



AFRL-RI-RS-TR-2022-131

SD2 CHALLENGE PROBLEM INTEGRATOR (TA5)

APTIMA, INC.

SEPTEMBER 2022

FINAL TECHNICAL REPORT

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED

STINFO COPY

**AIR FORCE RESEARCH LABORATORY
INFORMATION DIRECTORATE**

NOTICE AND SIGNATURE PAGE

Using Government drawings, specifications, or other data included in this document for any purpose other than Government procurement does not in any way obligate the U.S. Government. The fact that the Government formulated or supplied the drawings, specifications, or other data does not license the holder or any other person or corporation; or convey any rights or permission to manufacture, use, or sell any patented invention that may relate to them.

This report is the result of contracted fundamental research deemed exempt from public affairs security and policy review in accordance with SAF/AQR memorandum dated 10 Dec 08 and AFRL/CA policy clarification memorandum dated 16 Jan 09. This report is available to the general public, including foreign nations. Copies may be obtained from the Defense Technical Information Center (DTIC) (<http://www.dtic.mil>).

AFRL-RI-RS-TR-2022-131 HAS BEEN REVIEWED AND IS APPROVED FOR PUBLICATION IN ACCORDANCE WITH ASSIGNED DISTRIBUTION STATEMENT.

FOR THE CHIEF ENGINEER:

/ S /

WILMAR SIFRE
Work Unit Manager

/ S /

GREGORY HADYNSKI
Assistant Technical Advisor
Computing & Communications Division
Information Directorate

This report is published in the interest of scientific and technical information exchange, and its publication does not constitute the Government's approval or disapproval of its ideas or findings

REPORT DOCUMENTATION PAGE

PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ORGANIZATION.

1. REPORT DATE		2. REPORT TYPE		3. DATES COVERED	
SEPTEMBER 2022		FINAL TECHNICAL REPORT		START DATE AUGUST 2017	END DATE MARCH 2022
4. TITLE AND SUBTITLE SD2 Challenge Problem Integrator (TA5)					
5a. CONTRACT NUMBER FA8750-17-C-0294		5b. GRANT NUMBER N/A		5c. PROGRAM ELEMENT NUMBER 61101E	
5d. PROJECT NUMBER		5e. TASK NUMBER		5f. WORK UNIT NUMBER R2ET	
6. AUTHOR(S) Jared Freeman, Drew Leins, Sterling Wiggins, John Colonna Romano, Matt Wood					
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Aptima, Inc. 12 Gill Street, Suite 1400 Woburn MA 01801					8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) Air Force Research Laboratory/RITA 525 Brooks Road Rome NY 13441-4505				10. SPONSOR/MONITOR'S ACRONYM(S) AFRL/RI	11. SPONSOR/MONITOR'S REPORT NUMBER(S) AFRL-RI-RS-TR-2022-131
12. DISTRIBUTION/AVAILABILITY STATEMENT Approved for Public Release; Distribution Unlimited. This report is the result of contracted fundamental research deemed exempt from public affairs security and policy review in accordance with SAF/AQR memorandum dated 10 Dec 08 and AFRL/CA policy clarification memorandum dated 16 Jan 09.					
13. SUPPLEMENTARY NOTES					
14. ABSTRACT Engineers regularly use high-fidelity simulations to create robust designs in complex domains such as aeronautics, automobiles, and integrated circuits. In contrast, robust design remains elusive in domains such as synthetic biology, neuro-computation, and polymer chemistry due to the lack of high-fidelity models. DARPA's Synergistic Discovery and Design (SD2) program sought to develop tools to enable robust design despite the lack of complete scientific models. The performers in this program applied capabilities in the sciences to improve models (TA1), engineering expertise to enhance design (TA2), experimentation and testing to evaluate models and designs (TA3), and computer science (TA4) to enable collaborative research and development. The performers selected for SD2 were among the most capable in the world. However, capability alone is insufficient. DARPA retained Aptima as the prime for its Challenge Integration and System Evolution Oversight tasks (TA5) to ensure that SD2 performers addressed productive challenges, discovered and exploited fruitful methods, and collaborated deeply and productively within and across teams. To achieve each of these three goals, Aptima developed novel and effective methods, measures, and models of scientific productivity and team science. Aptima's work and products, reported here, enhanced the scientific and engineering achievement of SD2, helped establish an enduring Open Science community and toolset, and produced new methods based in Aptima's Team Vitals technology for managing scientific and engineering enterprises.					
15. SUBJECT TERMS Synergistic Discovery and Design (SD2), Robust designs, Technical Areas					
16. SECURITY CLASSIFICATION OF:				17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES
a. REPORT U	b. ABSTRACT U	c. THIS PAGE U			
				SAR	161
19a. NAME OF RESPONSIBLE PERSON WILMAR SIFRE				19b. PHONE NUMBER (Include area code) N/A	

TABLE OF CONTENTS

Introduction.....	1
1: On Selecting & Assessing Challenge Problems	4
Overview	4
Generating and Selecting Challenge Problems.....	4
Introduction.....	4
Methods, Assumptions, & Procedures	5
Results & Discussion	6
Conclusions & Recommendations	8
Assessing Performance on Challenge Problems.....	8
Introduction.....	8
Methods, Assumptions, & Procedures	9
Results & Discussion	11
Conclusions & Recommendations	17
2: On Discovering and Assessing Methods	18
Overview	18
Identify Emergent and Required Methods.....	18
Methods, Assumptions, & Procedures	18
Results & Discussion	20
Measuring Methods	30
Methods, Assumptions, & Procedures	30
Results & Discussion	30
Quantitative Measures.....	34
Insight in Discovery and Design.....	36
Conclusions & Recommendations	39
3: On Socio-Technical Systems in Scientific Enterprises	40
Overview	40
Methods, Assumptions, & Procedures	41
Program Structure.....	41
Collaboration Contracts.....	43
Challenge Problems.....	43
Activities	44
Training and Coaching.....	47

Assessing the STS.....	48
Results & Discussion	49
What is the SD2 STS?	49
What is the Role of Technology in the SD2 STS?	55
Alternatives to the SD2 STS	60
Deploying the SD2 STS.....	65
Conclusions & Recommendations	66
4: On System Evolution	68
Overview	68
Methods, Assumptions, & Procedures	68
Results & Discussion	69
Communicating System Status	69
Metrics	71
Managing Complexity and System Architecture	72
Process Automation	73
Data and Metadata	74
System Evolution Management	75
Data Sharing	76
Usability	78
Conclusions & Recommendations	78
5: On Metadata	79
Overview	79
Metadata Interviews	79
Introduction.....	79
Methods, Assumptions, & Procedures	80
Results & Discussion	80
Metadata Survey	85
Introduction.....	85
Methods, Assumptions, & Procedures	85
Results & Discussion	86
Conclusions and Recommendations	91
6: On Transition	93
Overview	93

Methods, Assumptions, & Procedures	94
Interview script.....	94
Results & Discussion	95
Transition of ESCALATE to Los Alamos National Laboratory (LANL).....	95
Transition of ESCALATE to Colorado State University (CSU).....	96
Transition of Tool-chain to Accelerate Synthetic Biology Engineering (TASBE) Extensions and AutoGater to EQAPOL	97
Transition of ETL / Data Analysis Platform Components to Xilis	98
Transition of Cell-Free (CF) Riboswitches Pipeline to Air Force Research Lab (AFRL)	98
Transition Pain Diary	99
Conclusions & Recommendations	100
References	102
Appendices	105
Appendix A: Selecting and Assessing Challenge Problems	105
Appendix B: Example Survey	117
TA1 & TA2.....	117
TA3 survey	117
Appendix C: Example Survey Responses	118
Summary of TA1 survey responses	118
Summary of TA2 responses	119
Appendix D: Decision Tree	121
Appendix E: CTA Outline	122
Appendix F: STS Video Transcript	124
Appendix G: Program Coordination Guidebook	127
Introduction.....	127
Organize People	127
Organize Work.....	129
Organize Events.....	131
Conclusion	133
References	134
Appendix H: Metadata Survey	135
Appendix I: Metadata Survey Responses	141
Acronyms List.....	150

TABLE OF FIGURES

Figure 1: SD2 achieved effects on science of 3x to 20x.....	2
Figure 2: A process for generating and selecting challenge problems	5
Figure 3: SD2 achieved effects on science of 3x to 20x.....	12
Figure 4: SD2 improved four stages of science and the full loop through them 7.5x to 30x.....	12
Figure 5: SD2 improved outcomes (effects) in program problems that spanned several scientific domains.....	13
Figure 6: SD2 had positive effects across the CPs, and different primary effects for almost all CPs.....	13
Figure 7: SD2 technologies were highly accessible to users, well documented, and generalizable across domains.....	14
Figure 8. Sample partial screen capture of Aquarium user interface.....	22
Figure 9. Count of lessons and sub-lessons by theme.....	31
Figure 10: The structure of SD2 Technical Areas and the teams that populated them, in both the biology and chemistry tracks of the program	42
Figure 11: The activity planning process for SD2.....	45
Figure 12: Top half of the TeamVitals dashboard.....	59
Figure 13: Bottom half of the TeamVitals dashboard	60
Figure 14: Dashboard indicating the state of experiments (on rows) through processing stages (on columns; only a sample of which are shown here)	71
Figure 15: Dashboard indicating the number of data uploads per lab.....	71
Figure 16: Redash chart showing the time between experimental request and latest data upload	72
Figure 17: Disciplinary composition of SD2 and ASIST survey respondents	87
Figure 18: Definitions of metadata across SD2 and ASIST.....	87

Figure 17. Disciplinary composition of SD2 and ASIST survey respondents	87
Figure 18. Definitions of metadata across SD2 and ASIST	87

LIST OF TABLES

Table 1: Development of measures and measurements can be centralized or distributed as a function of the diversity of program domains and technologies.....	9
Table 2: Measures of Technology and Levels of Accomplishment (sub-bullets)	11
Table 3: SD2 Challenge Problems rated their technologies similarly within each measure (i.e., column wise), with two exceptions (denoted with *)	14
Table 4: Measurements of SD2 technologies varied widely between measures within components, and within measures between components	15
Table 5: Classification of SD2 components.....	17

INTRODUCTION

DARPA established the Synergistic Discovery and Design (SD2) program in 2016 to “develop data-driven methods to accelerate scientific discovery and robust design in domains that lack complete models” (DARPA, 2016). Conceptually, the test of model completeness is the degree to which it supports the design of objects that have predictable characteristics or performance. This is common in domains such as aeronautics and the engineering of bridges. It is uncommon in some areas of chemistry and biology, where the number of interactions between components—such as neurons or amino acids—number in the millions or billions, or the mechanisms of interaction are not well understood.

SD2 set as its goal to “develop tools to enable robust design despite the lack of complete scientific models.” The term tool was construed broadly in SD2 to encompass software, data, and metadata standards, research techniques, and organizational structures and processes.

The program invested its resources, developed its tools, and demonstrated its effects on Challenge Problems of several types:

1. Constructing genetic circuits in yeast and in newly specified organisms
2. Creating proteins that are stable and bind reliably to DNA and to other proteins
3. Engineering cells that autonomously detect DoD-relevant chemicals in real world environments
4. Creating new perovskite crystals

Progress on these problems has generally been slow and hard won. The research has often been executed using the “individual investigator” model, in which a Principal Investigator and her laboratory team generate theory, collect data, perform analysis, and disseminate results. Success in that chain of activity requires that the team focus on aspects of the problem that their skills, technologies, and data can best address. This use of local resources can diminish reproducibility to the extent that materials and technologies are unique to the laboratory, and that experimental protocols, analytic techniques, and other features reside in the minds of the researchers or their lab notebooks rather than in open literature, databases, and software repositories. This approach is a modern instantiation of the European guild system, which codified the practice in which master craftsmen, their journeymen, and apprentices crafted individual devices.

DARPA designed SD2 to require collaboration and production across disciplines, and it rewarded development of infrastructure, software, standards, research techniques, and organizational processes that helped performers fulfill that collaboration requirement. These tools would codify the implicit knowledge of scientific craftsmen in data, code, and practices that were transferrable to new labs and research problems. Such transition would in turn enhance reproducibility, thus accelerating scientific progress.

DARPA’s design organized the program into two tracks—biology and chemistry. Populating each track were performers in five technical areas (TAs):

- TA1: Scientific Discovery Algorithms
- TA2: Design Algorithms and (in the biology track only) Experimental Planning Algorithms
- TA3: Experimental Facilities
- TA4: Data and Analysis Hub
- TA5: Challenge Problem Integrator

All five TAs had to interact in each track to fulfill the program's objectives. No single performer could execute all of discovery, design, experimentation, infrastructure, coordination, and evaluation. Crossing that complex structure were program-wide Challenge Problems (CPs), which defined specific scientific objectives (e.g., creating new perovskites crystals) to engage performers in advancing science and technology, and generate measurable effects of program impact. Three CPs, in biology, concerned protein design, engineering in yeast, and engineering in bacteria. The chemistry track addressed a single CP: perovskites science.

Performers on the program hailed from 23 unique academic and commercial institutions, 14 of which served as primes, 10 as subcontractors. (Two institutions, Massachusetts Institute of Technology [MIT] and the University of Washington, served on multiple teams).

DARPA engaged Aptima to serve as Challenge Problem Integrator. Our responsibilities, under the initial contract and modifications, were to:

1. Select Challenge Problems and evaluate performance on them
2. Analyze the methods by which performers accelerate discovery and design
3. Ensure continuous and productive collaboration across the program
4. Assess the technological systems of SD2 and coordinate improvements
5. Assess and document the overall socio-technical system of SD2
6. Review the metadata requirements and solutions of SD2
7. Assess the transition of SD2 solutions to military and civilian institutions

Aptima's assessments and evaluations helped DARPA and its performers to quantify the benefits of SD2 technology and organizational processes. Those benefits ranged from 3-fold to 20-fold improvements, at the median, in the speed of research, the quantity of research material processed, the complexity of research objects (e.g., proteins), the accuracy of research findings, and the efficiency of human labor (see **Figure 1**, and details in 1: On Selecting & Assessing Challenge Problems).

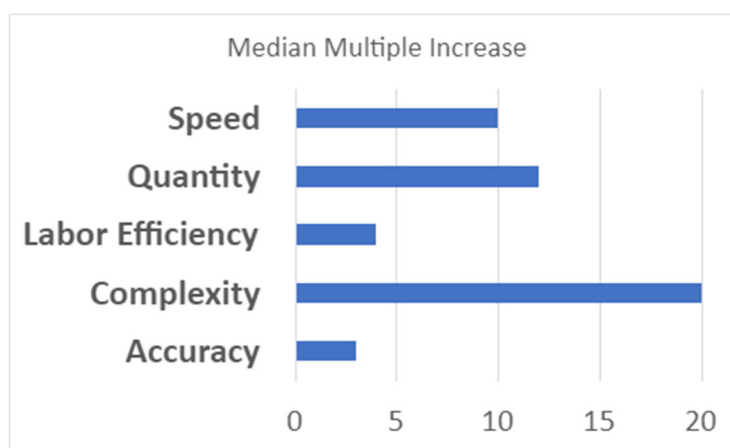


Figure 1: SD2 achieved effects on science of 3x to 20x.

Aptima developed and codified key practices that enhanced collaboration. These were the design and execution of collaboration contracts, challenge problems, program activities, and individual training and

coaching. While the program produced technologies for doing scientific design and discovery, Aptima developed technologies for monitoring productivity of and interactions within the scientific enterprises. (For detail, see 3: On Socio-Technical Systems in Scientific Enterprises).

Aptima's work as TA5 also produced insights concerning the development of research methods (see 2: On Discovering and Assessing Methods), metadata design strategies (see 5: On Metadata), and the transition of complex technologies and social practices (see 6: On Transition).

This report addresses each of the seven TA5 tasks, above, in a separate chapter, with one exception. We have combined our report of (3) collaboration methods with our analysis of their effect on (5) the socio-technical system of SD2.

We welcome discussion concerning the content of this report and our experience as evaluators and coordinators of other large, scientific programs and enterprises.

1: ON SELECTING & ASSESSING CHALLENGE PROBLEMS

Overview

SD2 recruited scientists from diverse domains of biology, chemistry, and computer science in a program intended to develop data and technologies that accelerate discovery and design. The program was intentionally structured to incentivize collaboration across boundaries between teams, technical areas, tracks (biology and chemistry), and professions.

SD2 defined several *Challenge Problems* (CPs)—well-specified missions such as “create reliable circuits in yeast”—that ran in parallel throughout the program. To define its CPs, SD2 created a method for generating and selecting engaging and productive scientific missions. To our knowledge, no such method existed in the literature prior to SD2. To assess the products of those CPs, SD2 developed and applied a method that produced quantitative measurements on multiple dimensions of the program’s diverse science and 54 technologies. Aptima led these efforts as the Challenge Problem Integrator (TA5).

We report on the advance in challenge problem design and the evaluation of the diverse progress and products of SD2, each in a chapter section below.

Generating and Selecting Challenge Problems

Introduction

Challenge Problems serve both technical and social functions in research and development programs. Challenge Problems (CPs) inspire and exercise technical solutions that advance the concrete cases that the problem specifies. When the solutions must serve multiple CPs, and those problems differ in scale, pace, focus, or domain, the exercise tests the generalizability of technical solutions. The scientific products of CPs also provide evidence of the effects and value of technical solutions. This garners support within the funding institution and among potential transition partners. CPs, when well designed, shape the social system of a program. They draw performers to the vision and goals of the program manager. Finally, CPs engage performers with each other, to the extent that the challenges require input from multiple technical areas or disciplines, and provide them all with victories to claim in publications and future proposals.

Given the centrality of challenge problems to the technical success and social integrity of programs, we would expect to find a robust literature concerning their design, selection, and use. To our knowledge, there is no such literature. Thus, the designers of CPs (primarily program managers (PMs) and their scientific staff) must work as artisans guided not by formal methods but by their vision and personal experience in the research domain. The research literature concerning the formulation of scientific problems offers lessons only indirectly to PMs.

- Scientific challenges are approachable only if there is adequate theory, technique, technology, and data to address them (Al-Ghassani et al., 2006; Polya, 1957). Accordingly, PMs must assess these attributes of candidate CPs.
- CP designers should clearly define the metrics of success (e.g., ‘engineer a bridge that is light and strong’). Ideally, separable sub-problems should also be measurable (McComb, Cagan, & Kotovsky, 2017).
- CPs should afford multiple representations, each of which potentially supports a unique solution strategy (Newell & Simon, 1972; Mohaghegh & Furlan, 2020).

- Case recognition and case-based reasoning drive problem-solving (Klein, 1993). This implies that CP designers should distinguish their challenges from prior cases, to focus performers on the distinctive issues ahead of them.
- Tactical decision making improves with theoretically grounded training in critical thinking (Cohen, Freeman, & Thompson, 1998). From this, we infer that CPs should present sufficient uncertainty, impact, and urgency to inspire performers to deep and critical thinking.

Some of these heuristics are familiar to PMs, notably the first two: assess the theoretical and technical attributes of a CP and clearly define metrics of success. All of the heuristics may be useful to CP designers. They do not constitute a method. In this section, we report a method for defining candidate challenge problems and selecting among them, developed in SD2, successfully applied in SD2, and published in Freeman, Leins, Bell, and the SD2 Research Consortium (2019).

Methods, Assumptions, & Procedures

We developed a method for generating and selecting CPs to ensure that SD2 performers could leverage their expertise to define the program's work and engage in that work productively. The method consisted of these steps (see **Figure 2**):

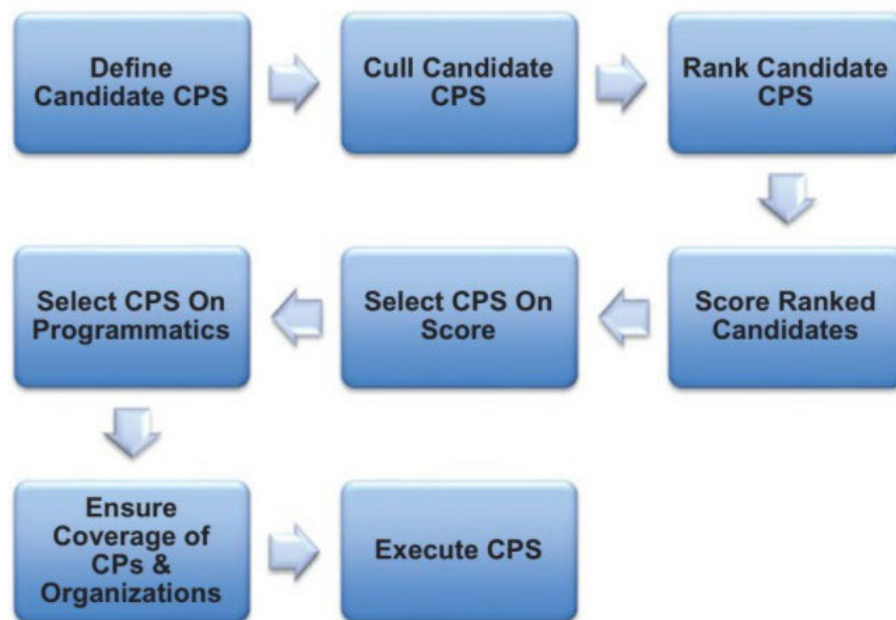


Figure 2: A process for generating and selecting challenge problems.

In practice, we applied the method as follows:

1. Sixty-six performers gathered at a workshop. These performers represented 18 geographically distributed organizations, ranging in expertise across biology and computer science, and representing academic and commercial sectors.
2. We charged them with generating a proposal for challenge problems, and documenting these on a template that required the CP name, the author and collaborators, short- and long-term

goals, automation requirements for execution of experiments, data processing and analysis steps, alignment with sponsor objectives, and relevant scientific literature.

3. The performers created about two dozen candidate CPs. Performers then read the collected set of candidate CPs, and met to cull those that were redundant of others or that were components of other CPs, yielding a refined set of 12 CPs for consideration.
4. We then asked a subset of 36 performers to rank order the five CPs they most preferred and provide a textual rationale for their choices.
5. We analyzed their rankings and named as winners the CPs that occurred most frequently and highest among the rankings. Conflicts in rankings (ties) were resolved by selecting CPs that were most frequently among the top two choices.
6. The DARPA program manager (PM) reviewed the top problems to ensure they conformed to her vision for the program. She amended the list with one highly ranked CP that closely aligned with program objectives.
7. We then assessed whether each performer had a role in at least one CP, and whether each CP had a sufficiently long bench of performers. To do this, we conducted a survey concerning potential engagement in the six CPs. Each participant entered a score to indicate whether they: '(1) definitely can work the CP now; (2) probably can; (3) cannot determine yet; (4) probably cannot; (5) definitely cannot'.
8. The survey data confirmed a suitable distribution of performers over CPs. These data also proved useful to the program manager for managing human resources, enabling her to identify which performers could support CPs in need of specific institutional capabilities, and which performers were spread too thinly across CPs to concentrate their attention and effort.

DARPA and Aptima then set to work forming Working Groups to plan and execute the six CPs.

Results & Discussion

The CP generation activity produced 24 candidate CPs, culled to 12, and selected five through rank order voting. The program manager amended that set with one CP from among the most highly rated remaining candidates. The survey of engagement opportunities revealed a reasonable distribution of scientists across CPs. The program manager made modest adjustments to their self-assignments based on these data and other information. These challenge problems were the focus of activity in SD2 for most or all of the program. In each, the members of the CP working group (typically drawn from multiple performer teams) developed techniques and technologies that would accelerate discovery and design with respect to their research objectives (as expressed by the CPs at the end of their work in SD2).

1. Novel chassis (NC) explored the construction of genetic circuits in new organisms
2. The protein design (PD) CP worked to create small proteins that are stable and bind to DNA and to proteins
3. Yeast states (YS) designed genetic circuits in yeast and assessed the robustness of those circuits
4. Cell-free riboswitches (CF-RS) sought to engineer cells that autonomously detect DoD-relevant chemicals in real world environments¹
5. Characterizing the operational envelope of yeast

¹ The Riboswitches CP evolved into the Cell-Free Riboswitches CP which tested riboswitches outside the cell, eliminating the need environment that maintains a cell. Cell-Free Riboswitches CP transitioned to AFRL.

6. The perovskites (PV) CP strove to create techniques and technologies that facilitated discovery of new perovskites crystals

The informality of the vote scoring process was a shortcoming of the method as described above. We subsequently defined a formula for computing winning CPs, one that performed accurately when rankings were incomplete (i.e., some performers ranked fewer than the allowed number of CPs) or redundant (i.e., some performers gave the same ranking to multiple CPs).

The ranked-choice voting formula computes a score for each of many candidate CPs from ordinal rankings by multiple rankers who may rank only a subset of candidates, may rank a unique subset of candidates, and may award some candidates the same rank (i.e., a tie). In this formula, the parameters are:

- nn denotes the number of rankers
- kk denotes the number of candidates to be ranked
- mm denotes the maximum rank assigned over all candidates and rankers

CC_{jj} denotes candidate jj for $jj = 1, \dots, kk$. The ordering of rankers and candidates is fixed and identified with the sets $\{1, \dots, nn\}$ and $\{CC_1, \dots, CC_{kk}\}$.

For each of the n rankers, we define a ranking function:

$$\begin{aligned} r_i : \{\text{Candidates}\} &\rightarrow \{\text{Ranks}\} \cup \{m + 1\} \\ \{C_1, \dots, C_k\} &\rightarrow \{1, \dots, m, m + 1\} \end{aligned}$$

Above, $mm + 1$ corresponds to a candidate that a ranker does not rank. This score represents a worst case, in which the ranker considered all candidates not ranked as equal in priority and minimally less desirable than the lowest ranked of the ranked candidates. We define a pre-weight as follows:

$$\tilde{w}(C_j) = \sum_{i=1}^n r_i(C_j)$$

We define the weight function $w(C_{jj})$ using the following normalization, in which CC_{ii} is the candidate with the lowest pre-weight:

$$w(C_j) = \tilde{w}(C_j) - \min_{i \in \{1, \dots, k\}} \tilde{w}(C_i)$$

We then applied this formula to the ranking data generated in the conference exercise, above. That post-hoc analysis confirmed the informal findings with the exception that one CP declined somewhat in the rankings, and the program manager's selection was found to be among the five highest ranked CPs.

Finally, we analyzed the 112 statements of rationale that performers provided with their votes. To do this, two researchers independently coded a subset of the rationales, met to resolve their emergent classification schemes into a single scheme, applied it to all statements, tested for high correlations between the classes (and found none), then presented these to DARPA for future use. These attributes (or dimensions) of CPs may be of value to future PMs. They were as follows:

- Capability: The CP creates a tool or technique that enables new methods of analyzing, designing, or producing parts, components, or systems (50 instances in CP ranking rationales).

- Speed: The CP enables the program to demonstrate success quickly (22 instances).
- Impact: Discovery, design, technology, techniques, and applications arising from this CP will be significant (e.g., publishable, award-worthy) (56 instances).
- Synergy: The CP builds on extant products of the program, resources of the performing organizations, commercial products, or open sources tools and data (52 instances).

Conclusions & Recommendations

SD2 was diverse in its scientific domains, spanning biology and chemistry. It developed challenge problems within each of those domains to leverage the expertise of its diverse performers and engage them in the program mission. The method developed here to define and select among those challenge problems was successful in both respects. We recommend that program managers with similarly diverse performers refine and apply the method described above and in Freeman, Leins, Bell, and the SD2 Research Consortium (2019). To our knowledge, the research literature provides no alternatives.

The analysis of challenge problem attributes or dimensions, though only briefly described above, may help future program managers distribute their funding judiciously over a portfolio of CPs. For example, a manager might wish to invest initially in CPs strong on synergy and speed. This would build collaborative relationships among performers, leverage scarce resources, and demonstrate progress rapidly. Later investments might emphasize capability and impact, to create transition-able product of value.

We also recommend that program managers gather performer preferences and capability for challenge problems, as we did in SD2. This analysis may help managers to understand the skills and resources of their performers and allocate those resources effectively across challenge problems.

Assessing Performance on Challenge Problems

Introduction

R&D programs vary in their degree of domain focus. Evaluation methods, assumptions, and procedures should vary accordingly. Programs with a singular domain focus can conduct evaluations of most or all performers with a single set of metrics and measures. Programs that encompass multiple domains may be able to standardize metrics across performers, but likely cannot standardize measures, because the object (or unit) of measurement may vary between domains.

To the extent that diverse domains are technically complex (as in the case of protein design, for example) or novel (as in the case of perovskites chemistry), measure design requires deep domain expertise. That expertise may only be available among the very performers the program must evaluate. In that case, evaluators must judiciously engage performers in collaborative design of measures, while managing the risk that performers will define measures that show their strengths (or “cherry pick”) while concealing their weaknesses.

These issues of metrics and measures effectively define two dimensions of challenge in designing evaluations: diversity of the objects of measurement and centralization of measure authorship. In **Table 1**, we dichotomize these rather continuous dimensions to describe specific evaluation design strategies, risks, and mitigations.

Table 1: Development of measures and measurements can be centralized or distributed as a function of the diversity of program domains and technologies.

		Diversity of Program Domains and Technologies	
		Low	High
Measure Author-ship	Centralized	Measures: Uniform measures across performers. Risks: Measures of diverse, enabling technologies may be harder to standardize than measures of effects. Mitigation: Critical review of measures by domain experts, potentially program performers themselves.	Measures: Domain-specific measures. Risks: Central authority lacks the domain expertise to design revealing measures of effects or technologies. Mitigation: Retain independent experts to design measures, at potentially high cost.
	Collaborative	Measures: Uniform measures across domains and technologies. Risks: Measure selection and definition may be contentious between performers whose solutions have distinct strengths because all performers may fail some measures. Mitigation: Assessment should be framed as diagnostic of distinctive solutions, rather than evaluative of performers.	Measures: Domain-specific or performer-specific measures. Risks: Performers cherry-pick complimentary measures. Mitigation: Critical review of performer-proposed measures.

Independent of the diversity of domains are two objects of evaluation: effects and the techniques or technologies that enable them. Challenge problems should be designed to test for effects in program’s research domains, ideally in domain niches most starved for advancement and most sensitive to the innovations of program performers. Metrics are also required to characterize the techniques and technologies that enable those effects, most especially if these are the transition-able products of an R&D program.

SD2 spanned a remarkable range of research domains, including the biology of proteins and cellular organisms and perovskites chemistry. These domains were deep and novel. SD2 generated a variety of analytic technologies that leveraged data science, artificial intelligence, and computer science. The program metrics and evaluation methods were adapted to these circumstances. We describe these in the following sections.

Methods, Assumptions, & Procedures

In collaboration with DARPA program leadership, Aptima defined two sets of metrics. The metrics of effects (MOEs) in scientific domains were accuracy, complexity (of the domain objects), labor requirements, quantity (of domain objects), and speed. The metrics of technologies (MOTs) were designed to support adoption decisions by potential transition partners. These metrics concerned the documentation, accessibility (to users), generalizability of the tool (value of output across domains), and usability.

Evaluation of Scientific Effects

The development of measures and measurements on the effects (MOEs) was a collaboration between TA5, DARPA, and other performers. Our team at Aptima presented the metrics categories to the scientific performers in each challenge problem working group. We requested measures and measurements of a baseline and endpoint in the program. Baseline measurements typically were performer-declared measurements drawn from experience or the literature of the state of the art at the start of the program, or from the start of the challenge problem or experiment. We then classified the delivered measures and their measurements using the scheme above (accuracy, complexity, labor requirements, quantity, and speed) and by the stage in a program-defined workflow: design (e.g., a biological circuit), build (e.g., the circuit in a host cell), test, and learn (e.g., the percentage of accurately functioning circuits given the experimental conditions). We then examined the range of measurement values, chose a descriptive statistic that represented those values sensibly—the n-fold effect (equivalent to percentage change)—and presented these summary statistics to DARPA and performers for use in program presentations and transition campaigns.

Measurement values ranged in type and size between and often within classes and research stages. We standardized the unit of measure by converting each value to a multiple of the baseline (e.g., 30 denotes a 30-fold increase or decrease from the baseline).

For example, the Novel Chassis working group reported a 7.5-fold increase in the number of circuit parts designed per study between program start and end. We classified this as an effect on quantity in the design stage of research. The Protein Design working group reported a 3.4-fold reduction in the time to analyze experimental results, which we classified as an effect on speed in the learning stage. The Perovskites working group reported a 10-fold reduction in human labor (staff or staff hours) required to characterize crystals. We classified this as labor efficiency in the learning stage.

We then computed the median value in each class or stage, to diminish the strongly biasing effects of large and small measurement values. For example, 22 measurements of speed ranged from a 2.3-fold increase in experiment execution rate to a 100,000-fold decrease in data validation time across multiple CPs using SD2 tooling. The median of the measurements was 10; the mean of 4615 was strongly skewed by the validation time measurement, and slightly skewed by two measurements of 540 and 724.

Measurements of effects were collected in the summer of 2021 and analyzed by September 2021, at the close of the Period of Performance for some performers.

Evaluation of Technologies

The development of measures of technology (MOTs) was centralized. Aptima defined these in collaboration with DARPA. Each measure concerned a single dimension of technology (e.g., its accessibility to users) and several increasing levels of achievement within that dimension (e.g., runs on a laptop for an individual, on SD2 shared infrastructure at TACC, in a Docker container that is by definition portable across operating systems though not necessarily integrated into scientific pipelines, or integrated into pipelines). We presented the measures to performers in the challenge problem working groups and asked that scientists and technologists collaborate to respond “yes” or “no” to each level for

each major technology developed by the group² (see **Table 2**). We then calculated the percentage of levels attained per technology, which standardized these measurements by adjusting for differences in the number of levels per measure. In rare cases, performers claimed achievements at a high level but did not claim achievements at one of the lower levels. These few cases were retained in the dataset and analysis. Measurements of technology were collected and analyzed in July 2020, when the technologies were, in the main, sufficiently mature to support the final year of experimentation.

Table 2: Measures of Technology and Levels of Accomplishment (sub-bullets)

Measure	Levels of accomplishment
Component Documentation	Checked into public repository for access by others
	Application Programmers Interfaces (APIs), other interfaces and build instructions are documented and available in the repo
	Tutorials are available
	Documentation and resources are sufficient for TA5 to build and run the component independently
Component Accessibility	Runs on individual's laptop or on hardware within that individual's lab
	Runs on TACC in an individual's directory
	Runs in a Docker container, Agave app, or as a sharable Jupyter notebook
	Integrated into the automated pipelines
Generalizability / Output utility	People use the component to address a single experimental type/thrust within a single challenge problem
	People use the component to address multiple experimental type/thrusts within a single challenge problem
	People use the component to address multiple challenge problems
Component usability	The author is integrally involved every time the component is run, and the author provides results and interpretations to others
	The author is integrally involved every time the component is run, and others are able to independently interpret the results or others are able to independently run the component, but the author is sometimes required to help interpret the results
	The user can sometimes run the component and interpret its results independently, but the author is often needed to reconfigure or add features to the component
	The system can run the component automatically; interpretation, bug fixes, and extension may require the author
	The user can run the component and interpret its results independently, and the author is only required for bug fixes and novel extensions of the component

Results & Discussion

Evaluation of Scientific Effects

SD2 increased scientific effects from 3x to 20x across the five program metrics. At the median, accuracy of designed scientific products (e.g., crystal stability, designs of stable proteins) or analytics (accurate machine-learning of protein stability) increased 3-fold, the complexity of experimental objects increased

² One additional survey item concerned the influence of the tool on scientific progress. The item garnered insufficient response to warrant inclusion in the analysis, and it is omitted from **Table 2**: “[This component] had slight / major influence on accurately, positively defining the design or predicting its effect.”

20-fold, labor efficiency increased 4-fold, quantity of scientific product (e.g., crystals discovered, proteins assayed) or analytics (e.g., characterization capacity) increased 12-fold, and the speed of science increased 10-fold (see **Figure 3**).

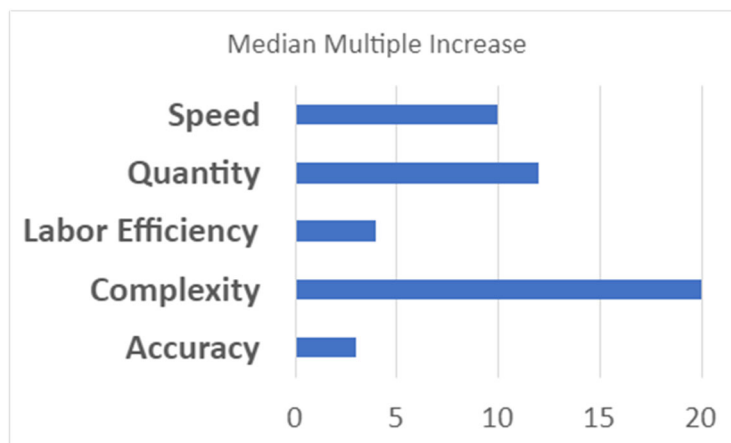


Figure 3: SD2 achieved effects on science of 3x to 20x.

These effects on science accrued most to the construction (or build) of experimental entities (e.g., biological circuits), but were considerable for the stages of design, testing, learning, and certain processes that spanned the full “DBTL” loop. At the median, SD2 enhanced accuracy, complexity, labor, quantity, and speed in the design phase 7.5-fold, in the build phase 30-fold, in the test phase 10-fold, in the learn phase 10-fold, in the DBTL phase 5.6-fold (see **Figure 4**).

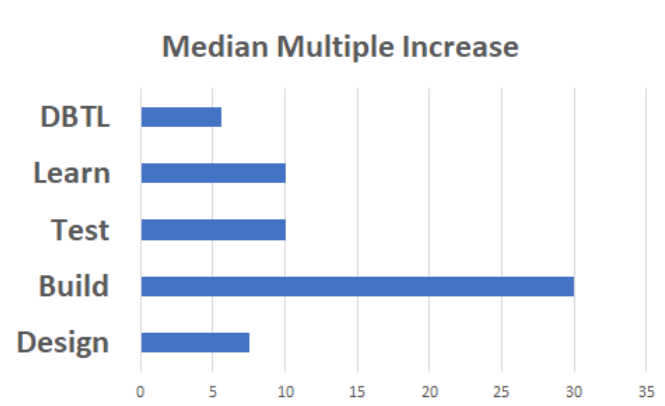


Figure 4: SD2 improved four stages of science and the full loop through them 7.5x to 30x.

SD2 brought benefits to diverse challenge problems in biology and chemistry. Measurements from the challenge problem working groups showed that the attributes of scientific work on Cell-free Riboswitches³ (speed, quantity, labor, complexity, accuracy) improved 4-fold at the median, on Novel

³ Recall that the Cell-Free Riboswitches CP arose from the Riboswitches CP that was defined early in the program. This evolution within a CP was common in SD2.

Chassis improved 10-fold, on Protein Design improved 11.5-fold, on Perovskites improved 4-fold, and on Yeast States improved 10-fold (see **Figure 5**).

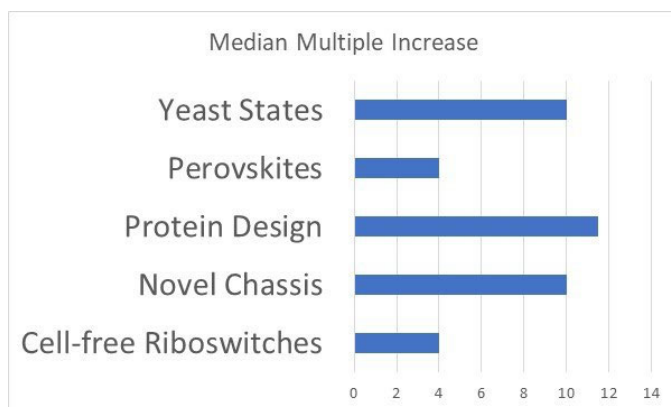


Figure 5: SD2 improved outcomes (effects) in program problems that spanned several scientific domains.

CPs differed from one another on the specific effects they garnered from SD2. The Novel Chassis (NC) CP improved 48-fold at the median on complexity; the Protein Design (PD) CP increased 42-fold on speed; the Perovskites (PV) CP grew 18-fold on quantity; and the Yeast States (YS) CP grew 40-fold on accuracy. The shorter-lived Cell-free Riboswitches (CF-RS) CP benefitted primarily on speed, which increase 17-fold (see **Figure 6**).

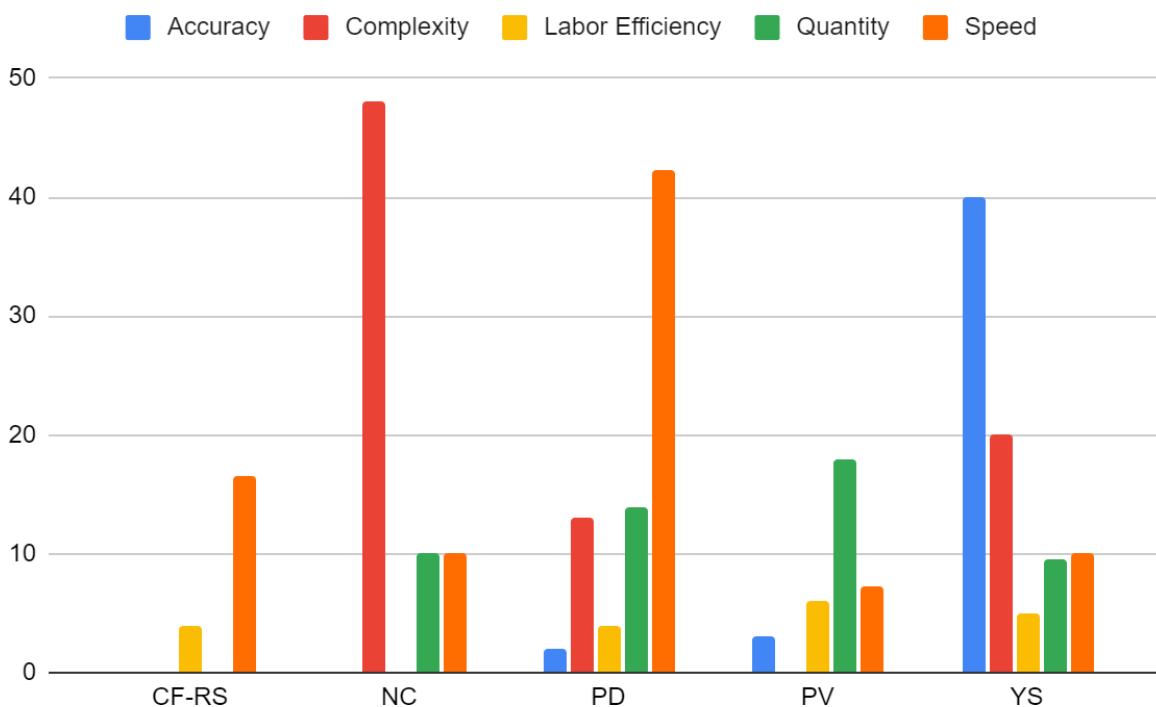


Figure 6: SD2 had positive effects across the CPs, and different primary effects for almost all CPs. The challenge problems (CPs) presented above are cell-free riboswitches (CF-RS), novel chassis (NC), protein design (PD), perovskites (PV), and yeast states (YS).

Evaluation of Technologies

SD2 performers were prolific developers of technology, creating some 54 components through the first three years of the program. These technologies were highly accessible to users, well documented, and generalizable, as assessed by measurements ranging from 79% to 87%. The usability of SD2 technologies was lower, ranging from 60% to 74%. However, technology efforts in the final year of the program often focused on usability and documentation, and so these measurements were likely higher at the end of the period of performance in 2021 than shown here (see **Figure 7**) in data from 2020.

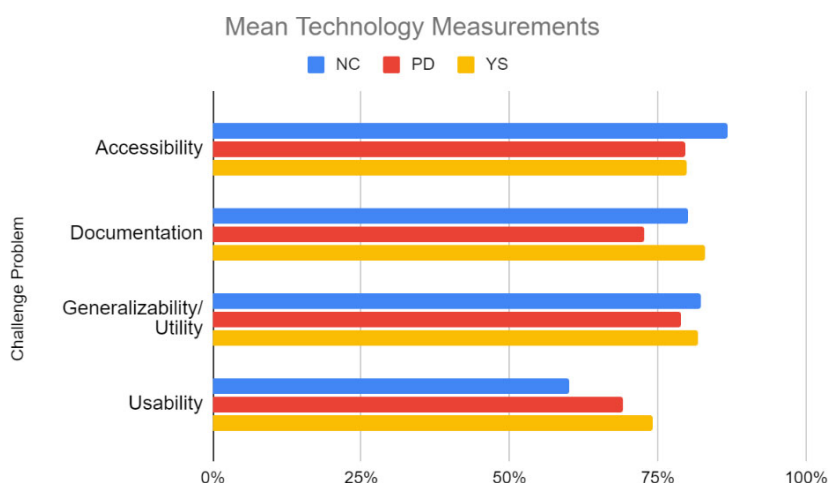


Figure 7: SD2 technologies were highly accessible to users, well documented, and generalizable across domains. Usability in 2020 received lower scores.

Measures of Technology from three challenge problems were evaluated. Their technology scores were similar between challenge problems, save that documentation was lower-scored on Protein Design (73% vs. an average over challenge problems of 80%), and usability was lower-scored on Novel Chassis (60% vs. an average of 69%; see **Table 3**).

Table 3: SD2 Challenge Problems rated their technologies similarly within each measure (i.e., column wise), with two exceptions (denoted with *).

Challenge Problem	Accessibility	Documentation	Generalizability / Utility	Usability	Grand Total
NC	87%	80%	82%	60%*	77%
PD	80%	73%*	79%	69%	75%
YS	80%	83%	82%	74%	80%
Grand Total	82%	80%	81%	69%	78%

Perovskites was excluded from this analysis because its data were preliminary and incomplete. The Perovskites team received its DARPA contract more than a year after other performers. Accordingly, its technology was immature. The Cell-free Riboswitches CP is excluded here because this study followed the closure of the Riboswitches CP and preceded the kickoff of the Cell-free Riboswitches CP.

