10th AIAA Aviation Technology, Integration, and Operations (ATIO) Conference
13 - 15 September 2010, Fort Worth, Texas

AIAA 2010-9263

Correct-by-Construction Design of Aircraft Electric Power Systems*

Alessandro Pinto †
United Technologies Research Center, Inc., Berkeley, CA

Sandor Becz ‡
Lenze SE, 630 Douglas Street, Uxbridge MA 01569

Hayden M. Reeve §
United Technologies Research Center, East Hartford, CT, 06108

We provide an optimization oriented formalization of correct-by-construction design following the principles of Platform-Based Design.[1] The design proceeds by refinement steps. At each step, a specification is given in terms of requirements to be satisfied by an implementation. The implementation choices are implicitly captured by a set of components, their properties, and their composition rules. A class of candidate implementations of the specification is derived by formulating and solving an optimization problem. The implementation becomes the specification for the next step in the design flow. We show how the optimization oriented formalization enables design space exploration, and we present the trade-offs involved in the selection of the refinement steps. We show how the methodology can be applied to the design of electric power systems by decomposing the design flow into the following steps: generator selection, generation of the connection configuration under faults, and topology design of the power distribution system.

I. Introduction

The design flow used today for electrical systems is mainly top-down and provides limited ability to predict, early in the design process, the consequences on system performance and cost of radical departures from known designs. This is why the design of aircraft secondary power systems has been for years a derivative process, where previous designs that are known to work undergo slight modifications to accommodate new features. Through the end of the Second World War, the 28 VDC system was typical. With the advent of the jet age, the increasing in power load led to the adoption of the more weight efficient 115 VAC / 400 Hz distribution system.[2] For the next four decades, this system dominated, typically using constant speed devices (CSDs) to ensure a constant 400 Hz frequency, and 2 or 4 channels. Research effort was directed mainly on the improvement of component level performance (weight and efficiency) rather than design methodologies and tools for automatic design exploration and verification. The arrival of new *"more-electric"* technologies such as electric main engine start, electrical cabin air pressurization, and electric primary flight control actuation has again increased the power demands on the electrical system and resulted in the adoption of higher voltage systems (270 VDC, 230 VAC Variable Frequency) in order to reduce distribution (feeder) weight. These changes have also brought system synthesis, evaluation, and verification challenges that are not well met by the legacy design system. For example, the 787 shows a fourfold increase in electrical power capability over the 777, threefold increase in the number of electrical buses, and a XXfold increase in the number of distribution states. Because the requirements imposed by these new applications are drastically different from the ones imposed on the previous generation of aircraft, re-use of known solutions

*Distribution Statement "A" (Approved for Public Release, Distribution Unlimited)
†Staff Engineer, Embedded Systems and Networks
‡Director of Engineering, AIAA Member.
§Staff Engineer, Thermal Management, Senior AIAA Member

and methods becomes inadequate. A major architectural re-design of the electrical systems poses challenges to engineers that find themselves engaged in manual exploration of a large design space constrained by many, and informally captured requirements and component performance limitations.

Typically, a new design is prototyped and tested. If the application requirements are not met, then the system is re-designed. The re-design cycle goes through the manual process of changing design decisions and producing a new prototype (or a change in the current prototype). Re-design is not unusual and is the direct consequence of difficulties to evaluate design solutions and predict the impact of design decisions made in the early stages of the design process on the performance of the final implementation. This problem can be attributed to several reasons, among which we mention the semantic gap between the specification of the system requirements and the details of the implementation platform, and the lack of methods, tools and formal models helping designers in marching from the system requirements to the detailed system implementation.

These two factors are not independent. System requirements are captured using documentations (assisted by requirement management tools such as DOORS[3]), and manually refined into several linked documents that capture not only the partitioning of the system into sub-systems (already implying the system architecture), but also the local performance constraints that each sub-system must satisfy. However, the number of possible choices for a system architecture is large which makes this process complex, and the solution sub-optimal at best. Perhaps, one simple complexity measure for a design can be defined as the number of requirements and the number of degrees of freedom in choosing the implementation. Further, because the high level architectural decision are based on non-executable (and non-analyzable) models, it is difficult to assess the behavioral properties of the system, and therefore impossible to look-ahead for possible emergent behaviors arising from the composition of sub-systems. For these reasons, some tools have been developed to help engineers evaluating the fitness of an architecture to a given application. The Design Structure Matrix (DSM)[4,5] and the Architecture Design Graphs (ADG)[6] have been used in aerospace (among other fields). However, these methods provide limited capabilities for efficient design exploration at different stages of the design process.

