Bringing Back the Social Affordances of the Paper Memo to Aerospace Systems Engineering Work

Author 1, Author 2, Author 3
Affiliations
e-mails

ABSTRACT

Model-based systems engineering (MBSE) is a relatively new field that brings together the interdisciplinary study of technological components of a project (systems engineering) with a model-based ontology to express the hierarchical and behavioral relationships between the components (computational modeling). Despite the compelling promises of the benefits of MBSE, such as improved communication and productivity due to an underlying language and data model, we observed hesitation to its adoption at the NASA Jet Propulsion Laboratory. To investigate, we conducted a six-month ethnographic field investigation and needs validation with 19 systems engineers. This paper contributes our observations of a generational shift in one of JPL's core technologies. We report on a cultural misunderstanding between communities of practice that bolsters the existing technology drag. Given the high cost of failure, we springboard our observations into a design hypothesis – an intervention that blends the social affordances of the narrative-based workflow with the rich technological advantages of explicit data references and relationships of the model-based approach. We provide a design rationale, and the results of our evaluation.

ACM Classification Keywords

H.5.0 Information interfaces and presentation: [General]

Author Keywords

CSCW; ethnography; design; technology acceptance

INTRODUCTION

The discipline of systems engineering emerged in the 1950s as an interdisciplinary, holistic way to manage the complexity of large aerospace projects [30]. With the ability to operate within and integrate findings across disciplines, the field has grown in size and scope, and has become a major contributor to projects in industries from software to agriculture [31].

Paste the appropriate copyright statement here. ACM now supports three different copyright statements:
• ACM copyright: ACM holds the copyright on the work. This is the historical approach.
• License: The author(s) retain copyright, but ACM receives an exclusive publication license.
• Open Access: The author(s) wish to pay for the work to be open access. The additional fee must be paid to ACM.
This text field is large enough to hold the appropriate release statement assuming it is single spaced.

Like many other engineering disciplines, the systems engineering workflow focuses around the memorandum. Memos become the authoritative source for requirements, constraints, architecture, designs, technical trade-offs, and decisions, and a conduit for communication and consensus (for example, the Cassini Control Analysis Book [1] shown in Figure 4). Critics of systems engineering point out that documents are by nature static, whereas engineering requires dynamism. Changes in one document do not propagate to others, and must be synchronized manually. Documents often become outdated, disconnected and even contradictory [26].

To address these challenges, some systems engineers (SEs) advocate a new perspective that relies not on documents but on formal computational models [9]. With Model-Based Systems Engineering, or MBSE, documents perform the same function, but are generated automatically from a model, instead of being directly authored or edited. In theory, the move from documents to models has several advantages. It can speed the sharing and communication of data, more dynamically maintain a correctness in the face of changes, and automate a chain of accountability [11].

Despite these promises, organizations have been slow to adopt MBSE. In this work, we investigate its barriers to technology acceptance. To do this, we conduct a six-month ethnographic field study of 19 systems engineers at the NASA Jet Propulsion Laboratory, during its experience of changeover to MBSE. We find substantial risk of organizational inertia [24].

This work makes two principal contributions. First, we characterize the cultural transition and find that JPL's systems engineers actually accept MBSE, but will not do it. They create documents with a correct picture of the state of the spacecraft, but lack the narrative explaining design decisions. Second, given the high cost of failure, we take an interventionist posture. We devise and conduct a preliminary evaluation of a socially-appropriate technology [26], borrowing the social affordances [16] of the paper memo [29] as a design metaphor. We create a prototype digital memo that captures rationale and decision-making, and still takes advantage of the clear traceability and provenance that derive from digital artifacts. Preliminary results show that a digital memo can bring the social affordances of paper memos into the digital world, and spur greater MBSE adoption.

In this paper, we describe our efforts to understand an emerging cultural phenomena, and to streamline its acceptance. We start by placing our work in a historical context.

HISTORY AND RELATED WORK

This section elaborates the history and roles of systems engineering, describes the emergence and capabilities of a model-based approach to systems engineering and its growth as a movement, and discusses the juxtaposition of the two approaches. The section then situates the contributions of this work in its historical context. We begin with a brief history of systems engineering.