Analysis across the 53 components indicated that 13 achieved perfect scores on all four metrics (accessibility, documentation, utility, usability), 27 achieved perfect scores on one to three metrics, and

13 achieved perfect scores on no metrics. Components achieved less than 50% of the available levels of accessibility in 4 cases, documentation in 7 cases, generalizability / utility in 10 cases, and usability in 16 cases (see **Table 4**). We note that several technologies—DSGRN, Intent Parser, QC, and XPlan—were used and reported by two Challenge Problem working groups: Novel Chassis and Yeast States. This is a testimony to their generalizability.

Table 4: Measurements of SD2 technologies varied widely between measures within components, and within measures between components. Green indicates higher values; red denotes lower values.

Component	Accessi- -bility	Documen- -tation	Generalizabili- ty / Utility	Us- -abilit y	Grand Total
Additive Design Model	50%	25%	100%	40%	54%
Aquarium	100%	100%	100%	60%	90%
Autoprotocol	100%	100%	100%	100%	100%
Bayes Database (BayesDB)	75%	25%	67%	40%	52%
Build request parser	50%	50%	33%	40%	43%
Cello	100%	75%	100%	100%	94%
Convolutional Neural Network (CNN) stability model	75%	25%	67%	80%	62%
CNN unfolded-state model	100%	100%	67%	80%	87%
DNA Assembler (DASi)	75%	75%	33%	20%	51%
Data Catalog	100%	100%	100%	100%	100%
Data Converge	100%	100%	67%	100%	92%
Design unit dependent on DSGRN	25%	50%	100%	60%	59%
Diagnose	75%	100%	67%	100%	85%
Dictionary Writer	50%	100%	100%	100%	88%
DNAseq	100%	100%	33%	0%	58%
Dynamic Signatures Generated by Regulatory Networks (DSGRN)	75%	88%	50%	40%	63%
Experiment Analysis Dashboard	100%	100%	100%	100%	100%
Flow ETL (TASBE Flow Analytics wrapper)	100%	100%	100%	80%	95%
Flow scoring technique	100%	100%	100%	100%	100%
Genetic Circuit Compiler	25%	25%	33%	20%	26%
Intent Parser	100%	100%	100%	100%	100%
Lab Trace Ingest	100%	50%	100%	100%	88%
Landing Site Recommender	50%	50%	33%	60%	48%
Live/Dead Model	50%	25%	100%	40%	54%
Mohammed's circuit-dynamics modeling	100%	100%	100%	40%	85%
Performance Metrics	75%	50%	33%	60%	55%
Persistence images	75%	100%	100%	100%	94%
Precomputed Data Table	25%	100%	67%	100%	73%
Preprocessing data pipeline	100%	100%	100%	100%	100%
PSAP (postprocessing pipeline)	100%	100%	100%	80%	95%

Component	Accessi- -bility	Documen- -tation	Generalizabili- ty / Utility	Us- -abilit y	Grand Total
pySBOL	100%	100%	100%	100%	100%
QC	88%	88%	100%	30%	76%
RNA sequencing (RNA-seq)	100%	100%	100%	60%	90%
Rosetta	50%	75%	100%	40%	66%
Synthetic Biology Open Language (SBOL) Project Dictionary	100%	100%	100%	100%	100%
Science Table	100%	100%	100%	100%	100%
Sensor Promoter Scraper Tool (KO)	75%	75%	67%	60%	69%
Sequence-based binding model	75%	0%	33%	40%	37%
Stability Assay Calibration	100%	100%	67%	100%	92%
Structured experimental request	100%	100%	100%	100%	100%
Structured experimental request - metadata comparison	100%	100%	100%	80%	95%
(Synthetic Biology Curation Tools) SYNBICT	25%	75%	100%	20%	55%
SynBioHub	100%	100%	100%	100%	100%
SynBioHub (SBH)	100%	100%	100%	100%	100%
SynBioHub Metrics	75%	50%	33%	80%	60%
SynBioHub Prospector	75%	50%	33%	60%	55%
Tool-chain to Accelerate Synthetic Biology Engineering (TASBE) Flow Analytics	100%	100%	100%	100%	100%
Terrarium	75%	100%	67%	20%	65%
Test harness	100%	100%	100%	80%	95%
TOPOPT + Cover-tree Differencing via Entropy Reduction / Topological Data Analysis (CDER/TDA)	75%	75%	100%	40%	73%
Unexpectedly Unstable Design Identifier	75%	25%	33%	40%	43%
VDR (versioned data repository)	50%	100%	100%	80%	83%
Xplan	100%	88%	100%	60%	87%
Grand Total	82%	80%	81%	69%	78%

SD2 components were categorized into one or more of three primary functions: design, data processing, and discovery. Average measurements of technologies were high, ranging from 75-100% in all functions, with the exception of components that served only discovery (67%), or both design and discovery (68%; see **Table 5**). It is likely that the particular, creative nature of discovery makes it more difficult or less desirable to mature the technologies that address it.

Table 5: Classification of SD2 components

Stages	AVERAGE of % of Levels Achieved
Data Processing	90%
Design	75%
Design & Data Processing	100%
Design & Data Proc & Discovery	100%
Design & Discovery	68%
Discovery	67%
Discovery & Data Proc	95%
Grand Total	78%

Conclusions & Recommendations

Program managers and evaluation teams should assess the diversity of program domains and technologies to determine the extent to which measures should be defined by a central authority or defined in a collaborative manner by performers expert in their disparate domains. Collaborative measure definition made evaluation feasible for the SD2 evaluation team (Aptima), which did not have organic expertise in the many domains of scientific research conducted under this program. The collaborative strategy was also productive and informative concerning the effects of SD2 on science.

The challenges to measurement of technology (MOTs) on a technologically productive program such as SD2 are timing and objectivity. MOTs should be taken at the point at which they are mature enough to support major experimental efforts within the program, transition outside it, and identification of the technology gaps that should be filled. The sweet spot is probably 1 year before the end of program Period of Performance, which is the timepoint we hit in SD2. We recommend that PMs refine the technology evaluation methods used here to enable programs to collect MOTs more frequently, including the end of the program when transition is imminent.

SD2 did not objectively verify performer measurements because the number, variety, and technical sophistication of the science and 53 technologies were extraordinarily high. We advise managers of diverse and prolific programs such as SD2 to consider whether the well-structured, rigorous, but subjective approach used in SD2 is sufficient, or a more detailed and objective evaluation is required at potentially higher cost.

2: ON DISCOVERING AND ASSESSING METHODS

Overview

The challenges of SD2 drove performers to innovate both automated and manual methods. They discovered gaps in their methods that called for automated solutions; and they introduced automation that generated unanticipated gaps in methods and workflows.

Aptima engaged in a variety of knowledge capture activities with SD2 performers to uncover core capabilities and limitations—for example, in data generation, processing, and sharing—and the impact of those capabilities and limitations on the emergent socio-technical system. Through these activities, we identified emergent methods, lessons learned, and opportunities for modifying and enhancing workflows (whether with automation or manual operations). We also captured the contexts in which performers applied novel methods, modified existing methods, or scrapped workflows altogether. Here, context includes organizational structure and processes, as well as scientific and engineering spaces in which they operate.

To address these challenges, we first analyzed some of the core capabilities that performers brought to the program. We then identified emergent needs as the program experienced roadblocks and bottlenecks. We measured the impact of some SD2 systems and methods on scientific productivity. Finally, we developed a framework to guide S&T enterprises in the use of these methods.

Identify Emergent and Required Methods

The intent of SD2 was to integrate multiple distributed capabilities to comprise a semi-automated design-build-test-learn (DBTL) loop to accelerate the pace of discovery and design. Although that end goal was relatively fixed, the program's architect realized that paths to it would be numerous and diverse. Accordingly, SD2 was purposefully designed as a modular, flexible socio-technical system that allowed performers to organize and reorganize themselves, innovate and abandon technologies, and engineer and re-engineer paths to achieve the program's goals. Thus, as performers discovered different paths to success, they developed, proposed, and implemented new methods to propel them forward. Aptima engaged regularly with performers to monitor these processes and help performers identify emergent needs and methods to address them.

Methods, Assumptions, & Procedures

Surveys

To elicit high-level details on characteristics of methods and workflows, Aptima engaged performers with short-form questioning methods, such as surveys and questionnaires. For example, we surveyed performers regarding where their methods were and were not applicable, whether their methods did or did not work in a particular workflow, and general strategies for improving methods and workflows. Typically, these procedures were short (e.g., usually no more than 10 questions) and focused on a specific topic (e.g., what method would you apply to a specific scientific or engineering question?). We applied survey methods to help identify candidate scientific or engineering methods, data and other resource requirements for those methods, and potential limitations associated with those methods (e.g., what methods supported or failed to support data accessibility). We also used surveys to surface methods that supported or inhibited progress and productivity at quarterly meetings. Aptima collected, analyzed, and summarized the results of these surveys in memos delivered to DARPA, or in

presentations to DARPA in intra-quarter meetings or to the broader SD2 community at program-wide quarterly meetings (e.g., see **Appendix B: Example Survey**).

Interviews

To surface more granular details about opportunities, issues, and limitations associated with methods, Aptima conducted in-depth, in-person interviews. In contrast with surveys, which are structured, interviews can be semi-structured and thus more flexible. This flexibility allows a range of topics to be discussed among participants. We applied interviews to learn more deeply about broad classes of methods to conduct science and support discovery, as well as to learn more about performer perspectives on the methods, state, and goals of the program. We analyzed interview content qualitatively and summarized the results in memos and presentations. We conducted formal interviews with multiple representatives from each TA, Challenge Problem working group, and technical working group (some individuals multiple times) throughout the course of the program; and we conducted informal interviews in sidebars and other activities at each quarterly meeting.

Cognitive Task Analysis

Some methods and procedures are difficult for performers to articulate completely because many components of those methods are represented as implicit knowledge. To reveal that implicit knowledge and make it explicit for the purpose of identifying inefficiencies or opportunities to improve, Aptima conducted Cognitive Task Analysis (CTA) activities. CTA comprises a suite of knowledge elicitation methods designed to "unpack" the thinking of experts, characterize it, operationalize it, and develop training around it to accelerate learning and develop expertise more efficiently. CTA methods combine in-depth interviews and activities that contextualize work operations, to help expert practitioners or performers recall and articulate the granular details of those operations, to identify the cues they attend to during task completion, describe how they interpret those cues, and reveal how their interpretations inform decisions. CTA can be useful in conserving resources that would otherwise be expended to learn the outcome of a procedure or of a modification to a procedure. For example, for SD2, we wanted to avoid the unnecessary consumption of resources to learn that an experiment should have been modified in-process or that a laboratory instrument was incapable of performing a necessary action. We wanted to characterize human operations to evaluate their candidacy for automation.

Small group working meetings

Methodological and workflow improvements required multiple stakeholders to convene to discuss issues that exist at the interfaces between TAs (e.g., data sharing across TAs). Thus, to surface methodological bottlenecks, solutions, strategies, and actions, Aptima participated in regular CP and technical working group meetings. We also convened meetings around special topics that required out-of-cycle discussions. These meetings offered forums for performers to discuss critical methodological or measurement issues and to work collaboratively toward solving them.

Quarterly integration/working meetings

Program-wide meetings were fertile ground for surfacing and discussing issues and solutions that involved or impacted the entire program (e.g., metadata issues). The purpose and methods of quarterly meetings are discussed in greater detail in 3: On Socio-Technical Systems in Scientific Enterprises. Of specific relevance to identifying methods, we leveraged quarterly meetings to surface program-wide issues in methods and metamodels, identify working groups to address those issues, and make plans for addressing tractable issues within the ensuing quarter(s).

Results & Discussion

Baseline/early capabilities and methods

In January 2018, the program negotiated a set of challenge problems (CPs) that would guide the work of the program through its first phase (18 months) and lead to derivative challenges in the second phase (1: On Selecting & Assessing Challenge Problems). Following the selection of six challenges, Aptima surveyed performers in TA1-3, to map out the CP-space in which each performer could operate; identify the methods, tools, and metrics that performers proposed to apply to each CP; and uncover the challenges—for example, materials and data sharing challenges—associated with distributing the workflow across design, build, test, and learn stages.

TA1 performers considered whether and how (i.e., with what method or tool) they could address the following CP-related questions:

1. Are there systematic problems that affect multiple circuit designs?
2. How does the circuit affect the host?
3. Which circuit designs exhibit environmentally resilient operating envelopes?
4. What features predict protein stability?

Their responses described the methods they would apply, the data requirements they would need to model or characterize a relevant construct, and their concerns about the feasibility of answering the question experimentally or computationally. These responses revealed concerns about the scope and cost of the work proposed to test circuit design. For example, TA1 estimated that the depth and breadth of data required to build a suitable machine-learning (ML) training set, and the cost of constructing and characterizing the diversity of genetic parts and sequences needed to predict systemic problems in circuit function, were high hurdles to overcome. In addition, TA1 performers identified the critical need to establish metadata processes, because the nature of the distributed workflow would make it difficult to discern experimental intent from a dataset that lacked such sufficient metadata. Without knowledge of experimental intent and associated metadata, TA1 would not be able to map systemic problems back to features of experimental designs. Lastly, multiple TA1 performers indicated they would need time-series data—that is, data collected at multiple timepoints in an experiment—to be able to assess the environmental resilience of circuits across operating envelopes and identify systemic experimental problems.

We asked TA2 performers to consider their proposed design tools and how they would measure the impact of those tools on CP outcomes. Potential metrics for assessing and improving their designs include planning duration, cost of design, proportion of design from automation, and accuracy of predictions. They also identified the CPs on which they would establish baseline measures (e.g., Quarter 0 [Q0] data, protein stability data generated in the first quarter) and the timelines for establishing baseline measures (e.g., by the end of Quarter 1 or Quarter 2).

We asked TA3 performers to consider the experimental designs required to address each CP. They commented on the feasibility of the designs and notional timelines; and they identified material, instrument, and coordination requirements to conduct the experiments. They expressed few concerns about their capabilities, except for one concern about limited availability of detection devices at the robotic lab.

Aptima coded and summarized performers' responses to enable DARPA to conceptualize group composition, computational/analytic coverage, data-related concerns, coordination and planning concerns, and experimental resources required for each nascent CP (see **Appendix C: Example Survey Responses**). Aptima also used this information to inform the content and process of subsequent quarterly integration (PI) meetings. For example, for the quarterly meeting that immediately followed this survey, we organized sessions to allow performers to surface and work through issues in defining experimental workflows; encoding manual workflows in software; generating, loading, accessing, and formatting data; auditioning TA1 algorithms for analyzing data; and soliciting expert feedback on designs and workflows.

Emergent needs

TA3 Methods, Capabilities, & Limitations

SD2 initially deployed three core laboratories to generate data at scale. These three labs deployed different methodologies and offered different capabilities for conducting experiments.

BioFab was an academic lab in which humans, supported by a software-based lab management system, built synthetic biology components and executed protocols at small scale.

Strateos (formerly Transcriptic) was a fully automated lab in which robots, driven by machine-readable code, executed protocols at scale.

Ginkgo Bioworks was a modular, high-throughput lab that flexibly combined human and machine capabilities to build materials and execute protocols at scale.

In theory, these labs offered opportunities to (1) distribute data generation across multiple sites, increase capacity, and generate data at scale, (2) distribute capabilities (e.g., in materials synthesis and experimentation) to allow more efficient end-to-end workflows, and (3) exercise "harmonized" protocols across multiple sites, to assess reproducibility.

The distribution of work in SD2 dictated that TA3 labs would generate data and deliver it to the shared data and computing infrastructure hosted by TA4. TA1 analysts would then access those data to surface discoveries and inform subsequent TA2 designs. However, interviews with TA1-3 performers in the first phase of the program revealed that TA3 labs were not generating enough data, and much of the data they did generate were of dubious value; therefore, TA1 performers were blocked from deploying their algorithms to discover anomalies and insight. This necessarily impacted TA2 performers' ability to refine their design algorithms, as they were unable to leverage TA1 discoveries. Several specific issues were raised that called for solutions: TA3 was not building strains and parts quickly or reliably; growth-condition data were incomplete because of delays to automated TA2 infrastructure; and some datasets (e.g., small RNA sequencing, Riboswitches) produced by TA3 were unusable. This led us to conduct interviews with performers at each of the TA3 labs to explore more deeply their capabilities and limitations. The following is a description of those capabilities and limitations, and proposed strategies for optimizing TA3 usage.

University of Washington Biofab

The BioFab is largely a human-operated lab. It employs students and early career scientists to execute protocols (i.e., build materials and run experiments). Thus, the work of the BioFab is limited by human constraints—for example, staff availability and staff facility with procedures. To help mitigate one limiting aspect—facility with workflows—the BioFab uses a software application, *Aquarium* (Keller et al., 2019; Vrana et al., 2021), that guides lab technicians (techs) through experimental activities. Aquarium supports standardized protocol authoring and execution by offering predefined, common experimental procedures that can be chained to generate simple or complex protocols. During protocol execution, Aquarium guides users, step by step, through the sequence of procedures (e.g., see *Figure 8*). At each step, it provides information about materials and actions required to complete the step. For example, it identifies a set of candidate materials to use, how much to use, and when and how to use them. Aquarium supports standardized protocol execution by reducing the need for human judgment in executing procedures.

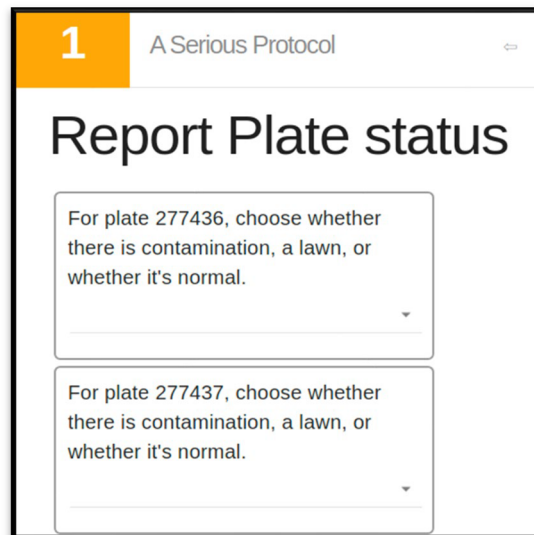
The image shows a screenshot of a web application interface. At the top, there is a header bar with a yellow tab labeled '1' and the text 'A Serious Protocol'. Below the header, the main title 'Report Plate status' is displayed in a large, bold font. The form contains two identical sections. Each section has a text prompt: 'For plate 277436, choose whether there is contamination, a lawn, or whether it's normal.' followed by a dropdown menu. The first section is for plate 277436 and the second is for plate 277437.

Figure 8. Sample partial screen capture of Aquarium user interface

Early deployment at BioFab revealed limitations of using

Aquarium. First, identifying problems in protocols and execution was an inexact and inefficient process. There is no automated gating or error-flagging in Aquarium—thus, if a protocol includes an error, either in the coded procedure or in the scientific methodology, it will be identified only by a human operator and likely only after the protocol has been initiated. In the best-case scenario in which a protocol has an error, Aquarium will crash, or the lab tech will realize that a procedural step is flawed, and the experiment will be postponed or terminated. In the worst-case scenario, errors in the protocol may not be revealed at all, and problematic data can be delivered to analysts.

Although Aquarium is designed to mitigate a lack of scientific proficiency—for example, a shallow understanding of principles and methods of biology—there is still some level of proficiency required to recognize methodological errors that propagate through an Aquarium-supported protocol. The BioFab, like many academic labs that employ technicians at various stages of their education, deployed a staff with varying degrees of competency. During the first year of SD2, the BioFab technical staff comprised four novice and three proficient technicians. According to lab managers, differences in technician proficiency, coupled with the difficulty in surfacing protocol errors, potentially accounted for delaying experimental production of some data by weeks. Lab managers sought to mitigate these issues by reducing errors in protocol execution (e.g., ensuring the right materials and instruments were used) and remediating lab tech performance. However, these mitigations take time and resources. Managers must monitor the technicians' work, engage in after-action reviews, and train (or retrain) technicians to an acceptable baseline of proficiency. This can take weeks or even months.

BioFab experiences errors and inefficiencies particularly when protocols are novel and complex, as was the case with some protocols written to support experiments in yeast. Thus, despite the BioFab signaling that it could participate in all yeast and protein challenges, it ultimately excelled on the Protein Design challenge for which it developed a relatively high-throughput (1,000-sample chip) assay for testing

protein stability. Although they had difficulty building strains and generating data for experiments in yeast, they were able to assay proteins with far more efficiency than could have been achieved by their TA2 counterpart in that CP, the Institute for Protein Design.

Ginkgo Bioworks

The foundry lab at Ginkgo Bioworks offers a hybrid model of experimentation. They have a base capability of human-executed, benchtop methods that they leverage when protocols rely on human discretion, for example, when protocols have not yet been vetted and may be subject to in-process modification, and when high-throughput testing is not necessary.

As with any human-operated workflow, errors can occur and may be discovered only after the work is complete. Ginkgo mitigates the impact of these errors by making their workflows modular. Different teams work on different components of a protocol, so, for example, the team building genetic parts is different from the team inoculating or incubating strains, which is different from the team measuring samples with a plate reader or sequencer. Thus, errors can be discovered midstream and corrected so they do not impact the entirety of the workflow and waste experimental resources. In this case, errors may be tolerable, as they result in only small-scale losses of time and materials.

Ginkgo further optimizes the use of time and resources by incorporating automated modules into their workflows. These modules leverage repeatable, machine-driven methods when protocols are vetted, unlikely to change, and call for high-throughput testing. As with their human-executed modules, each of Ginkgo's automated methods also represents only a fraction of a full protocol's workflow. Thus, a full workflow can comprise several human- or machine-operated modules, chained together to optimize accuracy, efficiency, and throughput.

When considering how best to operationalize a protocol, Ginkgo conducts a (sometimes informal) cost-benefit analysis that considers the flexibility required, estimated error rates associated with methods, and the cost of those errors. When the gain in productivity from automation outweighs the anticipated cost of errors, Ginkgo tends to opt for automation. Thus, even though the overall error rate of benchtop testing may be lower because of the flexibility it affords in sensing and mitigating problems before they devastate an experiment, the limited throughput of benchtop testing may ultimately be more costly.

Relevant to SD2, the high rates of error associated with unvetted protocols (or protocols that require greater flexibility or discretion), render them more suitable for benchtop versus high-throughput testing. By contrast, vetted protocols (or protocols for which flexibility comes at a cost), are better suited for high-throughput testing.

When issues of data quality occurred at Ginkgo, it was attributed to attempts to stand up new protocols—for example, novel RNA sequencing (RNA-seq) pipelines. However, such problems were sometimes difficult to diagnose because Ginkgo's Application Programmers Interface (API) for engaging their automated workflows is not publicly accessible. Thus, prior to the program adopting a standardized experiment request template, TA1/2 performers who wanted to submit experiments to Ginkgo had no window into how those protocols were translated into operations.

Strateos

Strateos offered fully automated, robotic experimental capabilities. Performers could author protocols in *Autoprotocol* (a machine-readable format for expressing experimental protocols;

<https://autoprotocol.org/>) and submit them through a web-based application (cf. the inaccessibility of Ginkgo's API).

Upon receipt and scheduling of an experiment, human lab technicians would load the samples to be run, and then robots would execute the experiments, end to end. This necessarily removed the risk of error and inefficiency associated with human-executed benchtop methods. For example, it allowed for precisely timed operations (such as plate transfer in hysteresis protocols) that humans may not be equipped to conduct. Furthermore, robots can be available to perform when humans may not be (e.g., overnight or for prolonged durations).

However, robotic operations introduce other limitations. Whereas human-operated, benchtop testing accommodates minor modifications to experiments in process, robotic operations are relatively inflexible. Protocols must be vetted and fixed (vs. dynamic) prior to execution. Thus, robot-executed protocols can be costly to stand up and implement. Moreover, those protocols must be written in machine-readable code, and they must contain only those operations that the machinery can execute. This made it difficult (if not impossible) to translate some traditionally human-executed protocols to the robotic lab, because the robots simply could not execute the human operation (e.g., vortexing).

To mitigate this issue, Strateos deploys some benchtop pilot testing prior to transitioning protocols to robot execution. However, all of these steps for generating a vetted, fixed, machine-readable, robot-executable protocol cost time and resources. Because these resources are finite, those spent generating and implementing one protocol can take resources away from another protocol. For example, it was determined early in the program that resources spent transitioning riboswitches protocols would take away from the resources needed for yeast protocols. Thus, tradeoffs were required to allow Strateos to conduct experiments in the most efficient way. They reported to the program that some desirable combination of parameters that optimize flexibility and robustness simply were not possible and that the program would have to navigate those regimes.

Critically, although robotic execution mitigates some of the error and bias associated with human judgment, it necessarily lacks the benefits of human judgment that can rescue a protocol from disaster. For example, a human technician can receive an instruction to place two samples in one well, realize this is an error, and avoid complying with the instruction. By contrast, the robot technician is not necessarily coded for common sense—it will not reflect on the accuracy or logic of every instruction it receives—and therefore will not recognize when a coded instruction contains an error. Moreover, whereas humans can sense when materials and procedures are problematic (e.g., a material smells contaminated, or an attempt at pipetting yields the wrong volume), robot technicians are likely insensitive to the problems unless they are specifically coded and instrumented to do so.

Solutions

Aptima began the task of identifying methods by searching for opportunities where automation would enhance (or replace) methods and move the program forward; however, it became clear that some of the early limitations in the program were a result of automation or poorly coordinated attempts to automate some methods at the experimentation stage. The program was not pushing data to TA1 because (1) protocols were not yet interoperable (i.e., easily ingested by multiple automated workflows) and (2) automated lab capabilities that were supposed to allow for efficient production of data at scale were slow to adapt to the methods that TA1/2 were proposing (specifically, human operations were not

easily coded into machine operations). Through conversations with TA1/2 performers, we identified an opportunity to help mitigate these problems and facilitate pushing data to these performers.

Early in the program, TA5 convened a subset of TA1-3 performers to discuss the limitations at TA3 labs that might impact the transfer of data (e.g., in protocols and plans from TA1/2 and in experimental data from TA3). For example, as noted previously, Ginkgo had no publicly accessible API, making it difficult to write protocols that could be easily ingested by the lab; and data coming out of TA3 were difficult to reconcile because they lacked requisite metadata. Moreover, different labs used operations that yielded data of different granularity; some labs simply could not conduct certain methods—for example, liquid transport—and thus could not support certain classes of experiments; and, critically, some experimental parameters could not be implemented across labs because of differences in instrumentation. These differences significantly impacted efforts to develop protocols that could be tested across sites. Consequently, the program was generating data at scales far below what TA1 needed to apply and refine their algorithms.

To alleviate the bottleneck in generating data for TA1, we explored the feasibility of allowing TA1 performers to run experiments locally, at their benchtop, to (1) generate and push data to TA1 analysts more efficiently and (2) vet protocols for translation into TA3 languages.

Aptima conducted a site visit to a TA1 benchtop lab, where multiple TA1 performers convened to harmonize a protocol for time-series experiments in yeast and explore the costs and benefits of adopting Aquarium. This led to a discussion of a benchtop route to generating data for SD2. The group discussed the potential role of benchtop experiments and the feasibility of developing machine-readable (specifically, Autoprotocol) scripts from benchtop procedures. This yielded a rationale for proposing benchtop experiments and a decision tree for evaluating when experiments should be conducted at the benchtop versus in a high-throughput lab (see **Appendix D: Decision Tree**).

We subsequently applied the decision tree to a question of whether time-series experiments should be executed at TA1 benchtop labs, the BioFab, the robotic lab, or at some combination of the three. This surfaced challenges in harmonizing a protocol. Although the time-series protocol could be executed at the BioFab, the lab would have to make significant changes to Aquarium (and its user interface) to enable time-cued experimentation. The protocol could not be faithfully deployed at the robotic lab because that lab uses different incubation and sampling methods. Whereas the authoritative, benchtop-based protocol called for “flask in a bath” incubation, the robotic lab deployed “on the plate” incubation and sampling. These different methods yield different temperature effects on samples. A flask in a bath can be kept at a constant temperature, whereas transfer of a plate from one environment to another for sampling will cause temperature fluctuations for the sample. In addition to the conflicting temperature parameter, high-throughput testing of the time-series protocol can be prohibitively expensive, especially if measuring performance using RNA sequencing (RNA-seq). Thus, performers derived the following principled strategy for coordinating time-series experiments: two TA1 benchtop labs would execute a small-scale flask-in-a-bath protocol and the robotic lab would pilot test a low-throughput plate-incubation protocol. This would allow for some data to be generated via the original protocol while a modified high-throughput candidate protocol could be tested for compatibility and generalizability.

Optimizing automated experimentation

The activities that generated the decision tree and associated guidance for proposing benchtop testing also generated discussion on how to support translating human-executed benchtop protocols into machine-readable, machine-executable code. This discussion specifically focused on the steps required to enable benchtop technicians to produce and use Autoprotocol scripts, and candidate operations for pilot testing a translated protocol.

At these meetings, Aptima and a TA1 performer attempted to translate a benchtop protocol into Autoprotocol. In doing so, we surfaced multiple hurdles to accurately and completely translating the protocol. First, there were procedures prescribed by the protocol that simply could not be translated into terms recognizable by Autoprotocol. For example, there are explicit stop rules in the benchtop protocol that could not be coded for robots which are not sensitive enough to know when to apply the rules (such as “stop extracting materials in the aqueous phase”). Thus, we had to generate terms (“boxes”) for protocol steps that did not exist in Autoprotocol. This raised a critical question of whether it is possible, practicable, or even desirable to attempt to translate flexible benchtop procedures into Autoprotocol and thus into rigid, potentially unfaithful, automated operations.

To address this question, Aptima, with the participation of TA1 and TA3 performers, proposed a test case for translating a flexible benchtop procedure to Autoprotocol for execution in the robotic lab: *Experimental Decision-Making Based on Optical Density Observation and Measurement*. To uncover the requirements for translating this typically human-executed operation to an automated process, Aptima applied Cognitive Task Analysis (CTA). We used CTA because we wanted to avoid the unnecessary consumption of resources—for example, setting up an automated protocol—to learn that a laboratory instrument was incapable of performing a necessary action. Moreover, we wanted to “train” robots to behave like expert scientists, to know when to modify a protocol or simply to know more precisely when to take subsequent steps in a protocol. CTA was ideal for eliciting and making explicit the implicit expertise in benchtop procedures.

To accelerate data generation (and thus experimental cycles), the program tasked automated TA3 labs with conducting the same experiments that humans conduct; however, this proved to be problematic. Human lab technicians often conduct experiments in which steps of the protocol are contingent upon observations and judgments that do not occur at standard intervals or in standard sequence. Materials to be measured will grow at different rates and therefore must be assessed for candidacy for measurement, otherwise one risks measuring materials that are unsuitable for measurement. Automated labs are not necessarily equipped to conduct experiments that require such flexibility, as they are only as flexible as their hardware (e.g., their sensors and moving parts) and their software (e.g., the code that operates the machinery) will allow. For example, once a protocol begins, automation will complete it unless there is some failure of the apparatus (e.g., mechanical failure or the code is unreadable). A problem with stimuli or materials (e.g., no live cells to assay) will not necessarily cause an experiment to be aborted. Automation also suffers from limited sensitivity. For example, machinery is not equipped to sense stimuli in ways required to assess the following:

The feasibility of continuing experiment based on a proportion of live samples

Whether to enact a subsequent step and which step to enact

Whether multiple samples on a plate need different treatments (e.g., some need concentration while others need to be diluted)

Consequently, early in the program, TA3 executed experiments in which they took RNA-seq measurements on wells that had no viable yeast cells in them, costing time and money. We sought to mitigate the risk and cost of this type of error occurring in future experiments.

For the CTA activity that informed the test case, the focal experimental procedure to investigate was human measurement of optical density (OD) of live cells in a colony of yeast to (1) determine the count (and concentration) of live cells and (2) trigger the next steps in a protocol. The CTA workflow is presented in **Appendix E: CTA Outline**. This activity revealed the following:

- Procedural steps for measuring optical density

- Sensors and senses used in the procedure

- Critical observations made by the human lab technician

- Cues to which the lab tech must be sensitive and respond

- Background knowledge required to interpret those cues

- Ranges of possible interpretations

- Possible courses of action given a particular interpretation

To determine whether OD measurement was a candidate for automation, we submitted the results of the CTA to TA3 performers and asked them to consider whether each element of the procedure could be coded for automation. Because OD measures hinge upon observation via microscope, a critical question was, how could machinery observe or sense the health of cells in requisite resolution and do so from multiple perspectives? TA3 proposed that current automated Live/Dead Flow Cytometry measurements could serve as a proxy for these observations, as there was an ongoing effort to apply machine learning to all channels of flow data to identify cells that were live, dead, and various states in between (e.g., sick, near death). Moreover, it was suggested that an imaging cytometer could be a proxy for a microscope and amenable to automation. These solutions proved sufficiently feasible to advance to detailed conversations about authoring and pilot testing a protocol.

Subsequent discussions among representatives from TA1-5 revealed that the harmonized Yeast States experiment was a suitable candidate for adopting an automated OD measure. Moreover, the automated OD measure could be tested as a mitigation step toward identifying and reducing variance in experimental results that, to date, were unexplained. Performers from all TAs contributed to this test case. TA1 described the experimental protocol in sufficient detail to be represented in Autoprotocol and provided growth-curve data to support decisions about the timing of sampling; TA2 identified the data requirements for submitting an experiment request (and associated changes needed for the experiment request template to represent this workflow); TA3 implemented the necessary changes to the harmonized protocol; and TA4 identified requirements for metadata processing. Ultimately, this activity generated a pre-experiment workflow for determining when to kick off an experiment. Conceptually, it was an automated method of measurement-based decision making. The output of the method was discovery of a source of variance—cell count—that informed new protocols to eliminate that variance. Thus, this activity convened performers from all TAs to identify automation-relevant friction points,

develop a solution for mitigating those friction points, and implementing a solution that led to reducing the amount of uncertainty in the experimental data.⁴

Program-wide automation

A fundamental goal of the program was to incorporate more automation into scientific methods, to enable more rapid production of data at scale. In many instances, automation did just that. For example, it dramatically reduced the time required to analyze proteins for stability and binding properties; it reduced the number of humans required to move data through a full experimental cycle of testing circuit performance in yeast; and it significantly enhanced the reliability of (and trust in) data made available for sophisticated analytics. However, automation came at a cost. To foster a balanced perspective of the costs and benefits of automation in SD2, Aptima hosted a session at the July 2020 quarterly meeting, titled “Tales of Automation.” This session featured several panelists representing all TAs, who shared their experiences developing or deploying automation on SD2. Aptima interviewed these panelists prior to the session to identify salient aspects of automation, positive and negative, that may not have been perceptible to the broader SD2 community. The following are insights gleaned from those interviews and the subsequent session.

Tradeoffs involving discovery

At the beginning of the program, automation was proposed as a solution to reproducibility challenges. It was hypothesized that automating design, build, test, and learn (DBTL) loops would enable reproducibility and enhance discovery. However, automation and reproducibility are not necessarily correlated. As the novelty of a protocol increases, so does the difficulty and cost of automating it. Costly investment in automating some protocols necessarily constrains the way those protocols are implemented and deployed. Those constraints translate to specific, rigid processes that can then constrain discovery modes.

Tradeoffs involving reproducibility

Whereas some biology—for example, molecular biology—is conducive to reproducible automated workflows, dynamic cell biology may not be. In molecular biology and biochemistry workflows, samples are well mixed and rapidly equilibrate. By contrast, in workflows with live cells, the cell states are constantly changing. In high-throughput workflows with multiple strains, cell phases or states vary between strains due to differences in dynamic properties (e.g., growth rate). Thus, to support reproducibility, one should consider (as the SD2 program did) adaptive protocols with measurement-based decisions (e.g., OD measures). The trade-off, however, is in potentially losing throughput and generating fewer data. Moreover, there are experimental procedures for which operational knowledge (e.g., how to hold a pipette) is implicit and therefore will not be codified in an automated workflow. Even if it somehow gets codified, it must be recodified every time it changes.

⁴ Of note, the activities described above, as well as other informal conversations, led to repeated calls for a method or application for translating across protocol languages. Toward the end of the program, an initiative began to develop an application to translate protocols between Aquarium and Autoprotocol, the *Open Protocol Interface Language (OPIL)*.

Tradeoffs involving human capital

Automating the various workflows of an experimental cycle can enhance efficiency and productivity. Ultimately, automation force multiplied humans working on SD2—for example, by helping to discover and solve mistakes, such as syntactic mistakes in protocols, and by freeing humans from spending time conducting mundane, repeatable tasks. However, the work of automating workflows is resource intensive. Automating protocols from designs, data-processing applications from requirements, and analytics from hypotheses takes many human labor hours, particularly as workflows require recoding across versions. Moreover, troubleshooting these efforts in automation is difficult, as it often requires human observation to reveal sources of error. Automating new processes introduces many new failure modes, and simply interrogating associated records is insufficient to identify the causes of errors. For example, adjusting flow rate and sampling mechanics (e.g., the depth at which a pipette draws a sample) in one protocol resulted in greater consistency in experimental results; however, identifying those sources of inconsistency was impossible simply by interrogating the data. Perhaps a more strategic use of automation is called for, for example, by first employing automation where its impact can be multiplicative, as in surfacing errors. Automation should absolutely support scientific discovery and it should support the people doing that work. Recalling the previously described tradeoff involving discovery: automation was a goal for enhancing discovery on SD2 and it may have been a suboptimal goal. Perhaps the goal for automation for SD2 should have been to build an environment that supports rapid, accurate execution of human-conducted work.

Data Accessibility

Roughly one year into the program, Aptima asked a subset of performers what they perceived to be the barriers to entry to using the program’s infrastructure. The most salient response, which represented the sentiments of many, was, “I do not know who to track down to find out where data are and what those data represent.”

In late 2018, Aptima conducted interviews with TA1 performers and members of the ETL/Data Representation working group. The interviews focused on issues of data accessibility and quality. The interviews revealed that using instruments set up for data accessibility (e.g., the data catalog) was sometimes difficult; for example, it was difficult to map data back to metadata, such as sample identifiers (IDs) in the catalog. In addition, there were multiple data formats in use, the conventions for generating them were unclear or inaccessible, and it was unclear whether the multiple data formats yielded similar file types (e.g., RNA-seq had multiple pipelines outputting potentially different file types). Thus, when data were available, TA1 analysts were unsure how to use them, and they needed domain experts to vet and approve the data before operating on them. Consequently, there was a request for methods to make explicit the implicit knowledge about data and metadata.

TA1 performers also expressed frustration that they lacked awareness of the status of experiments (e.g., requested, scheduled, and completed). Performers had access to experiment request documents; however, it was difficult to identify a subset of experiments associated with a request. Thus, TA1 performers formally requested an artifact for tracking experiments, and ideally an automated solution for surfacing the relationships between experiment identifiers (IDs) and their parent experiment request.

Concurrently, Aptima convened TA3 and TA4 performers to discuss data validation. TA3 reported, correctly, suspicions that some performers had developed a mistrust of data and a reluctance to use

data from the TA4 database (the data catalog). Thus, a validation scheme was proposed to help boost confidence in data. Aptima and TA3/4 discussed the following salient issues that needed to be addressed:

Lab-originated errors must be found quickly and addressed.

Products of data extraction, transformation, and loading (ETL) must be evaluated for their ability to address metadata-related questions.

Some TA1 pipelines needed tabulated data, but not necessarily standardized tabulated data. For example, data normalized in certain ways made it difficult to observe and understand nuances (e.g., gene expressions) in the data.

Some data-generation processes may not have been stable enough to ETL in an automated way.

One potential solution to many of these issues was to allow different groups to ETL data by their own processes, in their own lexicon. This was critical to allowing TA3 and TA1 to maintain their own data representation schemas. However, it required a post-processing effort to ensure data were findable, accessible, interoperable, and reusable (FAIR; Wilkinson et al., 2016). These post-processing efforts were time consuming and error prone. They often required a set of three people to implement, and because the process comprised several translation points, it was error prone. This represented an opportunity to improve processes through automation. Thus, TA4 and TA1 performers collaborated to spearhead ETL and metadata representation efforts. These efforts addressed issues in translating one group's representation scheme to another's. They also addressed questions of data validation, including how to collect and encode rules for data validation, how to make the process adaptable, when data should be validated, and who should be data stewards. Metadata processing and data quality checking eventually became automated processes that enhanced the speed at which TA1 received high quality data and metadata.

Measuring Methods

To better quantify the impact of research and engineering methods on SD2, Aptima worked with program performers to measure SD2 processes and outcomes. 1: On Selecting & Assessing Challenge Problems of this report comprehensively describes the methods we used to measure the program's achievements in its scientific and technological thrusts. Here, we report qualitative measures taken at the midpoint of the program and quantitative measures for one transition use-case/system stress-test that occurred near the end of the program.

Methods, Assumptions, & Procedures

At the midpoint of the program, Aptima captured qualitative measures of the impact of research and engineering efforts. Performers documented actions that were productive (termed "what we did well," below) and actions that should be taken to improve performance ("what we can do better"). Aptima aggregated the items in the documents, examined them for themes, and coded each item as a member of a single theme (see *Figure 9*).

Results & Discussion

SD2 performers generated 72 lessons and 96 sub-lessons, of which 49 captured "what we did well" and 119 captured "what we can do better." The majority of lessons learned concerned (1) establishing and negotiating metrics, goals, and priorities, (2) communications, (3), SD2 systems, and (4) data quality. The

most troublesome themes (those dominated by “what we can do better”) were about establishing metrics, goals, and priorities and improving SD2 systems. **Figure 9** presents the frequency of lessons learned by theme. Following **Figure 9** is a subset of items that were particularly relevant to the issues discussed in the previous section.

Metrics, Goals, & Priorities

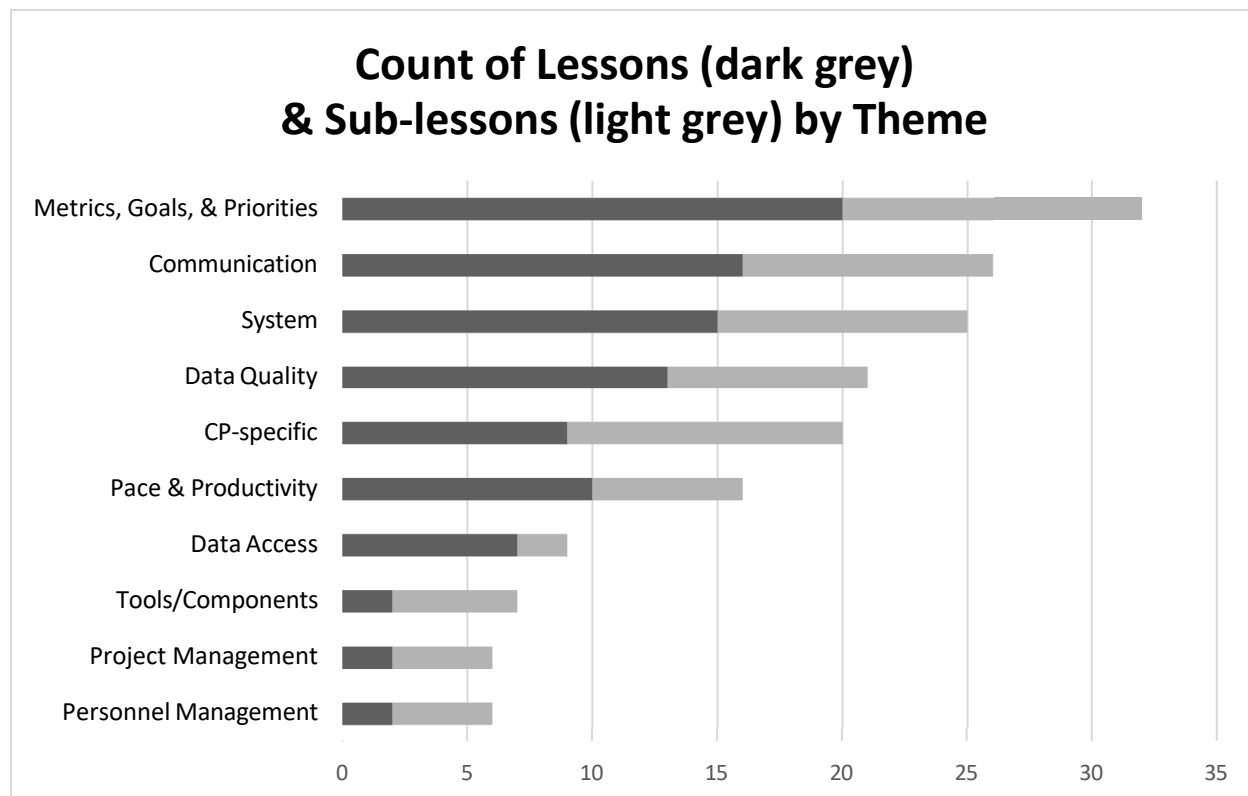


Figure 9. Count of lessons and sub-lessons by theme

What we did well

The Novel Chassis group reported having iterated within the bounds of a cohesive theme, allowing them to optimize processes (e.g., RNA-seq preps), protocols (e.g., time points, iterates), analytics (characterizations of circuit components), and design.

What we can do better

The Yeast States group reported that they could better leverage scrums and monthly coordination of working groups to better prioritize their goals and work.

The Perovskites group identified opportunities and benefits associated with promoting “Big Goal Conversations.” For example, generating ideas outside of PI meetings creates time for “vision” conversations on a quarterly time scale. They noted that a benefit of ensuring that the goal of the CP is well stated is the ability to measure progress. However, they needed better definitions of “success” within the scope of the research. For example, what is a “good” prediction? What is the impact of using new versus old data? They wanted to define “good” and revise their dashboard accordingly. They also

noted that experimental pauses—taking time discuss experiments in between cycles—can promote better research.

The Protein Design group identified a need for a 2-year roadmap, informed by expertise and the needs of all stakeholders.