In this article, we present a correct-by-construction methodology inspired by the Platform-Based Design (PBD)[1] methodology that has been successfully used in the automotive and consumer electronics domains . The PBD methodology provides an intellectual framework where a design flow that implements a specification proceeds through self-similar refinement steps. In this framework there is a clear distinction between the function (what the system is supposed to do, i.e. the requirements) and the architecture (how requirements are realized, i.e. the components and their interconnection that together implement the function) that allows for automatic design space exploration. Each refinement step consists in selecting a platform instance that correctly implements a specification. A platform instance is a valid composition of library elements that are characterized by their cost and performance metrics. Thus, a design step can be formalized by an optimization problem (in general multi-objective) whose solution (or set of non-dominated solutions) represents the functional specification to be implemented by the sub-sequent refinement step. This process repeats until the abstraction level is close enough to the implementation.

Key to the success of such methodology is the careful selection of the abstraction layers, i.e. the selection of the refinement steps. In fact, each step explores the design space along a subset of the axes representing the design variables. Thus, it is important to carefully prioritize the design choices and make sure that the performance and cost models are accurate enough for the level of abstraction such that design decisions can be made without compromising the quality of the final implementation. Ideally, if each refinement step is done by solving an optimization problem and if the models are accurate (with respect to the abstraction level), the verification effort is minimal because the implementation is guaranteed to satisfy the specification by construction.

II. Preliminaries

Formal treatments of the PBD methodology have been presented using different mathematical frameworks such as agent algebra[7] and labeled graphs.[8] In this section we provide an optimization oriented description to outline the trade-offs involved in the definition of a concrete instantiation of a PBD design flow.

Consider a library of parametric components that can be instantiated and configured by selecting the values of the parameters. Each instance s of a library element (e.g. a generator or a load) has a set Q_s of associated parameters. A parameter $q \in Q_s$ denotes a metric (e.g. the rated power of a generator) that

affects cost and performance of a design . Let $x_{s,q}$ be a variables associated with parameter q of component s. This variable ranges over a domain of values D_q. For instance, the availability a of a generator ranges in the closed interval $D_a = [0, 1] \subset \mathbb{R}$. A system comprises a set of component instances S and implicitly defines a set of decision variables $X = \{x_{s,q}\}_{s \in S, q \in Q_s}$ ranging over the domain $D_X = \times_{s \in S, q \in Q_s} D_q$. Parameters are very general quantities that can be used to model choices in the design of a system. For example, a binary parameter ι can be used to decide whether a component is really needed in a system or not. This parameter could be used to decided whether a system needs one or two generators. A designer may start with an instance that includes two generators s_1 and s_2, and then realize that one generator is sufficient to power all loads, in which case the value of the variable $x_{s_1,\iota}$ may be set to zero to denote that generator s_1 is superfluous and can be removed from the system (we will exercise this feature later in our examples). Also, some of the variables may be assigned as a result of the specification. For example, the power required by a load is given as input to the design problem.

The design space is a subset of D_X. In fact, a platform is defined by the library and by a set of constraints called *composition rules*. For example, in some power systems, generators cannot be connected on the same bus. Therefore, the design space, i.e. the set of valid assignments of the variables X is restricted by a set of platform constraints $C_p(X)$. The functional requirements are captured by another set of constraints $C_m(X)$ that defines those assignments that correctly implements the specification. For example, under all possible faults, critical loads must always be powered; the total power required by loads is provided by generators. Thus, the set of system configurations that are valid platform instances and that satisfy the specification is $C_p(X) \cap C_m(X)$. Finally, the cost of a system is in general a multi-objective function $F : D_X \to \mathbb{R}^f$. Thus, the optimal configuration problem can be written as follows:

$$\begin{aligned} \underset{X}{\text{minimize}} \quad & F(X) \\ \text{subject to} \quad & X \in C_m(X) \cap C_p(X). \end{aligned}$$

The complexity of this problem depends on the number of decision variables of the problem, i.e. $|X|$, the structure of the constraints, and the form of the cost function. If the library is defined at a very low abstraction level, with many components each characterized by many parameters, finding a solution to this problem becomes challenging. Imagine for example considering a library that includes wires, contactors, transformer-rectifier units (TRU), converters, inverters, generators, loads, batteries, circuit breakers, and all other detailed components of a typical system. To deal with this complexity, the design process can be divided into refinement steps where the set X is partitioned into sub-sets $X_1, \ldots, X_L$. At the i-th layer, the following problem is solved:

$$\begin{aligned} \underset{X_i, \tilde{X}_i}{\text{minimize}} \quad & F_i(X_i, \tilde{X}_i, X_1^*, \ldots, X_{i-1}^*) \\ \text{subject to} \quad & (X_i, \tilde{X}_i) \in C_{m_i}(X_i, \tilde{X}_i, X_1^*, \ldots, X_{i-1}^*) \cap C_{p_i}(X_i, \tilde{X}_i, X_1^*, \ldots, X_{i-1}^*). \end{aligned}$$

where $\tilde{X}_i$ is a set of additional variables that are used to capture the abstraction of the variables in the sets $X_{i+1}, \ldots, X_L$. These additional variables often represent virtual components. We will show an example of how the power distribution system is abstracted into point-to-point connections by introducing connectivity variables. The solution of this problem is the set of optimal values X_i^* and $\tilde{X}_i^*$. Clearly, $\cap_{i=1}^{L} C_i \subseteq C$ meaning that only feasible solutions should be explored. This formalization shows the choices that need to be made in the definition of a PBD flow, and interesting additional features that this methodology provides:

1. the set of variables X is far from being unstructured meaning that there are some additional constraints to take into account when deciding on the partition $X_1, \ldots, X_L$. For example, the topology of the power distribution system results as a consequence of the decision on the number of generators and the connectivity requirements between loads and generators. By the same token, the insertion of tie and circuit breakers can only be decided after the topology of the power distribution system has been designed. Structural constraints arise naturally from the notion of refinement where sub-systems are further decomposed into sub-systems.

