|  |
| --- |
| Force Strategy Division |
| Marathon 3.1415926535897932384Primer |
| Working Draft |
|  |
| **Tom Spoon** |
|  |

|  |
| --- |
|  |

# Introduction

The purpose of this paper is to provide a basis for understanding Marathon, and to shore up the argument for (somewhat drastic) changes to its design. The paper starts with a conceptual frame of reference for understanding Marathon. Using this frame of reference, the paper identifies inherent sources of complexity within the problem domain, and some proposed design philosophies that attempt to mitigate complexity. A discussion of Marathon in-theory follows, in which the major abstractions and “muscle movements” inherent to any Marathon implementation are enumerated. Finally, the reader will delve into implementations of Marathon, beginning with the current object-oriented implementation, and exploring how Marathon will look under functional and distributed computing paradigms.

**What is Marathon?**

Marathon is a mechanism for analyzing the effects of Army supply, demand, and policy variations, where supply is a set of potentially deployable units, demand is a set of activities requiring a unit, and policy is a collection of rules or constraints that determine a unit’s ability to fill a demand. As a design goal, Marathon seeks to validly simulate the physics of Army supply and demand, governed by policy, to analyze both the general behavior of such systems and the specific effects relative to changes in supply, demand, or policy. Ultimately, Marathon is an analytic sandbox for evaluating courses of action relative to the Army Force Generation domain.

**Army Force Generation**

Army Force Generation is a system for managing readiness, the ability for units to deploy to meet contingencies. In general, force generation is the structured progression of increased unit readiness over time, resulting in the periodic availability of trained, ready and cohesive units prepared for operational deployment in support of civil authorities and combatant commander requirements. The domain of Army Force Generation is enormous, encompassing the range of processes and resources necessary to man, equip, train, deploy, and sustain the Army’s supply of units.

Out of necessity, Marathon focuses on a subset of the Force Generation process, and generally holds many gross assumptions about the behavior of quite complex subsystems (such as training processes, manning, equipment, mobilization, etc.) Even with the Force Generation domain scoped to the unit level of detail[[1]](#footnote-1), and with complex subsystems like equipping and manning abstracted away, the variety of supply, demand, and policy options is still staggering.

**ARFORGEN**

In addition to the breadth of the force generation domain, the policies for managing the force generation process have historically ranged from various flavors of Tiered Readiness to a contemporary notion of Cyclical Readiness.[[2]](#footnote-2) Marathon was initially designed to analyze ARFORGEN, the Army’s contemporary cyclical force generation process. ARFORGEN seeks to synchronize individual sourcing, manning, equipping, and training processes to ensure a continuous supply of forces, and transitions the Army from a system focused on surging forces to war to a system for sustained operations. The goal of ARFORGEN is to provide a consistent and predictable supply of units ready to deploy to meet contingencies.

Under ARFORGEN, units constantly[[3]](#footnote-3) increase readiness as a function of time in a cycle. At the beginning of a unit’s ARFORGEN lifecycle, the Reset phase, the unit lacks equipment, personnel, and training, leaving it at the lowest relative state of readiness. Units accumulate resources and training as they accumulate time in the lifecycle, which propels them through the abstract ARFORGEN phases: Reset, Train, Ready, and Available. In the available phase, units are at the highest level of readiness, and are best suited to deploy. After the Available phase, units begin a new lifecycle, losing equipment, personnel, and readiness as they transition to the Reset phase. In theory, distributing units uniformly across ARFORGEN lifecycles – an ideal ARFORGEN state – simultaneously enforces equal opportunity for training, deploying, and resetting across the unit supply, and ensures a static (consistent) supply of units in each ARFORGEN phase.