The Novel Chassis group identified a need to iterate more while engaging less process. They also perceived the need to prioritize data streams (e.g., by leveraging expertise of others in the program). Finally, they identified a need to better communicate goals and expectations, stick to a theme, and accelerate cycles (e.g., by preparing analysis plans prior to data becoming available).

Communication

What we did well

The Yeast States group reported that that pre-scrum documentation of discussion topics, organized by task, allowed them to maintain focus during meetings and situation awareness of progress between meetings.

The Perovskite group reported that interactions with other TA1/2 performers—for example, at PI meetings and through weekly conversations—allowed them to have collaborative “think big” conversations, stay on the same page, and maintain progress.

The Novel Chassis group learned the value of well-written experimental requests. They also found value in developing an “ecosystem of ideas” that supports (1) freedom to suggest new experimental approaches, measurements, and analyses, (2) unrestricted authorship/ownership of experimental requests, and (3) democratic processes for choosing the group’s direction.

What we can do better

The Yeast States group learned that the process of sharing and discussing data is cumbersome and difficult. Data and discussions shared across multiple channels (e.g., Slack, direct messaging, and other documentation) are difficult to track. They identified a need for a single forum for data sharing and discussions.

The Perovskites group perceived the need to query the experimental team on what are the “views” of the data that best serve experimental needs, and supply experimentalists with analysis feedback to prepare them for forthcoming insights. They also identified the need to provide experimental scope and variety to analysts.

The Protein Design group noted that pipelines were being used by non-SD2 actors; hence, SD2 performers need to better understand the sources of their data.

System

What we did well

The Protein Design group reported that the lab pipeline and experimental assay worked well and that their ML tools for understanding proteins were successful.

What we can do better

The Yeast States group reported that the DNAseq pipeline was non-existent. They reported a need to get buy-in from labs to provide pipelines for full-assembly optimized for environment, to help

interrogate genomes. They expressed a need to include design as input to pipelines to allow analysis and discovery.

The Perovskites group reported that they would benefit from more automated approaches to assessing outcomes. Tasks such as stock solution preparation and outcome scoring were, at the time, human operations. They also reported that the velocity of getting their pipeline running was good, but it made some parts of the process brittle—they spent some time thinking about how to generalize, be more robust, and involve better engineering.

The Protein Design group noted that automating the processing pipeline would help support situational awareness about the availability and purpose of new data. They also wanted to automate data processing scripts and creation of structural metrics (that currently run manually).

The Novel Chassis group indicated a need for better integration of circuit design with the data infrastructure, to include stubs that ingest circuit part information, circuit design/interactions that are integrated and annotated in SynBioHub, and hooks for analyses to extract information.

Data Access

What we can do better

The Yeast States group identified a need to find a better process for capturing discussions, versioning data, tracing data, and optimizing workflows to search and discover plots and other analysis artifacts.

The Novel Chassis group identified a need for centralizing access to data frames and job status.

Data Quality

What we did well

The Perovskites group made consistent improvements in data quality, including more consistent experimental outcome characterization, record keeping, and retroactive analysis using metadata records.

The Novel Chassis group stabilized data acquisition, formatting, and metadata annotation processes. Stable data upload formats enable optimization of post-processing. Concerted data analysis efforts enabled metadata QC checking.

What we can do better

The Yeast States group identified a need to do data versioning, as dynamic data frames make it difficult to track new versus old data. They perceived a need for a pipeline solution and a reduced latency (< 24 hours) between lab and pipeline, and they wanted a hybrid between pipeline and data processing versioning.

The Perovskites group wanted to generate less data “for the sake of data.” They wondered, when does one know when they’ve reached the point of generating superfluous data? That is, when do you stop and just analyze the results that you have?

The Novel Chassis group requested metadata quality control (QC) checks. They perceived value in receiving a post-ingest confirmation that “the bits you flipped were the ones you wanted.” They suggested a “data nose,” comprising integrated flags in the post-processing pipeline, that determines the “smell of the data.”

Pace & Productivity

What we did well

The Perovskite group was the primary contributor to this category. They commented on the cycles of their weekly cranks that allow iteration on ideas on a short timeline, provide structure for experiments, enable planning and iterative improvement of experiments, ensure conversations between TAs, and ensure rhythm of experiments is suitable for high throughput. Quick software fixes enabled the *Escalation* dashboard to work well. They reported that ambitious software development in a short time span delivered world-class capabilities.

What we can do better

As noted above, under *System—What we can do better*, the Perovskite group recognized that the fast pace of getting the pipeline running was good but also made some parts of the process brittle. They were also generating data faster than they could write papers.

Summary

This exercise in identifying and discussing lesson learned was valuable, as it provided performers with insight into what worked, what had not worked, and what could work better through that point in the program. It resulted in concrete action plans to improve workflows and the metamodels (e.g., collaborative and integrative methods) that supported those workflows across technical areas. For example, the Perovskites group took steps toward making Experiment Specification, Capture and Laboratory Automation Technology (ESCALATE) more general (see also 6: On Transition); the Protein Design group further automated their analysis pipeline (resulting in dramatically faster data analysis); the Novel Chassis group successfully lobbied for an automated QC process; and the Yeast States group contributed to improved processes for capturing and tracking metadata. These last two enhancements to the SD2 system eventually supported an effort to transition (and test) a large chunk of the SD2 pipelines. We describe that effort in the next section.

Quantitative Measures

The data presented in *Task 1: Select & Assess Challenge Problems*, comprehensively describe the utility of automated tools developed on the program, as well as the impact of the methods of SD2 on the speed, accuracy, complexity, efficiency, and quantity of science conducted on the program. Here, we describe one additional experimental thrust for which we collected quantitative data.

Toward the end of the second phase of the program, members of TA2-4 composed a working group that engaged with a synthetic biology lab at the Air Force Research Lab, to stress test a substantive and transitionable set of SD2 pipelines “round trip” research concerning Cell-Free (CF) Riboswitches. The proposed test involved asking an external partner to submit an experiment request, deploy the robotic lab to generate data, and access the data via SD2’s data infrastructure. Aptima supported this working group by engaging with performers to identify potential sources of error in the workflow, monitor iterations of the workflow, and measure performance across those iterations.

The CF Riboswitches work involved multiple SD2 performers and pipelines. The high-level workflow was as follows:

1. Air Force Research Laboratory (AFRL) ordered DNA synthesis from Ginkgo.
2. Ginkgo ordered DNA synthesis from their partner, TWIST, and delivered it to AFRL for purification.
3. AFRL purified the constructs and shipped them to Strateos for testing.

4. Strateos tested the constructs and uploaded the data to the SD2 data infrastructure.
5. AFRL pulled the data from the SD2 data infrastructure for analysis.

The questions of interest, besides the focal scientific hypotheses, were how well the performers could coordinate across three labs, streamline through programmatic interfaces, and accelerate production of experimental results over time. Critical outcomes were delivery of good-quality data to AFRL and hardening of a robust set of data pipelines. Thus, we sought to investigate whether the pipelines supported the science and whether the science of CF Riboswitches is a good context for proving the pipelines. In observing and documenting this test, Aptima engaged performers in pre-submission interviews and post-hoc reflections of completed experimental cycles.

Performers wanted a relatively pure test of the pipelines. Whereas most of the work of SD2 occurred within the context of a comprehensive socio-technical system, including via collaborative communication (e.g., through regular scrums and communication platforms), this test of the pipelines was planned with no regular meetings and only minimal communications, primarily via Slack.

A critical metric of success was the time to complete a full cycle. The first cycle was expected to take roughly 10 weeks: four weeks to synthesize and deliver DNA to AFRL, four weeks to purify and deliver DNA to Strateos, and two weeks to run experiments at Strateos.

The first iteration took a total of 146 days to complete. However, that timeline includes 87 days that elapsed between when the design document was made public (i.e., posted to Slack) and when the experiment request was submitted. It also includes the following steps and elapsed times:

Time of submission of DNA request to Ginkgo until receipt of synthesized DNA at AFRL = 31 days

Time of submission of purified DNA and reagents to Strateos until experimental run = 18 days (Run 1) + 51 days (runs 2-4)

Time of final experiment run until data upload = 11 days

Among other issues in logistics, there was a problem with the first DNA order sent to Ginkgo. It contained repeated DNA requests, which took time to discover and troubleshoot. After this first issue, Ginkgo deployed a mechanism for checking the accuracy of the order (e.g., to ensure the requested DNA constructs were synthesizable, there were no repeat orders, and no forbidden structures).

There were delays in AFRL submitting the initial experiment request, which required gaining familiarity with the request template, interfacing with SynBioHub, and writing new code. Also, there was a metadata issue resulting from different time points expected based on the experiment request and those output from the Strateos protocol. A potential mitigation strategy proposed for future cycles was a dummy data frame that allowed the user to check expectations against the actual data frame.

After the first cycle, the AFRL user estimated that subsequent cycle times would benefit from more efficient communication and greater facility with the SD2 components and processes. An interesting issue of scale also arose. This exercise generated a large volume of data for a user accustomed to small volumes of data. It took time to recalibrate methods and expectations. The user commented, "I don't typically do as big of a batch of DNA preps at once; this will take time to increase efficiency."

After addressing issues with the workflow and reducing some of the socio-technical friction that Cycle 1 encountered (e.g., for troubleshooting template completion and experiment submission), Cycle 1.5 (a

slight modification of Cycle 1) took 47 days from the time the experiment request (ER) was submitted until data were processed through data converge (for quality checking). That cycle still incurred errors—the ER had to be submitted multiple times—however, it took less than one-third of the time of Cycle 1. The AFRL user estimates that the SD2 pipelines accelerated her experimental cycle by roughly two-fold. Thus, the test of this chunk of SD2 pipelines was a moderate success, as it (1) demonstrated that an external user could engage (albeit with some friction) with the SD2 infrastructure and (2) proved that the pipelines could enable acceleration of experimental cycles. However, this test also surfaced some limitations to transitioning these pipelines, as the use of Google as a platform for hosting some tools was problematic, and the effort required substantive input from multiple members of the SD2 community to troubleshoot emergent issues (see also **6: On Transition**).

Insight in Discovery and Design

Aptima engaged in a diverse set of activities to support collaboration and integration, and to help performers find the most propitious paths to discovery. We facilitated the selection and evaluation of the scientific challenges that guided the work of the program (see 1: On Selecting & Assessing Challenge Problems); we conducted knowledge elicitation activities to identify methods to advance that work (described in this chapter); we helped develop, maintain, and measure the socio-technical system (STS; see 3: On Socio-Technical Systems in Scientific Enterprises); and we oversaw the evolution of the technical infrastructure (see 4: On System Evolution). Through these activities, we learned where insight can be found, generated, and leveraged to advance discovery and design in the context of a distributed socio-technical system. Here, we present some of the more salient insights gleaned from that work.

First, the novelty of the structure of SD2 cannot be overstated. SD2 represented a paradigm shift for many performers on the program, who were accustomed to conducting science in an individualist paradigm. Individualist science is characterized by an individual or small group working within the resource constraints of their own organization to design, test, and analyze constructs. It is a useful paradigm for addressing relatively simple scientific problems but is inadequate for solving complex problems that require multiple experts, diverse resources, or data at scale. Such problems call for the collaboration and integration of capabilities and resources similar to that of the STS established in SD2.

The STS approach of SD2 was fundamentally different from individualist science because it enabled performers to share knowledge and insight at the level of data rather than at the level of conclusions. That is, it allowed performers to discuss the value and validity of methods, data, and interpretations before completing an experimental cycle. In theory, this allows performers to identify and remediate problematic methods before they consume valuable resources, and to identify problematic data before they produce invalid and irreproducible results. However, this was not always the case in SD2. In some instances, the program completed full experimental cycles only to discover that the data were faulty. In other instances, early in the program, data were generated but they were not interoperable because they lacked sufficient metadata. These issues arose in part because simply providing the platform for aggregating knowledge at the level of data is insufficient if those engaging at that level are unaccustomed to articulating their idiosyncratic conventions (e.g., lexicons, schemas, and workflows) and reconciling them with the sometimes-incompatible conventions of their counterparts. Thus, one discovery-related insight gleaned from SD2 is this: one can convene experts and integrate capabilities, but careful attention must be paid to the interfaces between them. That is, one must ensure that collaborators fully articulate their conventions and the requirements of those conventions, and that

counterparts understand and acknowledge those conventions and requirements. A program can support this engagement by inserting mechanisms (people and processes) to dive deeply into organizations' conventions, fully describe them, and curate them for integration. This activity should happen before substantive S&T work is done on the program.

In addition to the novelty of the paradigm, the size and scope of SD2 was larger than many performers had experienced previously. For example, materials chemistry is usually conducted by small groups (or even single investigators). The materials chemistry team on SD2 comprised nearly 20 members. Some working group leads (e.g., for the Yeast States CP) managed groups comprising members from more than half a dozen organizations whose work was distributed across the four stages of the DBTL loop. The learning curve for leading and operating in such large, distributed teams had to be overcome before cross-organizational friction could be reduced enough to allow coordinated science to occur. Therefore, it was critical to provide support—such as meeting facilitation and coaching for group leads—to help performers effectively manage the types of teams that convened under this collaborative paradigm of science.

Specific to distributed science, passing workflows across organizations introduces challenges that might not exist when a single organization conducts a workflow end to end. For example, pushing protocols to TA3 labs often blinded protocol authors to experiment execution, as they did not have access to observe experiments in progress. Thus, when protocols malfunctioned or data did not match expectations, the protocol authors were induced to execute and troubleshoot those protocols at their own benchtop. This strategy hindered diagnosis to the extent that benchtop conditions did not replicate foundry conditions, and at any rate, such duplicative work defeats the purpose of employing distributed capabilities. This issue recalls the decision tree Aptima developed for guiding selection of benchtop versus high-throughput modes of experimentation. Candidate protocols for pushing to external labs (high-throughput or otherwise) should be carefully vetted for the tradeoff between the cost (in time, resources, rework) and the yield of that push. This also highlights the importance of cross-education between performers about each organization's capabilities and those of other organizations.

The amount of data to be generated in SD2—petabyte scale—was enormous. Such enormity was required by analysts' ML algorithms to surface anomalies and yield discovery. However, an enormity of data is only as good as the quality of those data. In SD2, data quality was evaluated on at least two dimensions, including:

- Validity—the extent to which the data represented the constructs and outputs described in the experiment requests and the conditions of the source experiments.
- Completeness—the extent to which the data represented all measured samples and the extent to which the data were fully described and annotated.

Both dimensions are critical, and performers experienced issues with both. Validity issues occurred when uploaded data violated expectations and were discovered to be invalid (e.g., when ambient light at the robotic lab contaminated fluorescence measures or when the foundry lab measured wells containing no viable samples). Validity issues, especially those stemming from experimental procedures, were costly but also relatively straightforward to address (e.g., by adjusting instrument parameters or adding a layer of observation to ensure procedural accuracy). Issues of data-completeness were far more complex. Completeness issues stemmed from the challenges associated with making implicit

knowledge explicit and then reconciling those explicit knowledge structures (i.e., conventions) across organizations. These issues manifested specifically in the form of metadata issues (see also 5: On Metadata). Without metadata—for example, information about experimental intent, instrument parameters, or experimental conditions—analysts could not interpret the focal data and therefore could not operate on those data. These issues arose at the start of the program and persisted long enough to engender mistrust of the data, underuse of the data, and underuse of the infrastructure that hosted those data. This persisted until the Data Representation and ETL/Analysis Platform working groups developed solutions, including post-hoc translations of data conventions and automatic quality control measures. These solutions eventually enabled distributed analysts to access metadata and operate on unified data frames (i.e., to “compare apples to apples”). Trust in data is a necessary antecedent to discovery. SD2 performers established that trust only after they could confirm that data were valid and complete.

In SD2, flexibility also proved to be an antecedent to discovery. Flexibility of program goals and structure, scientific methods, and technical solutions, were critical to allowing SD2 performers to advance the science of synthetic biology and materials chemistry. Early in the program, automation was envisioned as the enabler of discovery, as it would reduce human engagement (and thus reduce associated biases and error) and accelerate the pace of science. Indeed, it enabled many of the measures of achievement reported in 1: On Selecting & Assessing Challenge Problems. It also came at a cost (see *Program-wide Automation*, this chapter), and it could not have advanced without the flexibility of the socio-technical structure of SD2—that is, of people engaging with people, updating architectures (forms, code), catching and correcting errors, committing time and resources to ensure systems worked as needed, and, critically, identifying manual solutions to complement (or enable) automated solutions.

The program also demonstrated flexibility in realizing that some solutions were untenable, working toward identifying more productive paths, and, critically, abandoning untenable solutions in favor of more promising ones. The program demonstrated that a one-size-fits-all approach to some elements of S&T work is suboptimal. For example, in attempting to pass all experimental protocols through one experiment-planning application, the program experienced a severe bottleneck that was alleviated by taking a more flexible, distributed approach to planning. To be flexible enough to generate data across diverse structures (i.e., organizations and their processes), the mechanism for planning data generation must also be flexible (or adaptable) enough to ingest diverse protocol structures and export plans compatible with those diverse recipient structures.

Moreover, no individual should be the sole source of operations, system modifications, or configuration necessary to run a critical step in a workflow (or set of workflows). In an STS, discovery should not be contingent upon the resources and availability of any one performer or organization. Regardless of the mechanisms by which data are generated and shared (e.g., by humans, robots, human-machine teams), there will be sources of error. No single organization will have the capabilities to conduct all possible workflows and do so without error. Hence, a flexible paradigm that leverages the expertise and outstanding capabilities of each organization in a consortium is optimal. In SD2, the socio-technical system embodied such a paradigm. It provided the platform and processes for collaborative discovery and design.

Conclusions & Recommendations

Aptima explored the methods that performers applied to their scientific and technological work in SD2. We also explored the metamethods—for example, the decision making, troubleshooting, and communicating—that evolved to enable productive S&T work. In doing so, we revealed limitations to capabilities, inefficiencies in workflows, and other hurdles to productive S&T work. However, we also helped reveal alternatives to suboptimal approaches, as well as opportunities to capitalize on untapped capabilities and resources. We found that science and engineering, even when applied through principled methods, can be unwieldy and error prone. Its conduct, especially when performed in a complex STS, requires a flexible structure, and regular care, maintenance, and recalibration. One must revisit goals and methods for achieving those goals at a cadence that balances the time needed to test solutions but also to fail fast and correct course.

In light of the lessons described in this chapter, Aptima recommends the following for future programs that endeavor to apply similar STS-based methods to S&T:

- Identify and deploy a performer (or organization) to curate capabilities early in the program. Front-end efforts to identify and organize the detailed capabilities (and associated limitations) that performers bring to the program should allow the program to diagnosis and mitigate incompatibilities and enable rapid synergy.
- Deploy a diverse set of knowledge capture activities to identify the details of performer methods that are critical to supporting productive S&T. In the same way that making implicit knowledge explicit is critical to sharing knowledge and enabling discovery at the level science, making methods and metamethods explicit is critical to sharing knowledge (e.g., procedural, semantic, and institutional knowledge) that enables collaboration.
- Build flexibility into the structure and goals of the program. Allowing performers to organize and reorganize in groups, develop, abandon, and redevelop solutions, and work creatively to identify and fill gaps in workflows will lead to more productive and successful groups.
- Allow performers to fail and encourage them to fail fast. Many professionals across industry and academia perceive failure as a negative outcome. This may obstruct their ability to view failure as an opportunity to improve or advance their work; consequently, rather than see and embrace failure, they push back and persist in an unrecognized failure mode. SD2 experienced both conditions; and, not surprisingly, when performers recognized failure, accepted it, and sought opportunities to improve, they experienced success much more rapidly.
- Carefully consider the goals of introducing automation as a solution. It can enhance efficiency and free humans to engage in more fruitful activities, but it comes at a cost in set-up time, diagnosis, and flexibility. Automation should support S&T outcomes and it should support the work of relevant human operators and stakeholders.
- Consider early in the program the generality of the tools and platforms developed and used by the program. If those tools and platforms are to be transitioned to external users, the ability of those users to engage with those artifacts on their problems is critical.

Science and engineering, even when applied through principled methods, can be unwieldy. If it is to be conducted via an STS, it requires deep forethought, regular maintenance, and recalibration. One must identify goals and methods early and revisit those goals and methods at a cadence that balances the time needed to develop and test solutions but also to fail fast and rapidly correct course.

3: ON SOCIO-TECHNICAL SYSTEMS IN SCIENTIFIC ENTERPRISES

Overview

All the great problems that face our world today have both technical and human content—the one intermingled inseparably with the other. (Simon, 1978).

Historically, DARPA programs have brought together the brightest minds to compete over profound scientific and technical challenges. Many programs are evolving to activities in which these minds collaborate, rather than compete. Inter-disciplinary teams working collaboratively can make discoveries that are impactful and rapid, arguably more than scientists and engineers working alone or in competition. The value of such collaboration is evidenced by the recent growth in the number and professional diversity of authors per scientific paper (Greene, 2007), attested by the National Research Council (2015), and evidenced in the effects of SD2 on many metrics (see 1: On Selecting & Assessing Challenge Problems).

DARPA conceived SD2 as an enterprise that leverages teams and technology to accelerate scientific discovery and design. Here, the teamwork consisted of scientific collaboration conducted in an interdependent fashion by multiple individuals, loosely structured cross-institutional teams, and teams of teams on difficult scientific Challenge Problems. The technologies spanned tools for collaboration, experimentation, and computation. Together, these constituted the SD2 Socio-Technical System (STS).

An STS is classically defined as a composition of organizational structure, actors, technology, and tasks (Leavitt, 1965). In finer detail, an STS is "any organisation [that] employs people with capabilities, who work towards goals, follow processes, use technology, operate within a physical infrastructure, and share certain cultural assumptions and norms" (Socio-Technical Centre at Leeds University, 2022).

The field literature concerning these complex systems emerged in studies of the interplay between humans and their techniques in manual labor, such as the organization of miners to harvest coal walls, where different mineral conditions (such as hardness) warrant different mining teams and techniques (Emery 1959). Later research focused on the evolution of factory work from systems of human-machine interaction to systems of human-computer interaction. Zuboff's (1996) analysis of this transformation at a paper mill is a seminal work (see also Klein, 2014).

To define and develop the SD2 STS, Aptima applied five techniques. We (1) evolved the program structure, (2) implemented collaboration contracts, (3) structured the design and selection of challenge problems, (4) designed and orchestrated program activities, and (5) provided training and coaching that fostered collaboration amongst inter-disciplinary teams and team members. These techniques helped SD2 systematically build a collaborative, interdisciplinary community of biologists and chemists, machine learning experts, and software and system developers. Future programs may benefit by replicating aspects of the structure, process, culture, and technologies, which bound that community together into a productive whole.

In this chapter we focus on the pragmatics of STS development and management. The methods section of this chapter describes the SD2 STS structure and the processes that enhanced collaboration across professional and institutional boundaries in this program. The results section presents performers' insights and feedback concerning the SD2 STS. We elicited those insights at a meeting held near the end of the program to define the SD2 STS from several perspectives, characterize its culture and

technologies, assess the limitations of that STS and the alternatives to it, and address its transition or deployment⁵. We close with recommendations to the organizers and sponsors of STS for future science and technology missions.

A brief video (Garrett, 2022) complements this chapter, and conveys the spirit of many of the performer comments below. The video presents interviews with selected SD2 performers and DARPA program leaders summarizing the nature and value of the SD2 STS (Garrett, 2022; **Appendix F: STS Video Transcript**). The video is publicly available at this website, where we hope it will inform and incite other programs to adopt the STS design and management methods of SD2: <https://www.aptima.com/sts/>. In addition, we have documented aspects of the SD2 STS in articles published or submitted (Freeman, Leins, Bell, et al., 2019; Leins et al., in review), and in a guide for program managers and coordinators (see **Appendix G: Program Coordination Guidebook**).

Methods, Assumptions, & Procedures

This section describes the work Aptima conducted to build the SD2 STS, in collaboration with DARPA and the program performers. We then describe the design of the STS Summit. It produced performers' observations that we present in the results section.

Program Structure

DARPA designed the foundational structure of SD2. DARPA organized the program into two tracks—biology and chemistry—each of which consisted of five technical areas, as follows (TAs; see **Figure 10**):

- TA1: Scientific Discovery Algorithms
- TA2: Design Algorithms and (in the biology track only) Experimental Planning Algorithms
- TA3: Experimental Facilities
- TA4: Data and Analysis Hub
- TA5: Challenge Problem Integrator

⁵ This chapter reports work executed in multiple contract tasks: Task 3: Enhance Collaboration, from the base contract; and Host the SD2 Socio-Technical Summit and Document the SD2 Socio-Technical System in a contract modification.

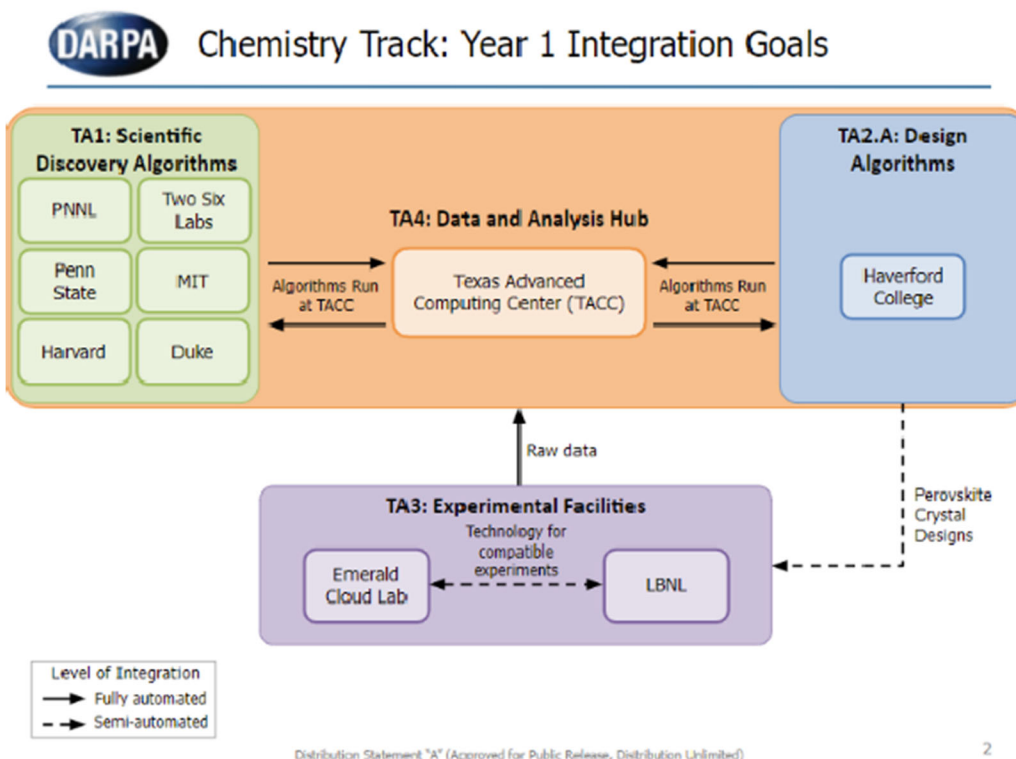
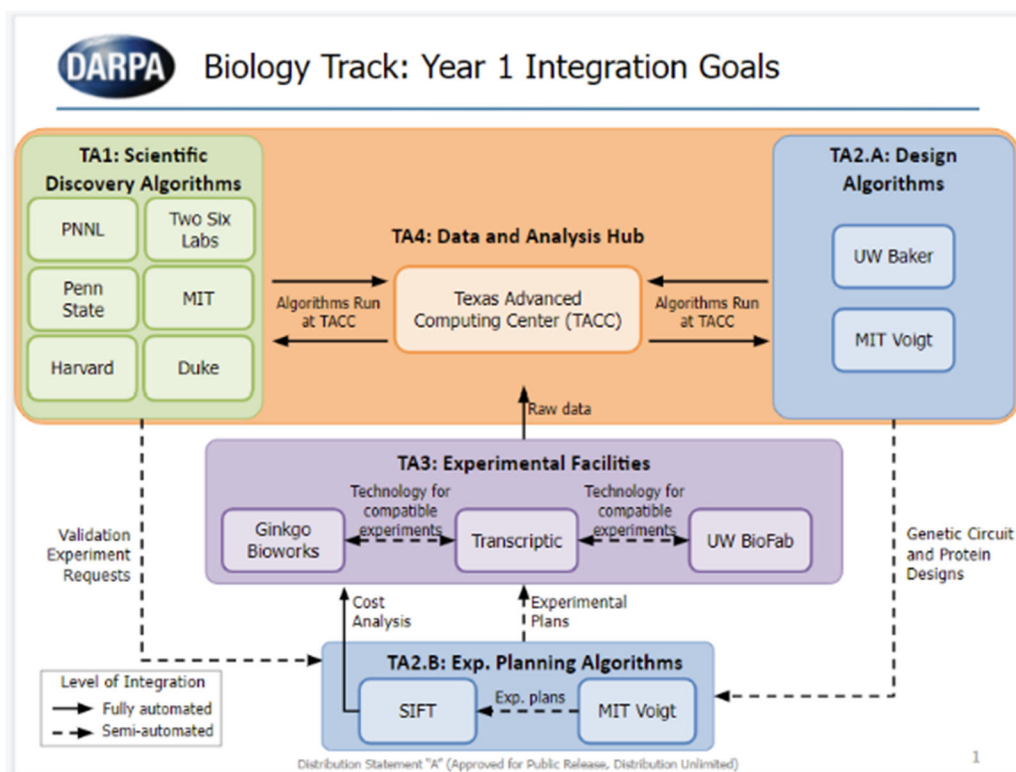


Figure 10: The structure of SD2 Technical Areas and the teams that populated them, in both the biology and chemistry tracks of the program.

DARPA's plan was to fully divide performers into these TAs to incentivize development of technologies that would share algorithms, experimental plans, experimental technologies, and data across the Technical Area (TA) boundaries. As the creator and original program manager of SD2 attested (Garrett, 2022; see also **Figure 10**):

“One of the ways that we address that challenge inside of SD2 was to very intentionally separate (a) people who were designing an experiment from (b) the ones who were running it (c) from the ones that were collecting and aggregating the data, and (d) the people who were ultimately analyzing it to understand what scientific insights could be drawn from that data. This exposed the places where all of this implicit knowledge needed to be explicitly represented.”

Crossing that complex structure were program-wide Challenge Problems (CPs). Three, in biology, concerned protein design, engineering in yeast, and engineering in bacteria. The chemistry track addressed a single CP: perovskites science.

Some 23 unique academic and commercial institutions participated as performers in the program, two of them (MIT and University of Washington) on multiple teams. In sum, SD2 supported 16 institutions as primes, and 10 as subcontractors (see **Figure 10**).

Aptima's role in structural design was to ensure that working groups were established to execute the scientific work of the program (the CPs) and the technology development that both supported that work and would, in some cases, transition to institutions outside the program.

Collaboration Contracts

The SD2 culture was built in part through contracts, such as the Associate Contractor's Agreement (ACA) and the SD2 Publication Consortium. Aptima negotiated the ACA in meetings represented by the Principal Investigators (PIs) and/or their institution's legal counsel. The basis of the agreement was a model document from the Air Force contracting office, which performers amended to satisfy their institutional constraints and objectives. The ACA spelled out the obligation to share data, and constraints in doing so (e.g., with respect to proprietary data).

The SD2 Publication Consortium specified the rules for allocating credit—authorship or acknowledgment—in publications funded by SD2. The primary objective was to fairly credit creators of data, tools, and infrastructure for their contributions, just as scientists receive credit through authorship for their development of theory, experiments, and findings.

Challenge Problems

Formulating challenge problems (CPs) was a key method for surfacing research and development opportunities that leveraged performer capabilities to meet program goals, and for evaluating the impact of SD2 solutions. As described in 1: On Selecting & Assessing Challenge Problems, we developed and applied a collaborative technique for defining CPs to engage performers in work that would reward both the individuals and the collective (Freeman, Leins, & Bell, 2019). The method paired bottom-up generation and selection of CPs by performers, with top-down control by DARPA. In an early program meeting, 38 of the program's biologists, chemists, mathematicians, and computer scientists generated roughly two dozen candidate CPs, pared them to a dozen to reduce redundancy and test generalizability, and ranked their top choices. A ranked-choice voting formula was applied. Five CPs were highly scored. The program sponsor subsequently directed

researchers to address six CPs, including the elected five. These CPs were the scientific focus of the program in its first two years, and strongly influenced subsequent, more organically defined CPs.

While executing these CPs, the CP teams evolved their focus and organizing structure to leverage performers' existing capabilities, optimize the speed toward solutions, maximize scientific and technical impact, and realize synergy across performers. For example, periodic revisions of CP leadership helped achieve these efficiencies and allowed the program to manage the workload imposed on leaders, offer leadership opportunities, and address team effectiveness issues.

Activities

To facilitate deep knowledge integration of the inter-disciplinary team members from different disciplines with different cultures, languages, and research practices, the program conducted a variety of activities in which team members could apply their unique knowledge and skills to a shared research problem. At the most granular level, Aptima orchestrated weekly program meetings, supported weekly working group meetings, designed and directed quarterly meetings, and held special events. Such events included a two-week hackathon at the beginning of the program, a data-sharing activity that applied the program's data analysis pipeline to data from potential transition partners, and an open science conference focused on making data and analytics as publishable and well-published as scientific methods and findings.

DARPA program leadership held regular one-on-one meetings with Principal Investigators and their teams to monitor technical progress, answer questions, and discuss performer concerns. These meetings also served as an opportunity for DARPA leadership to assess the spirit and frequency of collaboration of performers, to identify opportunities for collaboration, and reinforce the value of working collaboratively.

Regardless of the type of meeting, thoughtful planning and execution were critical to success. Aptima employed the following methods to plan productive meetings and enhance collaboration before, during, and after a meeting. We note that SD2 kicked off before COVID with live meetings but conducted its final 18 months of sessions entirely by video conference. The methods below were robust across in-person and virtual meetings.

Before the Meeting: Clear, well understood meetings goals are the sine qua non (i.e., the essential condition) for any successful and efficient meeting, particularly for meetings with large interdisciplinary teams. Aptima worked closely with DARPA to craft goals that advanced program aims while leaving room for performers to have agency (i.e., control) in the work that was conducted. Aptima engaged in the following activities for each meeting or event to address program and performer needs:

- **Define meeting goals:** Elicit meeting goals from the PM and PIs, top-down and bottom-up. Select those goals that support the mission of the program, as well as the aspirations and frustrations of performers.
- **Design the products of meeting activities:** Generate templates or clear instructions with which meeting groups can report the issues and recommendations they address in WG and breakout groups.
- **Design meeting activities:** Design sessions whose members, missions, instructions, and duration enable performers to generate the product you need.

- **Define the participants:** Identify the speakers, leaders, and key participants for each session in advance. Aim for a distribution of participants across teams. Contact them early to engage them in the planned activities.
- **Schedule “open time”:** Reserve considerable time on the agenda for ad hoc meetings of WGs, teams, and other interest groups to generate new research concepts while sitting side-by-side or in Zoom breakout rooms, to work out research processes on whiteboards, and to recruit collaborators into plans that benefit the program.
- **Calculate logistics:** Ensure that start and end times, room capacity and layout, audio/visual setup, dining time (for video conferences) and food and beverage options (for live meetings) are conducive to a productive meeting. It may sound banal, but our experience is that a hungry or under-hydrated group tends not to do their best thinking or collaboration.

Of these activities, the most critical was the design of activities to engage performers in meeting goals. For example, DARPA’s concern that performers candidly identify impediments to SD2 work led to the invention of sessions called “Pain Diaries”, in which performers told short stories concerning significant challenges, and attendees contributed diagnoses and potential cures.

We allowed 8-10 weeks to complete the activities above, using the workflow illustrated in **Figure 11** below. The activities in this workflow often involved cycles of review and revision concerning goals and activity design, which we addressed primarily in meetings between DARPA and Aptima.

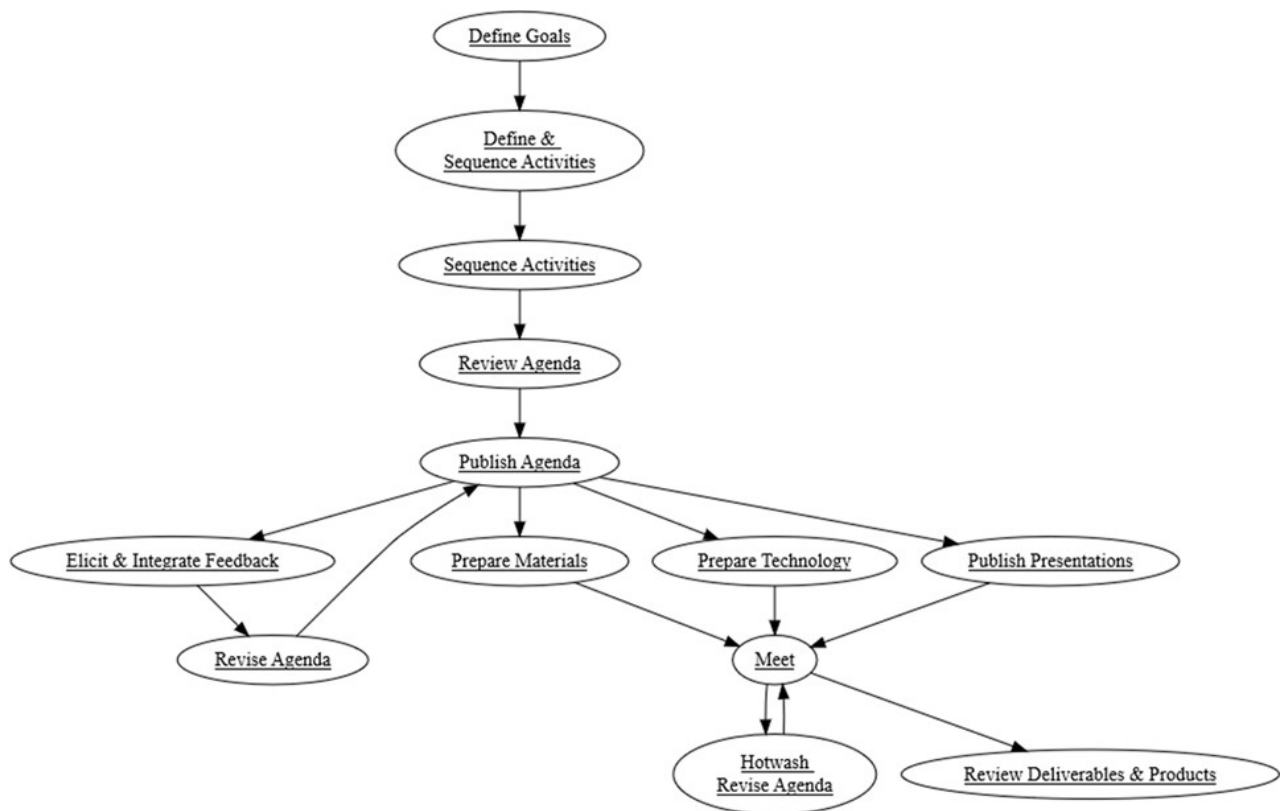


Figure 11: The activity planning process for SD2.

During the Meeting: Events such as hackathons and integration meetings are dynamic affairs. Logistical and interpersonal complications can arise. Therefore, creating the conditions for collaboration—clear goals, psychological safety to try new things and fail without negative repercussions, and to acknowledge not knowing something—are antecedents for developing innovative ideas and solutions. We used the following practices to conduct productive and collaborative meetings:

- Begin the meeting with a clear statement of the meeting goals, a review of the agenda, and ask for clarifying questions about the goals and agenda.
- As the event progresses, sense and respond to cues that indicate the group would benefit from deviating from the schedule. For example, a group that was animated and engaged but quickly becomes lethargic and disengaged may benefit from an unplanned break or change of topic.
- Try to stay on schedule. Inevitably, some speakers will run over their allotted time or a data analysis session will uncover issues in the data that must be resolved before follow-on sessions can begin. Consequently, you must determine whether the day's (or subsequent days') agenda should be revised. Consider whether and how revisions—for example, cutting sessions versus extending the day's end time—will impact performers.
- Do not let the schedule be oppressive. Discoveries get made, unexpected opportunities for collaboration arise, or perhaps differences of opinions need to be settled before the group can move forward. Any one of these can make a session go beyond its allotted time. Cutting the session short in these instances can do more harm to collaboration and esprit de corps than sticking to the schedule. These can be moments when the benefit of extending a session or adding a new session on-the-fly far outweigh sticking to the schedule. The long-term benefits and costs of going off schedule must be considered. Participants appreciate the willingness to flex the schedule so they can take advantage of opportunities to solidify their work. Decisions to make schedule changes need to be communicated clearly, quickly, and across several communications platforms (including the daily agenda) so all meeting participants can learn of the schedule changes.

Facilitating Interpersonal Interactions: To grow the spirit and practice of collaboration during meetings, we engaged in the following practices to optimize the interpersonal interactions that drove the SD2 work forward:

- Ensure that participants have a voice and are heard, and that all are respectful.
- Identify tensions within the group and determine how and when to address them. Note: The key to enabling and enhancing collaboration is to address tension and friction directly; most participants feel it, and left unaddressed it undermines safety, trust, and the desire to work closely together. This tension can diminish the willingness of individuals to externalize their cognition, which is what fuels identifying potential opportunities to collaborate.
- If a discussion becomes confused:
 - Stop the discussion, summarize the areas of agreement and disagreement, and ask for confirmation before resuming the discussion.
 - Ask speaker(s) to define important words and concepts, especially those that might have different meanings across disciplines.
- If a discussion becomes contentious:
 - Ask speaker(s) to summarize their understanding of what was previously said.

- Remind speaker(s) to ask clarifying questions before stating a contrary opinion.
- If a discussion is dominated by one perspective:
 - Ask performers who have not contributed recently to the conversation to share their thoughts.
 - Validate the contributions of the dominant group members and encourage them to reduce their talking time, to allow others to contribute.
- As sessions approach their end time:
 - Identify a good stopping point for the discussion, summarize progress, and articulate the prioritized list of topics for any follow-on discussions.
 - Clarify consensus decisions, actions, and actors.

After the Meeting

It can be difficult for those orchestrating and facilitating to learn well from events. They rarely see all of an event, and they can never see the event through the eyes of the participants, who the event serves. Participant feedback and feedforward helps address errors in the current event and plan the next one. Aptima followed these practices to learn from the meetings and events we planned and conducted:

- Elicit feedback after an event, feedback focused on the technical and emotional:
 - What went right? What activities or sessions produced good insights, products, or partnerships?
 - What went wrong? Which activities or sessions were unproductive?
 - Were the time and preparation sufficient? Did you have the time you and your colleagues needed to get work done?
 - What was the high point of the meeting? The low point?
- Elicit feedforward—objectives and advice—for the next meeting. We asked DARPA leadership, PIs, and WG leads what they want to accomplish or resolve, and what activities will help or hinder their progress. Notably, we have received at least as much feedback about past meetings by asking performers to look forward as we do asking them to look back.

Meetings at most levels—from program-wide to CP—benefit from ground rules concerning how decisions will be made. In SD2, these decisions often concerned which research paths to follow and where to invest resources. Aptima worked with performers to define decision-making processes that were appropriate for the group or task to establish those processes and reinforce them. For key decisions, each CP and WG often declared in discussion or agendas when, how, and by whom decisions would be made. This enabled performers to contribute to decisions and to know that they have been heard (even when decisions were made by another party). Some CPs used the same decision process over time, while at least one CP used a variety of decision processes (hierarchical; consensus; voting) depending on the team task and situation.

Training and Coaching

Any complex organization has roles and practices that can be difficult to learn and to put effectively into practice. Aptima provided explicit instruction about and modeled behaviors that promoted communication and collaboration in the WGs. These instructions and behaviors concerned:

- Expressing curiosity about the assumptions, mechanics, and effects of novel or challenging viewpoints.
- Listening closely and asking clarifying questions before disagreeing.
- Asking questions that open dialogue instead of making statements that close conversation prematurely.
- Identifying friction and using it as an opportunity to develop deeper understanding, even when agreement cannot be reached or is unnecessary to move forward.

When a WG was unable to prioritize its goals or actions, we coached the WG lead to bring them to DARPA leadership for discussion and prioritization. On the rare occasions when WGs struggled to build or maintain a collaborative spirit and meetings were characterized more by conflict than collaboration, Aptima interceded in public conflict and brought the individuals involved into private discussion to mediate the dispute.

The program did not offer formal training in leadership as recommended in the National Research Council report on team science (NRC, 2015). This was arguably a shortcoming of the program, and certainly an omission that might have handicapped the program if other scientists or engineers had taken the helm. However, the rare succession to leadership within the program did not test this gap in training.

Finally, the program developed a library of technical documents to facilitate both onboarding and cross-disciplinary communication.

Assessing the STS

To assess the SD2 STS built using the methods above, we designed and hosted a video conference to elicit responses from SD2 performers to several key questions concerning the SD2 STS:

1. What is the SD2 STS?
2. What is the role of technology in the SD2 STS?
3. Alternatives to the SD2 STS
4. Deploying the SD2 STS

We asked each of the dozen participants to focus their comments on one or two of these questions. We then facilitated discussion after the prepared remarks concerning each question.

The SD2 STS Summit participants were DARPA program leadership, Aptima program staff, and key members of the program's technical staff. Joshua Elliott (Program Manager) and Susan Buckhout-White (Systems Engineering and Technical Assistant, SETA) represented DARPA. The attending performers were Daniel Bryce of SIFT (TA2), Enoch Yeung of University of California, Santa Barbara (UCSB; TA1), Jacob Beal of Raytheon BBN (TA1), Jed Singer of Two Six (TA1), Joshua Schrier of Fordham University (TA1), Mark Weston of Netrias (TA1/4), Matt Vaughn of TACC (TA4), Matt Wood of Aptima (TA5), Niall Gaffney of TACC (TA4), Peter Lee of Ginkgo Bioworks and formerly of Transcriptic (TA3), and Steve Haase of Duke University (TA1). Jared Freeman, Drew Leins, Sterling Wiggins, and John "JCR" Colonna-Romano of Aptima, Inc. (TA5) designed, hosted, and facilitated the five-hour video conference on October 4, 2021.