2. Ideally, X^* should be equal to $(X_1^*, \ldots, X_L^*)$. However, this result depends on the quality of the abstraction, meaning how well the additional variables $\tilde{X}_i$, the constraints C_{m_i} and C_{p_i}, and the cost function F_i represent the lower abstraction levels. In fact, if the abstractions are not done carefully, the optimization problem solved at the i-th level may prevent the exploration of part of the design space by selecting a sub-optimal assignment of the variables in X_i.

3. Because the set of constraints C_{p_i} define the set of valid platform instances, it is possible to capture domain knowledge by restricting the class of architectures to be considered in the optimization problems. For example, it is possible to add constraints to only consider hierarchical systems divided into a primary and a secondary power distribution systems, or restrict the exploration to ring topologies only. Moreover, if the optimal configuration of some of the components is known, those design variables can be fixed in the optimization problem and treated as constants.

4. The optimization problems could in principle be automatically derived from a model-based description of the library elements. In a virtual engineering environment, the library may also contain components that do not yet exist, allowing to play "what if" scenarios and automatically compute the requirements that such components should be able to satisfy. These requirements would be provided in the form of values for the parameters of the virtual components.

These observations show the importance of understanding the structure of the design problem to build the right abstractions, and to use languages that allow to represent components and their refinements in a unified way. In this article we will show examples of how a design flow is broken into refinement steps. The definition of the right language to use is out of the scope of this article but it is a well explored and evolving research field. Many system-level design languages are available that provide the required features. Among these, Metropolis,[9,10] Rosetta,[11] Architectural Analysis and Design Language (AADL)[12] and SysML[13,14] are all good candidates for a correct-by-construction design methodology. Contrary to other methods such as DSM,[4] we do not aim at providing a way of documenting and analyzing the interactions in complex systems, but rather providing an organized design method to overcome complexity.

III. Correct-by-construction design of Electric Power Systems

In our design problem, the specification is given in terms of a set of loads together with their power and reliability requirements. The objective is to determine the architecture of an electric power system able to satisfy the demand of the loads. We start with a qualitative analysis of the main drivers of the overall system cost with the intent to partition the design decisions and define the refinement steps.

The efficiency of a generator $\eta(P, P_l)$ is a function of the the power P offered by the generator, and the total power P_l absorbed by the loads connected to it. By fitting data from a database of representative generators, it was found that the efficiency is a concave function of P_l/P meaning that the efficiency improves when the generator is fully utilized by the loads.

Observation 1. *The maximum efficiency of a power system is achieved when the rated powers of the generators are matched to the power requirements of the loads.*

The weight of a generator is a function of the rated power. The function $w(P)$ that links the power and the weight is a concave function and can be fitted well by a quadratic function. This means that in terms of watt per pound, generators with high rated power are preferred to small generators.

Observation 2. *The minimum weight of a power system is achieved by selecting generators with as high rated power as possible.*

To understand the trade off between efficiency and weight, consider the mission profile shown in Figure 1. In this simple UAV mission, the power consumption is not uniform over time. A peak in the power consumption, mainly due to the use of electric actuators during the persistence phase, can be observed. If we were to favour weight over efficiency, we would select a generator able to provide as much as $105kW$. However, this generator would be inefficient for the rest of the mission providing an efficiency of approximately 80%. If we were to favour efficiency over weight, then one choice would be to use two generators of 85 kW and 30 kW and use the smaller generator only in that phase of the mission where more power is required. In this case we would have a weight penalty of roughly 10 lb but without any loss in efficiency.

However, an additional metric to consider is the complexity of the power distribution system and the control and communication sub-systems required to manage redundancy and maintain the desired power quality. In fact, control complexity increases when generators are matched to the loads because of their limited authority in driving the voltage on the power buses. Further, increasing the number of generators would also require to increase the number of buses which has two effects: it makes the topology of the power distribution system more complex, and it increases the complexity of the state machines that control power transfers.