The origin of systems engineering

In the wake of the Second World War, the National Advisory Committee for Aeronautics (NACA), predecessor to NASA, was introducing experimental forms of automation to the cockpit. Systems engineering evolved as a way to manage the complexity that accompanied this high risk and poorly understood technology. Rather than engineering any aircraft part individually, the systems engineers defined relationships within and between parts of the aircraft [21]. They then left the design and engineering of the individual component assemblies to newly-evolved domain specialties. These engineers would own the development of subsystems, such as power, propulsion, and communications.

The technological change also brought with it social and organizational change. Systems engineers, with new backgrounds, educations, and technical values found themselves leading projects [18]. These engineers believed in and advocated cockpit automation [21]. The same automation that enabled space flight also threatened the autonomy, and even the existence, of the main users of the technology: pilots. Although they saw themselves as primary users of aircraft technology, new features made the highly skilled pilots feel replaceable [18].

Modern systems engineering has evolved to include the architecting, implementation, and deployment of entire space missions [29]. To accomplish this, systems engineers are responsible for managing the information that describes the mission and project assets. This information covers specifications, requirements, trade studies, and a host of analyses, and requires that SEs collaborate with multiple sub-systems engineers, managing such capabilities as propulsion, power, and telecommunications [20].

The dawn of model-based systems engineering

The majority of the information that SEs manage resides in various documents. Changes in one document would need to propagate across entire project teams, necessitating that downstream references and dependent calculations be updated for consistency. As systems engineering grew in scope, synchronization grew to become a major issue confronting the discipline [29].

Model based systems engineering emerged as a response to these foundational problems [20]. Modeling advocates argue that by moving from decentralized documents into a single authoritative source of truth, systems-based truth could always be carefully monitored [29].

Even with many large aerospace organizations actively investigating its potential (e.g., NASA [2, 10, 27], Lockheed Martin [9, 19, 23], and MITRE [29]), MBSE has been slow to make a significant impact. To facilitate the transition, proponents developed a set of international standards [3, 2], created a governing body [9], and received a boost with the introduction of a common modeling language [12], SysML, an offshoot of UML. The language allows for a model-based representation of hierarchical architectural and behavioral data and polymorphic relations. Main tools such as MagicDraw were extended to accommodate document and report generation [8] to support the current systems engineering work.

Contributions

Systems engineering is a field on the brink of a generational change. Given the potential impact of this transition, systems engineering might benefit from the shared organizational experience of the CHI community. To that end, we interview 19 SEs over a six-month period at the NASA Jet Propulsion Laboratory, home of US robotic space exploration [21]. Our work brings an analytical eye to this significant and impactful community of practice, and adds to a long tradition of field study of sociotechnical practice and technology acceptance [26]. Our work brings to light candidate road blocks to MBSE adoption.

Because our interests also include project success, we take an interventionary approach while following in the tradition of Design Research [6]. We translate our observations into a plan for the design of a software system. We rely on the thick description generated from our field work to develop insights into the underlying problems facing the field. We conclude with a design hypothesis, a strategic approach to address the situation.

METHODS

We studied SEs performing *in-situ* project work and model development. The goal of our research was to investigate the perceived hesitation of MBSE adoption.

To accomplish our goal, we focused on the current tools and practices and their interaction with the social context of systems engineering work. Our aim was to generate grounded observations informing hurdles to adoption, and to evaluate an appropriate intervention.

Our field work included ethnomethodologically-inspired ethnography [28] to understand systems engineering work in the context of the observed culture [25], technology, and workflow. Then, to vet our own understanding of the problem domain, we used a design research method called needs validation [7]. Through this, we evolved a broad picture of real user needs.

Subj.	FS	NV	Relationship	Role
P00	×	×	Systems	Core
P01	×	×	Systems	Core
P02	×		Systems	Core
P03	×		Systems	RW
P04	×	×	Systems	RW
P05	×		Systems	RW
P06	×		Systems	Core
P07	×		Systems	None
P08	×	×	Systems	RW
P09	×		Systems	Core
P10	×		Verification	Consumer
P11	×	×	Systems	RW
P12	×		Systems	Core
P13	×		Management	Support
P14	×		Subsystem	Consumer
P15	×		Subsystem	Consumer
P16	×		Systems	Support
P17		×	Systems	Consumer
P18		×	Systems	RW

Table 1. Subject breakdown by relationship and role in the ethnographic field study (FS, $N_{\mathbf{FS}} = 17$) and needs validation (NV, $N_{\mathbf{NV}} = 7$). P00 was primary stakeholder.