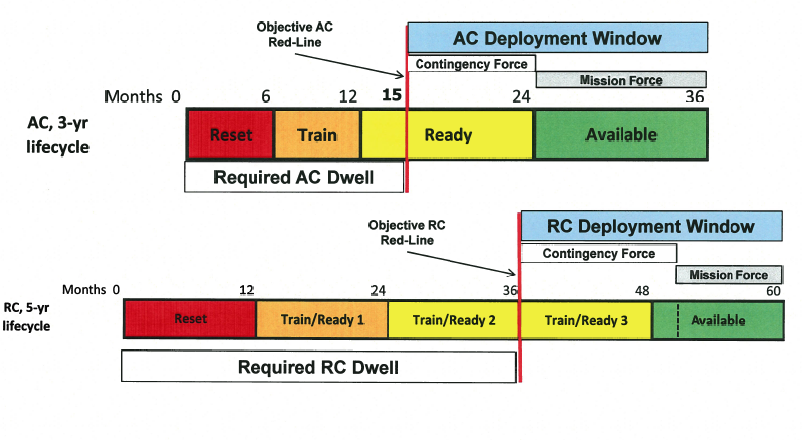
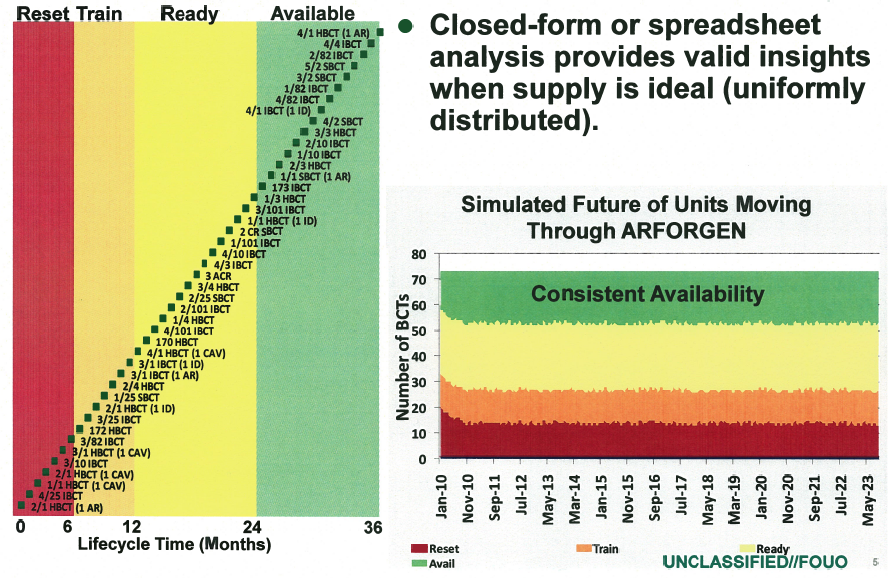


Figure : Notional ARFORGEN lifecycles for the Active Duty and USAR/ARNG

One should note that ARFORGEN is only one system of managing readiness, and that alternatives, such as Tiered Readiness, do exist. The relative value of a force generation system is affected by resource constraints (namely budget and end-strength), the magnitude of demand, and the duration of demand. ARFORGEN was enacted in response to the protracted wars in Iraq and Afghanistan, and under the auspices of large budgets and congressional mandates to grow structure. ARFORGEN may not be feasible or desirable under different circumstances.

**Static Rotational Analysis and Ideal ARFORGEN**

Rotational analysis is the notion (circa 2003-2012+) of units deploying for a fixed interval to fill demands, with a supply system producing new units to backfill demands under a cyclical readiness policy. This typically equates to an ARFORGEN force generation process.[[4]](#footnote-4) Fortunately, ARFORGEN (in the ideal state) is amenable to analysis due to its static nature. In the ideal state, units are evenly distributed across ARFORGEN lifecycles: 

Units are not deployed “out of cycle”, which keeps the quantity of units in each ARFORGEN pool constant (or static). The **theoretical capacity of the supply** – the expected number of units available at any time, relative to a total rotational supply and a unit lifecycle - can be calculated by multiplying a rotational discount by the total number of units in the supply. The **rotational discount** is a dimensionless quantity, ranging between 0 and 1, which represents the expected proportion of a unit’s available time relative to lifecycle time. Summing the factors that increase available time (e.g. time in the available phase) or decrease available days (e.g. time required to mobilize) provides a numerator, while the denominator is the duration of a lifecycle.