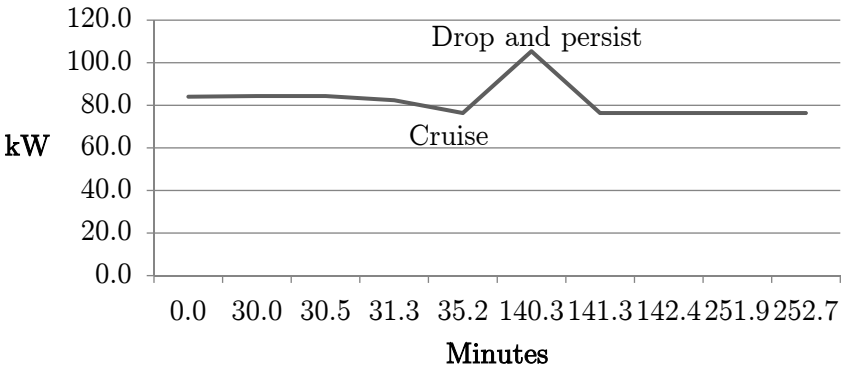

Figure 1. Power profile during a mission from take-off to landing.

Observation 3. *The costs of the power distribution system and the control system increase for more efficient electric power systems.*

Figure III shows the qualitative trade-off in the design of the electric power system. Few large generators will provide the best solution in terms of pounds per watt and in terms of the complexity of the power distribution system, denoted by *cplx*. However, many small generators will be able to deliver a very efficient solution allowing, for example, a UAV to fly longer for the same amount of fuel, while at the same time lowering the heat rejection requirements. The number of generators affects also the overall reliability of the electric power system. The Probability Loss Of Function (PLOF) decreases with the number of generators as more sources are available to power the system loads in the event of a generator failure. In order to maintain the reliability of the system above a certain value, more components need to be added and therefore the overall cost and complexity increases.

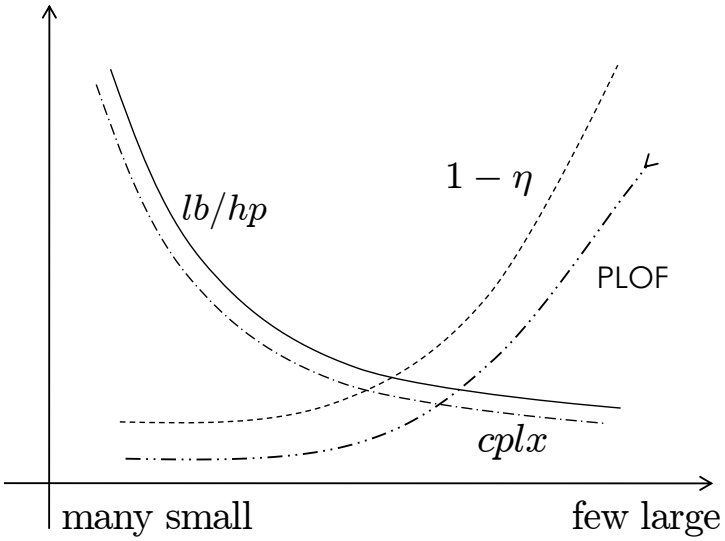

Figure 2. Trade-offs between weight, efficiency and complexity of the distribution and control systems.

From these observations, we conclude that the the selection of the number of generators and their rated powers drives the trade-off between cost and efficiency of the electric power system. It is reasonable to explore this trade-off first in the design flow. However, the cost of the power distribution system must also be taken into account. In our methodology, this objective can be achieved by including a virtual component in the library characterized by a few parameters that capture the cost and performance of the power distribution system.

Consider a power distribution sub-system that connects n generators to m loads. Because loads and gen-

erators may have different voltage interfaces, the cost of power conversion must be taken into account. This cost depends on which generator powers which load. Moreover, because of the reliability constraints imposed by the loads, each connection should provide a minimum level or reliability. The number of connections, i.e. the number of physical paths that must be provided by the topology of the power distribution system, affects the cost of the communication sub-system. Finally, the reliability levels of the loads also determine the cost of the communication sub-system as reliable connections cost more than unreliable ones. The cost model obtained from historical data shows that the weight of power conversion is a linear function of the power. Therefore, the total weight of the power conversion units is independent from the way in which loads are associated to generators. The efficiency of the power conversion units is fixed and therefore there is no trade off with weight.

Observation 4. *The cost drivers for the power distribution system are the number of generator-to-load connections and their reliability.*

Thus, the power distribution system can be abstracted by a set of parameters defining the reliability of the connections from generators to loads.

In summary, we justify the following design flow for aircraft electric power systems (depicted in Figure III):

Step 1: Generator selection . The specification is given by a representative power profile for each load together with reliability requirements. The library contains generators, loads and a virtual power distribution system. The synthesis problem is formulated as a multi-objective optimization problem that determines the size of the generators and the assignment of loads to generators such that the weight and the inefficiency of the system are minimized. Notice that the number of electrical power sources (engine driven generators, ram air turbine generator, batteries) is in general constrained by formal design rules. For example, a minimum number of power sources are required to meet safety requirements (primary flight control and cabin pressurization) and ensure high aircraft dispatch availability levels (main engine start). Furthermore the number of primary generators is almost always a multiple of the number of aircraft engines. The electrical loads are partitioned into groups based on the required power supply (28 VDC, 115 VAC, 230 VAC etc.) and the number of generation sources in use during typical operation. The power distribution system is abstracted by two set of variables: $\{y_{ij}\}$ indicating whether load i is connected to generator j, and $\{a_{ij}\}$ denoting the availability of the connections.