For the field study, fifteen subjects were selected for variability of job function (e.g., project management, systems engineering, subsystems engineering, verification and validation, project concept, etc.), technical knowledge, and modeling experience. One additional participant was recruited using purposeful snowball sampling. The subjects formed a representative cross-section of skilled systems engineering work on a specific mission architecture. Three of the subjects were female; the rest were male. This male-to-female ratio parallels industry ratios for technical engineering fields [15]. The level of experience with MBSE approaches varied widely from no experience to daily model development.

Each interview lasted at least one hour and occurred in the subject's office. To help subjects feel more at ease, no recording devices were used. We chose semi-structured interviews to allow the participants to lead the discussion, and to form a partnership in the investigation. Interviews typically began with a high-level introduction to the study and interviewer's background, followed by modeling and workflow to encourage emergent topics. Table 1 lists the subjects, their relationship to the project (whether the subject was a systems, subsystem, or verification and validation engineer, or manager), and their role of interaction with the data stored in the model, with read-only being on one end of an imaginary interaction spectrum, and building fundamental model elements such as library components on the other.

The beginning of the interview centered around models and modeling. Modeling refers to the process of creating the model-based representations of the system design (models). We asked broadly about modeling and inquired about the artifacts that constitute a model. Next, we discussed the process of modeling, nudging the conversation towards architectural models for system documentation. To assess the subject's place in the process of model-building, we asked the user about their level of interaction with the model, and at which stage(s) of completion their interaction takes place (with speculative or concept-level on one end and complete and deployed on the other end). We asked the subject how relevant data are gathered and input into the model, and we asked about where the model goes next to determine the consumers of the model and model workflow. Interview data were transcribed, and coded for emergent themes. Specific representative experts from the interviews were grouped separately.

Needs validation is a powerful asset when used prior to design and development. We generated a list of perceived technical hurdles with modeling using the data from the field study. To vet our understanding with the community, we used a method of user research called needs validation. This method is useful in using user-centered techniques to iteratively prune researcher understanding of the problem domain and to identify areas where more research would be required.

We generated a list of 16 perceived user needs. For each of the needs, we drew a two- or three-panel storyboard. In the first panel, we identified the user need as we perceive it, or the need under test. In the second panel, we drew a technology-agnostic proposed solution to the need (see Figure 1 and Figure 2). The storyboards were used in follow-on interviews as a communication aid to discuss the perceived needs and their proposed solutions. We reused some subjects from the field study pool, and continued interviews until repetition of data, which occurred after 7 participants. Interviews lasted around 90 minutes and occurred in the subject's office. No recording devices were used.

For each of the 16 needs storyboards, subjects were instructed to place themselves in the situation of the character drawn in the storyboard. Next, subjects were asked to give their overall impression of the need under test (e.g., "Is this a need?") along with the frequency and severity of its impact (e.g., "How often does this happen? Is it a big deal?"). We asked subjects for specific examples of the last time they felt themselves in the situation of the need under test. At the end of all storyboards, we asked subjects to rank their five most important needs from the storyboards, in descending order from most important to address. Finally, we asked participants for any other needs that the storyboards failed to address or any additional insights on user need.

FINDINGS

Our field investigations and needs validation revealed:

1. The emergence of two distinct user classes: We discuss the culture surrounding model-based systems engineering, and the miscommunication between the two communities.