We conducted a qualitative analysis of the recorded transcript of this meeting by identifying quotes that represented significant insights or that clearly articulated important aspects of each topic. The quotes

below are drawn largely verbatim from a transcript of the Summit, and in several cases from a transcript of the STS video (Garrett 2022; **Appendix F: STS Video Transcript**). We have intentionally not attributed quotes to individual participants. We present findings from the Summit below.

Results & Discussion.

What is the SD2 STS?

We turn first to definitions of the SD2 STS. We do so from three perspectives: its structure, its function, and its cultural character. We then present performer perspectives on the techniques we used to found and develop the culture of the SD2 STS.

Definition by structure

Performers' comments concerning the structure of SD2 concerned an organization deliberately designed by DARPA to require collaboration; the SD2 STS techniques and technologies helped performers meet that requirement. In their comments concerning this structure at the SD2 STS Summit, performers defined the SD2 socio-technical system in terms of the integration of individuals, organizational members, structures, processes, and tools. One performer, in an interview for the STS video, summed up the SD2 STS crisply:

A socio-technical system or STS is a collection of people and technology. The system is all about how they interact. The idea [in SD2] is to bring experts in machine learning and artificial intelligence (AI) to gather with those who are experts in synthetic biology and material science. The underlying assumption was, at the core, we need to be able to share data.

A more artful perspective on the STS resonated with several performers:

Taking the music analogy of what we're trying to do. We basically currently have people playing their own music in, you know, their different labs just across the country. It would sound discordant, right? You would have, you know, just everyone playing to their own tunes. How do we ... contribute to that larger symphony?

Two Summit participants characterized the STS in terms of its evolutionary character, as a convening of existing STS, or processes, as a systematically nurtured organization.

The SD2 socio-technical system is a collective of existing social and technical systems that coalesce into a larger one.

[The SD2 STS was a] network of collaborative relationships between people, the programmatic structures that encourage the formation of those relationships and nurtured them, and the technologies that empowered and facilitated those relationships.

Even components of the program normally considered purely technological had a distinctly social component. The provider of much of the computing infrastructure noted this:

Most of the resourcing [i.e., funding] was oriented toward staff time, so an objective was to provide white-glove access to a research computing ecosystem that had unparalleled capacity, so that people never felt limited by their computing infrastructure ... The overarching objective was to build the most integrated version of tech systems, services, and people that we'd ever attempted.

Definition by function

Summit participants also defined the SD2 STS in terms of the technical and social functions it enabled. One highlighted the services that SD2 technology provided, services that might normally be provided primarily by people, and services whose value and salience varied between performers.

All some performers know or see is that some samples were tested under a certain set of conditions. To some the STS is the ability to use Slack and email, and maybe some extra request documents, to collaborate with a bunch of other organizations. To others, the STS involves generating and providing access to metadata, common data frames, and analysis pipelines. Some see something even larger than that, such as the automated round-trip infrastructure, that's going from the inkling of an experiment design to execution, to tracked data, to analyses, and the next experiment planned in a multi month campaign.

Another participant noted that the SD2 STS served the interests of science (not just of program members) by creating metadata and tools that facilitate replication and extension of the SD2 research.

The experiment that you're doing today has to be discoverable by somebody else 10 years 15 years down the line, so they can understand what we did and what we learned from it. Making tools that make that possible is so important.

Other participants, focusing on social functions, reported that the STS enabled performers to define scientific goals that served science, the program, and themselves. The SD2 STS enabled performers to “negotiate common goals. In science, this in part means to create systems of hypotheses across performers. Such collaboration is ‘unnatural’ in the natural sciences, at least in biology.” The negotiation enabled performers to collaborate for the benefit of the program, “...on challenges that that an individual performer could not address by themselves, e.g., high-throughput experiment workflows that individual performers could not complete in a single lab.” While these collaborations served the program, they rewarded individual members as well. One participant noted that, “the structure of SD2 gave us the opportunity to do cool, interesting things that might not otherwise be possible.”

Definition by character

The SD2 STS, like any community, developed a culture. Several participants described the STS primarily in terms of three cultural attributes: collaboration, candor, and credit.

Collaboration was designed into the SD2 STS in the form of carefully delineated Technical Areas that required “cross-border”, technically-enabled communication; Challenge Problems that drove those communications towards scientific goals; and meetings that joined performers at the program level, the CP working group, and team. Collaboration and attitudes about this culture varied between participants. As one participant noted:

We had people who were extremely collaborative, who wanted to be ... sharing as much as they could, and being transparent about it. We also had people who were a bit more reserved, who wanted to ... develop their work internally, keep failure modes private, debug, and then present their polished output at a quarterly meeting. That traditional model is problematic because it delays others’ awareness of work that could be useful to other efforts. Fortunately, for many, the approach evolved over time to become incredibly transparent and coordinated.

The design of SD2 technologies also drove collaboration, in part by constraining teams from doing independently work that should be shared. As one participant noted:

You're not allowed to just do an experiment. You can do it because you received an experimental request that went through a structured system. You're not allowed to just analyze the data hot off the instrument. It has to go into some system and then come out. The practice for scientists in general is to cut corners. We don't even know if things are worth doing the first time, much less the 10th time and so there's this kind of culture of cutting corners. The fact that that was explicitly discouraged was really powerful.

A prominent and valued feature of culture on the SD2 STS was candid discussion, “a culture of discussion that valued disagreements, rather than sweeping them under the rug.” One might expect candor among scientists, whose methods focus on disproof of hypotheses. However, the marriage of candor and collaboration is not universal in large research programs. Participants described this culture of candor as follows:

A collaborative spirit that allowed communication of failures, to put them front and center. It changes the kind of scientific problem you tackle, because if you're worried about failure modes becoming public, then you don't take on big problems. Because of transparent failure modes, we asked ambitious questions, like can we do data-driven systems biology?

This notion of small components being well integrated and failing fast—we did that and that is not an academic way, you do not talk about your failures, you do not get promoted for your failures, ... yet that's the way that the real world does work. It was very important to this project that we were allowed to get two years into the program and chuck a whole system overboard and start over.

This culture was not an easy one to adopt, one performer noted, “As high functioning engineers and designers, it's often difficult to say we're stuck because we don't know what to do or because something's broken or we need something from someone else. It was nice to normalize that.” Nor was it present from the start of the program. Rather, it was engineered in part through facilitation practices that trained performers to both listen and question in different modes, such as listening for the gist, for logic, for emotion, for viewpoint, and for weaknesses to rebut (Gaunt, 2020); and asking “clarifying questions” to increase understanding. Said one performer:

With every group of people, there's going to be initial friction as they learn to communicate, to listen to, and to understand each other. The group aspect of the culture turned out beautifully. We developed a tremendous rapport and camaraderie as a consequence of the positive culture, a culture of listening to understand.

Candor arguably freed performers to radically revise their goals, techniques, and technologies. As one performer noted, “SD2 as a group was boldly unafraid to step back and try something new, to just wipe off the whiteboard and start over.”

The SD2 STS was also strongly characterized by a practice of giving credit and offering help. One performer noted that:

Stitching all of this together was an abundance of gratitude and credits. We had countless technical calls, a lot of high stakes conversations, but credit was always given. This is really validating, and it fuels a willingness to stay on ... and a willingness to just help. ... People reprioritized and deprecated their own work to help others, because the collaboration was important to them. It's really hard to maintain that level of engagement over four years.

In sum, these SD2 performers defined the program's STS as a fabric of social practices and technological artifacts from which emerged a culture of critique, credit, and collaboration.

Creating the STS culture

Summit participants remarked on the design techniques that fostered the program's culture: collaboration contracts, challenge problems, activities, and training and coaching. (Performer perspectives on program structure, the remaining element of our STS method, are presented above.)

Collaboration Contracts: The ACA succeeded in reinforcing DARPA's intent that performers share data and collaborate over it, so much so that the agreement was not invoked to enforce data sharing during the program.

The Publication Consortium, as noted above, specified the rules for allocating authorship and acknowledgment in SD2. As one Summit participant explained:

Different levels of work or effort could be fairly credited in different publication types: the individual level of contribution or authorship, working group level, and the consortium level of all SD2 performers. Consortium publications [which credited all performers (c.f., Freeman, Leins, Bell, 2019)] truly helped underscore the fact that some of the work of SD2 was so collaborative that the question who's done what is impossible to answer.

Performers' notices of publication plans, registered in the logs of the Publication Consortium and announced in Slack, cued peer review, because these notices alerted performers to authorship opportunities. Centralizing those publication and authorship plans also facilitated reporting of publication counts to DARPA.

Challenge Problems: The SD2 Challenge Problems fostered the program's culture, as well. Above, we described a structured method for eliciting CP ideas, collaboratively shaping them, and prioritizing them for PM review, revision, and approval. Performers noted that this CP definition method brought significant social and cultural benefits to the program, in addition to technical direction. As one performer explained:

We gave a three-minute pitch in front of the whole crowd, but it wasn't about the pitch, it was about all the discussions that happened before that pitch. It was getting to know who you want on your team and who are you need to pull off something.

Another performer noted that the CP's defined an incentive to collaborate, "organizing programs around challenge problems with voluntary participation is great. It's not a zero-sum game. You win if the other participants win as well."

Activities: Performers attributed the culture of this STS in part to SD2's many meetings at many levels. The personal interactions—whether in-person or on Zoom during Covid—communicated personal

competencies and interests, fostered commitments between performers, built trust, and generally incentivized teamwork and further collaboration.

A two-week hackathon, held early in SD2, was a powerful catalyst to collaboration in SD2, according to Summit participants. The event was designed with presentations to convey the competency and interests of performer teams; with analysis activities that gave proof of competency and demeanor under stress and revealed interesting scientific and technical requirements; and with other activities in which performers learned the program manager's values—a distribution of effort over fast wins and substantial scientific discoveries—by applying them to potential Challenge Problems. Said one performer:

The Q0 hackathon was a powerful and great experience. It was transformative and very different. Here was this this room of experts from lots of different backgrounds and expertise, presented with an opportunity to learn something new, to learn some new problem and to learn some potential new application of what we had. It was very fertile ground. The act of discovery or creation is something that we love, being in that space where there are tremendous opportunities and the ability to act quickly on them. There was also this freedom to find opportunities that unlock the most value for everybody. I didn't just shop my capability or widget, I felt I could make it better by working with your team. That sense of freedom and mutual growth was really important. It also felt totally voluntary.

The hackathon was disruptive as well as uniting. As a Summit participant noted, "a lot of the younger folks wanted out of the program after their first week. We had to calm them all down when we got there, but [the Q0 hackathon] was invaluable. Q0 allowed us to learn a lot about each other."

The program conducted other one-time events, such as a data sharing conference in which performers used the program's data analysis pipeline to analyze data from potential transition partners and generate insights. This event encouraged collaboration between performers and transition partners.

Program PI meetings were held with unusual frequency relative to other programs—once each quarter. The schedule presented the risk that SD2 would exhaust precious resources in mere discussion. The program avoided this by casting meetings as opportunities to debug social and technical systems (several meetings featured a "Pain Diaries" session to expose friction and resolve it) and to design the program's future. Intentional design of these meetings gave them value, as several various Summit participants noted:

On most other programs, when you get together for the program meeting, everybody's talking to DARPA about what they did the previous period, and everyone is talking past each other.

These meetings featured reports from cross-organization working groups rather than individual, contract teams. It reduced the focus on individual organizations and emphasized the working group.

The intentionality [of the quarterly PI meetings], the effort spent by DARPA and TA5 in tailoring the agenda of each meeting to the current and immediate future state of the program was critical. It wasn't the same meeting, over and over again. Every meeting was different. Without that intentionality it would have been less effective.

The program activity schedule also featured weekly meetings by the various CP and technology working groups (WGs). These meetings defined the detailed plans for the program and tracked progress with a strong focus on both crediting achievements, identifying failures and blockers, and solving problems. In addition, CP and working group leads, DARPA, TA5 and some other performers joined in a weekly scrum to summarize CP and WG progress and issues, and to adjust course (e.g., from scientific and technological progress in the first several years of SD2, to completing and transitioning data and technologies). As one participant noted, the weekly program scrum “was a great way for people to get genuinely useful progress reports and to understand why things weren't moving when they weren't moving. It really normalized publicly airing being stuck.”

Finally, DARPA program leadership used one-on-one meetings with Principal Investigators and their teams to monitor technical progress, answer questions, and discuss performer concerns. These meetings also served as an opportunity for DARPA leadership to assess the spirit and frequency of collaboration of performers, to identify opportunities for collaboration, and reinforce the value of working collaboratively, and assess the state of the SD2 STS.

Beneath these formal activities was a continuous “drumbeat” of communication and interaction. Said one Summit participant:

We had quarterly meetings (for integration and results reporting), we had weekly coordination meetings (for organizing the work of the CP), we had scrums (for regularly surfacing progress and blockers), and we had Slack messaging to debug issues. Regular communication established a drumbeat by which synergy was created, maintained, and course corrected. That regular communication was critical to coordinating day-to-day operations and maintaining synergy.

Training & Coaching: The SD2 STS was a dynamic social system, just as it was a dynamic technological system. Membership in the SD2 STS changed over time, particularly within the academic institutions. The responsibilities of leaders and individuals who filled those roles also changed. Summit participants noted the complexity of the CP and WG leadership roles they filled:

There is extra work that goes into being a leader. There's a lot of reporting, a lot of consensus building, a lot communication that goes into it. Having a network of great people is helpful. We all worked toward the same goal.

I felt more like a mediator than a leader ... I was trying to integrate everything that was going on, to figure out which parts of that were pertinent to SD2.

Performers in leadership roles are not always natural leaders. They may have difficulty identifying and organizing work, enlisting and motivating team members, and spotting and resolving conflict. Aptima participated in WGs to assess WG coordination and dynamics for emerging opportunities to counsel and coach. Aptima provided personal coaching to some leaders as they worked through these challenges. Aptima also intervened with several performers who had difficulty participating cordially and collaboratively in the voluntary groups. Two adopted new behaviors and continued as valued members of the program; one left the program.

Similarly, the program offered only modest support for onboarding new members (e.g., new graduate students and post-doctoral staff) to non-leadership positions. Said one Summit participant, “In terms of organization and onboarding, it can seem insurmountable for a new person to come in and get up to

speed. Without emphasis on that you tend to become insular.” To proactively address this, the program developed a library of scientific and technical material relevant to its Challenge Problems. That effort did not have obvious benefit or a vocal user community; it was abandoned halfway through the program. It remains unclear what materials would have smoothed onboarding.

What is the Role of Technology in the SD2 STS?

SD2 employed and developed four types of technology: standards for data and metadata, technology for science, technology for collaboration, and technology for measuring, monitoring, and managing collaboration. A Summit participant summarized the importance of the first two crisply:

One of the things we did really well was we adopted standards, but we didn't adopt them too tightly. We agreed, for example, your code goes in Github or Gitlab. We have common data integration and knowledge representation systems that people were encouraged to use. We had common automation and orchestration that were encouraged. It was the same for software portability best practices, and attempts to maintain good quality, accurate documentation and monitoring on hosted systems. We were far from perfect but making that sustained attempt over years allowed us to be nimble without incurring a lot of unknown technical debt that could have derailed the program.

Below, we explore the role of standards in the STS, as Summit attendees perceived it.

Standards for data and metadata

SD2 adopted standards for data and metadata and developed new ones when the need arose. For example, the program adopted the existing Synthetic Biology Open Language (SBOL). SBOL is a standardized format for the exchange of information about structural and functional aspects of biological designs. It employs a well-defined data model to describe biological designs unambiguously and provides best practices concerning how to use the data model. One performer noted two benefits of adopting this standard: “representation of biological entities using SBOL enabled design-build-test-learn well, and subsequently supported protocol language development.”

SD2 developed its own data and metadata standards when public standards were not adequate or available. Among those mentioned at the Summit were the homegrown standards implemented in the measurements table, data converge, and the precomputed data table. As one performer noted:

We don't have three different metadata formats, we have a single data format, regardless of the lab (benchtop or foundry). The experiment requests and all the ecosystem around that (data presentation, metadata format and schemas behind converting the lab traces from lab, the data catalog) leveraged this. ... So, we can analyze that data alongside something that we ran just a few weeks ago.

These data and metadata standards served performers, not just machines. SD2 developed a form called the experimental request. The simple document format, when processed, standardized the terms for biological components and materials; this information in turn populated complex experimental plans of materials and procedures for automated laboratories. Equally important, the document enforced a systematic workflow, “so we did not cut corners.” Finally, the experimental request expressed intent useful both for understanding and for replication by human researchers. As one performer noted:

When sharing protocols, academics often have a high-level conversation maybe with an email follow up, but never disclosing the level of detail that's required to document experiments for reproducibility. The experimental request forced us to articulate the questions we were trying to address. This process allowed us to capture and record the context and the intent of the experiment.

Technology for science

SD2 also borrowed and built technologies to accelerate science. One Summit attendee pointed to the SD2-created “automated round trip” technology as an example.

It enabled experiment execution by clicking on items in a Google Doc. That was an incredible thing. ... This technology, for doing the kinds of automated workflows in automated loops, was a really important enabling technology for the group as a whole.

Similarly, the SD2 automated planning system integrated the experimental requests with experiment execution, and thus served as an integrator or coordinator of parts of the SD2 system. (The report chapter concerning Challenge Problem Evaluation enumerates dozens of other tools developed in SD2.)

Most of these tools ran on infrastructure custom-configured for the program. As one performer explained, “the infrastructure at TA4 allowed us to switch to different execution systems, to leverage more memory, more GPU. All we had to do was containerize a bit of code and in a matter of hours or maybe a day or two we could run it across all experiments.”

Conventional commercial and open-source software tools served SD2, albeit their use had mixed effects. An example was Github, a software repository for more than 200 million projects that offers strong version control and project management features. The program also leveraged conventional databases and spreadsheets, among other tools. These were, on the one hand, familiar to users and easy to adopt. On the other, they tended to fossilize functionality, preventing new technology advances that could accelerate scientific progress. As one performer noted:

We suffered from thinking early on that we’d just make a spreadsheet as a first draft, and then that just got too cooked into the workflow and it was hard and somewhat painful to extricate ourselves from that. We also saw version control and long-term management issues. We wanted to prototype quickly, that was a good way to prototype quickly, but then things ossified around it, and so we had to break our dependence from that in a painful way.

Another performer cited a case in which the tension between familiar but inadequate and threateningly new technology was resolved using extreme methods:

Some of our software developers wanted to use only their software stack and their way of managing things If you believe that everything is a relational database because you've been building relational databases for a bunch of years, then Resource Description Framework (RDF) is scary and SPARQL Protocol and RDF Query Language (SPARQL) is scary. Despite prompting from the top, it's the technology stacks that people are used to which are the ones that are going to [be favored]. The perceived barrier to adopting other technologies was not overcome until the program down selected a performer and replaced them with performers aligned with SD2 goals.

The technologies for science also included laboratory tooling, ranging from manual devices used in benchtop laboratories in academic settings, to robotic laboratories such as Strateos, and foundries that incorporate manual and robotic technologies such as Ginkgo Bioworks. SD2 performers (Leins, et al., in review) explored the role of the benchtop laboratory in this program dedicated to automating science. They noted that the benchtop environment is a necessary complement to robotic labs in several respects. First, it allows researchers to design, test, and revise protocols rapidly and inexpensively. Said a participant, benchtop researchers “implement pilot protocols that could be executed and tweaked in a much more agile way before we took the [weeks or months of] time to implement it in a fully automated workflow.” For example, benchtop work enabled scientists to understand the variability in experimental materials and develop plans and protocols to manage it. As one performer noted:

It was important to do the type of bench scale, non-automated work because you know there's just so much variation in the kind of underlying materials that we're working with, even if the actions that are taken on those materials are the same. Bench-scale, tacit knowledge type work was invaluable.

The benchtop is not just a prototyping site for automated labs, however. Researchers can use the benchtop to test whether data standards and tools developed for robotic labs and foundries can be adapted and extended to address new research challenges.

Finally, the benchtop sometimes offers capabilities that the available automated labs do not. “One of them was counting colonies,” a performer noted. “We couldn't do that kind of gold standard work at the TA3 labs.”

Technology for collaboration

SD2 used Slack for messaging, and Google Drive for documents, spreadsheets, and slides. We used Zoom for videoconferences heavily during the final two years of the program that coincided with the Covid pandemic. Gitlab and Jupyter Notebooks supported collaboration concerning software. These provided common spaces in which performers could interact in conversation and code, and they supported asynchronous coordination (in place of real-time meetings) through their alerting and search features (with the exception of Zoom). Said one performer, “if somebody said something three months ago [in Slack or Drive], a year ago, you could find what they said and maybe even what the context was.” Additionally, the logging features of these tools enabled Aptima as the Challenge Problem Integrator to analyze interactions between individuals and working groups to identify the most strongly engaged performers and those whose membership in multiple groups positioned them to mediate cross-group issues.

Ironically, some of the commercial tools served performers but not program leadership, because the security protocols at DARPA prohibited access to certain SD2 tools on standard DoD work computers. Summit participants explained:

Slack, Git, ... Google Drive. Without these tools, a lot of our work would not be possible. However, those things don't necessarily work for every branch of the government. You can't replicate SD2 tools [across all stakeholders] even though there's a need for these types of things.

It may never be possible to implement a socio-technical system that you can plug everybody into, where all work is done.

In sum, SD2 leveraged standards and technologies—both off-the-open-source-shelf and custom-built—to further scientific progress, but grappled with familiar tradeoffs between the familiarity of tools and their fit to the program’s unique needs.

Technology for measuring, monitoring, and managing collaboration

Aptima developed software for support in measuring, monitoring, and managing collaboration across the SD2 STS. That software, called TeamVitals, monitors data that pass between performers and enables program leadership to assess and detect changes in the structure, interactions, and productivity of the program.

TeamVitals ingests Slack communications and Google Drive file interaction logs and monitors production of data and documents to help assess progress and identify bottlenecks in a workflow. It also monitors and measures interactions between performers to identify highly interactive individuals, boundary spanners, initiators, and gaps in an STS. It then organizes, filters, and visualizes those data in ways that help the user to understand the structure of the STS, and compare that structure to what is expected from initial design or the effect of challenge problems. TeamVitals helped Aptima to ask and answer questions such as:

- Who are the most (dis)engaged performers on the program?
- Which performers are best positioned to discuss the state of a program initiative, or to address an emergent issue?
- Which performers are engaged in collaboration across teams?
- What program documents are attracting the most/least attention and interaction?
- Is the pipeline for generating and analyzing experimental data productive or jammed?

TeamVitals presents a dashboard for exploring the realized, dynamic structure and process of a socio-technical system (STS) and compare that structure to what is expected from initial design or the effect of challenge problems and other mechanisms intended to influence that structure. The Dashboard, depicted in **Figure 12** and **Figure 13**, allows the user to filter by the type of data (**Figure 12A**, Slack chat, Google Drive file interactions), by time (**Figure 12B**), or by specific teams (**Figure 12C**) or team members (**Figure 12D**). These filters are toward the top and right of the display, and influence what data is presented through the bottom two-thirds of the dashboard. An “Experiment Roster” (**Figure 12E**) shows the name and start/end times of an experiment. A “Communications Graph” (**Figure 12F**) shows who is talking to each other. “Top Boundary Spanners” (**Figure 13A**) provides a summary count of in-degree and out-degree for those the most engaged members of the SD2 Program. “Top Slack Channels” (**Figure 13B**) indicates team engagement in each channel, with wings on each team dot in the channel graph showing whether the team’s participation is increasing (wings down, as if flying upward) or decreasing (wings up, as if descending) over the user-specified timespan, with the magnitude of that change indicated by the angle of the wings. “Timeline Document Interactions”, at the bottom (**Figure 13C**), lists the top documents by edits in Google Drive, and includes information about contributing teams and top individual contributors. TeamVitals software has been delivered as a 7-zip archive which can be unzipped and run on a local client using Docker Desktop.

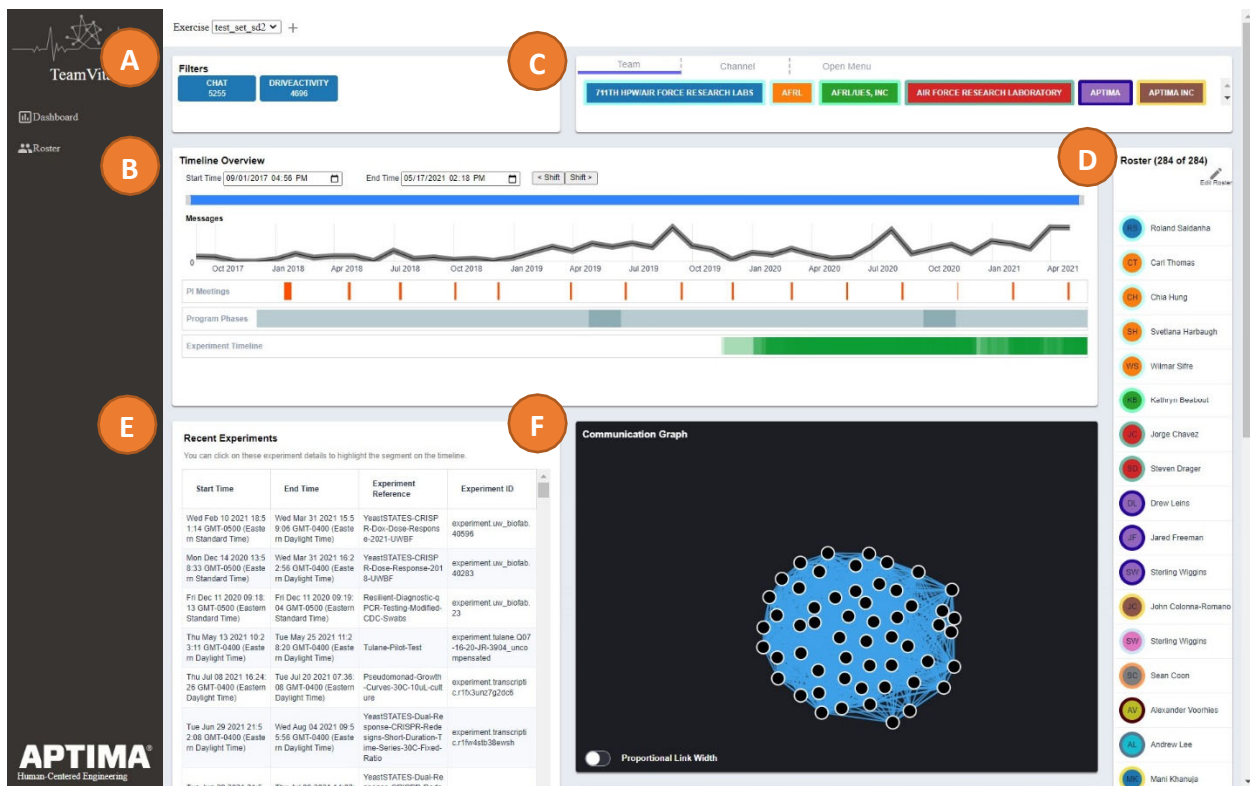


Figure 12: Top half of the TeamVitals dashboard.

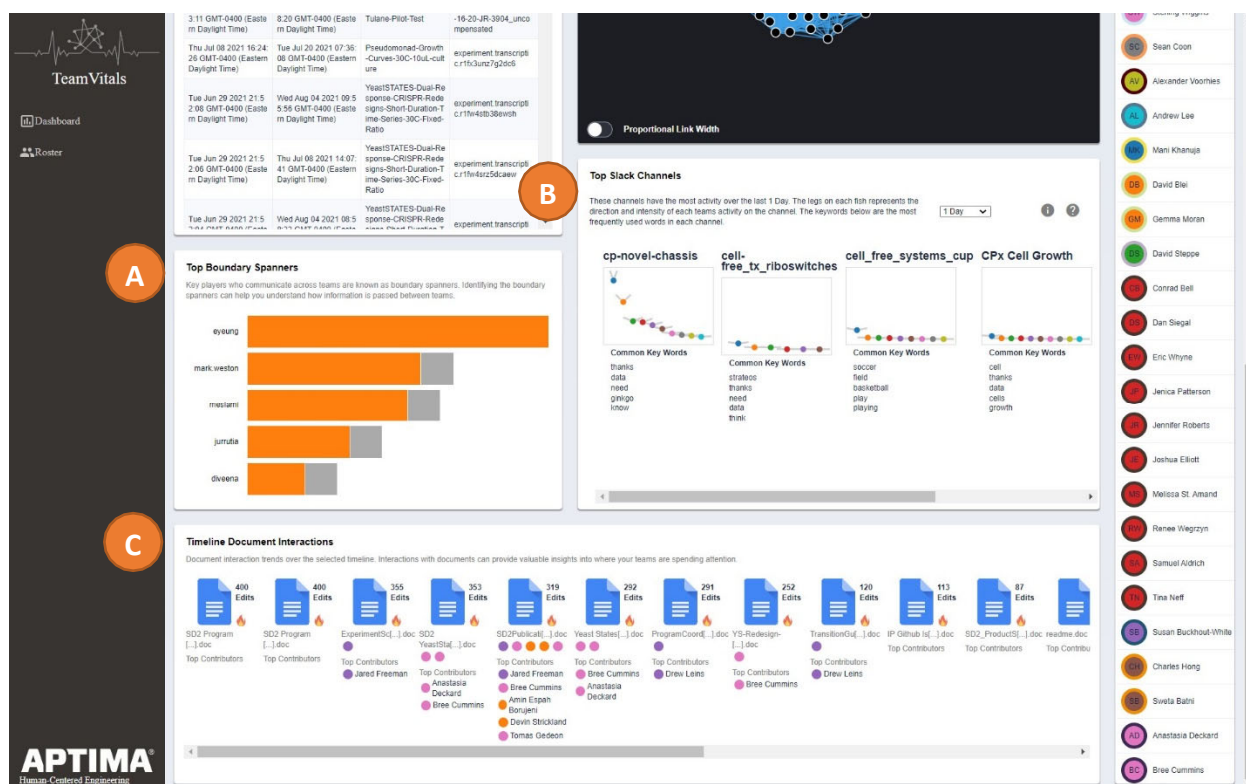


Figure 13: Bottom half of the TeamVitals dashboard.

Alternatives to the SD2 STS

The SD2 STS is characterized, above, as a carefully structured, highly collaborative enterprise whose work was facilitated by common language (data and metadata standards), technologies for science, and technologies for collaboration. The STS demonstrably accelerated both scientific progress (see 1: On Selecting & Assessing Challenge Problems) and development of many new tools. However, it had limitations that, at the least, require careful management in future STS, and, at the most, may lead some program leaders to adopt alternative enterprise models.

Limitations of the SD2 STS

Span of work: The SD2 STS pooled resources to advance science and technology, but those challenges were drawn from the artificially small pool of problems of common interest. As a Summit participant noted:

If I can't get everybody interested in some hypothesis I have then I have no shot at addressing it in an STS. Think about the challenge problems we did not address and the people who thought they were important. That's another limitation of the STS: everybody has to agree on those hypotheses. In the investigator-initiated model everybody has a bunch of different ideas and what you can do is search a very broad hypothesis- or idea-space with this model in a way that you can't with a large STS that focuses on more concise hypotheses. So, one of the limitations of an SD2-style STS is ... you get rapid results in a single or coordinated hypothesis set, but you don't get to search that space fully.

Cultural friction: STS are constrained by the reward structure and cultures of member's communities. Some scientists may not be able to resolve conflicts between the incentives of their local institutions and the goals of a program such as SD2. Summit performers said:

The challenges for junior faculty to participate in a program like this could be high. The need to carve out your own domain is important at that career stage. At other career stages, it's okay to encourage the work of others and help others do things.

The academic culture is one in which independence, independent thinking, and siloed thinking is required for tenure. The structure in which you're rewarded is based on demonstrating that independence and brilliance, so a program like this is going to be very difficult for pre-tenure folks ... However, if handled correctly and properly there's so much to be gained by young faculty. The amount that they can bring back to their own work is huge, the resources are the kind in a typical DARPA program that you would never have access to as a young junior professor. It's also a really rich training ground for students, because the nature of science is changing, I think, eventually we're going to stop being so focused on individual accomplishments and start rewarding the ability to work in these interdisciplinary teams.

Not just individuals, but institutions may decline to engage in an STS. One performer noted of their commercial lab:

One of the challenges that we had working with this external cloud lab was that they wanted to keep a deliberate distance from SD2. They were worried about technology rights and other types of issues. They didn't see the goals as clearly. They weren't indoctrinated in what that should be. That was a social fail in the STS.

An alternative interpretation is that the laboratory in question held values that could not be aligned with those of the sponsor and the STS, regardless of their indoctrination.

Cost of engagement: The culture of the SD2 STS was highly collaborative, and that collaboration came at a cost that warranted comment by Summit participants. A performer noted that "facilitating communication among performers is critical for any STS. Sometimes you get too much communication, and you want to turn it off, but it was the best thing we did." Others noted that this STS launched with some friction and miscommunication, until it developed a culture and cadence that enabled people to collaborate smoothly and see benefits from that collaboration. Individual performers also managed the cost of engagement by participating at different levels of frequency and intensity. As at least two Summit participants noted, it is critical to get folks over the initial and costly barriers to collaboration and communication.

Evolution: An STS evolves, like any social system. That evolution must be monitored and managed. Summit participants noted that, absent any design or management, an STS will emerge in any reasonably intact organization. Monitoring detects the form of the STS, designed or emergent, and management shapes it. A particularly difficult issue is changes in the personnel of the STS. Graduate students and post-doctoral fellows come and go, as do laboratory staff and principal investigators. This requires onboarding and training procedures. Those procedures were slim in SD2, as noted above. They are much more robust among some frequent performers for National Institutes of Health (NIH), which calls for systematic and repeated training in leadership and collaboration skills.

The turnover in SD2 membership presents as much opportunity as challenge. Summit performers noted that new members may reduce groupthink in an STS, bring new technical skills, and bring new perspectives. Just how existing STS members and culture interact with new members is itself a matter for careful monitoring and management.

Sustainment: The SD2 STS has not survived the end of the program intact. As the program drew to a close, its members largely withdrew from this STS and immersed themselves more fully in other communities. Similarly, some of the technologies SD2 developed will likely be unused because they do not generalize to new scientific problems or are forgotten from organizational memory. As one participant noted:

The thing that we're trying to get to in SD2 that I don't think we necessarily solved is, are there enough general modules that would allow us to get to something that's completely useful for every biologist? For example, an internet protocol was enough to get the Internet going. Is there a similar seed that would prompt researchers to pile into the system and say, 'I can do all of my science here'?

Design for sustainment and transition funding are two complementary solutions to this problem. The sustainment path was well described by one performer:

The SD2 STS is slowly spending down because we didn't focus on sustaining it on day one. There's been good effort from DARPA to transition key technologies and some approaches, but I think we've identified that this STS as a whole, is not quite so portable. ... From the government's perspective ... infrastructure is a side effect, and the primary outcomes are the science outcomes. Frankly, that's as it should be, but it makes it hard to build this kind of stuff to be self-sustaining (without returning to funders to ask for financial support) ... If you really want the government to fund value-add infrastructure—e.g., science-domain-focused support and capabilities—you need strong science partners to champion the infrastructure.

Sustainment of the technology and data products of an STS requires a strategic investment of resources in development for sustainability (as noted above) and engaging with transition partners to fund sustainment—partners such as DoD users in service labs, a non-profit organization, a substantial open-source community. SD2 invested in both.

SD2 devoted much of its last year to the sustainment activity of transition. On DARPA's instruction, performers invested those months in finalizing technologies, data, and the scientific findings that proved their value, and to talking about and testing technologies with transition partners in military, government, academic, and commercial labs. The 6: On Transition of this report documents the effects of that investment. A Summit participant summed up the transition outcome as follows:

What's happening is individual bits and pieces are transitioning off where they make sense, where they can fit into a specific organization that has a compatible scientific problem
Transition has been easier on other programs that tackle very specific, targeted problems.

Alternative models and methods

The primary alternative to the STS model is the individualist or investigator-initiated model, in which each "investigator or their team is creating a hypothesis, using that hypothesis to generate a protocol, doing that experimentation, collecting the data, and then analyzing the data and putting out

conclusions” (Garrett, 2022). This is the model used in many research programs by NIH, National Science Foundation (NSF), and the DoD labs. The individualist model arguably accelerates local progress, but limits the generalizability of data, techniques, and tools, as well as the agility of the overall program. As one Summit performer explained:

The research in a lot of large scientific consortia tends to run along threads that occasionally intersect where there's a common technical intellectual interest. That works okay for science, but that means those individual threads still advanced at the single lab pace. When there are cross-cutting problems or emergent opportunities, it's hard to switch gears and address those problems.

In addition, it is difficult for such programs to generate data and technologies that generalize, transition, and persist. Independent investigators have little incentive to sacrifice research progress for improved metadata and data standardization, and for designing and refactoring software and documentation for other uses and users. Federal and institutional funders have accepted these limitations for many decades, presumably because well-chosen independent investigators advance science and scientific technologies, and some of that S&T has broad application. Further, the individualist model does build socio-technical systems to the extent that members of a research or development team coalesce around common language, techniques, and technologies, and to the extent that these STS persist as students and staff move to different institutions over time.

There are models that lie between the investigator-initiated research and the form of STS implemented by SD2. One is the temporary, voluntary STS. An example is the International Genetically Engineered Machine (iGEM) competition, a contest running several months each year in which participants build biological systems from a kit of standard parts ('BioBricks'), operate those system, submit the products and findings to judges for medals and awards in categories such as “Best Measurement” or “Best Model,” and contribute their novel parts to the iGEM parts registry for future use. The competing teams (350 of them in 2021) are loosely bonded into an STS by the use of a standard parts kit; incentives to adopt common practices of registering new parts (which warrants a bronze medal), documenting them (silver medal), and showing a proof-of-principle (gold medal). They have the joint experience of convening (in non-Covid times) to present the work. The DoD has occasionally applied a similar strategy in creating challenge problems that are open to all to solve. However, this loosely knit, temporary STS is designed primarily for education (in the case of iGEM), or engaging new talent and sampling novel prototypes (in the case of DoD). The odds that a temporary STS will generate deep scientific discoveries or transition-able and enduring technology are slim. As one SD2 performer noted, “with the more high-intensity collaboration that we found in this program, it was really possible to do that.”

One member of both SD2 and the iGEM organizing committee noted additional frustrations of such “volunteer” STS:

You should expect most people in your volunteer projects to contribute nothing that is obvious to you. Understand that it's completely unpredictable who is going to become a strong contributor and who will not. ... You have to get really comfortable with having people around the edge who are just interested but will never do anything and might never actually even use anything [that the program produces]. ... Further, almost everything that you try will fail because almost all the people who are interested in it will drop out. Challenge problems are selected, but almost all of them fail because either you run into a technical challenge or the people who were

enthusiastic are no longer enthusiastic (or don't have the bandwidth). You have to get really comfortable with things dying."

The same performer offered a scheme for understanding the types of participants in funded and volunteer STS.

There are three groups of people who engage with STS: core performers, stakeholders, and fellow travelers. The volunteer organizations include all three groups, whereas the threshold-of-entry consortia essentially include just the first two groups. An STS like SD2 has a very large set of core (paid) performers, which can be very expensive. Your project stakeholders are the people who care enough to engage but they're not actually going to do the work. But they're the ones who are going to use the work and who are going to tell you where you're getting it wrong. SD2 had core and stakeholders. Fellow travelers seem to be useless but these are the people who are your ambassadors to the rest of the Community.

The SD2 STS invested modestly in fellow travelers. It sponsored a conference concerning data sharing that attracted funders of data-generating research, and publishers of that research who could, potentially, publish data as well to enable replication and extension of published findings. SD2 performers generated at least 140 articles, chapters, and conference presentations that potentially grew the ranks of fellow travelers, those who tracked the progress of SD2. The citations to SD2 publications will, over the coming years, identify the fellow travelers who made use of SD2 work, but it will not, of course, identify the others who merely observed the program's work and then moved on.

Funding institutions and the program managers have several mechanisms at their disposal for managing collaboration and the costs it imposes on an STS. They can monitor collaboration and reward the "boundary spanners" (Brown, et al., in review) whose work mediates distinct working groups, or those willing to execute work that is important to the program but that they did not propose. Programs can carefully design activities to ease performers into collaboration and coaching to help resolve unproductive friction in those collaborations. However, these methods must all aim to empower performers to make scientific and technical progress that is their intrinsic reward for costly collaboration.

Institutions can manage STS using the power of the contract and the scope-of-work (SOW). A Summit participant said:

On SD2, the stick was DARPA's requirements that you were not allowed to do certain things or else you might lose funding. There was also this wonderful carrot, which was the flexible statement of work that we all had.

More specifically, the contract and SOW can be written and enforced tightly or loosely to incentivize collaboration. In SD2, these documents segregated experimental design from execution and analysis, specifically to force collaboration. SOWs were written loosely enough to allow for performers to take on unforeseen missions in science (the CPs), data standardization, and software development. Institutions can also withdraw, award, or modify these documents address new challenges or, as was the case in SD2, to engage new performers as need arises.

Similarly, program managers can use the controls described above to incentivize collaboration: collaboration contracts, challenge problems, activities, training, and coaching. For programs of

significant duration, the PM can adjust these controls proactively and reactively to manage collaboration.

Deploying the SD2 STS

Is the work we do, the body of work we leave behind, going to be useful for future generations if we're not doing it in a more holistic sense, with contextual metadata and all these other concepts that we've developed within an STS that are necessary for model development?

This statement by a Summit participant draws attention to two distinguishing characteristics of an STS such as SD2. First, the STS prioritized transition of highly generalizable, interoperable, extensible data and technology. Second, it developed the capacity of performers and institutions to replicate useful aspects of the STS in future enterprises.

There is evidence that SD2 deployed aspects of its STS both at the personal and institutional levels. One performer asserted, "I feel like I was trained as a leader in the STS." We hope that this personal growth improves STS in future scientific enterprises. At the institutional level, we cite deployment of the SD2 STS concept to the NIH Bridge2AI program. The principals of that program turned to SD2 leadership and TA5 for advice concerning the design of the STS for their nascent program (NIH, 2021). Bridge2AI will employ a central team coordinating the standards and processes among diverse institutions that are striving to create datasets, standards, tools, training, and public resources that foster machine learning in biomedical research.

Summit attendees predicted that the SD2 STS would largely dissolve upon the completion of the program. They identified several strategies that future programs might use to manage the resulting impact on sustainment and transition.

One is to invest less program funding in scientific innovation, and more in routine documentation, robustification, and transition of program products. Said one performer, "surface the idea of self-sustainment earlier in the program. You have to start early because it takes at least a year, if not two years of lead time to pursue a long-term funding strategy."

Two concepts raised in the Summit would influence the accessibility of data and technology for scientific innovation. One is to coax publishers to acknowledge the value of scientific infrastructure (e.g., systems that host data and tools, and integrate them in workflows). New journals and wider use of emerging authorship standards such as Contributor Roles Taxonomy (CRediT; CASRAI, 2022) might accomplish this. A second concept is to fund and foster a DevOps movement in science, one that creates open-source tools that enable full and rapid cycles of design-build-test-learn in the sciences.

The most ambitious concept presented at the Summit was to view and fund scientific STS as a national resource, such as the National Guard. Said one performer:

The sustainable part of the infrastructure will be data storage and all the analysis tools, because when you have a new problem, you want to be able to respond. You want to dump data into a structure and have tools ready to analyze it. And in you want to have all the rules and conventions in the dictionaries. People need to be trained on that regularly, so anybody who's going to plug in needs to be actively interacting with this infrastructure.

This is a challenging proposal in part because the Federal Government largely pursues this goal by competing means. It creates national labs and DoD labs to harbor capabilities needed for emergency response to nuclear, chemical/biological, and other threats. It funds academia to build capability for science generally, but not necessarily for coordinated response to potential threats. The proposal is also challenging because the potential threats are myriad, the techniques and technologies to address one may not generalize to others, and the funding to build generalizable and specific capabilities is limited.

In this report, 6: On Transition, documents additional success and failures of deployment of the SD2 technical products.

Conclusions & Recommendations

In this chapter, we described the methods used to build the SD2 STS and use the words of performers to define it in terms of its structure, its function, and its cultural characteristics. We described several key actions that fueled deep knowledge integration across the program: reduce the barriers to team science, foster collaboration, build tools to monitor and manage collaboration. We described the technologies—standards, scientific technology, and collaboration technology – that helped bind performers into a productive enterprise. We identified challenges and limitations we encountered building and managing the SD2 STS: span of work, cultural friction, cost of engagement, evolution, and sustainment. We contrasted the STS with the primary alternative: the individualist model. Finally, we raised issues concerning deployment and strategies, small and large, for managing it. Among the many recommendations explicit and implicit in the text above, we believe the most influential are these:

- Leadership needs to consistently exercise programmatic authority to publicly declare the mission of the program, the values that behavior must fulfill, and critical decisions.
- Institutions should invest time and resources in designing and assessing collaboration contracts, challenge problems, activities, and training and coaching.
- Program leaders should thoughtfully trade off the convenience of familiar tools and local methods against the costly but beneficial creation and enforced use of program-wide tools, standards, and workflows.
- Leaders should attend continuously to the evolution of the STS; the dynamics of the social and technological system can drive it to success or collapse.
- Leaders concerned about performer engagement and contributions can reduce the need to assess this by attending some program meetings by monitoring communications and documenting interactions using tools such as TeamVitals. Such software should, in future programs, integrate a view of the production system (e.g., experiments requested and completed, datasets awaiting processing and completed) with a view of the social system.
- Program leaders and performers should encourage all participants in the STS to ask deep questions concerning goals and anticipate long term and often undesirable effects of unclear goals.
- Leadership must be unafraid of allowing performers to engage in difficult conversations, but must be quick to identify unproductive and destructive behaviors, and personally coach performers to behave collaboratively.
- Sustainment of the STS and its products must trade off premature investments in hardening and documenting immature solutions against late and insufficient investment in polished solutions.

