Feasibility of a 5x Speedup in System Development due to META Design

The MIT Faculty has made this article openly available. Please share how this access benefits you. Your story matters.

<table>
<thead>
<tr>
<th>Citation</th>
<th>De Weck, Olivier L. “Feasibility of a 5x Speedup in System Development Due to META Design.” Volume 2: 32nd Computers and Information in Engineering Conference, Parts A and B (August 12, 2012).</th>
</tr>
</thead>
<tbody>
<tr>
<td>As Published</td>
<td><a href="http://dx.doi.org/10.1115/DETC2012-70791">http://dx.doi.org/10.1115/DETC2012-70791</a></td>
</tr>
<tr>
<td>Publisher</td>
<td>ASME International</td>
</tr>
<tr>
<td>Version</td>
<td>Final published version</td>
</tr>
<tr>
<td>Accessed</td>
<td>Mon Feb 04 04:23:13 EST 2019</td>
</tr>
<tr>
<td>Citable Link</td>
<td><a href="http://hdl.handle.net/1721.1/116271">http://hdl.handle.net/1721.1/116271</a></td>
</tr>
<tr>
<td>Terms of Use</td>
<td>Article is made available in accordance with the publisher’s policy and may be subject to US copyright law. Please refer to the publisher's site for terms of use.</td>
</tr>
</tbody>
</table>
FEASIBILITY OF A 5X SPEEDUP IN SYSTEM DEVELOPMENT DUE TO META DESIGN

Olivier L. de Weck
Massachusetts Institute of Technology
Cambridge, MA 02139

ABSTRACT
The main goal of META design is to achieve a factor five (5x) improvement in product development speed for cyber-electro-physical systems compared to current practice. The method claims to achieve this speedup by a combination of three main mechanisms:

1. The deliberate use of layers of abstraction. High-level functional requirements are used to explore architectures immediately rather than waiting for downstream level 2,3,4 … requirements to be defined.

2. The development and use of an extensive and trusted component (C2M2L) model library. Rather than designing all components from scratch, the META process allows importing component models directly from a library in order to quickly compose functional designs.

3. The ability to find emergent behaviors and problems ahead of time during virtual Verification and Validation (V&V) and generating designs that are correct-by-construction allows a more streamlined design process and avoids costly design iterations that often lead to expensive design changes.

This paper quantifies the impact of these main META mechanisms with a sophisticated System Dynamics (SD) model that allows simulating development projects over time. META compares favorably against a simulation of a traditional design process due to the generation of late engineering changes in a traditional design-build-test-redesign environment. The benchmark case analyzed in this paper contained 3,000 requirements, and the results show a dramatic improvement for project completion schedule with a demonstrated speedup factor of 4.4 (70 months versus about 16 months). In the simulated META process we used 3 layers of abstraction, 50% novelty and a model library integrity of 80% with 70% of problems are caught early. The results were also validated against data from the B777 Electric Power System (EPS) design project at UTC where a speedup factor of 3.8 was demonstrated. The paper contains a useful sensitivity analysis that helps establish requirements and bounds on the META process and tool-chain itself that should enable the desired 5x speedup factor.

INTRODUCTION
It is well documented that the time and cost for developing major cyber-electro-mechanical systems in the defense and aerospace industries has been increasing sharply in recent decades. Examples of large scale programs that have come in significantly over schedule and cost are listed in Table 1.

Table 1: Examples of cost and schedule overruns for major defense and aerospace projects (to be completed)

<table>
<thead>
<tr>
<th>Program</th>
<th>Cost</th>
<th>Actual $</th>
<th>Schedule</th>
<th>Actual</th>
</tr>
</thead>
<tbody>
<tr>
<td>JSF</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>other</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

A recent report by the National Research Council (NRC) analyzed a set of about 80 Earth and Space observing missions and found that on average they were YY% overbudget and late. There is increasing evidence to suggest that this trend is mainly due to a collective inability to predict and manage the impact of product complexity on the design process. Interestingly other industries such as semiconductors and to some extent the automotive industry have been able to maintain shorter development cycles, despite increasing complexity. One of the major reasons in addition to increasing competition is the continued investment in abstraction-layer based design.

The META program’s goal is to develop and mature such a process for complex cyber-electro-mechaical systems such as aircraft, ships, ground vehicles etc…

This paper therefore starts from the following premise and asks the following questions:

- The main goal of META is to achieve a 5x improvement in product development speed over current practice
- Current practice is based on a sequential stage-gate approach with requirements definition, (little) architecture exploration, specification design, verification and validation (4).
- Current practice does not reuse much design knowledge and no (or few) component libraries between programs
One of the major reasons for delay is discovery of errors and rework late during integration and test.

