
- It keeps traceability to its operand origin
- Adapts possible values depending on engineering results.

Following, we present a formal definition to Alternative Quantities.

Definition 1: Alternative Quantity (AQ)
An alternative quantity with values belonging to a domain Q, is defined by a function *f* from Q to Ω, where Ω is a Boolean algebra such that:
The cardinal of Ω *is infinite*
There exists an infinite set of element ≠ {}, Xi (i=1, 2, ...) such that :
If i ≠j
then Xi.Xj = {}, Xi. ¬Xj = {}, ¬Xi.Xj = {}, ¬Xi. ¬Xj = {}.

Xi (i=1, 2, …) are called *principal elements* or *generators* of Ω.
The elements of Ω expressed as the multiplication of principal element by its negative element are called *irreducible.*

Definition 3: Alternative Quantity Measure
An AQ measure is defined by a function mapping each pair (Xi,⌐Xi) to a pair of values belonging to [0,1].

(Xi,⌐Xi) → (µi, □i) ∈[0,1]
where µi= 1 or □i=1.

µi is called measure of principal element Xi,
□i is called measure of negative element ⌐Xi.
We note |x| the measure of X.
With Alternative Quantity measure we are able to define a more expressive measure, which holds parameter values and their definition origins as well.

Definition 4: Alternative Quantity with finite number of values.
An AQ is said to be AQ with finite number of values if only a finite number of values from Q have an image different from the empty set. It could be expressed as:

$X = X_{\varphi(1)} /v1 + \ldots\ldots X_{\varphi(n)} /vn$. Where

- $X_{\varphi(i)}$ denotes the *irreducible* elements.
- The support of X is $\cup\, X\, \varphi(i)$. If the support of X is Ω , then X is called always defined.

Example: alternative quantities with values in $\mathbb{R}$ with finite number of values and always defined

X = X1/ 1 + ⌐X1/ 2. X is an alternative quantity evaluated to 1 or 2 and keeps the origin of its basic hypothesis X1 or ⌐X1 in simulation operations.

In addition, formal operations allow us, in practice, to manipulate alternatives quantities. For example, let Y= *X2/ 2 + ⌐X2/5, be an AQ*. We could calculate the sum of AQ (X+Y) by reducing to a common factor as following:

X+Y = X1.X2 / 1 + X1.⌐X2 / 5+ ⌐X1.X2 / 4 + ⌐X1.⌐X2 / 10

We could implement operations on QAR, by linking them to a mathematical library of alternative quantities. i.e. in object oriented programming, a QAR represents an instance of a class Alternative(float).

***Interpretation*: **we propose to use AQ as a structure that embody assumptions hypothesis and choices at a step of an ongoing engineering process.

- *Principal elements* Xi are used as symbolic elements to represent the set of all independent assumptions considered by system engineers until choosing the final engineering solution.

In an engineering situation Ω corresponds to an infinite set of assumptions. But in practical setting Ω is rather a finite set that represents engineer's assumptions underling different engineering choices.
Assumption measure |Xi| corresponds to the a priori acceptance of that assumption. This measure enables to assign preferences to alternatives solutions for an engineering problem.
The quantity |Xi| is used as a confidence measure for a particular belief.
Let's consider an engineering situation where two alternatives solutions are possible for a sub problem. We could put a superior confidence measure for the alternative that was used and verified in prior engineering situation. Else we could put 1 to mean that no preference is accorded.
In the example presented in section 2, the automated wagon consists of an *inertial station* and a *passer cell*. Each of these components can be represented by a variable having as domain a sub problem.
For instance the *propel mode* could have two values: *petrol* and *electric power*. We can assign a superior confidence measure to the solution *electric power*, if we have as soft goal *environment conservation*. This situation is formalized with an Alternative Quantity as following:

Propel mode = X1/petrol+ ⌐X1/ electric power with |X1|= 0,001 and |not X1|= 1

An alternative quantity is a parameter that enables to define a partially defined local solution. The major contribution is the facility of keeping different alternative element and reasoning on that structure as a well defined parameter i.e specific parameter value. An AQ could be all over defined if it s assigned to global engineering problem such as *vehicle propel mode*.
It could have limited scope if it is meaningful for a sub problem or dependent to specific choices. For example, lighting mode is meaningless if the propel mode is not *petrol*.

6.2 Real Alternative Quantity (QAR) algebra

This section describes an algebra for real alternative quantity i.e Q=$\mathbb{R}$. We argue that this algebra has calculation proprieties similar to those of classical real number. This leads to an easy transposing of classical calculus on AQ, enabling evaluation of global system proprieties although particular technical choices have not been done yet.