- Institutions should muster Federal support to sustain scientific infrastructure, techniques, technologies, and communities that could address probable and significant threats, such as a pandemic.

4: ON SYSTEM EVOLUTION

Overview

The overall objective of this task was to look across the entire SD2 system to identify areas where the system can be improved through system engineering; recommend system improvements to working groups, TAs, or performers; and assess those system wide improvements over time. This task focused on two areas: 1) identify technology gaps and requirements and 2) support SD2 technology design and development.

A variety of methods were used to monitor the development of the entire SD2 system and influence its direction so that it adhered to good system engineering principles. Measures were defined and collected to identify problem areas and indicate overall progress of the SD2 system. This report chapter describes our methods and findings.

Methods, Assumptions, & Procedures

This task was primarily an advising task which identified system wide issues and advised various elements of the program to make changes to solve the issue. The method generally followed the process as such:

- **Collect situational awareness on activities in the program:** We built situational awareness of the program by monitoring communication/collaboration channels such as Slack, attending status meetings of various program teams, and one on one meetings with team members throughout the program.
- **Identify issues:** System engineering and system architecture best practices were used as a guide to identify system issues that needed to be addressed. Different types of issues were identified including failures, process blockers, inefficiencies, inaccuracies, missing components, and system evolution impediments (such as current approaches that will make it more difficult to evolve the system in the future).
- **Design solutions:** To address identified issues, we worked with stakeholders to understand the impact of the issue and associated risks, defined the priority of the issue and the system implications of possible solutions, and proposed solutions that considered and traded off system-wide considerations.
- **Coordinate system wide implementation of the solution:** To realize solutions, we worked with stakeholders to implement the solution, while striving to minimize impact of the change to ongoing program operations.
- **Monitor the improvements in the system over time:** We identified measures to assess change in desired characteristics of the system, collected, and published over time.

During the execution of the above process, the guiding principles were derived from system engineering and system architecture best practices. These best practices included the following areas:

- **Generalizability** is the ability for a system to support other variants or domains from the primary target domain. The amount and direction of generalizability incorporated into solutions is constantly being evaluated against the cost of the generalizability and the probability of future requirements to generalize in a proposed direction and to a proposed level.

- **Evolvability** is the ability of the system to easily accommodate future changes. There are several complicating factors, especially in research-oriented systems. First, is the ability to predict in which direction the system might evolve and second, is the near-term, longer-term trade-off value to the program versus cost to the program.
- **Usability** is the ability to easily and efficiently utilize the various elements of the system to accomplish the desired task. A large program like SD2 will most likely include different types of users and the system needs to be able to support each type. If the system is not usable enough, program members will find other, easier ways to accomplish the task. Those alternatives may well sacrifice system value to minimize local efforts.
- **Availability** is the ability of the system to perform the requested function when it is called upon to do so. A common aspect of this is reliability and uptime to support production work in a timely manner. Another aspect of availability in an SD2-type research environment is the availability of capabilities to perform production work and availability of new capabilities to move the research forward.
- **Completeness** has several dimensions including relating to functionality and data. With respect to functionality, completeness involves identifying gaps in the capabilities of the system that are required by the program. Functionality has trade-offs for the value of the feature versus the cost of implementing it. With respect to data, the challenge is to identify the scope of the data that should be collected. This will include “metadata” to provide provenance and context of the data itself to support the use of the data downstream of the collection.
- **Scalability and Performance** allow the system to support the volume of the tasking and information to be processed through the system. This area also includes system throughput and the elimination of bottlenecks.

Another area of this task was to collect system measures and generate metrics of system performance. This activity also helped to identify system bottlenecks. Measures of various activities within the system were taken, which often required development of sensors to capture data and compute measurements. Specifically, we defined measures that indicated timing of activities and volumes of work product through the system. Some metrics were then rolled up to show the rate of change in timings (e.g., process step delay) and change of volume of work product over time.

Results & Discussion

Given the methods and system considerations described above, many system issues were identified and improved over the course of the program. Many different challenges were identified and solutions evolved with the efforts of stakeholders across the program. A discussion of the key system evolution issues is contained in this section.

The issues have been grouped into topics with a common theme. Each topic is described and a list of example system evolution issues within that theme are listed. Each issue contains an approach which describes what can be done about that issue. Some of the issues contain specific SD2 developments that were undertaken, and some provide general guidance that was used during the SD2 program but did not result in a specific outcome.

Communicating System Status

The SD2 program had a large and diverse collection of participants from many different organizations. They collaborated over a system that evolved significantly over time. This made it difficult for program

members to track the status of the system with respect to both its functionality (“what it is capable of doing”) and its processing (“what is the status of my experimental data?”). Below, we offer examples of issues and solution approaches related to communicating system status.

- **Issue:** Program performers sometimes struggled to keep current on the evolving capabilities of the system. For example, they might need to understand how to perform the functions needed this week to accomplish a task that was executed last week with different functions.
 - **Approach:** This is a difficult issue in any evolving system. The two main approaches that were taken to address this issue were the use of various communication mechanisms such as Slack and automating functions of the system. Automating functions of the system removed many of the details of running the system from the end user. This simplified system use for them, reduced human errors and allowed the process to be enhanced without change to the end-user interactions.
- **Issue:** Tracking and sharing process status information across the program was difficult. Because the experimental design, run, and analysis process spanned multiple organizations and could occur over an extended period of time, it was difficult to share process status across the program. For example, unobserved or unexplained lags might occur if the experiment request was submitted but the lab had not yet run the experiment, the experiment was run but the data was not uploaded, or the data was uploaded but the pipeline was not run and the data were not validated. At each of these steps there was the potential for a delay in the process because the human or automated agent responsible for running the next step was not notified that the prior step was completed. Also, from a program management point of view, the overall status of each of the experiments was needed to assess progress of the program.
 - **Approach:** As the SD2 system became more mature and various sections of the experimental process became more automated, the near-real-time status of an experiment could be gathered from the system (at various levels of granularity) and made available to program team members. Several approaches were used to disseminate process status information to the program., the first being Slack. When certain process events occurred (such as data being uploaded to TACC⁶), a note would be posted to Slack to indicate the availability of that information. The other approach we used was to create a dashboard developed using the Redash framework, which automatically updated with detailed status of each experiment that was processed through the TACC pipeline. For each experiment in the pipeline, the status of each step in the pipeline was reported along with the time and the success status of that processing step. **Figure 14** is a partial screenshot (the actual table is very wide) which shows the status of several experiments along the TACC processing pipeline as displayed in Redash.

⁶ Note that we use the acronym TACC (Texas Advanced Computing Center) to refer to the data hub in SD2. TACC provided and maintained the computing infrastructure that supported data transfer, storage, and processing.

experiment_id	experiment_reference	protocol	OvNgt	Recov Lps	SR_path	Submitted date/time	lab_status	date/time	Uploaded date/time	path
Search...										
experiment.transcriptic.r1fn8xa6nedjam	NovelChassis-Endogenous-Promoter	TimeSeriesHTP	18 hour	[3]	sr_path	2021-04-19 21:53	complete	2021-05-27 16:11	2021-05-19 17:34	path
experiment.transcriptic.r1fn874smnksb	NovelChassis-Endogenous-Promoter	TimeSeriesHTP	18 hour	[3]	sr_path	2021-04-19 16:38	complete	2021-05-27 16:11	2021-05-19 17:28	path
experiment.transcriptic.r1fn873gk9pu7	NovelChassis-Endogenous-Promoter	TimeSeriesHTP	18 hour	[3]	sr_path	2021-04-19 16:38	complete	2021-05-27 16:11	2021-05-19 17:25	path
experiment.transcriptic.r1fn86edtze22	NovelChassis-Endogenous-Promoter	TimeSeriesHTP	18 hour	[3]	sr_path	2021-04-19 16:29	complete	2021-05-27 16:11	2021-05-19 17:26	path
experiment.transcriptic.r1fn8683gyy6q2	NovelChassis-Endogenous-Promoter	TimeSeriesHTP	18 hour	[3]	sr_path	2021-04-19 16:26	complete	2021-05-27 16:11	2021-05-19 17:30	path

Figure 14: Dashboard indicating the state of experiments (on rows) through processing stages (on columns; only a sample of which are shown here).

Metrics

As SD2 is a large and distributed program, it was difficult to assess its progress towards a well-functioning system for discovery and design. It was necessary to collect metrics to assess the progress of the program (not just of specific experimental trials, as discussed above). A variety of metrics were collected at different levels and for different purposes. The challenges to collecting metrics involve first identifying what questions to ask about the program, then finding a source for the measures required to answer the questions, and finally calculating the metrics and presenting them. Below, we discuss specific metrics issues.

- **Issue:** The program wanted to know: what is the volume of data flowing from the labs into TACC? Are there bottlenecks with this data flow?
 - **Approach:** Automating the data upload process and logging the status of metadata positioned the program to measure this functionality. Redash was used to collect this data and present it in a bar chart. **Figure 15** shows an example of this chart, which shows the number of files uploaded (Y axis) to TACC from each lab over time (X axis).

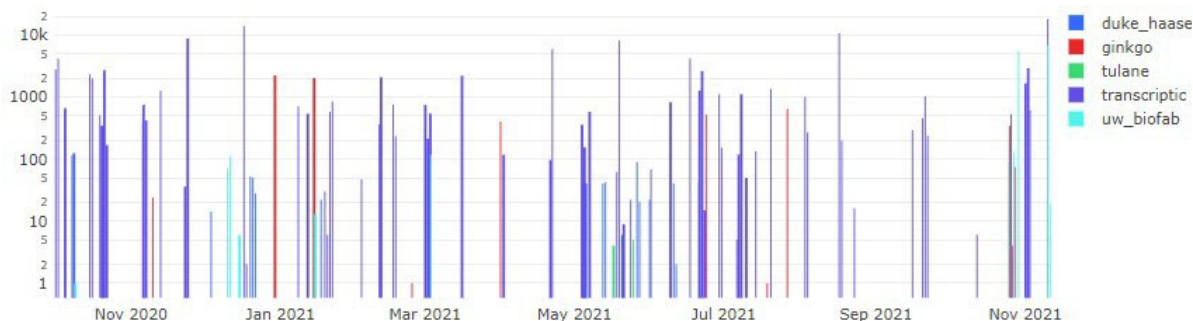


Figure 15: Dashboard indicating the number of data uploads per lab.

- **Issue:** The program wanted to know: has overall experiment cycle turn-around time improved over time? This was related to the time between the creation of the experimental request document and when the data from that experiment was processed and validated through the TACC pipeline.
 - **Approach:** Once the major part of the experiment processing loop was captured and managed by the SD2 system, there was a data source for measurements on this metric. For each experiment run, the system captured the time at which (1) the experiment request document was submitted and (2) the status of the experiment results data was complete and validated. The difference between these two times is the process time through a major part of the experiment processing loop. **Figure 16** depicts an example of the Redash chart that shows the time in days between the experiment request and data validation for each completed experiment. These time delays are plotted over time (X axis)

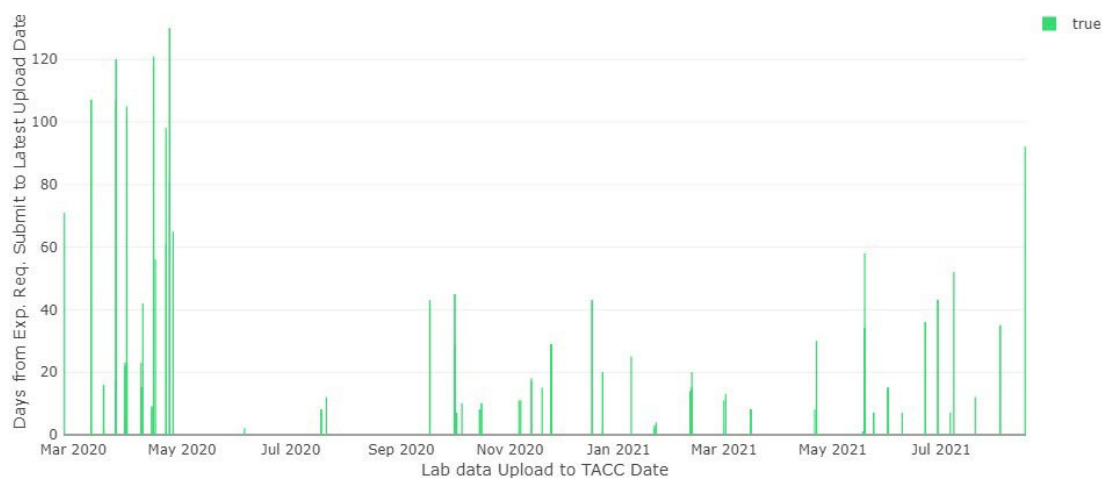


Figure 16: Redash chart showing the time between experimental request and latest data upload.

- **Issue:** The SD2 system was composed of several parts that were semi-independent. To link user activity across these different components, we needed to identify the same user in each of the different parts of the system. Several system components had separate user logins, hence the potential for different usernames for the same person. This made it difficult to link user activity in one system component to activity by the same user in another system component.
 - **Approach:** To accomplish the correlation of activities for users across these system components, a roster was setup which mapped the usernames for each system component to that user. Coordinating those rosters was needed so that activity and metrics could be collected across the system. This was a workable solution but less than ideal for a program of this size, as the roster needed to be updated often to reflect the change in program members.

Managing Complexity and System Architecture

The SD2 system was a complex system that was developed by people from different organizations. This led to challenges in fitting all the components into a coherent system that worked together and was usable by both the developers and the other program members. Several specific issues arose in this area of managing system coherence, as follows:

- **Issue:** The volume and complex structure of data created issues storing the data and making it available in a useable form for a variety of processing requirements. Also, the volume and structure of the data was constantly changing as the program progressed.
 - **Approach:** Several approaches were taken to mitigate these issues.
 - A non-structured query language (NoSQL) database was used to store metadata. This allowed for the structure of the data to evolve more easily as new data representations were defined.
 - Much of the raw experiment data was kept in files with the database used to capture metadata and then reference the files. This reduced the volume of data stored in the database.
 - The complex data structure was necessary to represent the relationships between the data in a general way. However, few users needed the full data representation of the data to accomplish their tasks. After data usage was observed, several application-targeted data views were created to give subsets of users more relevant data in a more relevant organization for their analysis.
- **Issue:** During the development of the SD2 system, there was always the discussion about the trade-off between building a more generalized system and building a more targeted system for the current need. The more generalized system would allow a wider range of uses of the system but add to the complexity of the system. A more targeted system would make the system easier to use for the current purposes, but more difficult to extend in the future to handle other uses.
 - **Approach:** This issue manifests itself in many areas. The first approach is to make the trade-off purposefully so that program members were always aware of where we were on the spectrum and could adjust as the system needs to be evolved. A second approach was to use good architecture principles and technologies (as described in previous and subsequent sections) to help the system evolve.
- **Issue:** The SD2 system was required to support data generated by multiple organizations and shared with multiple organizations. Each organization had somewhat different data structures and formats for similar information. This raised the challenge of sharing information across locales with varied standards.
 - **Approach:** In a distributed, multi-organization program, it is difficult to standardize on common data structures and formats. Therefore, instead of requiring multiple organizations to conform to a standard data structure and format, a “hub and spoke” approach was used to share data across organizations. This approach calls for the creation of translators between each organization with a different data format and a central common data store. With this approach N translators need to be created (one for each of the organizations to/from the common data store). In an alternative approach, in which data translators are created between each pair of organizations, as many as N^2 translators must be created. That alternative is untenable in a program with many performing organizations.

Process Automation

The SD2 system evolved along with the program. The experimental process started out mostly manual. Various parts were automated as the process evolved. Deciding what, when, and how to automate was a key issue during the evolution of the system.

- **Issue:** What to automate and when in the development of the system was a constant issue. There were two dimensions to this issue: 1) balance the needs of production and research and 2) reduce human time and errors.
 - **Approach:** In general, SD2 attempted to automate processes that were well defined and did not demand high flexibility. This purchased speedy execution for a large subset of processes but benefited production operations more than research operations. In addition, SD2 worked to identify the processes that required the most human effort and to automate those. This reduced processing time and human errors.
- **Issue:** As the experiment process became more automated, experimental data could be processed more quickly, but errors could also propagate down the processing pipeline without visibility. The completeness and correctness of the data with respect to the experiment request was also a concern.
 - **Approach:** To validate the data, there must be a definition of what the data should be. The experimental request provided a definition of what the resulting data should look like. With this definition, along with the data processing pipeline, the data could be validated automatically to determine if the data expected is present and complete.

Data and Metadata

- **Issue:** The right types of metadata were needed to make accurate and complete use of the experimental data. This included the ability to perform analysis of the experimental data (with the metadata providing important context), as well as the ability to reproduce the experiments.
 - **Approach:** The collection of metadata, the representation of metadata along with the data, and the availability of metadata were key characteristics of the SD2 system (see also 5: On Metadata). It is a challenging matter to define what metadata to collect and how to organize it, because the metadata requirements depend on how the data will be used. Those uses are not always discernible early in research, particularly in research conducted by future users of the data outside the immediate program. In addition, some metadata requirements cannot be easily identified until a problem is discovered. For example, variations in certain experimental results drove performers to capture metadata concerning the laboratory environment. With this, they discovered that the ambient humidity in the experimental chamber drove the experimental variations.
- **Issue:** It is challenging to determine whether a dataset is useful, particularly if it is incomplete or has some errors.
 - **Approach:** The usefulness of a dataset depends on the use. A given dataset, even though incomplete, could be useful for some users and not for others. The user of the data must be able to determine if the data can be used for their purposes. To make that determination, enough metadata about the data (including what might be incomplete or inaccurate) needs to be collected and stored with the data. SD2 used several mechanisms including Slack messages, metadata annotation, and validation status messages to indicate the correctness and completeness of dataset.
- **Issue:** During a multi-year project, data can become outdated, determined to be inaccurate, superseded, or irrelevant.
 - **Approach:** Data curation needs to be an ongoing process of the program. Data curation needs to be applied to both the data being generated in the program and data that is

generated elsewhere and is used on the program. Curation can be done either by automated means such as a validation script or by manual means such as visual inspection. These methods lead to more accurate data being used on the program as the program evolves and the use and purpose of data changes. SD2 curated data held in different components of the system used during different phases of the experimental process. including the data dictionary, the contents of SynBioHub, and validation of the data uploaded from the labs into the science table.

- **Issue:** Especially in the early phases of the program when data was being shared via files, there was a tendency to incorporate metadata into the file names. This led to long file names and difficulty managing the files. This also can lead to a wide variety of file names and difficulty in interpreting the meaning of the data in those files.
 - **Approach:** In general, sharing data by sending files (e.g., via e-mail or Slack) is problematic and should be avoided. However, if this method is used, the program should define some standard naming conventions so the metadata content of the file name can be interpreted correctly.

System Evolution Management

- **Issue:** The SD2 system was in constant change to make improvements and to support the evolving needs of the program. This creates a conflict between changing the system to improve it and keeping the system stable to support ongoing operations.
 - **Approach:** There are several approaches to mitigate this issue. First is communication to raise awareness of the changes so members of the program can prepare for the changes when they arrive. Slack was often used as the communication mechanism to share changes in the TACC system. Another approach used by TACC for a significant metadata upgrade was to run parallel systems, one for production and one for development. As the development system grew with useful capabilities and became stable, it was moved over to the production system.
- **Issue:** The requirements for compute and storage resources are continually increasing.
 - **Approach:** The software system architecture should support scalable computing resources. The TACC environment (as would most cloud environments) provided scalable resources. This allowed additional compute and storage resources to be easily added to the SD2 system. Above the level of the computing infrastructure, a scalable application infrastructure was used in SD2 to support the expansion and integration of new capabilities and data as they were developed.
- **Issue:** As the SD2 system evolved, and new capabilities were added and some parts of the process were automated, the workload shifted. This led to new bottlenecks emerging. Reliance on an individual or small group to perform key development or operational aspects of the system caused process bottlenecks as the system and process evolved. The example of this was the first attempt at utilizing XPlan to generate experimental requests.
 - **Approach:** Consider changes to the process carefully and consider the impact of those changes on other parts of the process. Monitor the process metrics to get early indicators of new bottlenecks so they can be addressed before they become significant. SD2 implemented this type of measurement and monitoring, as we described above. SD2 needed to take a step back when it was discovered that there was a major bottleneck using

XPlan (in the early stages of the program). To resolve this, there was a brainstorming meeting called with representation from across the program. At this meeting, the problem was characterized, and a way forward was defined. Later in the program, XPlan was incorporated into the experiment cycle with better process automation integration and was successful.

- **Issue:** In a multi-organization team, it is difficult to motivate groups to change to integrate with the program as a whole with respect to processes, tools, data formats, data structures, etc.
 - **Approach:** The system always needs to represent value to the members of the program. The cost of integration is real and program organizations are unwilling to take on that cost unless there is sufficient evidence that it will add value. Sometimes acceptance grows slowly at first until there is a tipping point where there is enough value in the system, because other groups are using it, to make it worthwhile for a given group to join. In the early stages of SD2, there was significant resistance to using the TACC infrastructure. However, as the SD2 pipelines evolved to incorporate more capabilities and became easier to use, acceptance grew, and the system became more utilized.
- **Issue:** System standardization brings benefits and detriments to system users. When it is stable, standardization enhances predictability and ease of integration. However, when it is dynamic, it generates work and consumes resources to maintain.
 - **Approach:** Choosing the right time to standardize some aspect of the system—be it an interface or a data format—is key to managing the acceptance of the system and its utility. Standardization can help with data sharing and ease of integration, but also causes effort to change to the standard. You do not want to standardize something too early because if the standardization is not correct and needs to be changed again, that will cause additional burden on the program. Using a phased rollout of the standard can help test out the utility of the standard and reduce future changes. On SD2, many items were standardized from names to interfaces to data formats and structures. One specific approach used was to bundle a collection of standards (and their changes) so that there are fewer times when changes are introduced (although the changes would be more significant).

Data Sharing

Data sharing was a key element of the SD2 program. There was a large volume of data generated by multiple organizations and shared with other organizations who performed analysis on that data. Sharing the data and associated metadata was key to improving the discovery process. The evolving system of data and processing also created challenges to sharing data as it evolved.

- **Issue:** In a large diverse and distributed program, many things need to be named so that they can be uniquely identified and referenced by other items (i.e., there needs to be a functional namespace). With multiple organizations, each might have their own, idiosyncratic namespace. Also, there may be name conflicts where one item has multiple names, or the same name is used for multiple items.
 - **Approach:** All the namespaces need to be managed. On SD2, there are multiple namespaces from different organizations, and as such, a super-namespace was created which mapped item names from one namespace to another. The SD2 Data Dictionary performed this function to map various names from one organization to another (e.g., between labs). The namespace needs to be curated and validated to make sure there are no duplicates and that names are a unique 1 to 1 mapping for an item.

- **Issue:** The SD2 program needed to support both highly-efficient processing of experimental data and innovation of new data and processes. These two goals can conflict between keeping the system stable for production and allowing the system to evolve for innovation.
 - **Approach:** SD2 system interfaces and data storage mechanisms were designed to support easy incremental additions of new data, which has minimal impact on existing data and structure. As the data innovations became more understood and their value determined, they were moved into the standard part of the data store and interface.
- **Issue:** It was sometimes difficult for users to determine whether a desired dataset exists and where it is located, given the large number and variety of data that was generated by the SD2 program.
 - **Approach:** The capture, storage, and linking of metadata to data is key in providing a means to find the appropriate data that was needed for a given experiment analysis. However, some common practices are also useful. For example, data should be located based on specific criteria, such as a set of conditions for an experiment. In SD2, data analysts could query the metadata based on a set of criteria and find data (and the files that contained the data) which was relevant to the analysis they wanted to perform.
- **Issue:** When data are shared outside the common process, it disrupts the process flow of the system and makes it difficult to share data, collect metadata, and determine the status of work more broadly.
 - **Approach:** Generally, out-of-process data sharing occurs when users do not know a process exists, or that process is difficult to use. These cases need to be identified and the causes understood. These issues can generally be eliminated by educating the team on the defined process, adding functionality to support the needed process, or making the process more usable. In the early stages of the SD2 program, Slack was used to share data. This was expedient but lacked the data collection and data management controls needed to run the program.
- **Issue:** In a long running program that is evolving, data is constantly changing. A mechanism is needed to identify and track the evolution of data.
 - **Approach:** In the data managed by TACC, almost everything contained a version tag. This includes documents, data, and the schema that define data. References between items linked to the specific version that were known to be correct. Processing pipeline and instances of running those pipelines also were versioned. When these requirements are met, it is at least theoretically possible to recreate a specific run of specific data through a specific pipeline, because the versions of these are aligned. Similarly, it is possible to assess the reliability of a dataset given known deficiencies in the versions of data and processing pipeline.
- **Issue:** Data, such as experimental data, goes through a series of processing steps to transform it from raw data to various levels of processed data. The type of data a given dataset contains needs to be identifiable.
 - **Approach:** In SD2, data was categorized into a series of levels that indicates the general phase along the processing pipeline the data was derived from. The data products level indicates what part of the process from “raw” to “final results” the data was generated from.

Usability

- **Issue:** The SD2 program involved several different disciplines and a diverse set of users. To support innovation and discovery across the program, it was important that tools be easy to use by the diverse user community, and not only by the developer of the tool or others in the same discipline. Easy-to-use tools allow users to experiment with different uses of the tools and potentially discover new things.
 - **Approach:** Tools were developed by members from most of the different disciplines on the program. Each tool developer should keep in mind who the potential users of the tools are and elicit feedback from them to build a more usable tool. Some important issues are avoiding domain-specific terminology, avoiding special skills to use the tool, and making inputs and outputs easy to interpret by a broad range of users. On SD2, tools were developed and tested first to understand their value and utility and then were often automated to make them more usable by a wider audience of users.

Conclusions & Recommendations

The SD2 program was organized with multiple TAs to perform different tasks (e.g., scientific, technological, analytic, and evaluative). DARPA foresaw the benefit of having created an objective system evolution oversight function to take a whole-of-system view and make unbiased recommendations in a variety of areas to improve the various systems of the program and the program itself. We recommend this approach to the managers of future programs in which multiple parties contribute to the design and implementation of a complex system.

In addition, many of the issues identified above are likely to occur in complex programs. We recommend that program managers be alert for these and consider applying the approaches tried and recommended in SD2.

5: ON METADATA

Overview

Metadata were essential to SD2 to enable a diverse set of organizations and roles to collaborate together in performing scientific discovery. In this chapter, we convey the different ways that SD2 performers defined and used metadata in the context of SD2. We captured these details through interviews with individual performers and a survey of the broader SD2 community. The purpose for doing this was to better understand the role and value of metadata on the SD2 program. To accomplish this, the Aptima team performed two subtasks that developed important insight into the value of metadata as used on the SD2 program, as follows:

- Aptima interviewed a variety of metadata users on the program to understand their connection with metadata. The metadata interview was used to gather some basic information about the use of metadata on the program which was then used to design the survey to get a broader sense of metadata use.
- Aptima designed and conducted a survey about metadata to get a broader sense of metadata across the program. A subset of the metadata survey was also given to members of the ASIST program to look for more global attitudes towards metadata.

Metadata facilitated moving data of all types through pipelines that linked the work of one technical area (TA) to another. This was true whether the data encoded design of a model construct, an experimental protocol, or experimental observations. Early in the program, it was clear that metadata were critical for enabling analysts—who did not necessarily design the experiments from which data were generated—to make sense of experimental data. It was critical to develop a method by which protocol authors, using diverse conventions for representing the content of protocols, could submit those protocols to the experimentalists and propagate the intent of those protocols through experimentation, data processing, and analysis. Without a shared system method of representing data, particularly metadata, the diverse organizations of SD2 would not have been able to collaborate on CPs and realize synergy. Results from these efforts revealed the value and challenges of metadata including:

- Metadata significantly increases the *interpretability* of the associated data
- Metadata significantly increases the *ability to analyze* the associated data
- Metadata is in the *eye of the beholder* and needs to be defined and shared with the usage in mind
- Having appropriate metadata maximizes its value. Too much increases complexity and lowers ease of use, and not enough impedes the ability to use the data.

Metadata Interviews

Introduction

To better understand the role of metadata on the SD2 project, we interviewed various stakeholders in different roles on the program. Metadata had different meanings and served different functions for different performers. However, many aspects of metadata, including definitions, gaps, requirements, and functions, were not always made explicit by performers or would not be known to performers until the need for those metadata arose. In keeping with the spirit of SD2, to identify instances in which

implicit knowledge should be made explicit, Aptima dug deeper into the landscape of metadata in SD2 to do the following:

- Define metadata
- Identify the functions of metadata
- Identify the domain-specific problems that each metadata solution addressed
- Identify the gaps in metadata solutions that the program did not address

In doing so, we aimed to reveal and document the role of what metadata in SD2 and the value of metadata to the program.

Methods, Assumptions, & Procedures

To gather definitions and insight into the functions and value of metadata on SD2, we conducted semi-structured interviews with a sample composed of a subset of all SD2 performers. We asked each interviewee the following three initial questions:

- What is your definition of metadata?
- What problems did metadata solve in SD2 (or what functions did it serve) and what database or tools addressed those issues in SD2?
- What metadata were not developed in SD2, but should have been?

Performers' responses to these questions often led toward follow-up and clarifying questions that revealed more detail and additional categories of information, including recommendations for the consideration and treatment of metadata on future similar or different programs.

Results & Discussion

In summarizing the insights gained from the metadata interviews, we organize the information by topic, and where appropriate, distinctions between the value of metadata with respect to TAs are noted.

Metadata defined

To understand the value of metadata, we first defined what *metadata* are. Different roles in the program had different definitions because of their use of metadata.

For reference, the literature defines the term metadata at varied levels of scope and specificity. National Institute of Standards (NIST; 2020, pg407; 2022) defines metadata as:

Information that describes the characteristics of data, including structural metadata that describes data structures (e.g., data format, syntax, semantics) and descriptive metadata that describes data contents (e.g., security labels).⁷

⁷ The current NIST statement is focused on security. Thus it puts considerable focus on "abstractions that represent the basic properties or characteristics of active and passive entities with respect to safeguarding information, "that can make metadata ... "first class" objects with respect to security policy."

Other sources present a finer-grained taxonomy of metadata, amending the NIST categorization of structural and descriptive metadata to include metadata that is administrative (Zeng, 2004), statistical (concerning the analyses that generate data; Directorate, OECD Statistics, 2018), and legal (concerning intellectual property rights).

For SD2 performers who *developed analytics to further discovery*, metadata describe any surrounding context that helps make sense of the data but are not the focal data. For example, metadata might include sample descriptions and experimental conditions. They might also include data file details, such as conversions, modifications, and normalizations to the focal data. Thus, metadata can describe features at the sample or experimental level and at the data processing level.

For performers doing *model design and experiment planning*, metadata are everything except raw data coming off experimental instruments or measurement devices. For example, the location of a sample in a well is a piece of metadata, whereas the measurement of that sample is a piece of raw data. Metadata are what we use to help describe and interpret results. We do not compute metadata to generate results (e.g., in contrast with independent variables), nor are metadata the results themselves.

Metadata are elaborative and specific versus collapsed and general. Metadata add information rather than summarize information. They describe choices one made in collecting or analyzing data; they are often a record of an analysis process. Thus, they are breadcrumbs, left on the trail toward results, that can be followed for replication.

Each of the following can be a type of metadata:

- Decision variable: something you chose, over which you had direct control
- Computed variable method: some function you applied to generate an outcome
- Random variables: neither controlled nor computed; rather, nature controls them and they can influence computed variables

For performers designing experiments, one must define data before metadata. For example, a hypothesis that defines independent and dependent variables also defines the data. All the conditions surrounding those data are metadata (such as context in which data are captured). How those conditions are defined or bounded is based on context. For example, some experimental conditions, such as the ambient temperature of a data-collection chamber, are considered data by some and metadata by others. As a specific example, growth-curve experiments conducted early in the program at Strateos yielded time-stamped metadata that became data in the later obstacle course experiments.

For performers doing data management, metadata represents the context around files or artifacts. In more nuanced terms:

- Metadata *is a language*, so it must evolve. The SD2 program evolved its metadata language as performers discovered and described data components on the program.
- Metadata evolution *is a process*. Moving data from platform to platform requires some decisions to be made, and some implied data can get lost or purposefully demoted.
- Metadata *captures a process*; for example, if one captures data and then translates the data to make them usable, metadata describe the translations.

Problems solved with, and functions made possible by, metadata

Gathering and making available metadata on the SD2 program solved a number of challenges and enabled the program to accomplish a number of tasks that would not have been possible without the metadata.

TA1: Discovery algorithms

Early in the program, TA3 labs would indicate that data were available and uploaded to the data hub. Performers often shared these data or processed versions of them by posting files on the Slack messaging system. While convenient in the moment, this practice of using Slack created several issues, including:

- Data findability was hindered because important metadata (such as concerning the specific experiment that generated the data) were expressed only as message text. Access required (a) membership in the SD2 program, and (b) ability to logon to Slack.
- Interoperability was reduced because this informal file transfer mechanism did not enforce any standard language or vocabulary of metadata or data.
- Reuse was impacted because the data files were not associated with provenance details (GO FAIR, 2022).
- Analysts had difficulty using data provided through Slack or the data hub because the data did not have associated context. For example, the analysts did not know what high versus low florescence values meant in the data. Moreover, analysts were often using different data sets that were purported to come from the same experiment; however, there was no method for validating that data were in fact from the same experiment. In the words of one analyst, they were “comparing apples to oranges.”

After surfacing this issue, one CP group agreed to generating quality control flags for accepting or rejecting data. Attached to datasets, these flags enabled analysts to know whether they were looking at data that had been checked for quality. Further, the program began standardizing some data processing (such as for RNA-seq) and data representation.

Three tools proved particularly useful to support accurate generation, capture, and propagation of metadata: the data catalog (or science table), data converge, and the structured experiment request.

- The data catalog (or science table), which linked data to common terms and back to experiment requests, allowed analysts to query the data for experimental conditions and metadata. It provided analysts access to reference files that helped them see the conditions of a given experiment. As a supplement to the data catalog, SynBioHub (SBH) helped identify the underlying details of any sequences not placed in the data catalog.
- The data converge tool further advanced data and metadata representation by unifying data and metadata into one format so that different analysts’ codes could operate on the same data. Prior to the data converge tool, analysts had to develop integrations with the TA3 labs to download data and make dataframes they could understand. The data converge tool standardized this and obviated the need for, and the confusion and redundancy of, multiple efforts to generate interoperable dataframes.

- The structured request tool forced authors to input protocol details in tables, to add structured information to what otherwise existed only as free prose. This structured information helped to identify experimental intent and important metadata for understanding results. The structured request allowed analysts to examine data against their expectations. Analysts could examine the experiment request, identify the expected metadata, then access the experimentally generated metadata, and compare the two.

TA2: Model design and experiment planning

Performers at TA2 recognized that many parameters were required to configure downstream analyses and that these parameters become metadata that describe those analyses. One particularly useful parameter to know when analyzing experimental data is the control sample(s) used in the experiment. Thus, TA2 developed a machine learning (ML) algorithm that used the values of a control sample to predict analytic output based on measurements. The control values were metadata for analysts to consider when examining and interpreting their analytic output. Having these controls enabled analysts to implement mitigations or corrections to their analyses.

TA3: Experimentation

Humans and robots have different ways of processing metadata. Metadata allows more explicit interpretation of the data, hence easier and more accurate data sharing. Harmonizing protocols (including metadata) so that they could be understood at different TAs and different labs enabled the process of passing protocols from TA1 to TA3. Engaging with a diverse set of computer scientists can easily share data, helping to construct experiments in ways that generated the data and metadata that TA1 needed.

Having metadata associated with the experiment requests supported running mock pipelines which helped to identify when expectations about data might be violated.

TA4: Data hub and system architecture

Creating the SD2 data dictionary was a critical step to supporting data ingest, data finding, and sharing.

The hackathon that kicked off the program (referred to as Q0) employed a dump of convenient but unorganized data. For example, file A was about an organism. Was file B about the same organism? Nobody knew because the data were not labeled or annotated well enough to support distributed analyses. Files were uploaded individually (or links to files were shared) and analysts could not trust the process of parsing a file name to discover what was in it. Thus, performers had to negotiate terms (labels) for data and metadata. Further complicating matters, each lab had its own convention for labeling data, and they could not initially agree on a shared lexicon. The solution was to allow each lab to maintain its labeling convention but require the lab to build a dictionary of the individual terms. After a lab uploaded data, TA4 converted the names of factors (strains, temp, time, etc.) from lab language to common language and input them into the data dictionary. This dictionary could be updated as the work of the program evolved, and it allowed for conflict detection. The data dictionary—and a subsequent, related process for quality control—was the foundation of a consistent and stable schema for organizing and naming data that gave analysts trust in the datasets.

Metadata challenges that were not resolved by SD2

The use of metadata on the SD2 program was constantly evolving. The evolution occurred in multiple dimensions including what metadata to collect, store, how to organize, and how to share it. The evolution was not complete, hence there were metadata solutions that were not undertaken by SD2 by

the end of the program. Below are several issues that arose during the interviews pertaining to value that additional metadata or metadata processing tools could have brought to SD2.

Discovery of data was complex

Performers noted that the number of parameters that one could query increased rapidly during SD2, and as dimensions were added, queries became more complex. Performers did not have a tool to simplify this complexity. Some analysts stopped using the rich and flexible science catalog to find data because it became too complicated toward the end of the program; they switched to using a less flexible but easier customized dataframes. More specifically, analysts noted that they prefer to use natural language searches, but the science catalog database (MongoDB) did not support a natural language query engine; thus users had to be deeply knowledgeable and skilled to specify the form and content of their queries. Because of these query-related conventions, it was difficult to query the database for the high number of dimensions of interest in SD2. For example, metadata for inducer concentrations were represented across several dimensions. Because it was too difficult to query for the full range of dimensions in the data catalog, analysts instead accessed the raw data files and attempted to reduce the dimensionality by querying across only two dimensions (inducer present vs. absent). This proved problematic, however, because those dimensions were not represented in the data files, thus requiring TA4 to manually amend those data.

In addition, performers noted that SD2 had no infrastructure for linking multiple datasets. For example, joining flow cytometry with RNA-seq data tables would have been useful, but was not attempted during the program.

Metadata definition and management was costly

Performers noted the challenge of defining stable metadata discovery research across roles. They noted that the metadata collected was always evolving based on discovered need. Thus, metadata were usually complete for planned analyses, but sometimes incomplete for unplanned analyses. The requirement for metadata was exacerbated by the SD2 structure, which placed responsibility for design and analysis in different teams (rather than combining those roles in a single person or group, as is traditionally the case). Thus, the designer and analyst had to share a structured request and collaborate to identify needed metadata. This was valuable, because it required teams to express and codify their metadata requirements. However, the coordination was costly.

Analysis chains were not easily discoverable

Performers noted that they often needed to identify and understand a chain of analyses, but SD2 systems did not describe these chains adequately. Thus, it was necessary for performers to discover a chain manually; this was not a simple task. Had it been automated and granular, performers could have more easily defined new analysis chains to address additional scientific questions in order to determine whether a question could be answered by chaining available analytics, or to improve on current analysis chains. For example, a more agile approach would have described the analysis with greater context (more metadata), similar to query planning. Performers could have computed the quality of each part, then do a hypothesis test or use some function to determine which parts were best. This capability for chain analysis could have been extended to represent the experimental plan and execution. With those data, performers could have identified suboptimal or irrelevant operations and eliminated them to streamline experiments.

Metadata concerning experimental process were insufficient

Performers stated that experimental protocols were not adequately represented in metadata, and they provided several examples. Protocol primitives (transfers, shakes, reads) could have been better represented in a language that harmonized protocols across labs. SD2 did not do this at the level of the protocol or the data output, and this hindered some analysis and discovery. One lab specified contents of concentrations but did not specify times of additions or other variables about those concentrations. This made it difficult to disentangle readout data (e.g., “When was component 2 added relative to component 1?”). Finally, performers noted that metadata concerning time could have been better formalized. For example, one lab tested two growth phases and used “timepoint zero” to describe two different conditions (post 16-hr growth and no 16-hr growth). It was hard to recover which timepoint zero was reflected in the data without a formal representation of the protocol, but this would have helped to identify which measures were taken at which time.

Metadata Survey

Introduction

The interview results reported in the previous section led to additional questions regarding the definition, role, processes, and value of metadata on SD2 and other S&T programs. For example, how might professionals outside of SD2 define metadata? What is the role of a program sponsor in governing metadata processes? What else do metadata allow scientists or analysts to do? To answer these (and other) questions and augment our understanding of metadata, we constructed a survey. We invited participation from the broader SD2 community, as well as performers on another DARPA S&T program—Artificial Social Intelligence for Successful Teams (ASIST)—to answer two items: metadata definition and respondent professional role.

Methods, Assumptions, & Procedures

We constructed an 18-item survey that addressed the topics of metadata definition, relevant programmatics, contextual factors influencing metadata, and specific benefits of metadata. The complete survey is provided in **Appendix H: Metadata Survey**, and responses are presented in **Appendix I: Metadata Survey Responses**. Twenty-six SD2 performers completed the full survey. Another 24 performers from the DARPA ASIST program—concerned with socially intelligent artificial intelligence—completed two items (#1 & #19 in the full survey), which enabled us to explore the generalizability of our definitions of metadata. The topics and items from the full survey are as follows:

Topic: Definitions of metadata:

- What is your definition of metadata? (This item presented multiple fixed-choice response options, allowed performers to select all that applied, and offered an “other” option for respondents to input their own response).

Topic: Programmatics that influence how metadata roles and processes are developed, organized, and instantiated. All of these were Likert-type items, with a 5-point response scale in which 1 = Completely Disagree and 5 = Completely Agree, and respondents could choose only one response:

- The program should identify a specific group (e.g., a working group independent of other technical working groups, to manage metadata activities).
- There should be a centralized authority that makes decisions on metadata processes.

- When possible, metadata should be collected automatically—that is, it should not be an effortful activity engaged in by data/metadata users.
- The program should define a common infrastructure for metadata collection, storage, and access.
- The program should impose a common format and definition for metadata.
- All performers should be compelled to adopt and exercise the program standards for metadata activities.
- A program needs metadata and associated systems and processes in place before program activities start.

Topic: Contextual factors that influence how metadata processes are developed and implemented (all Likert-type items with a 5-point scale):

- It is important that an organization's metadata schemas are flexible and adaptable enough to accommodate different external metadata conventions.
- It is important to have resources—for example, tutorials, external consultants, open-source software—that facilitate metadata use.

Topic: What does metadata enable? Items marked with an “*” presented multiple fixed choices from which respondents could choose all that applied; the remaining items were Likert-type with a 5-point scale:

- *Having metadata associated with my data improves my results by making them (organized, accurate, complete, etc.).
- It is easier for me to use other people's data when it has metadata associated with it.
- Others can use my data even if it does not have metadata associated with it.
- *Metadata is most useful to the program when it helps answer specific versus a broad set of scientific questions.

Topic: Recent work that involved metadata. We asked respondents to consider their most recent work when completing the following items (all items presented multiple fixed choices from which respondents could choose all that applied and complete an “other” option if necessary):

- I generated, shared, or used metadata that described (experimental conditions, materials, instruments, etc.).
- The metadata I generated, shared, or used allowed me to (share, transform, and analyze data, etc.).
- The metadata I generated, shared, or used allowed others to (share, transform, and analyze data, etc.).
- I identified the metadata I needed (prior to generating a protocol, during protocol development, during protocol execution, etc.).

Results & Discussion

The responses from the survey were gathered using a Google Docs survey capability and the results are summarized below.

Topic: Definitions of metadata (and generality). We surveyed performers across two programs, SD2 and DARPA's ASIST, and asked them to define metadata. The disciplinary composition of the two programs

differed in a potentially impactful way. Whereas SD2 predominantly comprised computer scientists, ASIST predominantly comprised social scientists (**Figure 17**).

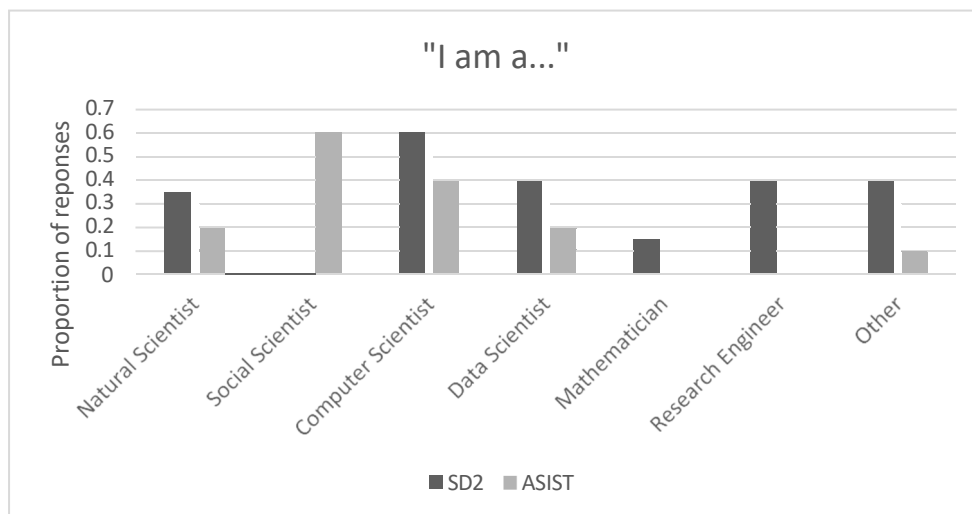


Figure 17. Disciplinary composition of SD2 and ASIST survey respondents

However, these two sets of professionals largely defined metadata similarly, agreeing that metadata primarily describe any surrounding context that helps make sense of the data, but are not the focal data (**Figure 18**).