Step 2: Topology design . The power distribution system is refined by instantiating buses and connections among them to form an optimal topology. Variables $\{y_{ij}, a_{ij}\}$ are refined into paths in the topology. In addition to busses and contactors, power conversion devices such as transformer rectifier units (TRUs) and inverters are instantiated to ensure that the different power requirements of the loads are meet. The topology of the electrical power system distribution architecture is optimized to minimize cost (weight, inefficiency, etc) and complexity while meeting the system level reliability constraints.

Step 3: Control design . Given the topology and the paths from generators to loads, and given fault conditions of the system, a state machine can be synthesized that controls circuit breakers and tie-breakers to guarantee that critical loads are always powered.

Step 4: Embedded system design . In this last step, the control functions are implemented on a networked system that comprises a network and a set of computation resources.

The last two steps are out of the scope of this article and they will be included in our future work.

IV. Step 1 : Generator selection problem

At this abstraction level, the library provides three types of components: loads, generators and a power distribution system. Several composition rules may be associated with the platform including connection rules (generators cannot be connected to other generators, loads can be connected to generators only through the power distribution system etc.). During this design step, we enforce many of this rules by construction as it will be clear soon. However, these constraints do not disappear but are propagated down to the lower levels of abstraction (see Step 2 of the design flow in Section V).

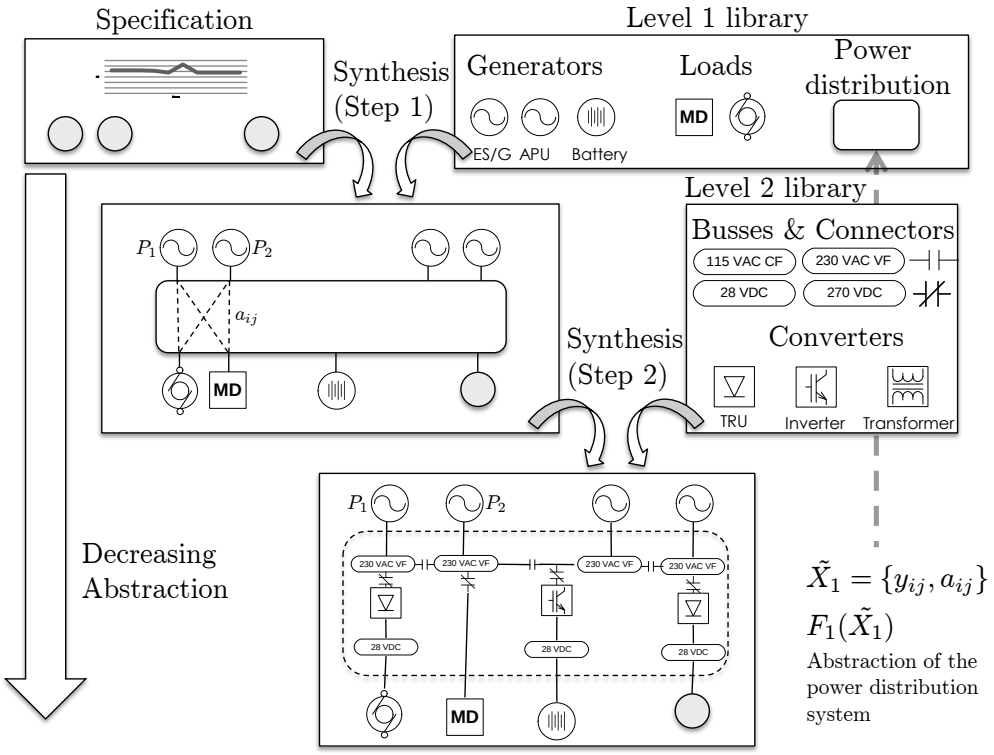

Figure 3. Detailed graphical rendition of the first two steps of the design flow.

The variables and symbols used in the definition of the optimization problem are shown in Table 1. The specification includes n loads and T mission phases. The power required by load i during phase t is denoted by $L_i(t)$. Moreover, let r_i be the reliability requirement of the i-th load. This set of variables have fixed values and capture the specification of the design problem.

Symbol	Domain	Meaning
i	$\{1, \ldots, m\}$	Load index
j	$\{1, \ldots, n\}$	Generator index
t	$\{1, \ldots, T\}$	Mission phase index
$L_i(t)$	$\mathbb{R}_{\geq 0}$	Power of load i at t
r_i	$[0, 1] \subset \mathbb{R}$	Reliability required by load i
P_j	$[0, 330e3] \subset \mathbb{R}$	Power offered by generator j
x_j	$\{0, 1\}$	Installation variable
$y_{ij}(t)$	$\{0, 1\}$	Load i connected to generator j
a_j	$[0, 1] \subset \mathbb{R}$	Availability of generator j
$a_{ij}(t)$	$[0, 1] \subset \mathbb{R}$	Availability of connection ij at t

Table 1. Symbols used in the formulation of the optimization problem.