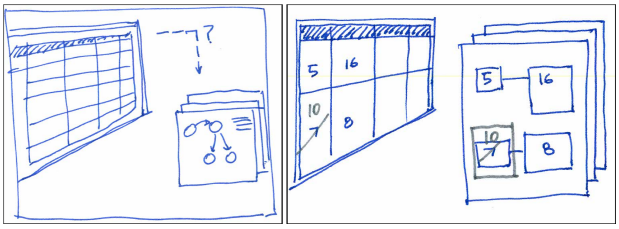


Figure 1. Needs validation cartoon depicting spreadsheet import need under test. Left: "I use spreadsheets, and find them terrifically useful – but how do I get them into the model?" Right: The spreadsheet data are in the model; assume shared data layer

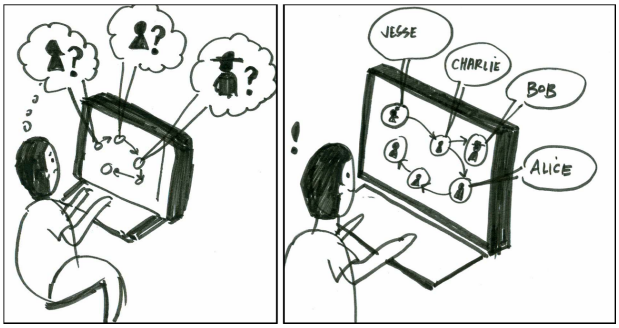


Figure 2. Needs validation cartoons depicting ownership need under test. Left: "Who is responsible for these model elements?" Right: Each model element's contributor, date, and authority is presented.

2. The acceptance of MBSE as an idea, but that some engineers will not do it: We present the two communities' thoughts on MBSE, and discuss the toolset surrounding the MBSE practice.
3. The changing role of the memo: We describe the historical significance of the memo, frame the memo as a social affordance, and discuss the future of the memo in a sociotechnological context.

Two Distinct Cultures Emerge

Across all participants in both the field study and the needs validation, two distinct user populations emerged. These groups were characterized not by their role in the project, but by how often and in what ways users interacted with or contributed to the system model. Interestingly, we found that the boundary between the cultures was not matched to job title because many of the participants had multiple roles on the project (e.g., model development, analysis, and design), and that most subjects took on several roles or responsibilities within a particular project. There was a spectrum of interaction among those not contributing directly to model development, yet the overall distribution was bimodal: model developers occupied a much more prominent role in model contribution (see Figure 3). The communities were divided in their opinions of modeling. A cross-cutting characteristic that separated the groups was who was doing the model development.

Broader communities of practice

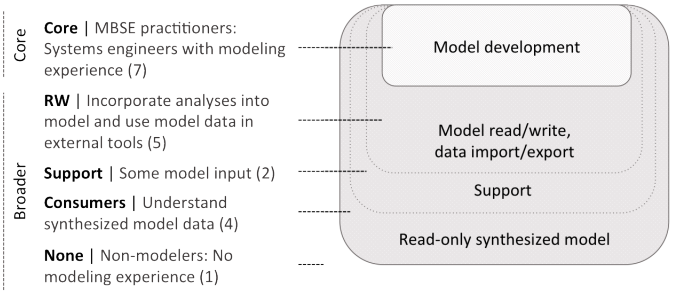


Figure 3. Model data proximity and subject breakdown for all parts of study ($N = 19$)

The main community that emerged in our findings consisted of some traditional and model-based SEs, subsystem engineers, managers, and program officers. SEs were experts in breadth across these numerous disciplines; subsystems engineers were depth experts in their specific domain. We called this group the *broader* community because it accounted for most of the user population. The members in this group had read-write, support, and consumer roles with the model data and typically had tools that they use in their own domain (e.g., simulation, mathematical analysis, computer-aided design). For example, P04 is a SE who needed read-access to many aspects of the data model in order to simulate the entirety of the project. Like many SEs in the broader group, P04 needed a separate domain-specific tool and access to model data to do his work.

We found that the broader group was made up of smaller subgroups, characterized by the ways in which they interacted with the model data. We call the first subgroup the *read-write (RW)* users, who needed to contribute to project data, including modifying parameters and adding and deleting elements as the system design changed. Most of the read-write group were traditional SEs and subsystems engineers with some or little formal training in a model-based approach, yet they had to interact with the model data directly. Next was a group of model *support*, for whom it was necessary to contribute to the model and understand modeling challenges to manage expectations. For example, the project manager needed to understand what goes into modeling so he could assign tasks and architect the documentation. The next group we call the *consumer* group. Members of the consumer group were read-only users that needed to understand the displayed, human-readable model and associated information; for example, in review meetings.