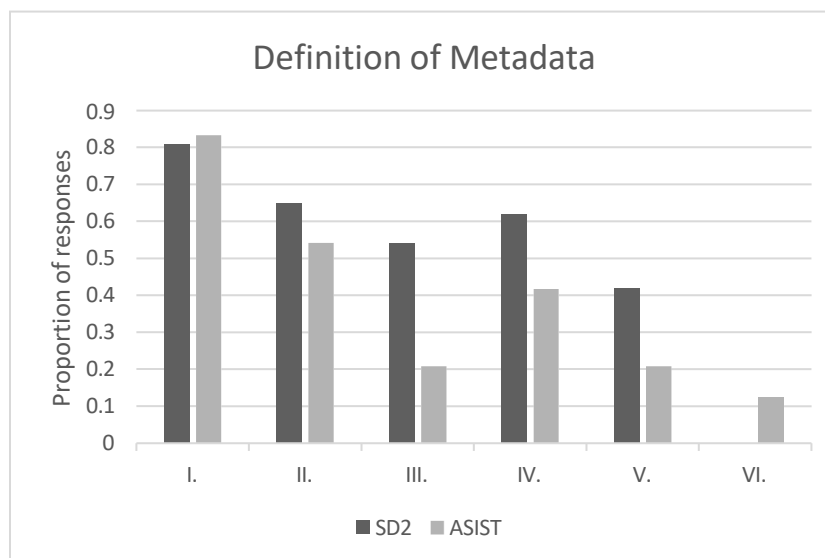


Figure 18. Definitions of metadata across SD2 and ASIST

Note. The X-axis labels correspond to the following definitions:

- I. Metadata describe any surrounding context that helps make sense of the data, but are not the focal data.
- II. Metadata are a set of data attached to a piece of data or function that gives some additional information.

- III. Metadata are everything except raw data generated from observation (e.g., from experimental instruments, measurement devices, algorithms).
- IV. Metadata is the language of describing where data came from, the state they are in, and where they are going.
- V. Metadata describes processes that generate, transform, and operate on data.
- VI. Other

The remaining items were completed only by the 26 SD2 performers. Therefore, although the items are general (i.e., they do not specify SD2 as the context), the results are from the perspective of SD2 performers and are likely informed by their experience on SD2. The following results are presented in figures in Appendix J: Metadata Survey Responses.

Topic: Programmatics that influence how metadata roles and processes are developed, organized, and instantiated. Regarding a centralized authority, half of respondents agreed that there should be a centralized authority that makes decisions on metadata processes (31% neither agreed nor disagreed). Similarly, half of respondents agreed that a program should identify a specific group—for example, a working group independent of other technical working groups, to manage metadata activities (50% responded $\geq$ “4” (agree or completely agree)), whereas 35% of respondents neither agreed nor disagreed with this statement.

Respondents largely agreed that a program should define a common infrastructure for metadata collection, storage, and access (80% either agreed or completely agreed), that a program should impose a common format and definition for metadata (85% either agreed or completely agreed), and that all program performers should be compelled to adopt and exercise the program standards for metadata activities (81% either agreed or completely agreed).

Specific to processes, respondents largely agreed that metadata should be collected automatically, when possible, and that it should not be an effortful activity engaged in by data/metadata users (77% completely agreed).

However, respondents were relatively evenly distributed across the response scale with respect to formalizing or concretizing metadata structure and processes a priori. Roughly one third of respondents reported disagreement (32%), agreement (28%), or neither agreement nor disagreement (32%) with the statement, “a program needs metadata and associated systems and processes in place before program activities start.”

Discussion: Respondents largely agreed that metadata should be managed by a central authority, that the program should define common infrastructure for metadata and that metadata should be automatically collected where possible. We believe these responses are based on the role of metadata as the backbone for information sharing across the multi-organizational, SD2 program, because it helped to communicate formats, structures, semantics, and context for the shared data. The lack of strong support for metadata structure and processes a priori to the project was most likely based on the impracticality of knowing what metadata were needed; the potential dynamism of that knowledge in this discovery program; and the high sunk cost to set up and revise metadata and data collection, organization, and management mechanisms.

Topic: Contextual factors that influence how metadata processes are developed and implemented.

Regarding the context that supports metadata process and usage, respondents largely agreed that it is important that an organization's metadata schemas are flexible and adaptable enough to accommodate different external metadata conventions (81% agreed or completely agreed). Respondents also largely agreed that it is important to have resources—for example, tutorials, external consultants, open-source software—that facilitate metadata use (85% agreed or completely agreed).

Discussion: During the SD2 program, metadata evolved based on two primary reasons—first because new metadata requirements and metadata content were discovered during the research activities, and second because metadata collection and management required development time to be implemented, hence becoming available to the program users. Therefore, this strong agreement about metadata evolving is understandable. Also, because of this constant evolution, it was always difficult for program members to keep up with the current state of the metadata (and other data and processes), therefore it was important to have resources to help facilitate the use of the metadata as it evolved.

Topic: What does metadata enable? Regarding the benefits of metadata, respondents reported that having metadata associated with their data improved their results by making them more:

- Interpretable (92%)
- Complete (73%)
- Credible (73%)
- Accessible (73%)
- Organized (69%)
- Accurate (69%)

Respondents largely agreed that it was easier for them to use other people's data when it has metadata associated with it (92% agreed or completely agreed). In adopting the perspective their data consumers, they saw similar value in metadata, largely disagreeing with the notion that others could use their data without associated metadata (58% disagreed or completely disagreed). Somewhat surprisingly, 31% of respondents neither agreed nor disagreed that others could use their data, even without associated metadata.

When asked to consider the value of metadata associated with questions of varying specificity (vs. generality), respondents tended toward finding the most value in metadata when they help answer a broad range of questions (39%) or neither specific nor broad questions (39%).

Discussion: This reflects the notion that metadata that are generalizable and can help answer more questions were valuable to the SD2 program, because the program sought to leverage a single platform for answering a multitude of diverse scientific questions. Different methods yielded different data that served to answer the same scientific question; thus, SD2 needed representations of metadata that addressed many questions without requiring a new metadata convention for each method/question.

Topic: Recent work that involved metadata. Respondents were asked to think about their most recent work involving metadata and then were asked to what that metadata described, what the metadata allowed them to do, what the metadata allowed others to do, and when they needed the metadata.

In recent work, respondents generated, shared, or used metadata that described the following:

- Data collection procedures—for example, protocol steps, sampling, training (79%)
- Data processing—for example, transformation, aggregation (75%)
- Experimental conditions—for example, temperature, time, duration (72%)
- Experimental materials—for example, concentrations, component parts, provenance (71%)
- Data collection instrument parameters—for example, setting, sensitivity, calibration, device model (54%)

The metadata these respondents generated, shared, or used allowed them to do the following:

- Analyze data (84%)
- Share data (80%)
- Interpret results (72%)
- Transform data (64%)
- Generate better data—for example, data that are more valid, accurate, or complete (60%)

These respondents reported that the metadata they generated, shared, or used allowed others to do the following:

- Analyze data (88%)
- Interpret results (88%)
- Transform data (71%)
- Generate better data—for example, data that are more valid, accurate, or complete (50%)
- Share data (50%)
- Transform data (71%)

Finally, respondents reported the time at which they identified the metadata they needed as:

- Prior to a protocol being generated (50%)
- While a protocol was being generated (54%)
- While a protocol was being executed (42%)
- While sharing data (46%)
- While analyzing data (67%)
- While sharing analytics or results (46%)

Discussion: In sum, respondents agreed on the categories of metadata used and on the value of that data, particularly for sharing and analyzing data, to themselves and to fellow performers. It is interesting to have a larger agreement on the analyze and interpret results categories than the sharing. This may indicate that even if you are not sharing the data that the metadata still improves analysis and interpretation of the data. Equally interesting was that metadata requirements (gaps) were most often identified while analyzing data. However, requirements were very frequently identified in other phases

of research from protocol generation to sharing of results; this speaks in part to the diversity of the use of metadata across the different roles in the program and their use of metadata.

Conclusions and Recommendations

The utilization of metadata on the SD2 program was an important part of the success of the program. In this task we carried out two activities to understand more about how metadata was used and what value it contributed to the program. Below we describe our conclusions and recommendations for metadata gleaned from these tasks and observations during the SD2 program.

Metadata and metadata-related processes, regardless of which nuanced definition one adopts, serve several critical functions on a multidisciplinary S&T program in which the disciplines and organizations bring different lexicons and conventions for representing data and metadata, as follows:

- They serve to deconflict and disambiguate language—for example, by standardizing terms (as was the case with the data dictionary in SD2).
- They enable multiple agents to reproduce, relate, or compare their analyses—for example, by specifying timestamped versions of data.
- They enable agents to select and understand an appropriate dataset—for example, a current, transformed, or unmodified set.
- They can specify the provenance of data, meaning how it was created. For example, one version of data may be raw or minimally processed data, whereas another may have been cleaned (e.g., outliers removed) and transformed (e.g., data types changed, missing data values imputed). In SD2, the data converge tool helped analysts identify the appropriate version of data to pull and analyze.
- They enable a system to store data and process data efficiently or accurately. For example, metadata describe the types of data and the range of eligible values for those data. This allows for reliable curation and quality control.

Recommendations

Below is a list of recommendations that were gleaned from the metadata interviews, the metadata survey, and observations during the SD2 program.

- Capture metadata from the start. That is not to say a formal standard must be in place at the start of the program; rather, simply having a plan to capture metadata in some way will go a long way toward mitigating metadata-related issues later in a program.
- Consider adopting existing ontologies and standards, or at least vetting them at the start of a program. This will preclude unnecessary work and redundancy with existing resources. For example, NIH's Library of Integrated Network-Based Cellular Signatures (LINCS) database already captures many of the terms the SD2 community defined.
- Be flexible about retaining versus omitting metadata processes. The SD2 infrastructure team was initially focused on automating the entire round trip (design-build-test-learn loop), but this turned out to be too large an undertaking for processes that were still evolving, and there was high potential for wasting time automating the nascent, volatile processes. Goals were reset and by the end of the program subsets of the process were automated with capture of appropriate metadata.

- Support users who want to query the sets of metadata that are captured. How can a naïve analyst query the data and learn the context of the experiment as efficiently as possible? In SD2, this was not an easy process, as the data catalog did not capture experimental intent. If experimental intent is critical to understanding and analyzing data, a program should explore whether and how experimental hypotheses can be represented in an ontology.
- In SD2, TA2 performers indicated that they would have benefited from having software engineering applied to metadata and data management prior to running experiments. The program could then have staggered performers' work. This might have mitigated some of the confusion and growing pains of Q0 (the hackathon that kicked off the program). Instead, the work required to stand up the SD2 framework delayed performers from doing the work they wanted (or needed) to do, and sometimes prevented them from doing that work.
- Reduce metadata to include only what is needed to perform the required processes. Metadata can describe the "world", but the question is how much is needed to provide the context for the work that must be done. Too much metadata increases the complexity of working with the metadata and data collected and utilized.
- Getting performer buy-in is critical. Future programs would benefit from sending a liaison to each organization to spend time with them, understand what they do, and introduce them to a metadata process. This liaison could be a performer who may be critical to establishing the relationships that make the work of negotiating metadata processes easier. For example, on SD2, there was a lot of lab-specific logic that needed to be translated into common terms. SD2 would have benefited from earlier outreach—for example, earlier visits to the TA3 labs for the purpose of understanding their methods and workflows. The value of these front-end activities notwithstanding, the metadata language will evolve over time. The experimental capture will look different over time and may look different for different classes of activity (e.g., different experiments). Thus, it is important to maintain flexibility.
- From the perspective of performers who develop and maintain data infrastructure, it is critical to better understand what analysts will do with a set of data (e.g., what subset of the data will they operate on?). For example, it would be useful to know whether analysts preferred a metadata standard (regardless of data provenance) to organization-specific conventions (e.g., parameter or instrument data). This knowledge might help determine the value of developing one fully query-able metadata resource.
- Identifying, mitigating, and solving metadata issues is a high priority, and it is worth spending time doing it. Solving metadata through automation can save a lot of resources. For example, RNA-seq tests cost thousands of dollars; thus, the cost of getting it wrong can be high. High-dimensional science is complex, with many inputs. Introducing (or not reducing) layers of uncertainty is problematic. Metadata need not represent one of those layers of uncertainty but rather can be used to mitigate errors before they occur.

Challenges associated with metadata

In spite of the high value of metadata in regard to the ability of organizations to collaborate on the program, it also carries some challenges.

Different measurement instruments and workflows require and generate different metadata.

Pipelines, rules, and jobs all capture information structure. However, what one records as metadata from one instrument is different from metadata from another instrument or measurement process. For example, a pipeline that reduces data can be represented one way (a reduction function), whereas a pipeline that uses ML to reclassify data is a different workflow and will be represented differently. One can attempt to mitigate this by standardizing metadata schemas; however, individual organizations with long-standing schemas may be reluctant to abandon or modify those schemas (and justifiably so). On SD2, metadata fixes, such as the data dictionary, were post-hoc because TA3 organizations refused to change their schemas. If TA3 labs had adapted to ingest TA2 artifacts (designs, protocols, requests) without dropping or modifying pieces, data sharing would have been easier.

Metadata are differently useful.

Data is the intersection of all user data needs. Metadata is everything outside that intersection. Defining those metadata is as difficult as anticipating the potential needs of unknown users.

Those who consume metadata differently will find different value in metadata. For example, TA1 analysts posit that metadata are important for analyzing and interpreting primary data, whereas TA3 experimentalists might suggest that metadata are more useful for debugging a protocol. Then there are those who simply want to receive and understand experimental results. There is value in providing metadata that help give a confidence interval for interpreting results. However, there is also value in withholding metadata that a user would not know what to do with even if they have it. For some users, metadata can become overwhelming. It seems that there is an optimal level of metadata that is contingent upon the needs of the audience; this level can be manipulated with respect to supply and demand.

You have to draw the line somewhere.

Although metadata can support experimental replication/reproducibility, they may not be sufficient for understanding experimental intent or results. Analogous to this is the qualification problem in AI: to make a decision or compute an outcome, one needs to qualify the condition under which actions are performed—that is, one must identify everything that has to be true to perform the action. The same largely applies to generating and processing experimental data. However, one can never generate an exhaustive set of metadata. One could identify all the observable conditions of an experiment, run it on different days, get different results, and the difference may never be explained. One can try to push metadata to a point to be sufficient and complete, but the theoretical limit is most likely impractical.

6: ON TRANSITION

Overview

The SD2 program developed new tools, data representations, processes, and systems to accelerate methods for designing biological and chemical constructs, building those constructs, testing them, and learning from them to initiate a new round of research (the Design-Build-Test-Learn loop). These tools and systems enabled SD2 performers to advance multiple scientific thrusts in dramatic ways—for example, by increasing the accuracy of predictions by a median multiple of 4x, the complexity of designs by a median multiple of 20x, and the speed at which they completed experimental cycles by a median multiple of 10x (see 1: On Selecting & Assessing Challenge Problems).

As impressive as these gains are, and as valuable as the tools that enabled them are, they do not fully represent the intended legacy of the SD2 program. The value of many DARPA programs is found not only in the advances in S&T generated during those programs but also in the transition of advancements beyond life of the program and beyond the community of program performers. SD2 sought to build an infrastructure that would persist beyond the program, to support a new paradigm of science that promotes collaborative work, facilitates access to resources, and can rapidly mobilize vast distributed capabilities in response to emerging high-dimensional problems that require application of sophisticated analytics to data at scale. To that end, SD2 performers made good progress transitioning selected technologies to service labs, government labs, and the broader scientific community. Near the close of the program, performers identified more than 50 tools developed or advanced in SD2 with transition value. They identified 18 different potential transition partners across government, industry, and academia. In this chapter, we discuss a few transition cases that illustrate some of the benefits that SD2 tools can bring to the broader scientific community, as well as some of the limitations of those tools, as perceived by scientists and engineers both inside and outside of SD2.

Methods, Assumptions, & Procedures

Aptima conducted meetings and semi-structured interviews with SD2 performers and a subset of five transition partners in summer, 2021, and winter, 2021/2022, to identify:

- The tools adopted or integrated by transition partners
- The workflow that called for SD2 tools
- Requirements for transitioning SD2 tools
- Friction points in the adoptive workflow that might lead to less use of the tools or require new solutions

Interview script

We applied the following script in semi-structured interviews with transition partners:

- What need did you have that led you to consider SD2 tools (e.g., components, systems, methods, data, data formats)?
- What SD2 tool(s) are you using?
 - How are you using it?
 - Is it installed locally or in a cloud environment?
 - Are you accessing tools hosted by an SD2 member?
- What kind of support are you getting from the tools author?
- Are you using it for a discrete application with a defined period of performance or do you expect extended use?
- Regarding integration of the SD2 tool:
 - What did it take to access and use the SD2 tool?
 - What were the infrastructure requirements?
 - For example, did it require upgrades, additions?
 - What were the critical dependencies?
- What issues arose with integration?
 - What amount of time and type of resources were required to learn the tool?
 - For example, did you access demos, documentation, or other training?

- What are some of the benefits realized with the tool?
 - What can you do now that you could not do before transitioning SD2 tools?
 - Have you realized any savings, for example, in time or cost?
 - Have you gained or lost any efficiency?
 - Have you realized any discovery-related impacts to your work?
- What are some of the limitations of the tool?
 - What gaps does it not address?
 - What do you need that SD2 components did not offer?
 - How did the output of SD2 tooling compare with your expectations?

Results & Discussion

Here, we present findings concerning five transition experiences:

- ESCALATE to Los Alamos National Laboratory (LANL)
- ESCALATE to Colorado State University (CSU)
- TASBE extensions and AutoGater to EQAPOL
- ETL / Data Analysis Platform components to Xilis
- Cell-Free (CF) Riboswitches Pipeline to Air Force Research Lab (AFRL)

Transition of ESCALATE to Los Alamos National Laboratory (LANL)

ESCALATE (Experiment Specification, Capture and Laboratory Automation Technology) is an ontological framework and open-source software package that allows scientists to specify an experiment, capture metadata, and structure data reports in ways readable by humans and machines. This enables scientists to author protocols that can be pushed to automated labs. ESCALATE simplifies data collection and machine learning (ML) integration (Pendleton et al., 2019).

At the time we interviewed a transition partner from Los Alamos National Laboratory (LANL), they were initiating a robot-executed chemistry workflow. They wanted to use ML and simulations to execute experiments and to select candidate experiments for future campaigns. They were seeking a way to move data back and forth between robots and machine learning. In their workflows, the robot does not have a programmatic interface; therefore, a human must set up the experiment. LANL believed that ESCALATE could address their interface gap, as it introduced a programmatic interface that enabled ML, rather than requiring a human, to specify an experiment.

The LANL use-case is somewhat unique because it introduces a tooling and a social element of transition. One of the members of the Perovskite CP working group, who worked on ESCALATE as both a developer and user, also transitioned to LANL, where he helped develop the experiment queue for candidate experiments in the human-in-the-loop workflow. Thus, he is in position to help analyze LANL's experimental requirements and communicate them back to the ESCALATE development team.

LANL has yet to complete a full experimental cycle using ESCALATE, as they were still bringing their robot online at the time of the interview. They had installed ESCALATE and were working through the initial learning curve, which included hosting and accessing the application locally. There is an instance of ESCALATE operating at Haverford College, which transition partners were invited to use; however, because of LANL's security posture, they were required to install their own instance locally. Thus, LANL staff can access a webapp version of ESCALATE that is hosted on an internal server.

To support access to and use of ESCALATE, LANL staff have an internal resource, as the former Perovskites CP group member is now a full-time employee there. As a first step to training LANL staff, he generated documentation for information-systems staff to install ESCALATE and for users to begin using it. Currently, documentation exists for some but not all of LANL's workflows. However, this is not problematic, as ESCALATE offers a relatively simple interface that is well documented. There are accessible notebooks that show how to use each endpoint, enabling a programmer to learn it without much difficulty.

LANL anticipates that the largest costs associated with adopting ESCALATE will be for maintenance of their web server. It will require periodic updates to stay current and secure. If they fall behind on organization-required updates, applications running on that server will be blocked. Other costs of deploying ESCALATE are not yet known.

If this transition succeeds, LANL will be able to bring together ML theory and automated experimentation to support work never done before. They seek to set up an autonomous discovery loop with only minimal human intervention. LANL anticipates that if data are stored systematically, as they are through ESCALATE, it will be easier to select from candidate experiments. They expect that a successful campaign, with ESCALATE driving robotic experimentation, should reduce human time and effort, as well as mitigate (or remove) human biases in discovery in liquid phase chemistry. Moreover, they expect ESCALATE to reduce the time and work involved in switching between scientific problems and reorganizing infrastructure, thus supporting a more general approach to discovery (e.g., across different chemistry phases).

As optimistic as LANL is about ESCALATE enabling an automated workflow, they see some challenges with using it. For example, ESCALATE might be too general. In updating ESCALATE from a previous version that was tailored to perovskites and thin film to a more general version that can be used in LANL, the data model became more open to interpretation by different users. For example, LANL will use the data ontology represent specific data from distinct experiments across a broad range of science. LANL anticipates that this will be a challenge. Their caveat to future users is to ensure those users understand the terminology and ontology with respect to both experiments and datasets. There are different ways of expressing experiments in ESCALATE, so one must pick an interpretation and use it consistently. Users can choose different meanings for different datasets over time, which could cause confusion and data incompatibility in the future.

Transition of ESCALATE to Colorado State University (CSU)

Similar to the state of transition at LANL, an academic chemistry lab at Colorado State University (CSU) had just begun to integrate ESCALATE when we interviewed them. This particular lab uses organic fluxes to synthesize solid state materials. They have run many reactions to make different materials, have collected large amounts of data, and they are using ML to extract insights about reactivity. For CSU, ESCALATE appeared to address a gap in streamlining data curation. They wanted:

- A system to organize and standardize reactions and store data
- To introduce ML to identify patterns worth examining experimentally

CSU is auditioning ESCALATE to test whether it can replicate the work of a lab member who already collected data, applied ML, and selected candidate experiments via "brute force" computing.

Similar to the LANL transition, the ESCALATE development team prepared a version of the app to run on a local server. From CSU's perspective, it was uncomplicated to set up and does not require much

support to use. To help CSU lab staff become more familiar with the application, ESCALATE developers demonstrated how to engage with ESCALATE's API. CSU lab staff commented that many of them know python, which made programming in ESCALATE relatively easy. However, CSU mentioned that there is a non-trivial level of effort to get started using ESCALATE. For example, it takes time to create new workflows for the oxynitride research CSU conducts. The time to plan and run experiments should decrease once CSU establishes a base of workflows (e.g., chemicals, reagents, experiment protocols) to choose from.

CSU hopes to benefit from ESCALATE initially by replicating experimental results (cf. the Perovskites initial use of ESCALATE to select experiments). Using ESCALATE in this way is a low-risk, exploratory effort, as the lab can work with what they know, get over the learning curve, and then apply ESCALATE to a new chemical system. Similar to LANL, CSU expects ESCALATE to yield reduced time and errors, especially when using it as a data storage tool. By admission, as in many academic labs that employ and train a diversity of students and scientists, some CSU lab members are not proficient at keeping records; thus, standardized data storage will be a boon.

CSU also expects to see benefits regarding data analysis and the directing of synthetic attempts. They expect ESCALATE to provide useful insights that will point to profitable areas of a research space and focus the work of junior scientists. However, CSU has also perceived limitations in ESCALATE. Currently, ESCALATE does not link to all of the CSU lab's algorithms. Thus, CSU is not able to query their data for properties of different reaction and materials and then pull those directly into ML algorithms. This is a future ask of the ESCALATE development team, who are currently engaged in this transition activity.

Transition of Tool-chain to Accelerate Synthetic Biology Engineering (TASBE) Extensions and AutoGater to EQAPOL

The External Quality Assurance Program Oversight Laboratory (EQAPOL) designs, implements, and executes quality control in laboratories that conduct HIV/AIDS research and vaccine trials internationally. These HIV/AIDS laboratories routinely apply flow cytometry, which measures the prevalence of fluorescent colors in live cells. It is a considerable challenge to compare flow cytometry measurements between laboratories (and often within them) given the variance introduced by use of different instruments or different calibration of the same instrument. Also challenging to standardization and reproducibility are differences in the skills and practices of human analysts who manually apply gates (i.e., criteria or thresholds) to assess measurements. EQAPOL and its client laboratories are also hampered by the time cost of flow cytometry analyses, which is driven in large part by scale of the task: 15-25 marginally distinct fluorescent colors must be distinguished in plots of approximately 20,000 cells.

Members of the SD2 performer teams at Duke, Raytheon BBN, and Netrias are working to demonstrate how specific SD2 tools can help EQAPOL solve these problems. For example, the SD2 extensions to TASBE analytics automate the manual gating procedure. Use of standard control beads, introduced by SD2, normalize the measurements from instruments that differ in capability and use because they are created by different manufacturers or are different models. The SD2 data ingest workflow will automate the onerous task of cleaning data and standardizing its presentation in data frames. Lastly, the SD2 infrastructure developed at TACC is available to host data ingest and analyses.

The SD2 performers and their partners from EQAPOL hope that a successful demonstration to NIH will persuade the Institute to increase its current funding to EQAPOL. The new funding would enable the team to integrate this SD2 tooling into the QC process. If that extension bid succeeds, then SD2 performers are positioned to develop demonstrations of additional capabilities needed by EQAPOL. For

example, an integration of TASBE with the industry standard FlowJo™ analytics package would both standardize data and automate gating in analyses. Adding SD2-developed AutoGater would exclude dead cells from flow cytometry data. This would increase the accuracy of analyses for two reasons. First, excluding data from dead cells reduces the color corruption that these cells introduce into a sample as they die. Second, excluding dead cells using AutoGater eliminates the need for colored stains that are used to discriminate dead cells from live cells, but that introduce spectral overlap between the stain and cell fluorescence, thus reducing measurement accuracy.

Transition of ETL / Data Analysis Platform Components to Xilis

Xilis develops precision therapy technologies and drugs to treat cancer patients. The discovery and design activities needed to develop these technologies are complex and executed on a large scale. Moreover, they are bifurcated at Xilis. The group that designs and conducts experiments is different from the group that analyzes the data—a distributed scientific workflow similar to that of SD2. As was the case in SD2, Xilis strives for robust communication between its experimentalists and analysts, a condition that is difficult to achieve. For example, documentation and communication about experimental intent convey the context that analysts need to make smart choices between analytics (e.g., statistical tests, machine learning algorithms) and return useful products. Yet, at Xilis—and at many other clinical labs that separate experimentation and analysis workflows—experimentalists often fail to reliably codify and communicate experimental intent for analysts. Thus, Xilis saw a solution in SD2's approach to communicating experimental intent across multiple distributed workflows. In addition to supporting better data and metadata sharing, Xilis is exploring better ways to automate its workflows.

A former member of the SD2 team from Duke recently joined Xilis and has begun a campaign to transition specific SD2 techniques and technologies to the organization. In his position as a leader and liaison between experimenters and analysts at Xilis, the former SD2 team member is applying the communication practices developed in SD2 to improve the understanding of experimental intent, experimental conditions, and other factors across the cultural divide. He is working to integrate several SD2 technologies into operations at Xilis. DataConverge will regularize data frames and enforce version control. It will serve as a flexible bridge carrying complicated and dynamic data between many sources and multiple human and machine data consumers that require structured, consistent data with provenance. The Pre-Computed Data Table is also a promising component for Xilis, as it standardizes the computation of measurements and their formats for analysis. For example, in SD2, the Pre-Computed Data Table was used to compute a standard growth statistic over time-series data. This eliminated variance in the specific growth statistic, algorithm, computational implementation, and resulting measurement values. Finally, an SD2 tool for quality control may have value to Xilis because it identifies variables with high or low variance that may explain failures in experiments or in data processing.

Transition of Cell-Free (CF) Riboswitches Pipeline to Air Force Research Lab (AFRL)

The transition of the CF Riboswitches pipeline to AFRL served two purposes. First, it was a stress-test of a pipeline that SD2 performers hoped would allow seamless integration between an SD2 lab and DoD service labs (and eventually extend to the commercial domain). Second, it was an attempt to efficiently fill a knowledge gap in engineering riboswitches for AFRL and for work funded under the Synthetic Biology for Military Environments (SBME) program. AFRL sought to deploy the SD2 pipeline (and performers) to initiate a design-build-test-learn cycle to characterize the sequence-to-function relationship between transcriptional riboswitch variants derived from synthetic serotonin aptamers and the natural *B. subtilis* pbuE expression platform. Notably, this transition activity occurred through much of the third phase of the SD2 program, thus ensuring that SD2 performers would be available to support

all aspects of the transition. In addition, AFRL reduced the risks associated with technology transfer by using SD2 tools through the program's infrastructure, just as an SD2 performer would do, rather than by attempting to replicate that infrastructure at AFRL.

AFRL confronted tradeoffs regarding the value of some components of the pipeline—for example, the Experiment Request and Intent Parser—compared to alternative authoring tools such as Benchling. The DoD uses Benchling because it facilitates collaboration over data. Benchling has a lab notebook and a repository with materials and one can link back and forth; for example, one scientist can see a strain made by Army collaborator and link to that collaborator's notebook. However, Benchling is costly. By contrast, the Google Docs-based tools developed on SD2 are inexpensive and offer interoperability with other Google-based artifacts. However, this can also be problematic, as the Google platforms for collaboration (e.g., Google Drive) are not accessible to all government personnel. Thus, large-scale adoption of SD2 tools by service labs would likely require migration to a different platform.

Although AFRL used many components of the SD2 pipeline, including the Experiment Request, Intent Parser, and post-processing tools, they experienced challenges with some and simply opted not to use others. For example, they used the Experiment Request document, but found it challenging to specifically identify what could be put into the requisite tables. They had to access the Data Dictionary to see what was available but could not find data representations or parameters they needed; for example, they did not know what timepoints in a time-series protocol were eligible for use. Consequently, they abandoned use of the Data Dictionary and opted instead to seek feedback from SD2 performers and to consult SynBioHub (SBH) directly, as they found that repository to be more intuitive to use. In response to the challenges experienced with completing an experiment request, they reported that some brief documentation on how to progress through the experiment request process (and ensuing processes) would be helpful.

Despite some logistical hurdles, AFRL and SD2 ultimately ran three cycles of CF-Riboswitches experiments. There were early challenges that slowed the pace of the first cycle (see 2: On Discovering and Assessing Methods), but they overcame those challenges and eventually cut their cycle time by more than two thirds. The AFRL transition partner estimated that SD2 tools reduced their data-generation and analysis time by two-fold.

Transition Pain Diary

Near the end of the program, as performers were in the midst of transition activities (e.g., identifying partners, demonstrating their tools, refining their pitches for their tools), Aptima facilitated a brief quarterly-meeting session titled, "Transition Pain Diary." The purpose of that session was to allow performers to share their transition stories, to identify transition difficulties ("pain"), and work to identify or develop strategies for addressing those limitations. Some of the more salient sentiments raised in that session align with and complement the sentiments raised by transition partners in the previously described cases. For example, during the pain diary session, performers commented that documentation, packaging, and security validation of software for transition were logistical challenges; and that front-end investigation of a transition partner's needs and capabilities requires significant time and effort yet is critical.

Conclusions & Recommendations

In the transition of ESCALATE, its developers leveraged comprehensive documentation and open engagement (e.g., in demonstrating the tool and training new users) to anticipate and mitigate perceived barriers to entry. In one case, they transitioned a developer to the transition-partner organization; in the other case, they engaged regularly with new users to help them through the learning curve. In both cases, new users cited comprehensive documentation and reach back to the development team as features that eased early transition and led to perceptions of ESCALATE as a potentially valuable addition to their workflows. Hence, a recommendation that follows is to make the learning curve for a new tool as painless as possible, through documenting the tool comprehensively and by being accessible to transition partners.

In the transition of data processing tools to EQAPOL and Xilis, SD2 performers engaged in outreach to describe and demonstrate the value of those tools. Again, SD2 transitioned a former member to the transition-partner organization (Xilis), easing the logistical challenge of troubleshooting emergent issues with the transitioned tools. In the specific case of EQAPOL, SD2 performers have engaged with the transition partner to identify specific needs that SD2 tools can address. The SD2 infrastructure comprises over 50 new or newly modified tools, not all of which will be useful to a potential partner. The teams transitioning tools to EQAPOL began by identifying a subset of tools to demonstrate against EQAPOL's priority needs (successfully compare data across labs and secure funding) and a secondary set of tools that can augment EQAPOL's infrastructure in the event that it receives additional funding. Successful transition in this case points to the need to fully understand and carefully consider a transition partner's needs, match tools to those needs, and advocate for auditioning those specific tools. This is more of a focused-targeting approach versus a scattershot approach to transitioning tools. Moreover, the tools that SD2 performer sought to transition to EQAPOL were largely computer science-informed tools (versus biology-informed tools) being introduced to biologists; therefore, it was important to include computer scientists and biologists in the outreach to be able to describe both how the tool worked and what it offered biologists. This type of multidiscipline outreach is important when engaging with transition partners who may not be familiar with the multiple disciplines that contributed to a tool's development and functioning.

In the transition of CF-Riboswitches pipelines to AFRL, SD2 performers recognized that they would not be able to port their tools to AFRL's infrastructure. Therefore, they offered the SD2 infrastructure and large parts of the socio-technical system in support of transition. Problems with accessing and using early-stage tools, such as the Experiment Request, led to a turbulent start to the transition activity. Fortunately, SD2 performers were actively engaged in supporting the transition, troubleshooting issues, and offering guidance for re-engaging with the infrastructure. This transition case reflects at least two challenges for transitioning SD2 tools: (1) security restrictions that prevent installation of SD2 tools and infrastructure (e.g., TACC computing infrastructure and Google Docs) at partner sites necessitate an added layer of support to help partners access the SD2 platform before they can engage with the tools, and (2) the human component of the socio-technical system may not be available in perpetuity. As was noted in 3: On Socio-Technical Systems in Scientific Enterprises, as SD2 ended, performers likely disengaged and immersed themselves in other communities and work. Therefore, the human support required to facilitate transition of SD2 tools are diminishing or simply unavailable.

We draw several recommendations from these SD2 transition stories. First, tool authors should ensure that their tools are comprehensively documented and that those documents have been tested by naïve users (as analogs to future transition partners). Such documentation will mitigate barriers to entry for some transition partners. A second recommendation, one that is more costly and must be weighed against potential gains, is to consider the relative costs of making tools accessible on their DARPA-funded platforms, and of packaging tools to make them portable to other platforms. SD2 performers did the latter with Aquarium, containerizing it for access via Docker, and noted that it was indeed costly, but significantly increased the tool's accessibility. Third, successful transition of technology requires deep support by people, even transition of people with technology. As one SD2 performer in the Transition Pain Diary session said SD2 performers need to "build the bridge and also be the bridge." In many cases, SD2 performers did this. They refined and productized their tools and engaged in the outreach necessary to put those tools in the hands of people who can use them.

REFERENCES

- Al-Ghassani, A.M., Kamara, J.M., Anumba, C.J., & Carrillo, P.M. (2006). Prototype system for knowledge problem definition. *Journal of Construction Engineering and Management*, 132 (5): 516-524.
- Brown, T. A., Braun, M. T., Perry, S. K. B., Orvis, K. L., McCormack, R. K., & DeCostanza, A. (In review). An intensive idiographic exploration of boundary spanning in a complex, real-world multiteam system.
- Cohen, M.S., Freeman, J.T., & Thompson, B.T. (1998). Critical thinking skills in tactical decision making: a model and a training method. In J. Canon-Bowers & E. Salas (Eds). *Decision-making under stress: implications for individual and team training* (pp. 155-189). Washington, DC: American Psychological Association Publications.
- CASRAI. (2022). CReDiT – Contributor Roles Taxonomy. CASRAI. Retrieved from <https://casrai.org/credit/>
- DARPA. (2016). Broad Agency Announcement: Synergistic Discovery and Design (SD2), HR001117S0003. November 22, 2016.
- Directorate, OECD Statistics. (2018). OECD Glossary of Statistical Terms - Reference metadata Definition. Retrieved from <https://stats.oecd.org>
- Emery, F. E. (1959). Characteristics of socio-technical systems: A critical review of theories and facts. Human Resource Centre, Tavistock Institute of Human Relations.
- Fiore, S.M., Smith-Jentsch, K.A., Salas, E., Warner, N., & Letsky, M. (2010). Toward an understanding of macrocognition in teams: Developing and defining complex collaborative processes and products. *Theoretical Issues in Ergonomic Science*, 11(4):250–271
- Freeman, J., Leins, D., Bell, & The SD2 Research Consortium. (2019). Selecting and assessing challenge problems. *Theoretical Issues in Ergonomics Science*, 20:1, 27-38, doi:10.1080/1463922X.2018.1485987
- Garrett, S. (Director). (2022, April 1). SD2: Developing a socio-technical system to advance science in complex domains. *Aptima*. Film. <https://www.aptima.com/sts/>
- Gaunt, D. (2020, August 13). Communication skills: Did you know there are 5 levels of listening? Retrieved from <https://blog.blackswanltd.com/the-edge/did-you-know-there-are-4-levels-of-listening>
- Greene, M. (2007). The demise of the lone author. *Nature* 450 (1165). doi:10.1038/4501165a
- Keller, B., Miller, A., Newman, G., Vrana, J., & Klavins, E. (2019). Aquarium: The Laboratory Operating System version 2.6.0, doi:10.5281/zenodo.2583232
- Klein L. (2013). What do we actually mean by 'sociotechnical'? On values, boundaries and the problems of language. *Appl Ergon*. 45(2):137-42. doi:10.1016/j.apergo.2013.03.027
- Klein, G. (2013). *Seeing what others don't*. Philadelphia: Public Affairs.

- Klein, G. A. (1993). A recognition-primed decision (RPD) model of rapid decision making. In G. A. Klein, J. Orasanu, R. Calderwood, & C. E. Zsombok (Eds). *Decision making in action: Models and methods* (138-147). Westport, CT, US: Ablex Publishing.
- Leavitt, H. J. (1965). Applied Organizational Change in Industry; structural, technological and humanistic approaches. In J. G. March (Ed.). *Handbook of organizations* (1144-1170). Chicago; Rand McNally & Company.
- Leeds University. (2022). Socio-technical systems theory. Retrieved from <https://business.leeds.ac.uk/research-stc/doc/socio-technical-systems-theory>
- Leins, D., Haase, S., Schrier, J., Eslami, M., Yeung, E., Freeman, J., & The SD2 Research Consortium. (In review). Collaborative methods to enhance reproducibility and accelerate discovery.
- McComb, C., Cagan, J., & Kotovsky, K. (2017). Optimizing design teams based on problem properties: Computational team simulations and an applied empirical test. *Journal of Mechanical Design*, 139(4), 041101-01— 041101-12.
- Mohaghegh, M. & Furlan, A. (2020). Systematic problem-solving and its antecedents: a synthesis of the literature. *Management Research Review*, 43(9), 1033-1062. doi:10.1108/MRR-06-2019-0284
- National Institutes of Health. (2021, October 8). Bridge to Artificial Intelligence (Bridge2AI). *NIH*. Retrieved from <https://commonfund.nih.gov/bridge2ai>
- National Research Council, Committee on the Science of Team Science, and Board on Behavioral, Cognitive, and Sensory Sciences, Division of Behavioral and Social Sciences and Education. (2015). *Enhancing the Effectiveness of Team Science*. Washington, DC: The National Academies Press.
- Newell, A. & Simon, H.A. (1972). *Human Problem Solving*. Upper Saddle River, NJ: Prentice Hall.
- NIST. (2020, September). *Special Publication 800-53, Revision 5. Security and privacy controls for information systems and organizations*. Retrieved from <https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-53r5.pdf>
- NIST. (2022). *Computer Security Resource Center*. Retrieved from <https://csrc.nist.gov/glossary/term/metadata>
- Pendleton, I. M., Cattabriga, G., Li, Z., Najeeb, M. A., Friedler, S. A., Norquist, A. J., Chan, E. M., & Schrier, J. (2019). Experiment Specification, Capture and Laboratory Automation Technology (ESCALATE): A Software Pipeline for Automated Chemical Experimentation and Data Management. *MRS Communications*, 9(3), 846–859.
- Pólya, G. (1957). *How to solve it*. Garden City, NY: Doubleday.
- Simon, H. (Dec. 10, 1978). Rational decision-making in business organizations. Speech made upon acceptance of the Nobel prize, Stockholm, Sweden. Retrieved from <https://www.nobelprize.org/uploads/2018/06/simon-lecture.pdf>.

- Vrana, J., de Lange, O., Yang, Y., Newman, G., Saleem, A., Miller, A., ... & Klavins, E. (2021). Aquarium: open-source laboratory software for design, execution and data management. *Synthetic Biology*, 6(1), doi:10.1093/synbio/ysab006
- Wilkinson, M., Dumontier, M., Aalbersberg, I., *et al.* (2016). The FAIR guiding principles for scientific data management and stewardship. *Sci Data* 3, 160018. doi:10.1038/sdata.2016.18
- Zeng, M. (2004). Metadata Types and Functions. *Metadata Basics version 2.0*. Retrieved from <https://marciazeng.slis.kent.edu/metadatabasics/types.htm>.
- Zuboff, S. (1988). *In the age of the smart machine : the future of work and power*. New York: Basic Books.

APPENDICES

Appendix A: Selecting and Assessing Challenge Problems

THEORETICAL ISSUES IN ERGONOMICS SCIENCE
<https://doi.org/10.1080/1463922X.2018.1485987>



Selecting and assessing challenge problems

Jared Freeman^a, Drew Leins^a, Conrad Bell IV^b and The SD2 Research Consortium¹

^aAptima, Inc., Arlington, VA, USA; ^bECS Federal LLC, Fairfax, VA, USA

ABSTRACT

Organisations conducting research programs often focus the work of their scientists and technologists on challenge problems (CPs). These challenges are designed to ensure that progress is measurable and relevant to the goals of the program sponsor. Generating and selecting pertinent CPs is difficult, as is assessing their value. We describe a method of generating and selecting CPs and its application in a highly collaborative, multi-organisation research program. Thirty-eight biologists, chemists, mathematicians and computer scientists across academic, commercial and government organisations generated and ranked their top choices from among 12 richly described candidate challenge problems. A ranked-choice voting formula was applied. Five CPs were highly scored; the remaining seven were distributed across a lower range of scores. The program sponsor subsequently directed researchers to address six CPs, including the elected five. Analysis of the rationales that participants offered for their CP rankings revealed four domain-independent dimensions of value: capability, speed, impact and synergy. These dimensions of value can help managers of interdisciplinary research programs systematically select a portfolio of CPs that will efficiently apply utilise resources towards program goals and facilitate measurement of scientific progress.

ARTICLE HISTORY

Received 6 March 2018
Revised 20 May 2018
Accepted 4 June 2018

KEYWORDS

Challenge problem;
problem definition; research
policy; voting

Relevance to human factors/ergonomics theory

Doing so addresses the significant human factors challenge of managing complex research teams and tasks in a manner that returns high research value for the funder's dollar.

Introduction

We choose to go to the moon in this decade and do the other things, not because they are easy, but because they are hard, because that goal will serve to organise and measure the best of our energies and skills.

John F. Kennedy, President of the United States, 12 September 1962.

CONTACT Jared Freeman freeman@aptima.com Aptima, Inc., 1010 North Glebe Rd. Suite 700, Arlington, VA 22201, USA

¹Members of the SD2 Research Consortium are listed in the Acknowledgements section.

© 2019 Informa UK Limited, trading as Taylor & Francis Group

Challenge propels discovery and innovation. President John F. Kennedy set the bold challenge of a manned moon-landing in May of 1961. Success on scientific, technical and political fronts followed an investment of eight years and \$155 billion dollars (adjusted for inflation). Today, many Federal research organisations routinely define scientific *challenge problems* (CPs) to guide research. For example, one public notice from the Intelligence Advanced Research Projects Activity (IARPA) states:

IARPA intends to measure the utility of proposed solutions [on the HECTOR program] by trying to solve IARPA-furnished challenge problems. An example challenge problem might be to perform various properly authorised statistical analyses on census data without compromising privacy of any individuals ...

Intelligence Advanced Research Projects Activity (2017).

CPs serve at least two major functions. First, they drive investment and progress towards some distribution over basic research, use-inspired basic research (*Pasteur's quadrant*, per Stokes 1997) and applied research. Kennedy's moon-landing challenge focused primarily on achieving a single, practical mission; as a side effect, it produced fundamental advances in geology, chemistry, medicine, material science, physics and other sciences (National Aeronautics and Space Administration 2004). Another applied research effort, the Manhattan Project, had a similar effect in advancing our fundamental knowledge of physics. The program from which we report findings has a portfolio of CPs with long-term goals of creating new scientific methods and technologies and accelerating discovery in complex areas of biology and chemistry. However, each of these CPs will also advance understanding and generate solutions for a more immediate, practical problem. In sum, programs define CPs to distribute science and technology investments across multiple levels of research in a planned way.

The second function of CPs is to support measurement of progress. CPs typically define very few goals; for example, get a man to the moon. Therefore, the metrics of progress can be standardised across research participants. Defined goals are often tangible. Therefore, the metrics can be easily understood and communicated. Measurements help program managers and their institutions promote and defend funding decisions. Measurements also provide evidence of the effectiveness of research products, evidence that can persuade operational and industrial users to fund 'last mile' transition from the laboratory to the field.

If CPs drive and justify investment, then it is important to select relevant CPs and to understand the source of their value. Doing so addresses the significant human factors challenge of managing complex research teams and tasks in a manner that returns high research value for the funder's dollar. There is, however, a dearth of research that directly provides insights on how to generate, select and value scientific problems. Consequently, we turn for inspiration to literature in the closely-related fields of defining problem spaces, solving problems and discovery.