The binary operations: addition and multiplication are defined on QAR. Division is defined if the QAR has no possible null value.

> ***QAR addition****: commutativity, associativity holds for QAR. QAR addition has an identity element witch is zero.i.e QAR equal to zero for whatever hypothesis. Each QAR has an inverse element for addition.*
>
> ***QAR multiplication****: commutativity, associativity and distributivity over addition. Each QAR has an inverse element for the product.*

We present an example for QAR addition. This example highlights the transposing of classical calculus. In opposition to, fuzzy real where calculation is different from real calculation, QAR algebra gives similar results to those of classic real numbers. QAR is a structure encapsulating its possible values and the assumptions network on witch depend these values, as well.
For example,
Let's consider a fuzzy number having two possible values 1 and -1: x= 1/ (-1) + 1/ (+1)
Let's consider its inverse element for the addition: -x= 1/ (+1) + 1/ (-1),
The addition (x+ « -x ») gives three possibilities which are: -2, 0 and 2. This result contrasts with classical real calculus.
i.e: (x+ « -x ») must be evaluated to zero.
On the contrary, with QAR, we have the following results:
The same fuzzy number is expressed as:

x = X1/ (-1) + ⌐X1 / (+1) has as inverse element for the addition: x' = X1/(1) + ⌐X1 / (-1)

The result of (x + x') is evaluated to X1/0 + ⌐X1/0 = Ω /0 = 0, witch is conform to classical calculus of real.

6.3 Real Alternative Quantity (QAR) comparators

We could define a predicate θ on two QAR

> θ:X, Y → {true, false}, where θ is a relation such as equality, inequality , superior to etc.

Let's present some examples for QAR comparison

- *Let x be a QAR, having value x= X1/0.8 + ⌐X1/1.1. (x < 1) is evaluated to X1/true + ⌐X1/false*
- *Let x,y,z be QAR, having respective values :*
  - *x = X1/0.8 + ⌐X1/1.1*
  - *y = X1/0.9 + ⌐X1/1.0*
  - *z = X1/1.2 + ⌐X1/1.3*

  (x < y) is evaluated to X1/true + ⌐ X1/false

  (x < z) is evaluated X1/true + ⌐X1/true = □/true = true

- *Let x, y be QAR having respective values*
  - *x = X1/0.8 + ⌐X1/1.1*
  - *y = X2/0.9 + ⌐X2/1.0*

  (x<y) is evaluated to X1.X2/true + X1.⌐X2/true + ⌐X1.X2/False + ⌐X1.⌐X2/False
  = X1. (X2 + ⌐ X2)/true + ⌐X1. (X2+ ⌐X2)/False
  = X1. Ω /true + ⌐X1. Ω /False
  = X1/true+ ⌐X1/False

This result is intuitive. In fact, (x<y) is dependant of the hypothesis X1. The comparison result sets that if X1 is true then (x<y) else the result is evaluated to false.

We could define an algorithm to automate the testing process of QAR. This would enable to border hypothetical solution space providing a set of constraints. Following a sample illustrating this principle:

x = X1/0.8 + ⌐X1/1.1

y = X2/0.9 + ⌐X2/1.0

if (x < y) then x = x + 1

(x < y) is evaluated to X1/true + ⌐X1/False

(x < y) is supported by X1

x is evaluated to x+1 = x + Ω /1 if X1is true and remains if ⌐X1is true

x = X1.(X1/1.8 + ⌐X1/2.1) + ⌐X1.(X1/0.8 + ⌐X1/1.1) = X1/1.8 + ⌐X1/1.1

7. Application: Modeling a Transportation sub system with Alternative Quantity

As mentioned in the previous section, the AQ represents particular sets of possible system proprieties values associated with underlining hypothesis at a given step of an engineering process. This section details shows how AQ can be effectively used to support system evaluation and verification process.

We use the case study presented in the motivation section to explore the advantages that can be derived from the use of AQ. The study shows how, by employing such a modeling structure, it is possible to reduce the complexity of the simulation process.

The transportation sub system is a self piloted wagon. We focus on the synthesis phase and we discuss decisions related to component allocation choices and parameter configuration. Figure 4 presents an allocation sample for the automated transport system. The set of parameter for the physical solutions are presented.