Core group

A smaller emergent group group we call the *core group*. This group's members possessed two qualities that shaped their interpretation of the model-based approach to systems engineering both inside their own community and outside of it. First, all individuals had training and extensive education both in each of their domains and in modeling. Second, the core community showed the properties of dedicated early adopters as they joined together

to form a support group for modeling SEs (modelers) to meet, interface, and learn from each other's work. Members ran MagicDraw classes to teach aspiring modelers how to create system diagrams. An international community of SEs spanning several companies formed an MBSE usability group[1] in order to meet and discuss issues related to best practices, use cases, MBSE process, and the evolution of modeling tools.

The core group consisted mainly MBSE practitioners who were SEs with modeling education and experience and expert modelers with daily model development tasks. These individuals mainly used the main modeling tool While modeling undoubtedly affected all participants' jobs, only the core group were model developers.

Engineers believe in MBSE, but don't do it

The two distinct cultures that emerged each had a belief about the other, which led to a misunderstanding.

P00 and several other core group members expressed the overall goal being to "encourage people to think of the model(s) as the first choice in capturing and presenting information." P00 said the community currently thinks about modeling "as an imposed afterthought."

Yet we found that the broader group was not rejecting the model-based approach, but the software. In fact, broader group users implicitly accepted the concept of an underlying data model. Our experience with two needs validation storyboards (spreadsheet import and ownership) illustrates the broader community's position.

Spreadsheet import was described as the need to get the spreadsheet data into a model (Figure 1). The storyboard showed a spreadsheet and a model with no link between them. The researcher described the scenario as one in which the model does not have any communication to the spreadsheet: "I have this great spreadsheet. How do I get it into the model?" P01 and P04 described the current process of importing model data as "extracting" the data from e-mails, spreadsheets, and slide decks by hand, and manually entering it into the model using the modeling tools. The solution frame showed an edit to the spreadsheet data corresponding to a simultaneous edit in the model, with either the spreadsheet or the model being the primary recipient of the change. Additionally, the researcher would add: "The spreadsheet is a stand-in for exposing some kind of interface that lets you edit the model data in your own tool." In this way, we wished to imply the need for SEs to continue using their tools, methods, and practices, rather than forcing them to adopt the new methods and tools. All participants expressed agreement with the implicit and explicit needs under test; the primary goal was to get the data into the model, not to convert all users to the modeling toolset. P00 called the spreadsheet import need a "core issue of systems engineering."

Ownership was described as the need to assign responsibility to model elements (Figure 2). The storyboard showed an MBSE practitioner working on the model and confused about who was responsible for each of the model elements. The solution frame showed the same engineer with each of the model elements' responsible parties explicitly identified. The problem of ownership was seen as "hugely important" (P18). P17 said that in addition to showing who is responsible, SEs need to know (and the model needs to capture) authority, responsibility, rationale, and "we are missing the way to show provenance of that data." P18 said, "It's not enough to see that the model changed, you also need the rationale of why the model changed. It is important to record the history now, or else you come back to it 8 years later and you have no idea what happened." Every participant expressed enthusiasm about spreadsheet import and ownership; acceptance of these needs shows an implicit acceptance of the model-based approach.

Evolution of the tools

The main modeling tool was MagicDraw[2], a tool that uses boxes and arrows to express architectural and behavioral characteristics of a system. It uses SysML as the underlying language to build the model.

The software was excellent for capturing and representing the model in a very specific format and presentation. Additionally, the core group had built a set of plug-ins for the modeling environment to facilitate document generation directly from the model data. The document generation affordance was wildly popular. However, we learned that there were two main problems with the tools: hiding the data, and not capturing the metadata. Data were hard to retrieve from the model, as evidenced by P04's complicated workflow involving transformations in several programming languages. The metadata, including ownership, permissions, rationale, review, and data provenance, were either not captured, or not visible to the user.

Yet for P04, getting the data out of the system model and into his tool was a cumbersome task that involved several scripts written in three domain-specific languages. "I thought it was a thing I would only have to do once," P04 said. "I've had to do it about a half-dozen times so far." On the other hand, P15, a subsystem engineer, frequently needed to update the data in the model with new parameters, especially when considering variations on designs. His current work process involved keeping spreadsheets that he "borrowed from [previous projects], or copied someone else's, with my own color-coding." At certain milestones, he would e-mail his spreadsheets to a SE whose job, according to P01, was to hand-transcribe the data into the model.