There is no (or little) deliberate use of layers of abstraction and contracts between layers during design and only limited ability to manage design artifact complexity.

META intends to use a powerful combination of layered design with overlapping activities, virtual verification and validation and certificates of correctness, specifically it aims at:

- Faster design due to extensive model libraries (C2M2L)
- Layers of abstraction coupled with extensive system architecture exploration
- Lower rates of rework due to early verification and validation and “correct by construction” system synthesis

This paper attempts to answer the following questions:

1. To what extent is the main goal of META, the achievement of a 5x speedup, realistic?
2. Which of the META design process factors are the most important and how sensitive are schedule and cost to these factors?
3. Can we calibrate the model against past experience to predict future performance and speedup factors on real programs?

To answer these questions the paper is structured as follows. In the next section we develop a System Dynamics (SD) model that attempts to capture the dynamics of current practice (linear, waterfall-type) projects as well as META-enabled projects. Next the model is exercised for a hypothetical program that seeks to develop a complex cyber-electro-mechanical system that satisfies a set of 3,000 requirements. Next we conduct a sensitivity analysis to find which META factors are most influential for speedup. This is followed by an application of the META System Dynamics model to the development of the Boeing 777 Electrical Power System (EPS) to observe what kind of speedup META might have been able to achieve in that program. We conclude by summarizing our findings and recommendations for future work.

SPEEDUP RATIONALE

We illustrate the main process differences between traditional design, as exemplified by the conventional product development process flow in MIL-STD-499, and META in Fig.1. The total design time for a conventional program, $T_C$, is:

$$T_C = T_1 + T_2 + T_3 + T_4$$  

and is comprised of requirements definition ($T_1$), concept design ($T_2$), preliminary design, detailed design and system integration ($T_3$) and system validation ($T_4$). Note that the time required for manufacturing post the prototype stage and initial deployment is not accounted for here and is assumed to be similar for Conventional and META design. The vision for META is that the following relationships will hold true:

$$T_{META} = T_1 + \frac{T_3}{10} + \frac{T_4}{5}$$  

and

$$5 \cdot T_{META} \leq T_C$$  

Whereby the speedup factor (sf) of 5 is achieved with equal or superior results in terms of system performance and requirements satisfaction.

![Fig.1: Conventional (left) vs. META Design Flow (right)](image)

The main differences between Conventional Design and META Design are as follows:

- Conventional Design proceeds linearly and spends relatively little time during Concept Design ($T_2$)
- Most of the time in Conventional Design is spent during $T_3$, whereby System Integration consumes a larger portion of time due to discovery of emergent behaviors and generation of design changes
- META Design combines Concept, Preliminary, Detailed Design and System Integration and replaces it with Design Space Exploration at increasing levels of fidelity (Layers 1, 2, …N)
- System Validation is accelerated ($0.2T_4$) because it serves mainly the purpose of confirmation, rather than problem discovery

Table 2 gives an example for representative times in Eq. (1) and Eq. (2) such that the inequality in Eq. (3) is satisfied:

Table 2: Representative Times for Conventional versus META-Design Processes
Thus, the main time-savings are expected in the area of preliminary and detailed design and especially in system integration. The next section develops a System Dynamics Model to quantify and simulate both the conventional and the META Design Process.

META SYSTEM DYNAMICS MODEL

System Dynamics is a well established modeling technique for capturing the non-linearities and evolution of complex systems over time (5). Key elements of a system dynamics models are causal loops between variables describing the system as well as the notion of stocks and flows which are influenced by reinforcing and balancing loops. Figure 2 shows an initial version of the META System Dynamics Model that was developed and presented as part of the META Program (2010-2011). The model contains the following key stocks and flows that represent the accumulation and depletion of key quantities during system development:

- Requirements elicitation leads to defined requirements
- Concept exploration leads to explored architectures
- Design and integration lead to completed system specifications
- Verification activities lead to tests performed
- Validation activities lead to validated requirements

As discussed earlier, in an ideal world, all activities would proceed from start to finish and a certificate of completeness could be issued once all the defined requirements had been validated. In reality design changes are generated throughout the development effort due to exogenous requirements changes (not modeled here), discovery of errors and oversights etc (6). Thus an important additional flow that is included in the model is the generation and resolution of engineering changes as well as the accumulation of Non-recurring engineering (NRE) costs during the project:

- Change generation creates pending engineering changes for subsequent implementation or disposition. The model contains mechanisms for basic Change Management as describe in (6). Note that some changes simply lead to redesign, while other changes may also impact the original requirements set. The ability to generate changes in the model is set via the binary “Change Flag”.
- System engineers, designers and testers working on the program leads to a spending rate which accumulates to the total Non-Recurring Engineering (NRE) cost.
Beyond these two features, the other items shown in red in Figure 2 highlight features of the model that were introduced in order to enable the characteristics of the META design process in the process model. These are as follows:

- **META Flag (on/off)**: a binary flag that allows to turn on or off the mechanisms of the META design approach. If the flag is set to zero, the model simulates a traditional waterfall design process according to MIL-STD-499.

- **Levels of Abstraction**: rather than waiting until all requirements have been completely defined, in META conceptual exploration can begin once all requirements at that level of abstraction (1, 2,...N) have been defined. The number of layers of abstraction is defined by the parameter designated as “cognitive bandwidth” (7) and the total number of requirements that must be satisfied.

- **A measure of Structural Complexity** is introduced to capture the impact of system complexity on productivity (the more complex the less productive the workforce will be) and change generation (more complexity yields more changes). Exploring more architectures is beneficial in reducing structural complexity and finding simpler designs that will satisfy the same set of requirements.

- **Model Library**: in the case that the META flag is set to 1, a model library becomes available. This model library may not be complete and may only have partial coverage (e.g. model library coverage of 0.8 means that there are existing models available for 80% of all components in the final design). In META designers can import these models rather than having to generate all component level information from scratch.

- **Certificate of Completion**: A certificate of completion will be issued when a prescribed fraction (e.g. 99.5%) of all requirements have been validated and when there are no pending changes.

This initial version of the model was presented to the META performer community between May and July 2011 and as a result of the feedback received it was decided to implement a set of changes to the initial META process model. These changes are shown in Figure 3 in green and are summarized as follows:

- **Schedule Pressure** is a way that projects can be accelerated with or without availability of a META design process and tool-chain. Management exerts schedule pressure through typical actions such as overtime, hiring extra people, checking more often on progress etc… We assume that schedule pressure acts as a multiplicative speedup factor with respect to the rate at which work is being done in the various project phases (e.g. design and integration, testing etc…). There is a price to pay for speedup which is typically a higher burn rate ($/month) as well as some inefficiencies...
Novelty is an important concept as described by Carlile (8). Depending on how new a system design challenge/project is for a particular organization and design team there will be more or less novelty. We calculate novelty as 1 minus the existing (C2M2L) model library coverage. If the coverage is 100% then all components that are needed can simply be pulled from the library and we assume there is no novelty. If library coverage is 50%, then novelty is also 50% meaning that half of the components and related interactions have to be designed from scratch and are therefore new. A high degree of novelty affects a project in two ways. First it decreases productivity because the team is not as efficient and has to learn. Second the more novelty, the more chance for errors and discovery of unexpected (emergent) behavior. This novelty increases the change generation rate.

The third new feature of the model is the (C2M2L) Model Library Integrity. For example if integrity is set to 80% it means that 20% of the components that are stored in the library have some error or bug in them which has to subsequently be repaired and corrected. This impacts the change generation and implementation rate of the project.

The revised and final model is shown in Figure 3. This model is used in the subsequent section to analyze the impact of META on a design project which has to satisfy 3,000 design requirements.

PROJECT SIMULATION RESULTS

Figure 4 shows the results from running the model for three distinct cases:

a) Green: An idealistic conventional project (with no change generation allowed) according to MIL-STD-499
b) Red: A realistic conventional project with change generation and change “ripples” occurring as in the projects described by the NRC and GAO reports
c) Blue: A project with META design mechanisms enabled

This comparison of an ideally planned project (green) where no changes or error are assumed versus a realistic project with changes (“turn-backs”) shown in red versus a META enabled project in blue. This project assumes a baseline of 3,000 individual requirements that have to be satisfied. The following assumptions were made:

Simulation Assumptions:
All: Schedule Pressure = 1.5
META: 3-layers of abstraction (CB=9)
META: C2M2L library coverage: 50%
META: Novelty: 50%

META: C2M2L library integrity: 80%
Problems caught early: 70%

The results in Figure 4 show a dramatic improvement for project completion schedule with a demonstrated speedup factor of 4.4 (70 months versus about 16 months). Here the schedule pressure was set to 1.5 and in META we use 3 layers of abstraction, 50% novelty and a library integrity of 80% and 70% of problems are caught early.