Theoretical capacity is built upon ideal assumptions like perfect availability, and does not reflect the complexities of unit histories, transformation periods, changes to rotational policy, and lifecycle sequencing. **Static rotational analysis (or static analysis)** examines force generation through the calculation of theoretical capacities and variable rotational discounts, where idealized assumptions guarantee the system is in a static, uniformly distributed state. Static analysis is a highly useful tool for determining feasibility and providing upper bounds on the capacities of various ARFORGEN schemes.

**Dynamic Rotational Analysis and Real ARFORGEN**

Ideal ARFORGEN has yet to be achieved[[5]](#footnote-5), let alone maintained, so static rotational analysis cannot validly answer questions about real ARFORGEN, where the state of supply, demand, and policy is dynamic or changing. **Dynamic rotational analysis (or dynamic analysis)** accounts for changes in the force generation system and non-ideal states, to bridge the gap between theoretical ideal and the empirical reality. Dynamic analysis enables observation of supply, demand, or policy scenarios that vary as a function of time or event, and often illuminates unforeseen consequences via second and third order effects.[[6]](#footnote-6)

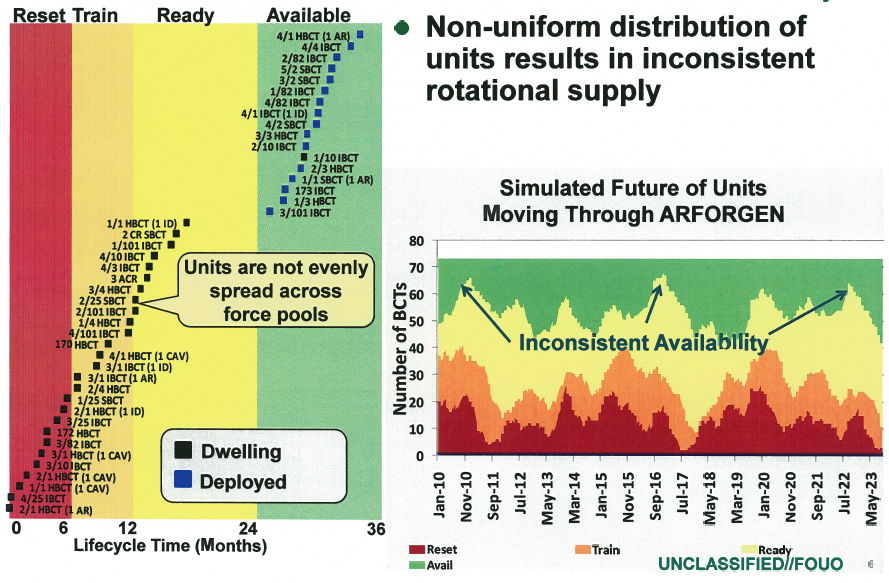


Figure : Reality is not ideal.

Dynamic analysis utilizes some form of simulation, optimization, or difference equations to generate multiple sequentially-dependent states of the rotational system. The sequence of states resulting from the application of a transition function to an initial state forms the “history” of the system. System history is the primary output, and serves as the basis for multiple forms of focused analysis. Incorporating dynamics allows for a more nuanced and contextual analysis (particularly when analyzing temporal phenomena[[7]](#footnote-7)). 

Figure : A possible workflow for dynamic analysis.