The broader community members took classes in model-based systems engineering approaches, participated in

[1] MBSE Usability community page: `http://www.omgwiki.org/MBSE/doku.php?id=mbse:usability`

[2] MagicDraw by No Magic, Inc.

the core group's tutorials and workshops on the methodology and the tools under development. But members from the broader group expressed mistrust in the data emerging from the tool, because the tools left the data invisible, without rationale, and difficult to verify. The mistrust manifested itself when participants said, "You cannot trust the diagrams" (P09) and "Are the data a fiction?" (P07). Relationships between SEs were not always smooth, partially because of data opaqueness. P03, a core group member, said other engineers and especially non-modelers often say, "I'm not entirely sure I trust what you're doing." P13 said, "I don't care if it's in the model. That's not enough," adding that he would need to see verification and rationale associated with the data.

Because they understood MBSE, the core group were asked to build extensions to support their own, and the broader community's, work. One core group member called the practice of using the tool itself to build the necessary extensions "Eat[ing] your own dog food" (P00) or "Develop[ing] with what you fly with" [4]. This created a discontinuity because core group members were mainly domain experts, not software engineers, interaction designers, or programmers. The extensions they created were cumbersome and weighty. "How do you give someone else this capability ... without needing eight years of programming experience?" asked P03.

We found that the broader group rejected the tools, including the main tool and the extensions that the core group was building, but not MBSE as a practice. For example, P14 said, "We would do all our reports in this thing" if the tools met his needs. Evidence for the rejection came as the threat of falling back on traditional systems engineering practices. Prior to models, spreadsheets were the way for SEs to store their information and findings, but they lacked central storage and good versioning and collaboration options among and between systems. "We need to pry spreadsheets out of their hands," P00 said.

Furthermore, the main tool was seen as "too complicated for non-modelers" (P12) and would "overwhelm [subsystems engineers]" (P08). The core group created an extension to the main tool for subsystems engineers to be able to contribute to the model, but one subsystems engineer (P14) exclaimed, "[The core group] built this for us, but we can't use them!" This left broader group users with limited ability to contribute.

Resistance to the toolset

Resistance to the main modeling tool was universal. Both the core community and the broader group were hesitant to accept it fully. "The way tools are used now is 99% a complete waste," one subject said (P11). P14 said, "The concept [of MBSE] is great. It's the tool, or how we use the tool."

Subjects remarked that the tools were not flexible enough to allow creative engineering design and expression, yet not rigid enough to allow structuring of data for analysis. For example, P01 said, "SysML is based on a bad standard. I think it's too rigid... [Traditional SEs] look at SysML and give up – partly because of the tooling, partly because of the language." The tool "has the potential to be a schematic capture program," a need integral to P14's daily task, yet it did not. In the interview, P14 opened his go-to schematic capture program[3] and explained about the structure of the data; for contrast, he opened a diagramming tool[4] and discussed the flexibility that it affords.

Figure 4. Cassini-era memoranda for Control Analysis compiled into a three-volume set

Later P01 added that it's not just that the tool is too flexible – "it's so flexible there's no one way to do something. You can find many bad ways to do something... [The tool] doesn't allow me to have creativity in the way I need." P14 also said the tool was "... too flexible. We need to be more disciplined or more rigorous in how we use it." P14 explained that flexibility in a modeling tool is a negative, requiring little initial investment yet "slowing you down" as the model becomes more complicated and requires increased precision.

The changing role of the memo

We learned that the memorandum contained the locus of the rationale about system design, and that somehow its writing, review, and cross-reference vanished as a practice.

Historically, formal documentation was always "a big deal. People put a lot of priority on documenting what [they] did," (P16) and the document was a source of pride for the engineer. Hundred-page interface specifications were a common sight. P11 said, "I miss that stuff." According to P16, the memorandum was one of two necessary artifacts needed to close an action item. The second was the handoff, in which the artifact was presented as an exchange product or deliverable when transferring responsibility to the next phase in the workflow. Figure 4 shows one such memo compilation.