We start by observing that the optimization problem is formulated in terms of the least constraining platform instance, meaning a platform instance with the maximum number of generators m that a designer considers appropriate for the application. An upper bound for m is n. However, not all generators will be actually used by loads, and some of them will be removed as a result of the synthesis procedure. A binary variable x_j is used for this purpose. The value of x_j is equal to one if a generator is needed, and zero otherwise. Each generator is associated with a parameter P_j which denotes the value of its rated power. We also include the virtual power distribution system as part of the platform instance to be optimized. Binary

variable $y_{ij}(t)$ is equal to one if load i is powered by generator j during phase t, while a_{ij} is the availability of the connection. The two composition rules included in C_{p_1} are the following:

$$y_{ij}(t) \leq x_j \qquad \forall\ i,\ \forall\ j,\ \forall\ t \tag{1}$$

$$a_{ij}(t) \leq y_{ij}(t) \qquad \forall\ i,\ \forall\ j,\ \forall\ t \tag{2}$$

meaning that node i can be connected to generator j only if generator j is actually present in the architecture (Constraint 1), and that the availability of a connection is zero when the connection is not active (Constraint 2)

The set of implementation constraints C_{m_1} is the following:

$$\sum_i L_i(t) y_{ij}(t) \leq P_j \qquad \forall\ j,\ \forall\ t \tag{3}$$

$$\sum_j y_{ij}(t) \geq \lambda_i(t) \qquad \forall\ i,\ \forall\ t \tag{4}$$

$$\sum_{i,j} \ln\left(1 - a_j a_{ij}(t)\right) \leq \lambda_i(t) \ln r_i \qquad \forall\ i,\ \forall\ t \tag{5}$$

where $\lambda_i(t)$ is equal to 1 if $L_i(t) > 0$ and it is equal to zero if $L_i(t) = 0$. Constraint 3 requires a generator to be able to power all loads connected to it. Constraint 4 requires that a load be connected to a generator whenever it needs power during the mission. Constraint 5 imposes that the aggregate reliability of the power sources connected to the load satisfies its reliability requirements.

The multi-objective function for this problem includes weight and inefficiency components $F_1 = (W, 1 - \eta_1(1), 1 - \ldots, 1 - \eta_m(T))$ defined as follows:

$$W = \sum_j w(P_j) x_j \tag{6}$$

$$\eta_j(t) = \eta(P_j, \sum_i y_{ij}(t) L_i(t)) x_j \tag{7}$$

In this formulation we did not consider storage elements which is part of our future work. Storage can be considered in this formulation by adding a vector of parameters $\Delta(t)$ denoting the amount of time the system spends in phase t of the mission. Energy balance constraints can the be added to the formulation. The optimization problem is mixed-integer, non-linear and multi-objective. It is therefore a hard problem to solve. In the next sections we propose some variants of the problem that can solved using standard optimization methods.

A. Problem variants

The first problem variant that we consider is to remove the dependency from variable t in the formulation of the optimization problem. Removing the time dependency has two effects. The number of decision variables is reduced by considering one configuration that satisfies either the worst case or average case scenario. The second effect is the simplification of the controllers that handle the switching of the contactors to disconnect and reconnect loads during the mission. This simplification results in a lower complexity and cost for the distribution network and software development. Together with the elimination of the variable t, it is possible to further reduce the complexity of the optimization problem by considering the reliability of connections $\{a_{ij}\}$ to be the same for all connections, say a_c. The resulting optimization problem becomes the following:

$$
\begin{aligned}
\underset{\mathbf{x},Y,\mathbf{P}}{\text{minimize}} \quad & C \\
\text{subject to} \quad & \sum_i \max_i L_i(t) y_{ij} \leq P_j && \forall\ j \\
& \sum_j y_{ij} \geq 1 && \forall\ i, \\
& y_{ij} \leq x_j && \forall\ i,\ j, \\
& \sum_j y_{ij} \ln\left(1 - a_j a_c\right) \leq \max_i \ln r_i.
\end{aligned}
$$

Perhaps, the most important abstraction that needs to be sought is one that reduces the complexity of the optimization problem coming from the cost function. Consider the rated power of generators to belong to a finite set of values $D_{P_j} \in \{p_1, \ldots, p_g\}$, $\forall j$. This will allow us to define a finite set of weight coefficients $w_h = w(p_h)$ and a set of binary variables u_{jh} that is equal to 1 if generator j has rated power equal to p_h. Therefore the total weight of the architecture can be expressed as follows:

$$W = \sum_j \sum_h u_{jh} w_h \tag{8}$$

with the additional constraints that $\sum_h u_{jh} = 1$, $\forall j$, meaning that a generator can only be of one type.