Al-Ghassani and colleagues (2006) proposed a rational process for defining a problem space. They suggested that the research begin with a loose characterisation of the problem. Specific components and processes for addressing the problem are then identified, including key theories, methods, technologies and datasets. Finally, a

determination is made regarding how these components and processes fit together. This survey of problem and solution information tends to define the problem space crisply enough that the research can, consistent with Pólya's dictum (Pólya, 1957): understand the problem, plan a route through it, travel that route and test for success, failure and discoveries. One implication for challenge problem designers is that they pilot these steps to ensure that: (1) the problem can be expressed and comprehended; (2) the solution components – theory, science and technology – exist and are discoverable and (3) any technical interfaces between the components exist or can be constructed.

McComb, Cagan, and Kotovsky (2017) characterised the space of structural engineering problems quantitatively in three ways. First, there must be alignment between all objective functions (e.g. 'achieve a design that is light and strong') and that these functions may influence the shape of the design space. Secondly, the *local* structure of the design space indicated the sensitivity of solutions to parameter values and, thus, the likely efficiency of local searches. Thirdly, the *global* structure of the design space bounded the number of potential solutions. Accordingly, CP design should include a careful examination of goals or objective functions by which solutions will be measured. Another implication is that CP design should construct a map of the problem space, one that measures the domain and, if possible, partitions it into areas that are quantitatively or qualitatively distinct.

In the rich literature concerning human problem solving, we find rationalist and naturalist threads that provide some guidance concerning CP design. Newell and Simon (1972) typify the rationalist approach. They proposed that problem solving is a function of the solver's choice of problem representation (or model) and of one or more strategies (or *programs*) from a catalogue of heuristic search strategies. The main implication for CP design is that there be multiple distinct and promising representations of the problem (and participants willing to use them). This will increase the variety of solution strategies and the likelihood of discovering a successful solution.

Newell and Simon noted that their information processing approach did 'not shed much light' on the matter of how humans select models and strategies. The naturalist school of problem solving research attempted to reverse engineer this black box. The theory of recognition-primed decision making (Klein 1993) asserted that domain experts should recognise the problem at hand as an instance of a case in memory, retrieve a solution and apply mental simulation to test and revise the solution. The implicit guidance for CP design is that the problem be carefully distinguished from similar but solved cases to ensure that researchers solve for the new, distinguishing features.

Cohen, Freeman, and Thompson (1998) described a process of metacognitive control of recognitional decision making and generation of novel solutions under uncertainty, high stakes and time constraints. A training method based on this theory had large, reliable and positive effects among relatively expert decision makers. The key implication for CP design derives from the framework: CPs should present sufficient uncertainty, impact and urgency to inspire deep and critical thinking.

Klein (2013) presented a theory of discovery (a phenomenon adjacent to problem solving) to account for historic cases that seem to violate the simple model by Wallas

(1926), in which the discoverer engages in preparation, incubation, illumination and verification. Klein argued that discoverers attend to surprising connections and contradictions and use these as cues to test and revise theories or mental models. This guides us to design CPs in which extant theory or methods fail to explain, predict or generate relevant data. This implication would be minor if CPs typically generated data sufficient to produce surprise and drive progress, but many CPs do not.

In summary, the literature concerning CP design is scant or non-existent, although related literature offers some guidance for CP designers. At least some solution components (theory, methods, data) should populate the problem space. It should be possible to map that space, to draw or represent it in multiple ways to ensure multiple solution opportunities. The CP should be well-distinguished from solved cases and sufficiently rich in uncertainty, impact and urgency to inspire focused and critical thought.¹ A view across this literature reveals the common requirement for domain expertise in CP design. In multi-disciplinary research programs, it is generally not the case that an individual or group from one scientific discipline can design a good CP. For example, it might be unreasonable to expect a biologist to generate a CP of significant interest and value to a computer scientist. In the program on which we report, researchers had such domain expertise. Consequently, the program manager tasked them with generating a set of CPs relevant to the program's goals. They more than obliged by generating a set of CPs far exceeding the research program's limited pool of resources (e.g. time, money and labour). This raised a significant question. By what means could the set of CPs be evaluated to identify a subset of those most worthy of investment?

Below, we describe a method for systematically selecting CPs and report an analysis that will help program managers to characterise the value of a portfolio of challenge problems.

A method for collaborative generation and selection of challenge problems

The method reported here grew out of a research program with aggressive scientific goals coupled with a high degree of collaboration. Researchers in the program consisted of biologists, chemists, mathematicians and computer scientists arrayed in more than a dozen teams. These researchers gathered together at a recent workshop, coming from 18 geographically distributed organisations of different types from both the academic and commercial sectors. The program sponsor asked workshop participants to generate CPs that engaged each other in significant discovery and invention in biology, chemistry and analytics. Below, we present the methods and findings of this effort to generate and select challenge problems, and findings from a subsequent analysis of the value that researchers placed on these CPs. This process is illustrated in Figure 1.

The researchers generated 24 candidate CPs independently and in small teams. They described CPs using a template that required: the CP name; the author and collaborators; short- and long-term goals; automation requirements for execution of experiments, data processing and analysis steps; alignment with sponsor objectives and relevant scientific literature. They were well-prepared to provide this information



Figure 1. A process for defining, selecting and characterising the value of challenge problems.

because many participated in drafting a guiding document that laid out a range of potential applications; experimental issues; data screening and analysis issues; product components; issues of experimental reproducibility and initial concepts for challenge problems.

As an example, one researcher proposed a candidate CP to develop a detector for a dangerous chemical agent. The CP's first quarter goal was, 'Design and test [products] that detect [chemical components of threat X] with the maximum possible dynamic ranges'. The first year goal was, 'Repeat workflow on up to 100 [chemical components of threat X] in parallel'. At four years, the goal was to detect and neutralise the threat. Data from this CP, noted the author, would be processed by existing software workflows as well new analysis software being developed by another program researcher.

Researchers independently read most or all of the CPs. In a large meeting, they collectively culled 12 CPs that were redundant of others or that were subsets of other CPs.

Participants were then asked to rank order the five CPs they most preferred from among the remaining 12 CPs. Thirty-six researchers completed this task, although some ranked fewer than five CPs. We performed an informal analysis to identify five highly ranked CPs. Problems that were most frequently among the top five selections were prioritised above all others. Conflicts in rankings (ties) were resolved by selecting CPs that were most frequently among the top two choices. The top five problems were reviewed by the sponsor, who recommended extending the initial list to include a sixth highly ranked CP that closely aligned with program objectives. This informal method was fast, but not necessarily reproducible.

A formal method was subsequently applied to verify the rankings before the sponsor and research team invested heavily in the selected CPs. This algorithm, reported in [Appendix A](#), accounted for the incompleteness of the rankings. This incompleteness arose because each participant ranked only their top five (or fewer, in some cases) of 12 CPs, not all 12.

Table 1. Quantitative analysis of CP rankings, ordered by a score that incorporates rank and frequency data, and compensates for rankings of a subset of all candidates by all participants.

CP ID	Formal score (rank)	Informal rank findings
h*	0 (1)	2
x*	5 (2)	4
a*	19 (3)	1
u*	36 (4)	3
c*	58 (5)	6
M	80 (6)	n.a.
K	88 (7)	n.a.
v*	93 (8)	5
b	97 (9)	n.a.
j	98 (10)	n.a.
d	106 (11)	n.a.
g	107 (12)	n.a.

CPs marked with "*" were selected for execution.

Shortly thereafter, a survey was conducted to elicit information about the level of engagement in the six selected challenge problems. This survey was presented as a table of CPs by performers. Each participant populated cells of their row with a score to indicate whether they: '(1) definitely can work the CP now; (2) probably can; (3) cannot determine yet; (4) probably cannot; (5) definitely cannot'. The survey results were used to ensure that each researcher could engage in at least one CP, and to identify any CP that was without a research team. The results were also used by the sponsor as a resource management tool to avoid potential resource conflicts or shortages. For example, a researcher might be directed to engage in a particular challenge problem as a subject matter expert or because a required capability was housed at their institution. Alternatively, a specific combination of researchers might be directed to engage because they were all required to successfully complete a challenge. Finally, the program sponsor would direct certain researchers away from one or more challenges to avoid the possibility of resource shortages, as would be the case if researchers distributed their resources too thinly across too many CPs.

Table 1 lists, for each CP, (1) the formally computed score and (2) the rank generated from summary ranking statistics. The analysis largely confirmed the informal findings with the exception that one CP (ID = v) was not among the five best ranked, but was roughly equivalent in its middling rankings to five others. In addition, the sponsor identified one CP (ID = c) to pursue that was not among those identified informally but that later emerged in formal analysis as the fifth highest ranked CP. The survey of engagement opportunities revealed a reasonable distribution of scientists across CPs. The program manager made modest adjustments to their self-assignments based on these data and other information.

Analysis of the dimensions of CP value

An analysis was conducted to identify the dimensions of value on which workshop participants ranked the candidate CPs.

Table 2. Four dimensions of CP value rarely co-occurred.

	Capability	Speed	Impact	Synergy
Capability	1	–	–	–
Speed	–0.12755	1	–	–
Impact	–0.32329	–0.04495	1	–
Synergy	–0.15177	0.035406	0.035806	1

Thus, they were weakly correlated and relatively independent of one another.

Method

In addition to ranking CPs, the scientists submitted rationales for their selection of each CP they ranked (in the task, above). These rationales varied in depth. A striking example of a compelling rationale was the following:

This project has a good chance of yielding a high impact publication by broadly advancing the state-of-the-art in [component] design... There are good opportunities here to combine biophysical models and [machine learning] to improve [component] function predictions, which can broadly utilise [other teams'] capabilities. Once the data tables are created for this effort (could be a Q2 milestone), the [computer scientists] will be able to dive deep with [machine learning] to identify patterns of interest.

Two of the authors independently coded a random sample of 20 of the 169 rationales that the 36 program members submitted with their CP rankings.² These authors then negotiated four codes that expressed most of the dimensions each had independently observed in the data, and jointly applied those codes to the same 20 rationales. One researcher coded the remaining rationales. Each CP rationale was coded as belonging to from zero to four dimensions.³

Results

We identified four dimensions of value⁴ in CPs:

- Capability = The CP creates a tool or technique that enables new methods of analysing, designing, or producing parts, components or systems (50 instances in CP ranking rationales).
- Speed = The CP enables the program to demonstrate success quickly (22 instances).
- Impact = Discovery, design, technology, techniques and applications arising from this CP will be significant (e.g. publishable, award-worthy) (56 instances).
- Synergy = The CP builds on extant products of the program, resources of the performing organisations, commercial products, or open sources tools and data (52 instances).

These four dimensions were independent of one another. An analysis (see Table 2) over the 112 rationales in which at least one dimension was present found only weak correlations ($r < |0.33|$) between the dimensions. Thus, coders who use these dimensions in the future can do so with some confidence that they will not conflate the dimensions one for another.

Discussion

A procedure for generating and down-selecting CPs was developed and applied. It was sufficient to enable program scientists to specify CPs, to eliminate redundant or low-value CPs and to down-select among CPs using rank-based voting. A survey of participant capability to engage in each CP was also conducted to confirm that no researchers or CPs were 'orphaned'. The method drove investment decisions by the sponsor and engagement of the program team. The method did not fully capture scientific or programmatic concerns that led the sponsor to amend the results with one additional CP, as noted above. Thus, we advise users of this method to ensure systematic review and revision of CP selections by program leadership.

An analysis was conducted on the rationales that the scientists offered for their CP rankings. That analysis revealed four independent dimensions of CP value: capability, speed, impact and synergy. These dimensions have useful applications. First, a program sponsor might build a portfolio of CPs whose distribution over dimensions matches their investment strategy. For example, a portfolio whose CPs maximise speed and synergy would execute quickly and potentially at low cost. To build a 'biggest bang for the buck' portfolio, the sponsor would select CPs that maximise impact and synergy. An investment in scientific techniques and technology might involve CPs that score highly on capability development. In sum, different sponsors will likely want to use mixtures of CPs based on the overall program goals and funding.

Second, a program sponsor might analyse the relationships of dimensions, researchers and CPs to understand and manage the complex socio-technical system spawned by a multi-million dollar research investment. A correlational, cluster or principal components analysis of scientists by their use of CP dimensions in rationales should reveal their preferences for building new capability, publishing at speed, attaining high impact or building research networks. An analysis of researchers by CPs will shed light on their perceptions of their technical capabilities. Analysis of CPs by CP dimensions should reveal the perceived potential of each CP to develop new methods or technologies, demonstrate progress quickly, discover new scientific or technical phenomena, or build the research community.

In future research, we plan to pursue the applications and analyses, above. This will require a more fine-grained dataset, either historical or newly created. We plan to have each scientist rate each CP (e.g. from 1 to 7) on each of the four dimensions, as well as rank their preferred CPs. The correlation between value ratings and CP rankings will provide some validation of the dimensions, indicate which dimension(s) drive rankings of CPs,⁵ and enable the correlational and dimensionality reductions described above.

The financial cost of CPs is, of course, a factor in calculating return on investment. This method did not directly capture such costs. It does give some indirect guidance: the survey of potential researcher engagement with CPs can help a program manager to manage redundant investments, and the synergy dimension may help the sponsor and scientists distribute tasks to those who can execute them most cost effectively. In the CP assessment exercise, above, neither the sponsor nor the scientists could assess the cost of the CPs, given the nascent state of the problem descriptions. Precise cost estimates require investment in deep design of research studies and, potentially, pilot

testing. The program at hand is developing automated cost estimation technologies (e.g. for laboratory costs) that will partially address this aspect of RoI.

Finally, we note that the CPs referenced here have, as of this writing, been selected but not executed. Once the CPs have been executed, a post-assessment will provide data to determine whether the program participants selected CPs that met their claims for timely execution, synergy, creation of new capabilities and impact.

Notes

1. A range of interesting research bears on the design of organizations for solving CPs, such as teams of researchers. Narayanamurti and Odumosu (2016) studied successful research institutions to define aspects of culture, policy, management, and investment. The National Research Council (2015) recommended that collaborative research programs apply lessons from the rich literature concerning teamwork. CPs, particularly interdisciplinary ones, should be designed with some expectation of the need for and cost of these practices.
2. Some scientists submitted fewer than the requested five ranked CPs, and some scientists omitted rationales for some rankings. Thus, the number of rationales is lower than expected from 36 individuals ranking five CPs.
3. A rationale was coded as having zero dimensions when the text could not reasonably be considered a rationale, or was blank.
4. Note that the name of a dimension indicates only that the rationale addressed the issue (e.g. impact) but not whether the rationale was positive or negative (e.g. high or low impact) on that dimension.
5. The dichotomous dimension data developed here did not support this type of analysis.

Acknowledgements

This material is based upon work supported by the Defense Advanced Research Projects Agency (DARPA) and the Air Force Research Laboratory under Contract No. FA8750-17-C-0294 (and related contracts by SD2 Publication Consortium Members). Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the Defense Advanced Research Projects Agency (DARPA), the Department of Defense, or the United States Government. This article was approved for Public Release, Distribution Unlimited (Distribution Statement 'A') by DARPA on 18 May 2018. Opinions, interpretations, conclusions and recommendations are those of the authors and are not necessarily endorsed by the US government. This document does not contain technology or technical data controlled under either US International Traffic in Arms Regulation or US Export Administration Regulations.

We thank Jesse B. Freeman for developing the mathematical formalisation of the rank voting algorithm. We thank Jacob Beal and this journal's anonymous reviewers for insight and recommendations that improved this article.

The members of the SD2 Research Consortium, as of the date of manuscript submission, are: William J. Allen, Texas Advanced Computing Center; Carrie Arnold, Texas Advanced Computing Center; Andrew Avila, Pacific Northwest National Laboratory; Bryan A. Bartley, Raytheon BBN Technologies; Jacob Beal, Raytheon BBN Technologies; Diveena Becker, Ginkgo Bioworks; Conrad Bell IV, ECS Federal; Vanessa M. Biggers, Transcriptic; Amanda Brower, Massachusetts Institute of Technology; Tracy Brown, Texas Advanced Computing Center; Daniel Bryce, SIFT, LLC.; Diogo M. Camacho, Wyss Institute, Harvard University; Richard Cardone, Texas Advanced Computing Center; James P. Carson, Texas Advanced Computing Center; Daniel Cetnar, Pennsylvania State University; Emory M. Chan, Lawrence

Berkeley National Laboratory; Yang Choo, Transcriptic; Katie J. Clowers, Ginkgo Bioworks; James J. Collins, Wyss Institute, Harvard University/Massachusetts Institute of Technology; Alexander Cristofaro, Massachusetts Institute of Technology; Anastasia Deckard, Geometric Data Analytics; Douglas Densmore, Boston University; S. Charlie Dey, Texas Advanced Computing Center; Yuval Dorfan, Massachusetts Institute of Technology; Robert G. Egbert, Pacific Northwest National Laboratory; Hamed Eramian, Netrias, LLC; Mohammed Eslami, Netrias, LLC; Amin Espah Borujeni, Massachusetts Institute of Technology; Erik S. Ferlanti, Texas Advanced Computing Center; John M. Fonger, Texas Advanced Computing Center; Jared Freeman, Aptima; Sorelle A. Friedler, Haverford College; Dany Fu, Boston University; Niall Gaffney, Texas Advanced Computing Center; Tomas Gedeon, Montana State University; Christopher Geib, SIFT, LLC; John Gentle, Texas Advanced Computing Center; Sarah Goldberg, University of Washington; Robert P. Goldman, SIFT, LLC; D. Benjamin Gordon, Massachusetts Institute of Technology; Steven B. Haase, Duke University; Sean M. Halper, Pennsylvania State University; Nathan O. Hodas, Pacific Northwest National Laboratory; Ayaan Hossain, Pennsylvania State University; Anagha Jamthe, Texas Advanced Computing Center; Manu Mary John, Texas Advanced Computing Center; Christopher Jordan, Texas Advanced Computing Center; Michael Keller, Texas Advanced Computing Center; Benjamin J. Keller, University of Washington; Eric Klavins, University of Washington; Ugur Kuter, SIFT, LLC; Peter L. Lee, Transcriptic; Sarah Leinicke, Boston University; Drew A. Leins, Aptima; Sahil Loomba, Wyss Institute, Harvard University; Julia Looney, Texas Advanced Computing Center; Eriberto Lopez, University of Washington; Chun-Yaung Lu, Texas Advanced Computing Center; Narendra Maheshri, Ginkgo Bioworks; Vikash K. Mansinghka, Massachusetts Institute of Technology; Paul Maschhoff, Ginkgo Bioworks; Joseph Meiring, Texas Advanced Computing Center; Benjamin N. Miles, Transcriptic; Vincent Mirian, Massachusetts Institute of Technology; Thomas Mitchell, Raytheon BBN Technologies; Francis C. Motta, Duke University; Tramy Nguyen, Raytheon BBN Technologies; Alexander J. Norquist, Haverford College; Rhys A. Ormond, Transcriptic; Michael Packard, Texas Advanced Computing Center; Ian M. Pendleton, Haverford College; Alex Plotnick, SIFT, LLC; Andrew Punnoose, Ginkgo Bioworks; Alexander C. Reis, Pennsylvania State University; Nicholas Roehner, Raytheon BBN Technologies; Howard M. Salis, Pennsylvania State University; Ulrich Schaechtle, Massachusetts Institute of Technology; Joshua Schrier, Haverford College; Jedediah M. Singer, Two Six Labs, LLC; Jawon Song, Texas Advanced Computing Center; Devin Strickland, University of Washington; Joseph Stubbs, Texas Advanced Computing Center; Xun Tang, Pennsylvania State University; Steve Terry, Texas Advanced Computing Center; Lisa Tiberio, Raytheon BBN Technologies; Virginia Trueheart, Texas Advanced Computing Center; Joshua Urrutia, Texas Advanced Computing Center; Matthew W. Vaughn, Texas Advanced Computing Center; Christopher A. Voigt, Massachusetts Institute of Technology; Mark Weston, Netrias, LLC; Weijia Xu, Texas Advanced Computing Center; Yaoyu Yang, Ginkgo Bioworks; Enoch Yeung, Pacific Northwest National Laboratory; Jing Zhang, Massachusetts Institute of Technology; Gregory J. Zynda, Texas Advanced Computing Center; Gary Cattabriga, Haverford College.

Disclosure statement

No potential conflict of interest was reported by the authors.

Notes on contributors

Jared Freeman, Ph.D., is Chief Scientist of Aptima and a Principal Investigator on the SD2 program in which this work was developed. His research addresses problem solving, human learning and organisational assessment and design.

Drew Leins, Ph.D., is an experimental psychologist specialising in legal and cognitive psychology. He focuses on decision making in high-stakes environments and has worked with federal law enforcement and the military to conduct research on psychological issues critical to national security.

Conrad Bell IV, Now in his second career, Mr. Bell is a Ph.D. candidate in Computer Science and a Systems Engineering and Technical Advisor. In his first career of 25 years, he leveraged the enterprise-wide computer systems he designed and implemented into large-scale process re-engineering efforts that resulted in improved organisational efficiency in Fortune 50 companies.

Funding

This work was sponsored by the Department of Defense under contract FA8750-17-C-0294 (and related contracts by members of the consortium members).

References

- Al-Ghassani, A. M., J. M. Kamara, C. J. Anumba, and P. M. Carrillo. 2006. "Prototype System for Knowledge Problem Definition." *Journal of Construction Engineering and Management*, 132 (5): 516–524.
- Cohen, M. S., J. T. Freeman, and B. T. Thompson. 1998. "Critical Thinking Skills in Tactical Decision Making: A Model and a Training Method." In *Decision-Making Under Stress: Implications for Training and Simulation*, edited by J. Canon-Bowers and E. Salas. Washington, DC: American Psychological Association Publications.
- Intelligence Advanced Research Projects Activity. 2017. *HECTOR*. Accessed 28 February 2018. <https://www.iarpa.gov/index.php/working-with-iarpa/requests-for-information/hector?id=984>.
- Klein, G. A. 1993. "A Recognition-Primed Decision (RPD) Model of Rapid Decision Making." In *Decision Making in Action: Models and Methods*, edited by G. A. Klein, J. Orasanu, R. Calderwood, and C. E. Zsombok, 138–147. Westport, CT: Ablex Publishing.
- Klein, G. 2013. *Seeing What Others Don't*. Philadelphia: Public Affairs.
- McComb, C., J. Cagan, and K. Kotovsky. 2017. "Optimizing Design Teams Based on Problem Properties: Computational Team Simulations and an Applied Empirical Test." *Journal of Mechanical Design* 139 (4): 041101-01–041101-12.
- Narayanamurti, V., and T. Odumosu. 2016. *Cycles of Invention and Discovery*. Cambridge, MA: Harvard University Press.
- National Aeronautics and Space Administration. 2004. *NASA Facts: Benefits from Apollo: Giant Leaps in Technology*. Accessed 5 February 2018. https://www.nasa.gov/sites/default/files/80660main_ApolloFS.pdf
- National Research Council. 2015. *Enhancing the Effectiveness of Team Science*. Washington, DC: The National Academies Press. <https://doi.org/10.17226/19007>
- Newell, A. and H. A. Simon. 1972. *Human Problem Solving*. Upper Saddle River, NJ: Prentice Hall.
- Pólya, G. 1957. *How to Solve It*. Garden City, NY: Doubleday.
- Stokes, Donald E. (1997). *Pasteur's Quadrant: Basic Science and Technological Innovation*. Washington, DC: Brookings Institution Press.
- Wallas, G. 1926. *The Art of Thought*. New York, NY: Harcourt, Brace & Company.

Appendix A

The ranked-choice voting formula developed in this work computes an interval score for each of many candidate CPs from ordinal rankings by multiple rankers who may rank only a subset of candidates, may rank a unique subset of candidates, and may award some candidates the same rank (i.e., a tie).

The parameters are:

- n denotes the number of rankers,
- k denotes the number of candidates to be ranked,
- m denotes the maximum rank assigned over all candidates and rankers.

We will let C_j denote candidate j for $j = 1, \dots, k$. The ordering of rankers and candidates is fixed and identified with the sets $\{1, \dots, n\}$ and $\{C_1, \dots, C_k\}$.

For each of the n rankers, we define a ranking function:

$$\begin{aligned} r_i : \{\text{Candidates}\} &\rightarrow \{\text{Ranks}\} \cup \{m+1\} \\ \{C_1, \dots, C_k\} &\rightarrow \{1, \dots, m, m+1\} \end{aligned}$$

Above, $m+1$ corresponds to a candidate that a ranker does not rank. This score represents a worst case, in which the ranker considered all candidates not ranked as equal in priority and minimally less desirable than the lowest ranked of the ranked candidates. We define a pre-weight as follows:

$$\tilde{w}(C_j) = \sum_{i=1}^n r_i(C_j)$$

We define the weight function $w(C_j)$ using the following normalisation, in which C_i is the candidate with the lowest pre-weight:

$$w(C_j) = \tilde{w}(C_j) - \min_{i \in \{1, \dots, k\}} \tilde{w}(C_i)$$

Appendix B: Example Survey

TA5 distributed this survey to all TA1-3 performers in January 2018, to elicit details on where (i.e., in which challenge problems) and how performers could apply their tools and capabilities.

TA1 & TA2

- TA1
 - For each question, below, (1) can your TA1 team address the question? (2) What technique would you apply? (3) What concerns do you have?
 1. Are there systemic problems that affect multiple circuit designs?
 2. How does the circuit affect the host?
 3. Which circuit designs exhibit environmentally resilient operating envelopes?
 4. What features predict protein stability?
- TA2
 - What metrics will you apply to assess & improve your design algorithms?
 - When will you have a baseline measurement on those metrics?
 - What challenge problems will you use to establish that baseline?

TA3 survey

- TA3:
 - For each Challenge Problem / Experiment you will execute:
 - How much will this CP / Experiment cost? (Tell DARPA only, please)
 - What protocols need to be harmonized before we start
 - Across Experiments: Which can you support at maximum capacity? To clarify, if you are engaged in multiple experiments, which combination(s) of experiments do you have the capacity to support? (E.g., We can do experiments A, B, C; our lab has capacity for A or B+C.)

Appendix C: Example Survey Responses

TA5 aggregated and summarized performer responses to a survey that elicited details on capabilities and requirements per proposed challenge problem (CP), to help DARPA visualize and plan CP participation and group composition.

Summary of TA1 survey responses

Data needed to characterize/model

Question	Parts	Circuit	Parameter Space	Cell State	Cannot Address
1	Pennsylvania State University (PSU), MIT, Pacific Northwest National Laboratory (PNNL)	PSU, Two Six	Two Six, MIT, PNNL, Duke	Duke	Harvard
2	PSU,	PSU, MIT	PSU, MIT	PSU, Two Six, MIT, PNNL, Duke	
3	Duke	PSU, MIT, Duke	PSU; Two Six, MIT, PNNL, Duke	PNNL	Harvard
4					PSU, PNNL

1. Are there systematic problems that affect multiple circuit designs?
2. How does the circuit affect the host? (Sub-bullet #2 is the concern for Q2)
3. Which circuit designs exhibit environmentally resilient operating envelopes?
4. What features predict protein stability? (Method variation included with concern)
 - a) PSU Tool: identify cause-effect relationships to predict genetic circuit function;
 - 1) Need many genetic circuits with varying compositions and part sequences
 - 2) Should be reframed as “circuit-host-environment” interactions
 - 3) Our model is well-targeted to answer Q3
 - 4) Not addressed
 - b) 2-6 Tool: predict circuit output or cell state based on target measurements
 - 1) Need representative training data: breadth and depth of data capture
 - 2) need representative training data
 - 3) need representative training data
 - 4) Same method; inconsistencies with how Rosetta’s parameters relate to reality
 - c) MIT/Vikash: Bayes Database (BayesDB): predict relationships between design features and circuit output
 - 1) Need many parts, many circuits and analysis-ready dataframes

- 2) Need many parts, many circuits that include host effects, experimental intent, and analysis-ready dataframes
- 3) Needs circuit functionality defined, operating envelope related metadata, and analysis ready dataframes
- 4) Combine BayesDB with Edward (Blei lab) to model input-output behavior of Rosetta; need Rosetta's parameters: protein identity, folding trajectory, stability prediction
- d) PNNL Input-Koopman: identify relationships between test conditions and output.
 - 1) None
 - 2) Need time-series data for RNA-seq and proteomics
 - 3) Need time-series data for plate reader, flow, RNA-seq, and proteomics
 - 4) Stochastic dynamic mode decomposition; algorithm still in development
- e) Duke (DSGRN) fixed-circuit parameter space analysis
 - 1) Need refined notion of robustness and parameter space sampling strategy to identify unplanned circuit space
 - 2) Gene regulatory interaction inference using sparse time points and multi-omics data to characterize global impact of circuit
 - 3) Need refined notion of robustness, expanded list of supported logic and relate parameter space to circuit output or cell state (not clear which)
 - 4) Provide topology-based measures of protein dissimilarity to enhance Rosetta; need flexible representations that can account for incorrect model (costly to correct)
- f) Harvard
 - 1) No response
 - 2) Take genomic sequence of novel host and identify novel ribosome binding sites, regulatory architectures, etc.
 - 3) No response
 - 4) Pro2Vec: embed amino acid sequence to explore determinant features; no concern

Summary of TA2 responses

1. What metrics will you use to assess & improve your design algorithm
2. When will you have a baseline measurement on those metrics
3. What challenge problems will you use to establish that baseline

TA2D

Voigt

- 1) a) gates per cell b) co-variation between genes and metabolites c) genomic response to stimuli, cellular capacity at varying growth rates d) time to find error
- 2) a) no short-term plan b) dependent on RNA-seq data, 2-4 weeks c) 2 weeks dependent on RNA-seq and Ribosome-seq d) once data is sent to TA1 for analysis
- 3) Cell state (b,c), robust circuits – new (b,c), Chemical, Biological, Radiological and Nuclear (CBRN) sensing – new (a), Novel Chassis (a-d), Yeast Gates (a,d), bio-cyber (d)

Baker

- 1) Correlation between computational prediction and experimental measurements; tested via experimental based model and computational model

- 2) 2-3 months when all 80K proteins have been synthesized and tested
- 3) Protein stability

TA2P

Densmore/Gordon

- 1) a) percent increase in circuit functionality, b) reduction in experiments needed, c) analysis pipeline automation, percent d) reduction in reagent use and time for DNA assembly and characterization
- 2) a) data available need to integrate, b) available for simulated datasets, needs experimental validation c) Integration with TACC – end of march d) after April meeting
- 3) Yeast Gates (a-d), Protein stability (b), round trip (d)

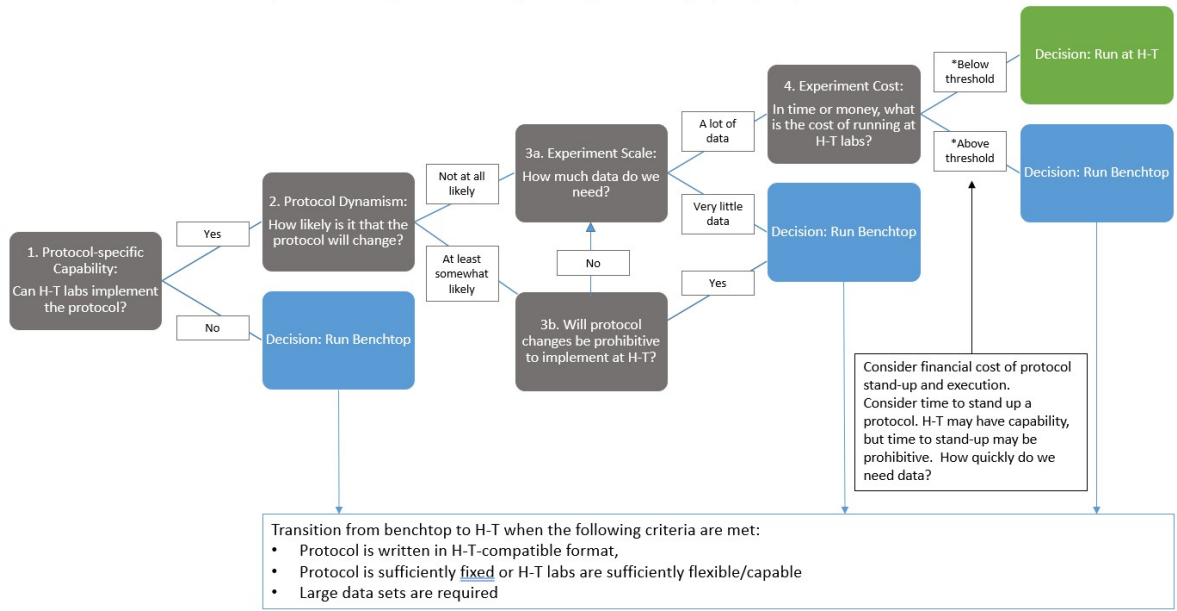
SIFT

- 1) Plan time: plan to experiment, plan size: number of samples, plan cost, information value: how useful was an experiment
- 2) Using analysis of Q0 data, end of February for time size and cost. Use uninformative and bio-literature formative priors – end of march
- 3) Rule30, Yeast Gates and Protein Stability

Appendix D: Decision Tree

Decision tree for evaluating whether to execute an experimental protocol at the benchtop or at a high-throughput laboratory.

Decision Tree for Experimenting at Benchtop vs. High-Throughput (H-T)



Appendix E: CTA Outline

Outline for cognitive task analysis for experiment decision-making based on optical density (OD) observation/measurement

- Purpose: Generally, "unpack" the thinking of experts, characterize it, operationalize it, codify it, develop training around it, to accelerate learning and develop expertise more efficiently. CTA seeks to mitigate the unnecessary consumption of time and resources during on-the-job training. In our case, we want to avoid the unnecessary consumption of resources to learn that an experiment should have been modified in-process. Here we want to "train" robots to behave like expert scientists, to know when to modify a protocol or simply to know more precisely when to take subsequent steps in a protocol.
- The decision environment:
 - Human scientist: Conduct an experiment in which steps of the protocol are contingent upon observations and judgments that do not occur at standard intervals or in standard sequence. Materials to be measured will grow at different rates and therefore must be assessed for candidacy for measurement, otherwise one risks measuring materials that are unsuitable for measurement. For example, early in the program we made RNA-seq measurements on wells that had no cells in them, costing time, money etc.
 - Automation: We want automated labs to conduct the same experiments that humans conduct; however, automated labs are constrained. They are only as flexible as their hardware (e.g., their sensors and moving parts, their machinery) and their software (e.g., the code that operates the machinery) and will allow. What have we learned about the limitations of automation in this environment as it is currently constructed?
 - Limits to flexibility: Once a protocol begins, automation will complete it unless there is some failure of the apparatus (e.g., mechanical failure or the code is unreadable). A stimulus or materials problem (e.g., no live cells to assay) will not necessarily cause an experiment to be aborted.
 - Limits to sensitivity: Machinery is not equipped to sense stimuli in ways required to assess
 - The feasibility of continuing experiment
 - Whether to enact a subsequent step and which step to enact
 - Whether multiple samples on a plate need different treatments (e.g., some need concentration while others need to be diluted)
- For our investigation, the focal task is to measure optical density (OD) to determine cell count/concentration and to trigger next steps in a protocol.
 - The steps of the task:
 - Measure OD at time $t=0$
 - Use pre-computed growth rate to determine dilution required for each strain to achieve target density at time $t=S$, where S is the start of the experiment.
 - Dilute each strain to target density
 - Measure OD again at $t=S$ and determine whether experiment goes forward given predetermined parameters (e.g., 80% of samples are within 10% of target OD)
- Some contextual knowledge is critical for knowing what to do at each step:
 - Number of cells diluted to start
 - Growth rate of cells
 - Medium in which cells are growing
 - Health of cells

- What does the scientist observe/sense in this protocol and what are the relationships between sensing/observing and decision making? (e.g., What do "healthy, happy" cells look like?)
 - Sensor used: Microscope
 - Cue (what you observe/sense):
 - Ratio of current cells to starting count [?] Healthy cells appear "phase bright" under phase contrast. High percentage of budded cells is associated with rapid/log-phase growth.
 - Threshold of live-dead
 - Sensor: flow cytometer
 - Cue: What you observe: health (what does healthy look like in a flow cytometer?) and number of cells
 - Assessment/Interpretation (what it means): Growth is [] too slow [] good enough.
 - Deviation from expected growth rate may be an indication that something went wrong.
 - Course(s) of Action (what you do in response): well-by-well decisions or plate-level decision?
 - Option 1: wait until cells grow out (and tag for metadata)
 - Option 2: Density can be adjusted if concentration is off: Dilute and wait if cells are too dense and look unhealthy (and tag for metadata)
 - Option n...Abort
 - If too dilute AND they look unhealthy
 - If really unhealthy for any reason
 - Candidacy for automation:
 - How can machinery observe or sense the health of cells:
 - Flow cytometry: Our Live/Dead measurements may do this step. We are trying to use ML on all channels of flow data to identify cells that are live, dead, and other statuses in between (sick, on the way to dead).
 - High-definition imagery from multiple perspectives?
 - Imaging cytometer is a proxy for a microscope and may be amenable to automation?
 - How do we code for the assessment/interpretation?
 - How do we code for executing each course of action?
 - Code for data/metadata tagging?
- To be elicited from TA3: Potential "training" strategies for enhancing automation to be more flexible:
 - Recommended hardware additions/extensions:
 - Recommended software for enabling sensors to observe critical stimuli:
 - Recommended parameters for coding experimental protocols:

Appendix F: STS Video Transcript

Transcript of the video SD2: Developing a Socio-Technical System To Advance Science in Complex Domains

Steve Haase:

My name is Steve Haase. I'm a professor of biology secondary in medicine here at Duke University. Science is incentivized from funding and also from academic institutions to really do science in this individualist or investigator-initiated model. Each one of those investigators or their team is creating a hypothesis using that hypothesis to generate a protocol, doing that experimentation, collecting the data, and then analyzing the data and putting out conclusions. When you get to aggregating our knowledge at the level of conclusions, invariably, we've all come to different conclusions. The more we learn, the more we generate expertise that's siloed. And that means in order to deploy on these really high dimensional problems, we really need to bring those experts together. And give them a platform and guidance and how to interact and how to go forward and solve those problems.

Dr. Jennifer Roberts:

The SD2 program was designed in order to pursue these scientific endeavors, much faster, much more accurately than we have in the past. By creating this interplay of technology and human capital.

Sterling Wiggins:

This program is about understanding how to build an infrastructure that allows us to be agile and to address issues, problems, challenges that face the nation that will evolve over time.

Steve Haase:

A socio-technical system or STS, is a collection of people and technology. The system is all about how they interact.

Dr. Susan Buckhout-White:

So the idea was to bring experts in machine learning and AI together with those who are experts in synthetic biology and material science.

Dr. Jennifer Roberts:

The underlying assumption was, at the core we need to be able to share data. One of the ways that we address that challenge inside of SD2 was to very intentionally separate out the people who were designing an experiment from the ones who were running it, from the ones that were collecting and aggregating the data. And the people who were ultimately analyzing it to understand what scientific insights could be drawn from that data. It exposed the places where all of this implicit knowledge needed to be explicitly represented.

Dr. Susan Buckhout-White:

SD2 required groups that may not normally talk to each other, to work together.

Speaker 5:

We had this hackathon on the current data sets. Data means nothing without metadata, you need to know what it's about. And it was about how do you capture and communicate experimental intent.

Dr. Peter Lee:

It raised a bunch of issues. Lab A is storing their data formats in row major sort of orientation, so like rows were first. Lab B was storing it in column major, right? When we were trying process that data, we were making assumptions about how we were storing it, and we were getting very weird plots.

Steve Haase:

The program while it had broad goals, didn't have specific scientific goals. Those goals were decided on by the group.

Dr. Susan Buckhout-White:

We looked at four scientific areas within the program, protein design, engineering in yeast, engineering in bacteria, and perovskites science.

Dr. Joshua Elliot:

What we're trying to do, is figure out how do we systematize the research that's coming out of those use cases in a way that makes it reusable, reproducible, shareable, et cetera.

Dr. Susan Buckhout-White:

When we break apart sort of the scientific process with each of these individual functions, we separated the design teams, the discovery teams, the experimental laboratories that created the data and the data infrastructure within each of those roles, there were sort of two typical loops that happened. With the experimental laboratories and the data infrastructure at the center, the experimental laboratories would create the data and that would be uploaded to the data infrastructure. That data then went into the discovery teams or the analysis teams. That data would come in and go back into the experimental infrastructure, which would then lead to the design teams to be able to in the next layer of experiments, informed by the data to create this iterative cycle that continued to repeat.

Steve Haase:

A single laboratory could not have performed end to end the number of experiments that we did.

Dr. Susan Buckhout-White:

We increased the speed of which we could complete an experimental cycle by tenfold. While we increased the speed ten-fold we also increased the complexity of the experiments we were able to run.

Dr. Peter Lee:

Taking the music analogy of what we're trying to do. We basically currently have people playing their own music in their different labs, just across the country. It would sound discordant, right? You would have just everyone playing to their own tunes. How do we unify distributed science from across the nation? Can we use all of the instruments and all of the pieces that we're playing out there to contribute to that larger symphony and get towards a much better understanding of the world that we're living in, the biology that's underlying all of us, and how we can use that to make our lives better.

Steve Haase:

The STS really stands as a model for how we can change the way we do science to focus on the tough problems that we have ahead.

Appendix G: Program Coordination Guidebook

A Concise Guide to Enabling Collaboration on DARPA Programs

Jared Freeman, Sterling Wiggins, Drew Leins, John Colonna-Romano, Matt Wood

Aptima, Inc.

Corresponding author: Jared Freeman, freeman@aptima.com

Introduction

Multi-disciplinary teams working collaboratively can often make impactful discoveries more quickly than can scientists and engineers working alone or in competition. The value of such collaboration is evidenced by the recent growth in the number of authors per scientific paper (Greene, 2007). DARPA programs are, accordingly, evolving from activities in which the brightest minds compete, to activities in which they collaborate.

Executing a collaborative program poses challenges to some performers and to the program managers (PMs) who lead them. Performers are typically incentivized by work models that place them in competition for resources that support their research and development efforts, for space in forums that could publish their work, and for access to markets in which to sell their work products. Their collaborations are often small in scale, relative to the size of DARPA programs. Thus, some performers are unfamiliar with the structure, processes, or tools that support collaboration across teams and institutions. Within DARPA, program managers and their support teams may find that designing for and managing collaboration pose complex problems. This is evident to project leaders when they must:

- Spot an impending conflict between performers, diagnose its cause, and remediate it
- Host a virtual meeting that engages performers in doing collaborative work, not just reporting it
- Identify an opportunity for collaboration across teams, and foster it
- Monitor and manage collaboration in several programs simultaneously

We offer the following guidance to support DARPA personnel who aspire to conduct collaborative S&T within their programs. This advice is based on our experience facilitating two large, multidisciplinary DARPA programs: SD2 and ASIST. The guidance focuses on techniques and tools for organizing people, work, and events.

Organize People

Goal: Optimize performance by placing performers in contexts that allow them to define and coordinate work that accomplishes the program mission, identify and resolve conflicts, and discover complementary capabilities and compatible philosophies.

Structure: You may have already organized performers into technical areas (TAs). However, you will likely benefit from further organizing performers into working groups that comprise performers from across multiple TAs. Working groups (WGs) will serve the mission of the program (e.g., an administrative, scientific, engineering, or other well-defined purpose) and coordinate performers. Several types of WG

may benefit the program. For example, enduring WGs focus on long-term S&T challenges (e.g., developing measures). Subgroups within these may address special issues (e.g., statistical techniques for evaluating AI influence on human systems). Fast Track WGs organize to rapidly address emergent problems.

Process: The need for a WG will become evident as PMs and performers identify discrete S&T or programmatic challenges to be addressed. WGs can form organically as performers learn more about one another's interests, capabilities, work styles, and personalities, or they can be dictated by the PM. We caution PMs to consider the rationale for forcing group membership. It can be a catalyst for identifying compatibilities and incompatibilities early in a program, and it can disrupt dynamics (positive or negative) later in a program. Regardless of how they are formed, WGs should be organized to enable members to do the following:

- Formulate issues and help resolve them.
- Implement workflows that serve all performers (e.g., proposal and development of performer-authored extensions to a common testbed)
- Engage performers continuously in collaboration with their peers.

PMs should ensure that WGs have the following features:

- Well-defined mission, duration, and leadership
- Representation from all or essential TAs and teams
- A venue and medium for communicating and documenting issues, ideas, resolutions, etc.
- A published meeting schedule for conducting routine business
- A periodic assessment of the value of each WG and of leadership succession

We recommend that PMs convene all WG leaders periodically to identify emergent issues and collaboration opportunities, and to share program news that can be disseminated among all WGs.

Coaching. Performers in leadership roles are not always natural leaders. They may have difficulty identifying and organizing work, enlisting and motivating WG members, and spotting and resolving conflict. PMs and program coordinators should participate in WGs to assess these factors, and provide 1:1 coaching to leaders to help them develop and apply these skills or find lieutenants who can do so. There is no one-size-fits-all approach to coaching and it may take time to tailor an approach for a specific program dynamic. Generally, coaching will succeed if it helps to ensure all voices are heard, and that none are given weight due solely to volume. We recommend the following processes for coaching:

- Request, model, and reward behaviors that promote communication and collaboration, such as:
 - Expressing curiosity about the assumptions, mechanics, and effects of novel or challenging viewpoints
 - Listening closely and asking clarifying questions before disagreeing
 - Asking questions that open up dialogue instead of making statements that close conversation prematurely
 - Identifying friction and using it as an opportunity to develop deeper understanding, even when agreement cannot be reached or is unnecessary to move forward

- Help WG leads develop a charter, goals, and etiquette for interaction that help to set and manage the purpose, tone, and flow of working group meetings.
- Assess WG coordination and dynamics for emerging opportunities to counsel and coach. For example:
 - If WG members are in public conflict, counsel each individual to join you in a private discussion to mediate the dispute.
 - If a WG is unable to prioritize its goals or actions, coach the WG lead to bring them to the PM for prioritization. This transfers to a higher authority the decision and the responsibility to state the values by which similar decisions should be made in the future.

Organize Work

Goal: Optimize performance by establishing program processes that enable the PM and performers to identify and raise issues, negotiate differences and decisions, announce insights, and manage task execution.