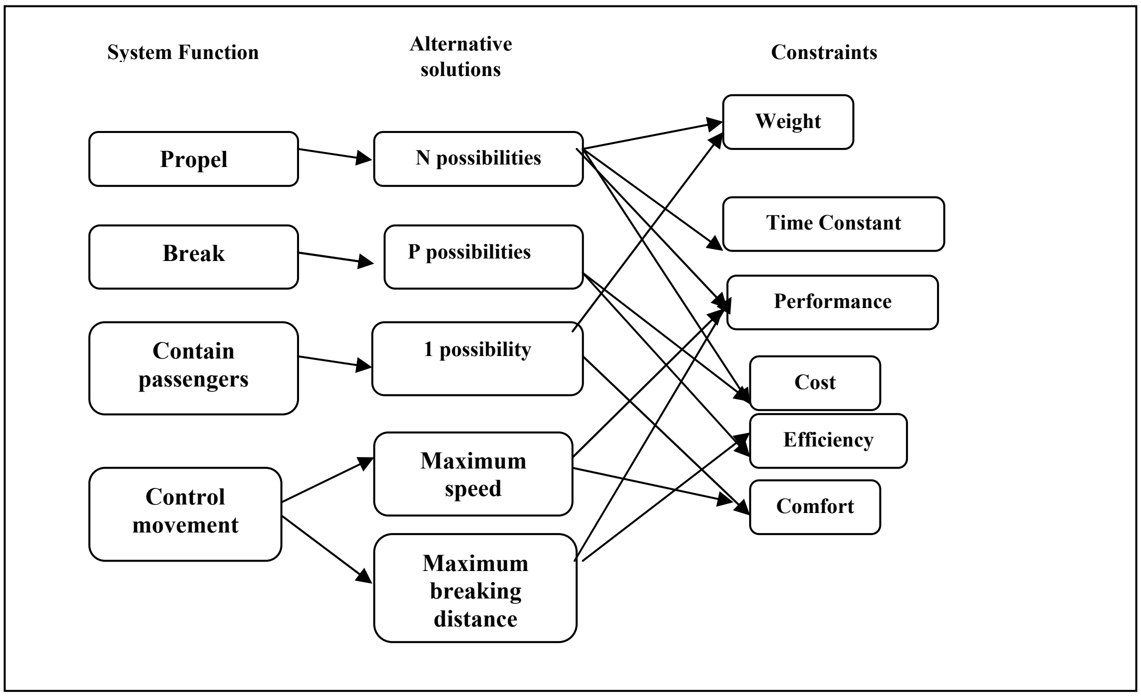

Fig. 4 allocation process sample for an automated transport system

Alternative designs are created and are evaluated based on performance, schedule, cost and risk. Then, concurrently, models should be constructed and evaluated, simulation data should be derived and prototypes should be built and measured.

Let's consider for example the following simulations scenarios:

-*Control component (a software in an embedded calculating)*: having as
Inputs: measured speed and measured position.
Outputs: acceleration control and breaking control
Parameters: maximum speed, maximum acceleration and maximum breaking distance (with values within a possible interval)

-Engine component: having as
Inputs: acceleration control
Outputs: acceleration
Parameters: engine type

-Braking component: having as
Inputs: deceleration control
Outputs: deceleration
Parameters: breaking component type

-Vehicle Kinematic: without parameters as the vehicle is already chosen at this stage.
-Transducers: that are assumed to be perfect, characterized by: measured speed = real speed and measured position = real position.
Concerning the *control*, we have a succession of four phases corresponding to different system states. In each phase, we consider a defined control law, witch controls the acceleration in the phase acceleration, controls the acceleration and the braking in the phase constant speed and controls the braking in the phase deceleration. Law successions could be modeled as a state machine diagram, where the transitions could be either events e.g. departure or predicates on reached speed, reached position etc. e.g.(acceleration phase transition-deceleration phase) or (constant speed transition- deceleration phase). A new system state is a function of current state and input. i.e. state n+1= F (state n, input).

For the vehicle we use the kinematic equations e.g. $S(n+1).dt = S\ n.dt + \gamma.dt$

$P(n+1).dt = P\ n.dt + S\ n.dt$

Where dt denotes the time pace of the simulation and γ denotes the acceleration or the deceleration.
For the engine, we use an approximation By a first order transfer function with a delay T0Engine. This function gives the following recurrent formula: $\gamma\ (n+1).dt = A.\gamma\ n.dt\ +\ B.Un.dt$.
Where u n.dt is the acceleration control at the instant n.dt
A is fixed by the formula: $e^{-T0.dt}$ and B = 1 – A.