This formulation does not help in simplifying the expression of the efficiency of a generator. However, a similar approach can be followed. The total power assigned to a generator can be divided into l consecutive intervals $\hat{L}_k = [q_k, q_{k+1}]$, $k = 1, \ldots, l$, $q_1 \geq 0$, so that efficiency numbers can be precomputed as follows:

$$\eta_{jhk} = \eta(p_h, q_k) \tag{9}$$

The inefficiency of the system is the sum $\sum_{jhk}(1 - \eta_{jhk}) z_{jhk}$ where variables z_{jhk} is equal to 1 if generator j is used (i.e. $x_j = 1$), has type h and has a total load attached to it in the interval $\hat{L}_k$. Additional constraints are required to define the variables z_{jhk}. However, this procedure can be automated and the size of each interval can be defined based on the required approximation accuracy.

With this formulation, we reduced the problem to a binary problem (i.e. one where each decision variable is binary) that can be solved using standard pseudo-Boolean solvers, genetic or evolutionary algorithms.

V. Step 2: Power distribution design problem

The input to the power distribution design problem is the set of parameter values $\{y^*_{ij}\}$ and $\{a^*_{ij}\}$ (i.e. the value found as solution to the optimization problem defined in Section IV), together with the specification used as input to the generator selection problem. Topology design is a known problem and can be formulated as a multi-commodity flow problem. However, we will see that a pre-processing step is needed to guarantee that the controller design problem (Step 3 not explored in this paper) is feasible[a].

Consider a set of nodes $V = G \cup L \cup B$ in the architecture of the electric power system that comprises a set G of $m^* \leq m$ generators from Step 1, a set L of n loads, and a set B of b buses, where b is an upper bound on the number of buses in the system. Further, the set of loads G is partitioned in the set of AC loads L_{AC} and DC loads L_{DC}. Similarly, the set of buses is partitioned in the set of AC buses B_{AC} and the set of DC buses B_{DC}. For $u, v \in V$, let the binary variable e_{uv} be equal to 1 if node u is connected to node

[a]Recall that from the discussion in Section I, we must ensure $\cap_{i=1}^{L} C_i \subseteq C$

v and 0 otherwise. The following composition rules must be considered in the definition of C_{p_2}:

$$e_{uv} = 0 \qquad \forall u, v \in G \tag{10}$$
$$e_{uv} = 0 \qquad \forall u, v \in L \tag{11}$$
$$e_{u_1 v} + e_{u_2 v} \leq 1 \qquad \forall u_1, u_2 \in G, u_1 \neq u_2, \forall v \in B \tag{12}$$
$$e_{uv} = 0 \qquad \forall u \in G, v \in L \tag{13}$$
$$e_{uv} = 0 \qquad \forall u \in L_{DC}, v \in B_{AC} \tag{14}$$
$$e_{uv} = 0 \qquad \forall u \in L_{AC}, v \in B_{DC} \tag{15}$$
$$e_{uv} = 0 \qquad \forall u \in G, v \in B_{DC} \tag{16}$$
$$\tag{17}$$

These constraints impose that generators cannot be connected to generator; loads cannot be connected to loads; generators cannot be connected directly on the same bus; generators cannot be directly connected to loads; DC loads cannot be connected to AC buses; AC loads cannot be connected to DC buses; and generators cannot be connected to DC buses.

To define the implementation constraints C_{m_2} we introduce the notion of a path in the power distribution system. Consider a set of connectivity requirements $F \subseteq \{(i,j) \in L \times G | y^*_{ij} = 1\}$ between generators and loads. For a requirement (i,j), let π_{uvij} be a binary variable that is equal to 1 if the path from i to j uses the connection from u to v. Obviously, the following must hold: $\pi_{uvij} \leq e_{uv}$, $\forall u, v \in V, (i,j) \in F$. A unique path exists between generator j and load i if and only if the following conditions are satisfied:

$$\sum_{v \in V} \pi_{jvij} = 1 \tag{18}$$
$$\sum_{v \in V} \pi_{uiij} = -1 \tag{19}$$
$$\sum_{u \in V} \pi_{uvij} = \sum_{u \in V} \pi_{vuij} \tag{20}$$

The reliability provided by a path must satisfy the following constraint:

$$\sum_{u,v \in B} (\ln a_{uv} + \ln a_u) \pi_{uvij} \geq a_{ij} \tag{21}$$

where a_{uv} is the availability of a connector (e.g. a TRU, power converter, contactor), and a_u is the availability of a bus.

The cost function is a multi-objective function that takes into account the weight and the inefficiency of the power distribution system. Both these functions depend on the set $E = \{e_{u,v}\}$ of connectors instantiated in the architecture, the number of buses used by the power distribution system and and the number of buses crossed by paths from source to destination. Thus, an optimization algorithm that solves this optimization problem will provide an architecture with the least amount of buses and connections, and with the shortest path possible. This is no surprise and it is in accordance with standard architecture where power distribution systems are organized into a two level hierarchy.