Structure: You have outlined the critical problems that performers should address. It will benefit you (and performers) to provide detailed guidance on how to address these problems collaboratively. This does not mean that you will dictate how performers conduct their work; rather, it means that you will facilitate their work by crisply defining goals, etiquette for collaborating on the program, and a schedule by which goals should be achieved.

Process: We recommend the following processes for organizing the work of your program.

- Define goals. Some programs will benefit from immutable goals, whereas others will benefit from adaptive goal definition. An example of an adaptive approach is one that accounts for shifts in performer dynamics or capabilities and redefines goals as the program evolves. Regardless of the approach, PMs will benefit from being clear about the long-term goals of the program and any short-term goals of segmented work cycles. A well-defined goal can foster shared mental models, reduce confusion over expectations, and lead seamlessly to defining the bounds of work that will allow performers to achieve the goal (see Develop challenge problems, next).
- Develop challenge problems. As you define or redefine goals, you will surface opportunities and objectives for achieving those goals. One can codify these opportunities and objectives in the form of one or more challenge problems (CPs). The one published process for collaborative CP definition (Freeman, Liens, & Bell, 2019) reports the steps that were productively applied early in DARPA SD2:
 - Performers generate candidate challenge problems and advertise them to all performers.
 - Performers rank-order the subset of CPs they prefer.
 - A ranked-choice voting formula is applied to prioritize the full set of CPs from the ranked subsets.
 - Highly ranked CPs are validated and adjusted by the PM.
 - Working Groups are formed to execute against those CPs.

Analyses of the CPs generated and prioritized in this way indicated that performers prioritized and selected for their potential to (a) leverage performers' extant capabilities, (b) optimize the speed

toward solutions, (c) maximize scientific and technical impact, and (d) realize synergy across performers.

- State how key decisions will be made: Explicit decision processes within WGs and TAs enable members to contribute to decisions in the right ways at the right times, and to know that they have been heard (even if the decision is, explicitly, made by another party). Many modes of decision-making exist, for example, centralized versus distributed, and static versus dynamic (i.e., rolling or partial decisions). What matters here is that performers understand how, when, and by whom decisions will be made on issues they can raise and inform.
- Schedule activities that support collaborative work: It is useful to schedule at least the following activities (see also, Organize Events, below):
 - Weekly (or bi-weekly) WG scrums
 - Monthly program scrums
 - One-on-One meetings between the DARPA PM and Principal Investigators or teams
 - Program-wide meetings, such as integration events, hackathons, demonstrations, evaluations, and experiments or evaluations
 - Revisions of program leadership (e.g., turnover / election of WG leads)

Tools: Several tools exist to help you organize the work of your program. For example, some tools support collaboration among performers, some help to ensure adherence to contractual obligations, regulations, and norms; and some facilitate the routine tasks of program management.

“Contracts” for Collaboration. Contracts are agreements that make collaboration safe by protecting proprietary information, ensuring credit for work, respecting regulations, and publicizing intent.

- Associate Contractors’ Agreement (ACA): The ACA requires data sharing and protects proprietary information, which makes collaboration over information safer.
- Publication Consortium Agreement (PCA): These agreements give performers a venue to negotiate for authorship credit, which makes collaboration safer.
- Human Subjects Research (HSR) protocols: These protocols, reviewed by an Institutional Review Board (IRB) and Human Research Protection Office (HRPO), ensure that human subjects are protected during experiments, and their data are protected (i.e., accessible only to those who are trained and certified to handle those data).
- Pre-registration of scientific hypotheses and planned technical capabilities: These Open Science research descriptions declare intent, methods, data, and analyses. Cross-team reviews of these documents educate performers about their collaborators’ intent and collaboration opportunities.

Tools for collaboration. Platforms for communication and collaboration enable performers to educate themselves and coordinate their work.

- Workflows: Jira and similar kanban tools
- Schedules: Google calendar
- Chat system: Slack, Teams, and other chat platforms
- Document collaboration: Confluence, Google Drive, SharePoint, and similar services

- Data repositories: Amazon Web Services (AWS), Google Cloud Storage (GCS), and similar services
- Code repositories: GitLab, SourceForge, and other platforms
- Video meetings: Zoom, Teams, Hangout, and other platforms

Tools for managing collaboration. Applications for monitoring collaboration and the data that pass between performers enable program leadership to assess and detect changes in the structure, interactions, and productivity of the program.

- Production monitors: These tools monitor production of data and documents on collaboration platforms, to help assess progress and identify bottlenecks in a workflow.
- Interaction monitors: These tools monitor and measure interactions between performers on messaging and editing platforms to identify high performing individuals, boundary spanners, initiators, and gaps in a socio-technical system.

Such collaboration tools enable PMs to ask and answer a variety of questions, such as:

- Who are the most (dis)engaged performers on my program?
- Which performers are best positioned to discuss the state of a program initiative, or to address an emergent issue?
- What performers are engaged in collaboration across teams?
- What program documents are attracting the most/least attention and interaction?
- Is the pipeline of data / technology / publication productive or jammed?

Organize Events

Goal: Optimize performance by providing forums for performers to convene, perform the work that most benefits from co-located collaboration (e.g., plan tasks, analyze data), report progress to the entire program, and receive critical feedback/guidance from PMs.

Structure: The events that support program goals will vary by program. The events we found most useful are these:

- PI Meetings enable performers to present progress and accomplishments, identify and address emergent challenges, and coordinate over longer term activities such as Challenge Problems.
- Integration meetings convene performers to test the interfaces between technologies, and the human workflow that applies them. These meetings may precede an evaluation event or transition demonstration.
- Hackathons present problems and data with which performers can exercise ingenuity and demonstrate their capabilities. Several types of hackathons have value, including:
 - Challenge type: PMs or performers introduce a novel, difficult problem. Participants collaborate to create solutions. Some programs host such hackathons as often as weekly. For example, MediFor reportedly hosted a weekly image analysis challenge.
 - Transition type: A transition partner brings a new problem and data corpus.

- Evaluation type: Performers convene to review and hack data representative of a planned evaluation problem, then report findings and recommendations for review by PM and evaluation TA.

Process: Regardless of the type of meeting, thoughtful planning and execution are critical to success. The following processes are particularly important, in our opinion.

Planning. We recommend allowing at least 10 weeks to identify meeting goals, plan activities that address them, and settle the logistics (from DARPA facilities and infrastructure to performer flight and hotel reservations). The following are some specific recommendations:

- Define meeting goals: Elicit meeting goals from the PM and PIs, top-down and bottom-up. Select those goals that support the mission of the program, as well as the aspirations and frustrations of performers.
- Design the products of meeting activities: Generate templates or clear instructions with which meeting groups can report the issues and recommendations they address in WG and breakout groups.
- Design meeting activities: Design sessions whose members, missions, instructions, and duration enable performers to generate the product you need.
- Define the participants: Identify the speakers, leaders, and key participants for each session in advance. Aim for a distribution of participants across teams. Contact them early to engage them in the planned activities.
- Make open time: Reserve considerable time on the agenda for ad hoc meetings of WGs, teams, and other interest groups to generate new research concepts while sitting side-by-side, to work out research processes on whiteboards, and to recruit collaborators into plans that benefit the program.
- Calculate logistics: Ensure that start and end times, room capacity and layout, audio/visual setup, and food and beverage options are conducive to a productive meeting.

Execution. Events such as hackathons and integration meetings are dynamic affairs. Logistical and interpersonal complications can arise. To help avoid or mitigate these issues, we suggest processes that largely fall within the scope of *facilitation*, that is, creating the conditions that enable performers to work together to develop innovative ideas and solutions:

- Begin the meeting with a clear statement of the meeting goals and a review of the agenda and meeting etiquette.
- As the event progresses, sense and respond to cues that indicate the group would benefit from deviating from the schedule. For example, a group that was animated and engaged but quickly becomes lethargic and disengaged may benefit from an unplanned break or change of topic.
- Try to stay on schedule. Inevitably, some speakers will run over their allotted time or a data analysis session will uncover issues in the data that must be resolved before follow-on sessions can begin. Consequently, you must determine whether the day's (or subsequent days') agenda should be revised. Consider whether and how revisions--for example, cutting sessions versus extending the day's end time--will impact performers.

Specifically for facilitating interpersonal interactions:

- Ensure that participants have a voice and are heard, and that all are respectful.
- Identify tensions within the group and determine how and when to address them.
- If a discussion becomes confused:
 - Stop the discussion, summarize the areas of agreement and disagreement and ask for confirmation before resuming the discussion.
 - Ask speakers to define important words and concepts, especially those that might have different meanings across disciplines.
- If a discussion becomes contentious:
 - Ask that speakers summarize their understanding of what was previously said
 - Remind speakers to ask clarifying questions before stating a contrary opinion
- If a discussion is dominated by one perspective:
 - Ask performers who have not contributed recently to the conversation to share their thoughts.
 - Validate the contributions of the dominant group members and encourage them to reduce their talking time, to allow others to contribute.
- As sessions approach their end time:
 - Identify a good stopping point for the discussion, summarize progress, and articulate the prioritized list of topics for any follow-on discussions
 - Clarify consensus decisions, actions, and actors

Closing the loop. It is difficult for event planners and facilitators to learn well from events. They have an author's bias; they rarely see all of an event; and they can never see the event through the eyes of the participants who the event serves. Participant feedback and feedforward helps address errors in the last event and plan the next one.

- Elicit feedback after an event, feedback focused on the technical and emotional:
 - What activities or sessions produced good insights, products, or partnerships? What went right?
 - Which activities or sessions were unproductive? What went wrong?
 - Did you have the time you and your colleagues needed to get work done? What work was that?
 - What was the high point of the meeting? The low point?
- Elicit feedforward -- objectives and advice -- for the next meeting. Ask WG and team leaders what they want to accomplish or resolve; and what activities will help or hinder their progress. Ironically, we have received at least as much feedback about prior meetings by asking performers to look forward as we do asking them to look back.

Conclusion

The guidance we present here is intended to facilitate collaboration in DARPA programs, and in so doing to accelerate scientific and technical progress. This guidance was developed from our experience coordinating

DARPA programs, from our empirical studies of organizations, and from the research literature of organizational science.

To implement the recommendations above requires, we have found, an array of skills. These are rarely found in a single person. The skills are:

1. Logistics: Executes logistics such as setting meeting venues and registering program members.
2. Planning: Elicits goals across the program, designs tasks to achieve them, documents those tasks and orders them in an agenda.
3. Critical thinking: Asks deep questions, from experience, concerning goals, and infers long term (often unintended) effects of the means.
4. Facilitation: Facilitates difficult conversations, distinguishes, and declares productive and destructive behaviors, and personally coaches performers to behave collaboratively.
5. Leadership: Exercises programmatic authority to publicly declare the mission of the program, the values that behavior must fulfill, and critical decisions.

Staff your program with people who have these skills and who, critically, enjoy developing and applying them.

We hope that what you find here is useful; that you will raise with us what you find puzzling or questionable; and that you will share with us any issues or techniques that you, as a DARPA PM, have used to further collaboration between your performers.

References

- Greene, M. (2007). The demise of the lone author. *Nature* 450 (1165). <https://doi.org/10.1038/4501165a>
- Freeman, J., Liens, D., Bell IV, C., & The SD2 Research Consortium. (2019). Selecting and assessing challenge problems. *Theoretical Issues in Ergonomics Science*. 20 (1), 27-38. <https://doi.org/10.1080/1463922X.2018.1485987>

Appendix H: Metadata Survey

3/22/22, 1:04 PM

Metadata Survey

Metadata Survey

We are interested in learning more about the role metadata plays on distributed science-and-technology programs (and similar activities). Please think about related work you do or have done that involves metadata and complete the following items accordingly. The first 14 items ask for your opinion on metadata-related issues in general. We use the term "program" to refer to an entity that governs the work of a distributed group of contributors (performers, scientists, engineers, etc.). This does not refer to any specific program. The subsequent 4 items ask you to reflect on specific work you have done involving metadata. The last item will help us categorize survey responses.

Thank you for your time. Please send any questions or feedback to dleins@aptima.com

1. **What is your definition of metadata? Please check all that apply and add your own definition at "Other" if it is not reflected in the existing options.**

Check all that apply.

- ☐ Metadata describe any surrounding context that helps make sense of the data, but are not the focal data
- ☐ Metadata are a set of data attached to a piece of data or function that gives some additional information.
- ☐ Metadata are everything except raw data generated from observation (e.g., from experimental instruments, measurement devices, algorithms)
- ☐ Metadata is the language of describing where data came from, the state they are in, and where they are going.
- ☐ Metadata describes processes that generate, transform, and otherwise operate on data.

Other: ☐ _____

2. **The program should identify a specific group, e.g., a working group independent of other technical working groups, to manage metadata activities.**

Mark only one oval.

	1	2	3	4	5	
Completely Disagree	☐	☐	☐	☐	☐	Completely Agree

3. **There should be a centralized authority that makes decisions on metadata processes.**

Mark only one oval.

	1	2	3	4	5	
Completely Disagree	☐	☐	☐	☐	☐	Completely Agree

4. **When possible, metadata should be collected automatically, that is, it should not be an effortful activity engaged in by data/metadata users.**

Mark only one oval.

	1	2	3	4	5	
Completely Disagree	☐	☐	☐	☐	☐	Completely Agree

5. **Having metadata associated with my data improves my results by making them:**

Check all that apply.

- ☐ More organized
- ☐ More accurate
- ☐ More complete
- ☐ More interpretable
- ☐ More credible
- ☐ More accessible to others (i.e., more shareable)

Other: ☐ _____

6. **It is easier for me to use other people's data when it has metadata associated with it.**

Mark only one oval.

	1	2	3	4	5	
Completely Disagree	☐	☐	☐	☐	☐	Completely Agree

7. **Others can use my data even if it does not have metadata associated with it.**

Mark only one oval.

	1	2	3	4	5	
Completely Disagree	☐	☐	☐	☐	☐	Completely Agree

8. **The program should define a common infrastructure for metadata collection, storage, and access.**

Mark only one oval.

	1	2	3	4	5	
Completely Disagree	☐	☐	☐	☐	☐	Completely Agree

9. **The program should impose a common format and definition for metadata.**

Mark only one oval.

	1	2	3	4	5	
Completely Disagree	☐	☐	☐	☐	☐	Completely Agree

10. It is important that an organization's metadata schemas are flexible and adaptable enough to accommodate different external metadata conventions.

Mark only one oval.

	1	2	3	4	5	
Completely Disagree	☐	☐	☐	☐	☐	Completely Agree

11. It is important to have resources--for example, tutorials, external consultants, open-source software--that facilitate metadata use.

Mark only one oval.

	1	2	3	4	5	
Completely Disagree	☐	☐	☐	☐	☐	Completely Agree

12. Metadata is most useful to the program when it helps answer:

Mark only one oval.

	1	2	3	4	5	
A specific question	☐	☐	☐	☐	☐	A broad range of questions

13. All performers should be compelled to adopt and exercise the program standards for metadata activities.

Mark only one oval.

	1	2	3	4	5	
Completely Disagree	☐	☐	☐	☐	☐	Completely Agree

14. A program needs metadata and associated systems and processes in place before program activities start.

Mark only one oval.

	1	2	3	4	5	
Completely Disagree	☐	☐	☐	☐	☐	Completely Agree

Please think about your most recent work involving metadata.

For each item, please check all responses that apply. Please add a response to "Other" when prepopulated responses are insufficient.

15. I generated, shared, or used metadata that described:

Check all that apply.

- ☐ Experimental conditions (e.g., temp, time, duration)
- ☐ Experimental materials (e.g., concentrations, component parts, provenance)
- ☐ Data collection instrument parameters (e.g., setting, sensitivity, calibration, device model)
- ☐ Data collection procedures (e.g., protocol steps, sampling, training)
- ☐ Data processing (e.g., transformation, aggregation)

Other: ☐ _____

16. The metadata I generated, shared, or used allowed me to:

Check all that apply.

- ☐ Generate better data (e.g., data that are more valid, accurate, or complete)
- ☐ Share data
- ☐ Transform data
- ☐ Analyze data
- ☐ Interpret results

Other: ☐ _____

17. The metadata I generated, shared, or used allowed others to:*Check all that apply.*

- ☐ Generate better data (e.g., data that are more valid, accurate, or complete)
- ☐ Share data
- ☐ Transform data
- ☐ Analyze data
- ☐ Interpret results

Other: ☐ _____**18. I identified the metadata I needed:***Check all that apply.*

- ☐ Prior to a protocol being generated
- ☐ While a protocol was being generated
- ☐ While a protocol was being executed
- ☐ While sharing data
- ☐ While analyzing data
- ☐ While sharing analytics or results

Other: ☐ _____**This question is for categorization purposes only.**

Please check all that apply.

19. I am a:*Check all that apply.*

- ☐ Natural Scientist (e.g., biologist, chemist)
- ☐ Social Scientist (e.g., psychologist, sociologist)
- ☐ Computer Scientist (e.g., software engineer)
- ☐ Data Scientist (e.g., analyst)
- ☐ Mathematician
- ☐ Research Engineer (e.g., human factors, mechanical engineer)

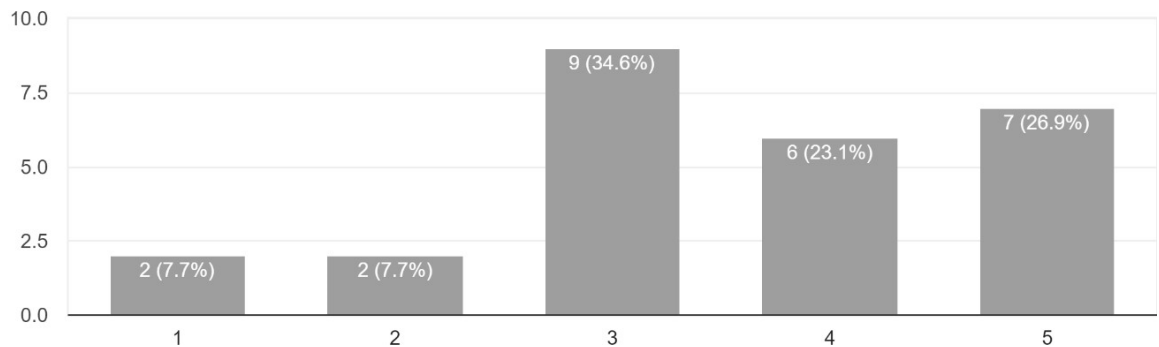
Other: ☐ _____

Appendix I: Metadata Survey Responses

These graphs depict responses by 26 SD2 participants to the 18 questions of a survey concerning metadata. The survey item text is presented at the top of each graph. On all vertical bar graphs, response distributions are presented on a Likert scale on the X axis in which 1 represents “Completely Disagree” and 5 denotes “Completely Agree”. All horizontal bar graphs present frequency counts on the X axis.

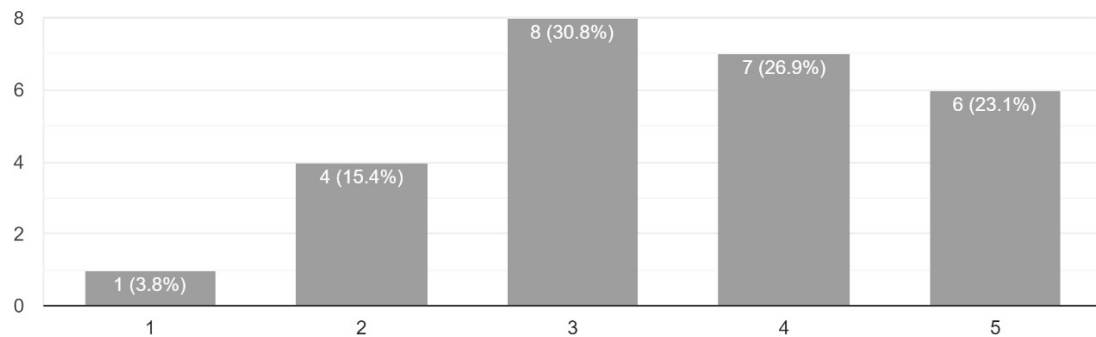
The program should identify a specific group, e.g., a working group independent of other technical working groups, to manage metadata activities.

26 responses



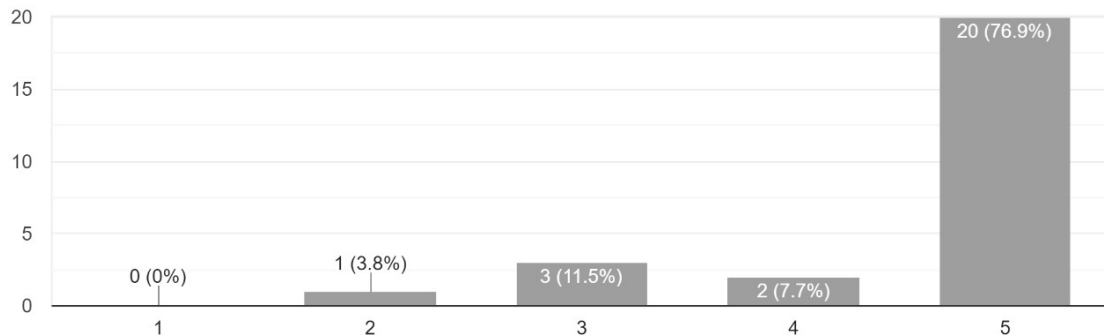
There should be a centralized authority that makes decisions on metadata processes.

26 responses



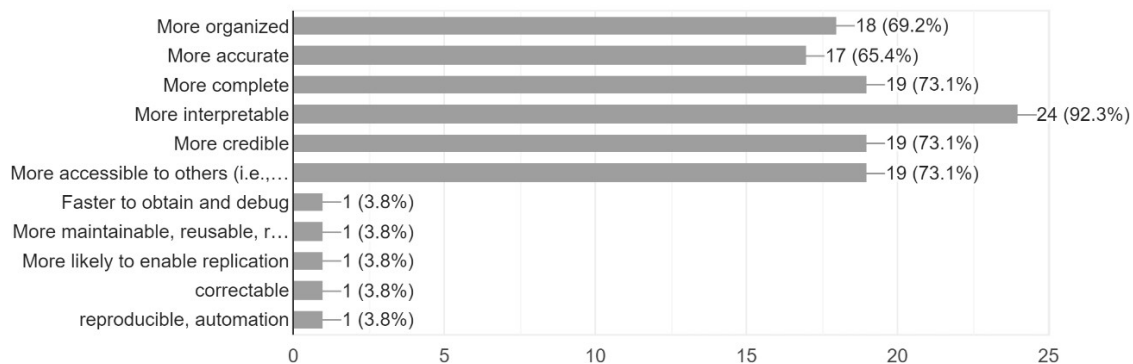
When possible, metadata should be collected automatically, that is, it should not be an effortful activity engaged in by data/metadata users.

26 responses



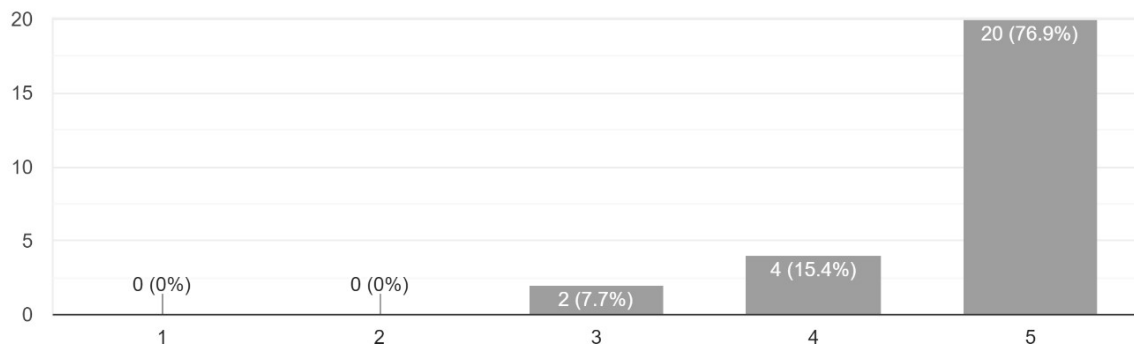
Having metadata associated with my data improves my results by making them:

26 responses



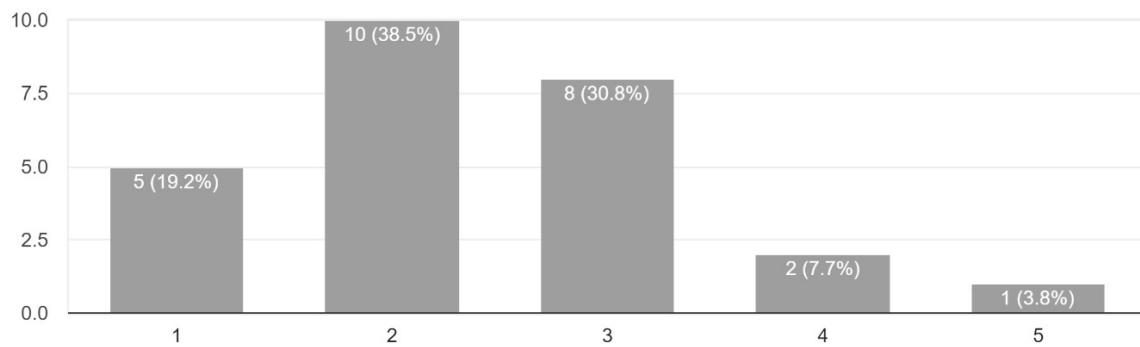
It is easier for me to use other people's data when it has metadata associated with it.

26 responses



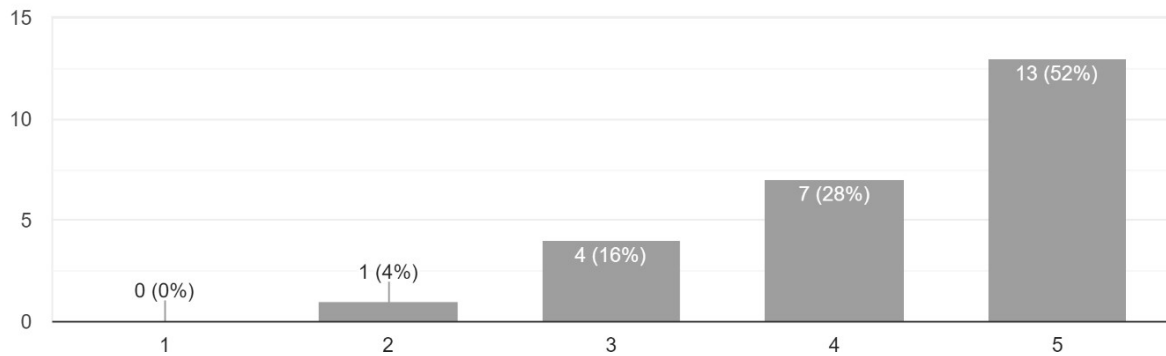
Others can use my data even if it does not have metadata associated with it.

26 responses



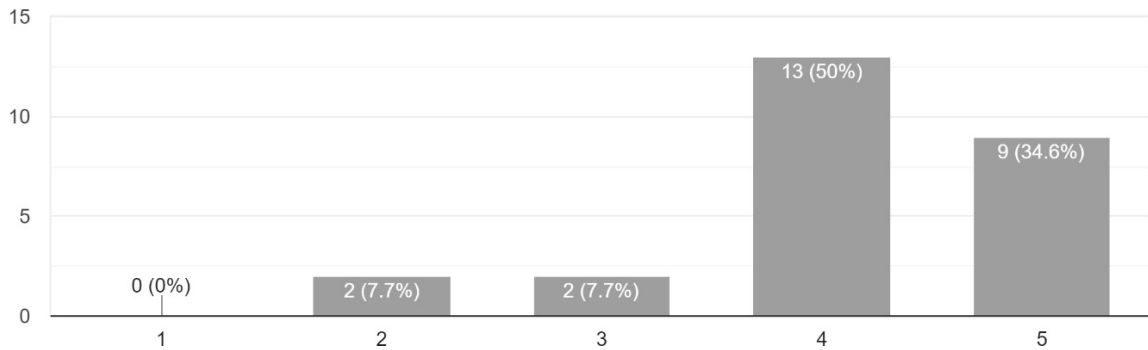
The program should define a common infrastructure for metadata collection, storage, and access.

25 responses



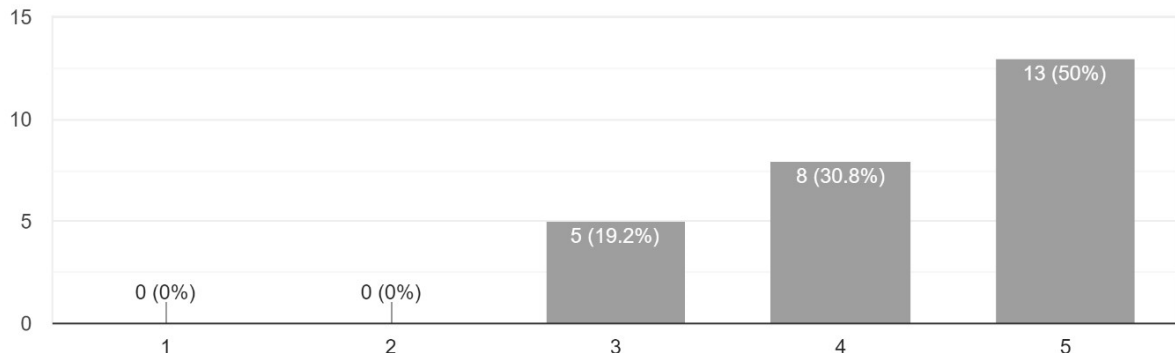
The program should impose a common format and definition for metadata.

26 responses



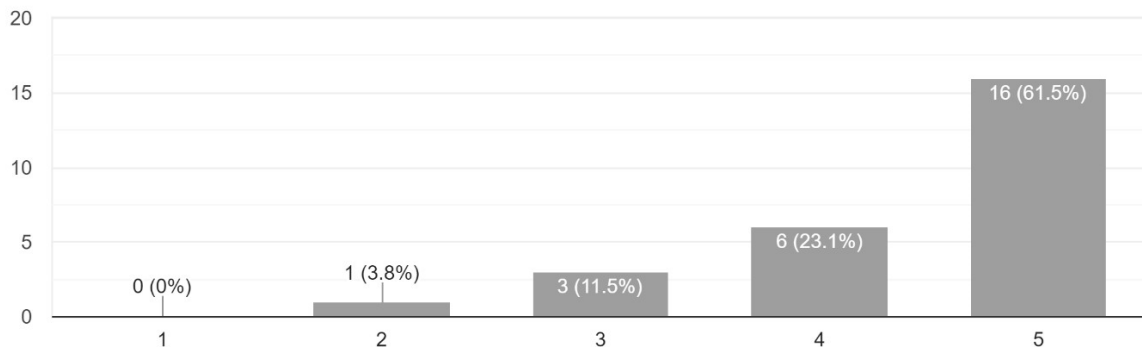
It is important that an organization's metadata schemas are flexible and adaptable enough to accommodate different external metadata conventions.

26 responses



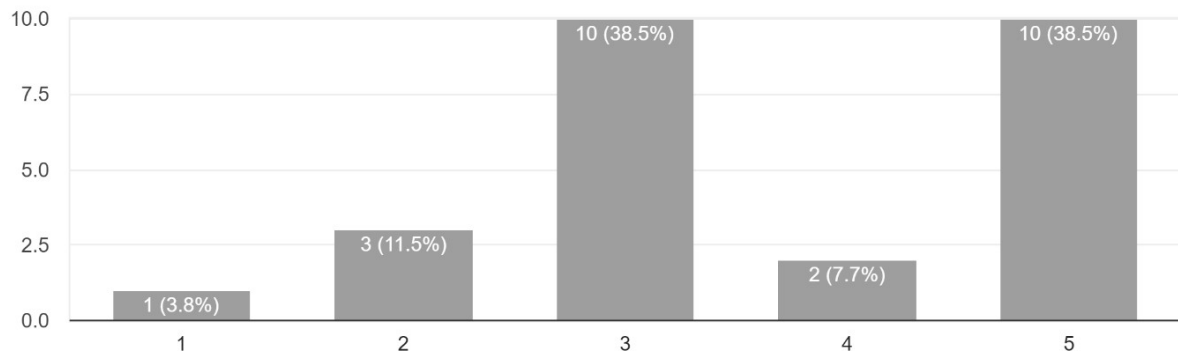
It is important to have resources--for example, tutorials, external consultants, open-source software--that facilitate metadata use.

26 responses



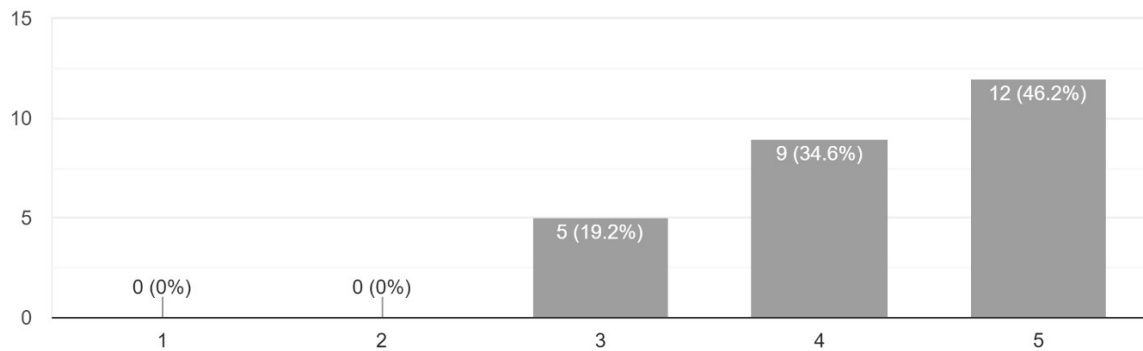
Metadata is most useful to the program when it helps answer:

26 responses



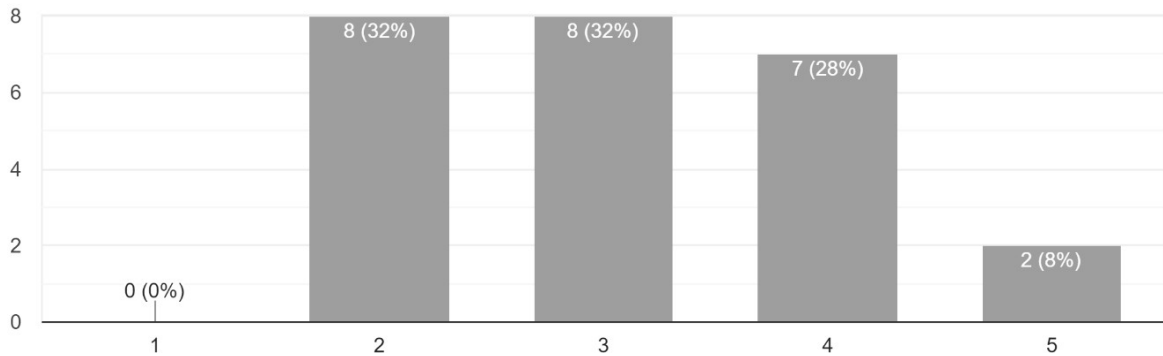
All performers should be compelled to adopt and exercise the program standards for metadata activities.

26 responses



A program needs metadata and associated systems and processes in place before program activities start.

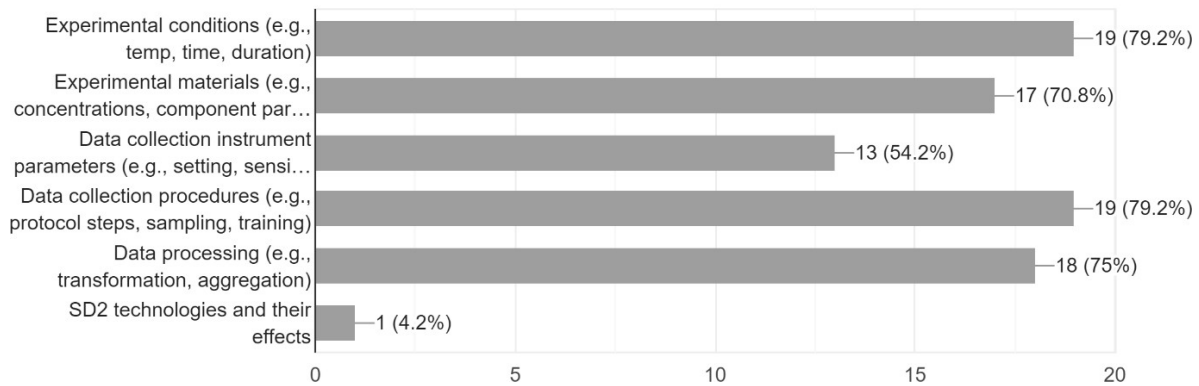
25 responses



Please think about your most recent work involving metadata:

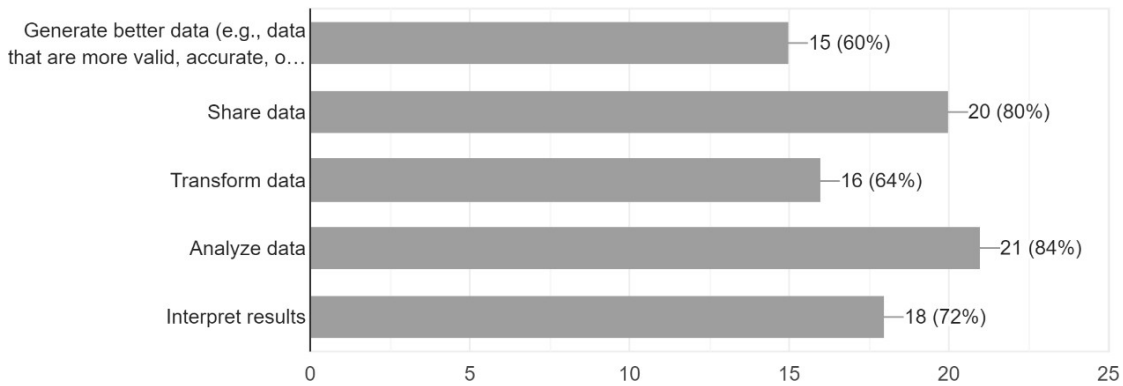
I generated, shared, or used metadata that described:

24 responses



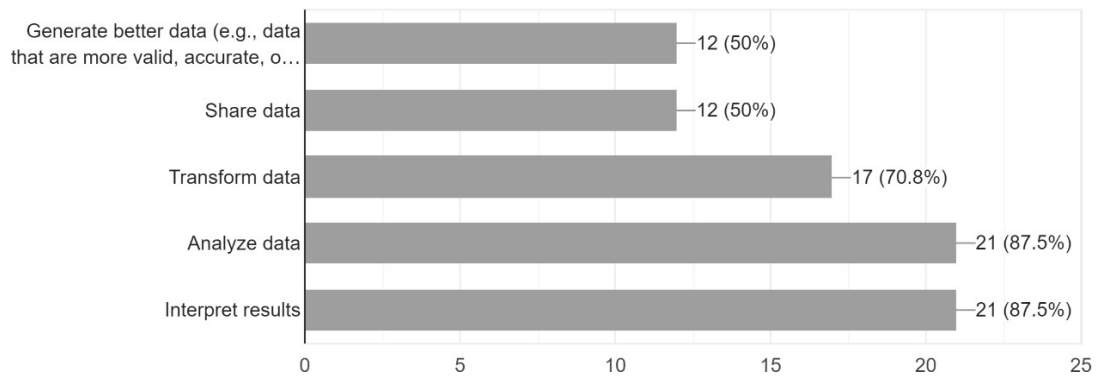
The metadata I generated, shared, or used allowed me to:

25 responses



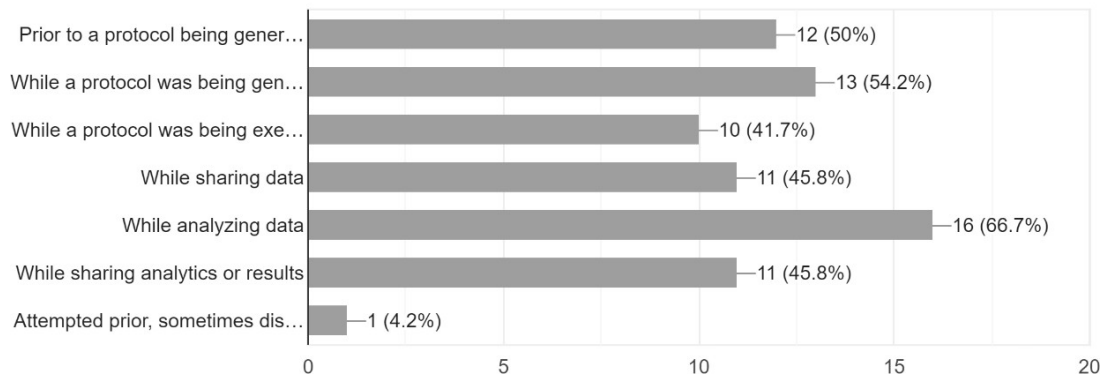
The metadata I generated, shared, or used allowed others to:

24 responses



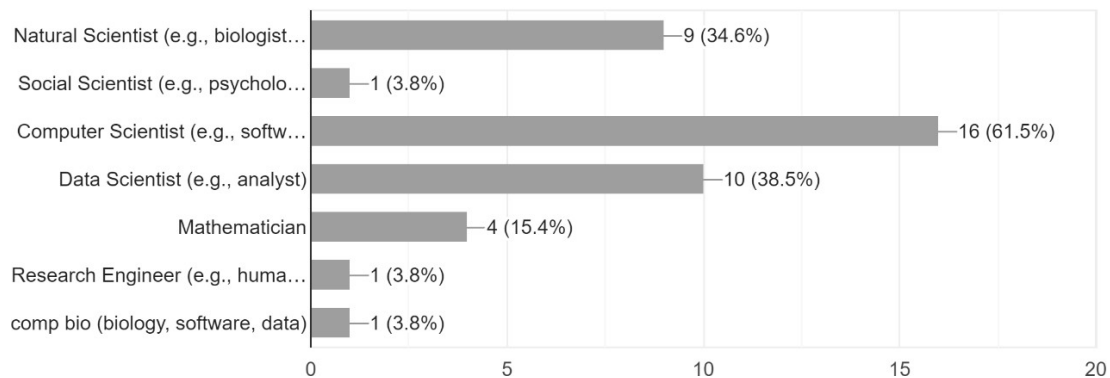
I identified the metadata I needed:

24 responses



I am a:

26 responses



ACRONYMS LIST

Terms	Definition
ACA	Associate Contractor's Agreement
AFRL	Air Force Research Laboratory
AFRL/RI	Air Force Research Laboratory Information Directorate
AI	Artificial Intelligence
API	Application Programmers Interface
ASIST	Artificial Social Intelligence for Successful Teams
AWS	Amazon Web Services
BayesDB	Bayes Database
CBRN	Chemical, Biological, Radiological and Nuclear
CDER/TDA	Cover-tree Differencing via Entropy Reduction / Topological Data Analysis
CF	Cell-Free
CF-RS	Cell-Free Riboswitches
CNN	Convolutional Neural Network
CP	Challenge Problem
CPU	Central Processing Unit
CRedit	Contributor Roles Taxonomy
CSU	Colorado State University
CTA	Cognitive Task Analysis
DARPA	Defense Advanced Research Projects Agency
DASi	DNA Assembler
DBTL	Design Build Test Learn
DNA	Deoxyribonucleic acid
DoD	Department of Defense
DSGRN	Dynamic Signatures Generated by Regulatory Networks
EQAPOL	External Quality Assurance Program Oversight Laboratory
ER	Experimental Request
ESCALATE	Experiment Specification, Capture and Laboratory Automation Technology

ETL	Extraction, Transformation, and Loading
FAIR	Findable, Accessible, Interoperable, and Reusable
GCS	Google Cloud Storage
GPU	Graphics Processing Unit
GUI	Graphical User Interface
HIV/AIDS	Human Immunodeficiency Virus/Acquired immunodeficiency Syndrome
HRPO	Human Research Protection Office
HSR	Human Subjects Research
IDs	Identifiers
iGEM	International Genetically Engineered Machine
IRB	Institutional Review Board
LANL	Los Alamos National Laboratory
LBNL	Lawrence Berkeley National Laboratory
LINCS	Library of Integrated Network-Based Cellular Signatures
MIT	Massachusetts Institute of Technology
ML	Machine Learning
MOEs	Measures (or Metrics) Of Effects
MOTs	Measures (or Metrics) Of Technologies
MQTT	Message Queuing Telemetry Transport
NIH	National Institutes of Health
NIST	National Institute of Standards
NoSQL	Non-Structured Query Language
NRC	National Research Council
NSF	National Science Foundation
OD	Optical Density
OMB	Office of Management and Budget
OPIL	Open Protocol Interface Language
PCA	Publication Consortium Agreement
PD	Protein Design

PI	Principal Investigator
PM	Program Manager
PNNL	Pacific Northwest National Laboratory
PSU	Pennsylvania State University
PV	Perovskites
Q0	Quarter 0
QC	Quality Control
R&D	Research & Development
RAM	Random Access Memory
RDF	Resource Description Framework
RNA	Ribonucleic acid
RNA-seq	RNA Sequencing
S&T	Science & Technology
SBH	SynBioHub
SBME	Synthetic Biology for Military Environments
SBOL	Synthetic Biology Open Language
SD2	Synergistic Discovery and Design
SETA	Systems Engineering and Technical Assistant
SOW	Scope of Work
SPARQL	A recursive acronym for SPARQL Protocol and RDF Query Language
STS	Socio-Technical System
SYNBICT	Synthetic Biology Curation Tools
TA	Technical Area
TACC	Texas Advanced Computing Center
TASBE	Tool-chain to Accelerate Synthetic Biology Engineering
UCSB	University of California, Santa Barbara
UI	User Interface
UW	University of Washington

VDR	Versioned Data Repository
WG	Working Group
WSL	Windows Subsystem for Linux
YAADA	Yet Another Analytic Dataflow Architecture
YS	Yeast States