In our approach, the classical formula, detailed above, we substitute the classical reals values or the system state with real alternative quantities or by alternative states quantities.
We have implemented this approach, by developing a mathematical library with basic alternative quantities classes. Each alternative object is an AQ class instance representing a set of alternative values. The operators, detailed in section 6, e.g. addition, multiplication, inverse, and QAR comparators are used to evaluate alternative results. In this way, alternative quantities allow to considerably optimize and reduce the complexity of the simulation process.

8. Conclusion

During component allocation process, system engineers typically have to find configurations of parts that implement a particular function. This process is constraint orientated, and requires the recognition, formulation and satisfaction of constraints.
In this paper we have introduced a novel formal approach to support efficient product configurations. We have discussed the contribution of alternative quantities i.e. to offer sound reasoning mechanism on product parameter although particular technical choices have not been done yet. We are currently investigating application of the proposed formalism in real engineering projects.

9. References

1. J.Meinadier. Le métier d'intégration de systèmes. Hermes Science Publications, décembre, 2002.
2. A. T. Bahill and B. Gissing, Re-evaluating systems engineering concepts using systems thinking, IEEE Transaction on Systems, Man and Cybernetics, Part C: Applications and Reviews, 28 (4), 516-527, 1998
3. International Council on Systems Engineering (INCOSE). [On line] www.incose.org
4. Shinya Tarumi et al. Development of a Design Supporting System for Nano-Materials based on a Framework for Integrated Knowledge of Functioning-Manufacturing Process. In Proc. of the 10th IASTED International Conference Intelligent Systems and Control(ISC2007),Cambridge, Massachusetts, USA, Novemb 19-21,pp.446-454, 2007
5. V. Kumar. Algorithms for constraint-satisfaction problems: A survey. AI Magazine, pages 32 – 44, Spring 1992
6. Freuder, E. 1989. Partial Constraint Satisfaction. In Proceedings of the Eleventh International Joint Conference on Artificial Intelligence, 278–283.
7. Gu, J. 1989. Parallel Algorithms and Architectures for Very Fast AI Search. Ph.D. diss., Computer Science Dept., Univ. of Utah.
8. Jon Doyle. Truth Maintenance Systems for Problem Solving. Technical Report AI-TR-419, Massachusetts Institute of Technology, Artificial Intelligence Laboratory, Cambridge, January 1978
9. de Kleer, J. 1986a. An Assumption-Based TMS. Artificial Intelligence 28:127–162
10. de Kleer, J. 1986b. Problems with ATMS. Artificial Intelligence 28:197–224
11. de Kleer, J. 1989. A Comparison of ATMS and CSP Techniques. In Proceedings of the Eleventh International
Joint Conference on Artificial Intelligence, 290-296. Menlo Park, Calif.: International Joint Conferences on Artificial Intelligence
12. Sébastien ROBIDOU, Yann POLLET. Imprecision and Uncertainty in Object Oriented Databases. 5th European Congress on Intelligent Techniques & Soft Computing, EUFIT'97. Aachen, Germany, September 1997.
13. L.A. Zadeh (1965) Fuzzy sets. Information and Control 8 (3) 338-353
14. Zadeh, L., Probability measures of fuzzy events, Jour. Math. Analysis and Appl. 23, 421-427, 1968.
15. Dubois and Prade, Possibility theory and data fusion in poorly informed environments. Control Engineering Practice, 2(5):811-823.
16. Duboi et al. Readings in Fuzzy Sets for Intelligent Systems. Morgan Kaufmann, 2929 Campus Drive, Suite 260, San Mateo, Californie 94403, USA.
17. Yann POLLET, Eric RICARD et Sébastien ROBIDOU. A fuzzy Spatio Temporal Data Model for CIS. In "Fuzzy Logic and Soft Computing", edited by B. Bouchon-Meunier, R. R. Yager and L. A. Zadeh. World Scientific, 1995.
18. Yann POLLET and Sébastien ROBIDOU. An approach for the representation of multi-valued attributes in Fuzzy Databases. FUZZ-IEEE/IFES'95 Workshop on Fuzzy Databases and Information Retrieval. Yokohama, Japon, mars 1995.
19. Yann POLLET et Sébastien ROBIDOU. STORM : une approche pour la prise en compte de l'imprécision et de l'incertitude dans la gestion des grands projets". 11ème Colloque National de Fiabilité et Maintenabilité (LambdaMu11), session "Mathématiques appliquées à la SdF". Arcachon, septembre 98.