However, in this formulation, we have not considered the role of failures and the fact that not all paths are active at the same time. In fact, the result of the synthesis problem from Step 1, may require the same load to be powered by more than one generator to satisfy reliability constraints. This set of generators are not connected to the load at the same time, otherwise they would be also connected to each other violating on of the constraints of our platform. For this reason, the power distribution synthesis step must be preceded by a partitioning algorithm that generates sub-sets of the connectivity requirements $Y = \{y_{ij}\}$ from Step 1 under fault conditions. This problem can be cast into a bin packing problem that aims at generating one sub-set $Y_F \subset Y$ for each fault condition such that all loads are powered and generator efficiency is maximized. The power distribution system design can then be formulated as an optimization problem with the additional constraint that for each pair of generators, the paths departing from them be disjoint. This condition will guarantee that a contactor configuration can be found so that generators never share the same bus at the same time. The result of Step 2 can then be used to synthesize a state machine that handles power transfers of the electric power system.

VI. Conclusions and future work

In this article we presented a formalization of the design exploration activity for complex systems in the context of the platform-based design methodology. The methodology is general and the advantages are numerous as it allows correct-by-construction design, thereby reducing the verification effort, and allows to explore large design spaces to improve optimality. However, the major challenge to overcome for a successful adoption of the methodology is the articulation of the design flow into refinement steps such that the complexity of the design exploration problem is contained while maintaining the optimality of the result. These process also requires to understand the structure of the problem and to build abstractions of the system components to be exported at the highest level of the the design flow to make informed decisions in the early stage of the design.

We used this driving principles in setting up a design flow for aircraft electric power systems. We present two refinement steps: generator selection and topology design. For these two steps, we also formulated the synthesis problems together with ways of dealing with their complexity.

We plan to extend our work in two directions. First, we plan to include storage elements in our library. These elements can be used not only to guarantee safety, but also to store energy that may be regenerated by actuators. Second, we plan to expand the approach to capture behavioral properties of the system such as power quality. This second extension include the automatic synthesis of discrete controllers used to command switches in the system, as well as continuous controller to guarantee the required power quality on each of the buses.

VII. Disclaimers and Acknowledgments

The development of the methods and results contained in this study were sponsored under the DARPA contract: "Abstraction Based Complexity Management" #FA9550-07-C-0024.

The views, opinions, and/or findings contained in this article are those of the authors and should not be interpreted as representing the official views or policies, either expressed or implied, of the Defense Advanced Research Projects Agency or the Department of Defense.

References

[1]Sangiovanni Vincentelli, A., Carloni, L., De Bernardinis, F., and Sgroi, M., "Benefits and Challenges for Platform-Based Design," *Proceedings of DAC*, June 2004, pp. 409–414.

[2]Moir, I. and Seabridge, A. G., *Aircraft Systems: Mechanical, Electrical and Avionics Subsystems Integration*, AIAA Education Series, 2008.

[3]IBM, "Rational DOORS," .

[4]Steward, D. V., "The Design Structure System: A Method for Managing the Design of Complex Systems," *IEEE Transactions on Engineering Management*, Vol. 28, No. 3, 1981, pp. 71–74.

[5]Browning, T. R., "Applying the design structure matrix to system decomposition and integration problems: a review and new directions," *Engineering Management, IEEE Transactions on*, Vol. 48, No. 3, August 2002, pp. 292–306.

[6]Simmons, W. L., *A framework for decision support in systems architecting*, Ph.D. thesis, Massachusetts Institute of Technology, 2008.

[7]Pinto, A., Bonivento, A., Vincentelli, A. S., Passerone, R., and Sgroi, M., "System-Level Design Paradigms: Platform-Based Design and Communication Synthesis," *ACM Trans. on Embedded Computing Systems*, Vol. 5, No. 5, May 2006.

[8]Pinto, A., Carloni, L. P., and Vincentelli, A. L. S., "A Methodology for Constraint-Driven Synthesis of On-Chip Communications," *IEEE Transactions on Computer Aided Design*, Vol. 29, No. 3, March 2009.

[9]Balarin, F., Watanabe, Y., Hsieh, H., Lavagno, L., Passerone, C., and Sangiovanni-Vincentelli, A., "Metropolis: An Integrated Electronic System Design Environment," *Computer*, Vol. 36, 2003, pp. 45–52.

[10]Davare, A., Densmore, D., Meyerowitz, T., Pinto, A., Sangiovanni-Vincentelli, A., Yang, G., Zeng, H., and Zhu, Q., "A Next-Generation Design Framework for Platform-Based Design," *Conference on Using Hardware Design and Verification Languages (DVCon)*, 2007.

[11]Alexander, P., *System Level Design with Rosetta (Systems on Silicon)*, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2006.

[12]Aerospace, S., *Architecture Analysis and Design Language (AADL)*, SAE, January 2009.

[13]OMG, *OMG SysML v. 1.1*, November 2008.

[14]Weilkiens, T., *Systems Engineering with SysML/UML: Modeling, Analysis, Design*, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2008.