CAA created Marathon to analyze dynamic force management under ARFORGEN. Marathon performs dynamic analysis through a combination of discrete-event simulation and on-line optimization[[8]](#footnote-8). Specifically, Marathon simulates the deployment of rotational operating force units to meet operational demands over time, based on specified ARFORGEN rotation policies. Marathon illuminates the differences between theoretical rotational capacities, and rotational capacities stemming from non-static system conditions, such as historic deployments and surges, and dynamic changes in supply, policy, or demand.

**How Does Marathon Work?**

Marathon simulates the force generation process through a coordinated set of supply, demand, and policy simulations.[[9]](#footnote-9) The supply system acts as a coordination point for polling unit availability, a dissemination channel for simulation supply events, and a general container of units. Thousands[[10]](#footnote-10) of unique unit entities follow rotational policies that are either global (shared) or local (unique to the unit), and are directed by one or more supply systems to execute the “supply physics” dictated by the corresponding policy. Each unit’s simulated history can be traced, recorded, and reacted to within the simulation ecosystem.

Unit rotational policy generally consists of a directed sequence of states and durations.[[11]](#footnote-11) Units also have a behavior, which interprets policy to implement the desired supply-side and deployed actions. Policies are entirely modular and variable, as are individual unit behaviors. The decoupling of behavior and policy allows for both homogenous sets of units that appear to behave identically, as well as a diaspora of independent singletons that can apply similar behavior to different policies or interpret the same policies (via different behavior) to simulate radically different populations. The potential for unique entities allows Marathon to flexibly and modularly account for the legion of subtleties and corner-cases in the force generation problem domain.[[12]](#footnote-12)

Demands are activated, and slated for filling, based on a - potentially sophisticated - user-defined priority function. A fill system matches the highest priority demand to the most suitable supply as needed, and directs the transition of units from the supply system to deployments or other states. The fill system also accounts for potentially complex unit substitution rules, demand preferences, and almost any value function associated with the selection of units to fill demands.

Finally, a policy system accounts for changes to policy (such as ARFORGEN suspension, variation in lifecycle length, and changes in deployment time) by enacting system-wide policy changes in response to either time or event.[[13]](#footnote-13) Policy changes automatically filter down to subscribing units, enabling a rich and diverse simulation of the supply-policy-demand dynamics.

1. Rather than the individual soldier. [↑](#footnote-ref-1)
2. According to Rand (citation needed). [↑](#footnote-ref-2)
3. We assume constantly, but the increase in readiness is more closely related to a step function of varying step width, which is validly approximated linearly. [↑](#footnote-ref-3)
4. ARFORGEN is the institutional model for force generation, although hybrids and alternative models are being examined. [↑](#footnote-ref-4)
5. As of Jan 2012. [↑](#footnote-ref-5)
6. One consequence is that ARFORGEN is dynamically unstable, and will not tend toward an ideal state without control. [↑](#footnote-ref-6)
7. Temporal questions are incredibly common, if not mandatory, in most force generation analysis. [↑](#footnote-ref-7)
8. Optimization performed iteratively in response to events within the simulation, usually used for dynamic control. [↑](#footnote-ref-8)
9. For purposes of abstraction, the three simulations are viewed as completely independent, although the current implementation subordinates them to a single, overarching simulation engine. [↑](#footnote-ref-9)
10. Perhaps millions in the very near future….if for some reason it’s needed. This is currently a limitation of the VBA host platform. [↑](#footnote-ref-10)
11. Policies are intended to look structurally similar to the ARFORGEN policy depicted earlier, but they can be entirely different as well. For some units, the structure of the state transitions intentionally looks nothing like an ARFORGEN cycle, and may not even be cyclical in nature. The policy system is truly general. [↑](#footnote-ref-11)
12. Corner cases and unanticipated changes imposed by sponsor requirements invalidated Marathon 2 and the initial VBA port. Unable to adapt, the designs had to be abandoned. [↑](#footnote-ref-12)
13. This is somewhat circular, since time is technically an event… [↑](#footnote-ref-13)