The simulation of Non-Recurring Engineering Effort ($) shows that the improvement relative to the realistic project with changes is only 1.5. This is because work is done at a higher rate and more people are needed, despite the automation provided by META. It is also important to note that the META project is more expensive than the ideally planned project where no changes (turn-backs or iterations) are accounted for. Thus META’s main strengths is speeding up the design process, not saving development money.

The simulation of Non-Recurring Engineering Effort ($) shows that the improvement relative to the realistic project with changes is only 1.5. This is because work is done at a higher rate and more people are needed, despite the automation provided by META. It is also important to note that the META project is more expensive than the ideally planned project where no changes (turn-backs or iterations) are accounted for. Thus META’s main strengths is speeding up the design process, not saving development money.

<table>
<thead>
<tr>
<th>Simulation Case</th>
<th>Schedule to complete</th>
<th>NRE $ to complete</th>
</tr>
</thead>
<tbody>
<tr>
<td>Idealistic Project</td>
<td>42.25 months</td>
<td>$27.9M</td>
</tr>
<tr>
<td>Realistic Project w/changes</td>
<td>70 months</td>
<td>$51.9M</td>
</tr>
<tr>
<td>META-enabled project</td>
<td>15.75 months</td>
<td>$31.5M</td>
</tr>
</tbody>
</table>

Figure 4: Project Simulation Results with a conventional idealistic project (green), realistic conventional project (red) and META-enable project (blue).

Another important finding is that these benefits appear to be achievable by META, despite the fact that the tool-chain and META process is not perfect. I.e. the model library coverage of 50% with 80% integrity and ability to detect problems early (70%) were sufficient to generate the 4.4 speedup factor observed in Figure 4. This raises the question of how sensitive the speedup factor is to the META model assumptions.

SENSITIVITY ANALYSIS

We took the baseline model (assumed baseline numbers are shown in Fig. 5 on the left in bold) and varied each of the 7 META factors up (+) and down (-). For each one-at-a-time perturbation of each factor we simulated the project and recorded the impact on project schedule and cost. Recall that the baseline META model took 15.75 months (Fig. 4) to
complete at a NRE $ cost of $31.5 million, while the traditional project (w/o) META took 70 months to complete at and NRE cost of $51.9 million.

The sensitivity results are normalized and shown in Fig. 5, so for example a value of -40% means that if the factor is increased by 100% (doubled) then the impact on cost or schedule will be a 40% reduction. First we notice that most sensitivities are negative. Increasing the number of layers of abstraction from 2 to 3 leads to a significant reduction in schedule, as does applying schedule pressure. Also increasing model library coverage and fractions of problems caught early by the META tool-chain leads to speed-up. Increasing the architecture exploration rate and maximum number of architectures that can be carried does not lead to a speedup (but improvement of technical performance and slight budget reduction).

This sensitivity analysis leads to the following conclusions:

- Working in layers of abstraction is extremely important for speedup. There seems to be a sweet-spot around 3 layers of abstraction (for a system with about 1,500-3,000 requirements), adding more layers does not yield much more benefit.

- The C2M2L model library is very important for both schedule and cost. The completeness of the library (“coverage”) seems to be a more important factor than ensuring that it is completely error free. In other words for META it appears to be better to have a near complete library even if it has bugs in it. We suspect that the reasons that model library errors are not that important is because the rest of the META tool-chain is able to find and resolve such errors quickly.

- The ability of the META tool-chain to catch emergent problems early is very important, especially for the NRE budget. The reason is that problems that are caught during virtual V&V can be discovered quickly, but they do cost effort for their resolution.

Traditional Schedule Pressure by management does also lead to a speedup in META, but it has limits.

**NOMENCLATURE**

- C2M2L Component Model Library
- NRE Non-Recurring Engineering Effort
- UTRC United Technologies Research Center
- T Project Completion Time

**ACKNOWLEDGMENTS**

We acknowledge collaboration with Dr. Brian Murray at United Technologies Research Center (UTRC and financial support from U.S. government contract FA8650-10-C-7080.

**REFERENCES**


**Fig. 5 Sensitivity Analysis Results**

Normalized Sensitivity Analysis:

A 100% change in a META process parameter will cause a X % change on schedule and NRE cost

- Effect on Schedule +
- Effect on Schedule -
- Effect on Budget +
- Effect on Budget -