According to Sellen and Harper [26], there are five main affordances of the paper memo, which are both physical

[3] Orcad
[4] Omnigraffle

and social, promoting effective workflow and collaboration: *a*) the paper memo supports document authoring, as authors are more likely to refer to paper than to electronic documents; *b*) document review is much easier on paper, especially where annotation and comment is necessary; *c*) planning and organizing was preferred to be done on paper; *d*) paper supports collaboration, especially in meetings and review sessions; *e*) paper was preferred to electronic means for knowledge sharing.

From conversations with P16, we learned that two major things happened to change the role of the memo. The first was the advent of computing. Whereas before computing, typing up and distributing a memo took weeks, e-mail systems made transferring information that much faster. However, the archival format and the rigorous process was lost. Accompanying the technological change was a culture shift that took engineers away from their desks and relocated them into meeting rooms. The need to communicate design and its decisions did not vanish, but instead of sending inter-office memos, SEs started meeting and presenting their work to one another. For example, P16 explained, "We spend all our time in meetings because we have to communicate. It's inefficient... 90% of meetings is for people to go and listen to see if they can understand what's going on." P16 said that the writing process was crucial to understanding for the writer, and reading the memo was faster for the recipient to absorb the information. "If you ask them to write and to write about what they're thinking, that's a completely different thing... It's the thinking we're trying to capture," said P16. Moreover, participants said that the terseness of the written communication that comes with presentations (i.e., slide decks) was a detriment to the process. For example, "The other bad thing that happened was Power Point," P16 said; "[Now,] things get reduced to bullets," said P07.

SEs noted that role of the written documentation describing the system changed from primarily a written narrative punctuated by figures and tables to a set of figures and tables with occasional descriptions. Some SEs criticized the process: "We've replaced a well defined process that works with one that's broken and is a complete waste... The information is of much lower quality. [Now,] there is no structure of understanding where you are compared to your goal" (P11).

We found that as a result of these technological, cultural, and process changes, presentation slide decks, e-mails, spreadsheets, and other non-archival files were among the only written remnants of system design. Each member of the core group described the fragmented process of gathering project information from various sources. With MBSE, the individual sources have to be re-entered into the model, which is expensive: P01 said, "it takes a week across 3 people working half time" to make and verify changes to the model given a specific document. It became more difficult to determine the responsible party for a specific question. For example, P07 wondered, "Has someone else thought about this?" and P10 said, "Everybody thinks they're Leonardo Da Vinci, but really everyone is reinventing the wheel when they do this work." P00 recounted a story in which he spent two weeks gathering information about the system in order to run verification on it, but then discovered someone else had performed the same analyses months prior.

Model-based systems engineering (MBSE) was created as a way to address the problem of non-rigorous, non-centralized system representation. While model-based approaches have centralized the information surrounding the system design, they have yet to pay tribute to the profound importance of narrative as a primary focus of system documentation. Capturing the story of system evolution, along with the design decisions, critiques, analyses, and context was seen as "a big need" (P17); indeed, P16 said that for 20 years MBSE had been making slow headway: "It was a slow slog... the basic underpinnings [of MBSE had] to be created from whole cloth."

Software digitized the current practice, but the current practice maintained the fractured narrative. As the system design changed more and more rapidly, secondary data such as parameters became primary. Model-based approaches digitized the changed technique, and though the data were centralized, the narrative was still largely unaccounted for. "Long, thoughtful narrative is extremely important," P16 said. Yet the narrative was often overlooked: P01 said the description fields for model elements were often insufficiently documented, or blank.

The memo as a social affordance

The term "social affordance" draws on the definition of a physical affordance [13, 22]. An object affords a specific action by its shape, texture, presentation, and other features. For example, a door typically affords pushing or pulling to open it. The social aspect of affordance is an important field of study, as certain systems cannot exist without their social context [16, 17]. Social affordance was first described in the computing environment as "the relationships between the properties of an object and the social characteristics of a group that enable particular kinds of interaction among members of that group" [5]. We continue to use social affordance, but apply it to the paper memo: the relationships between the paper memo and the culture of engineers who come together to collaboratively author and review it.

DISCUSSION

We observe a cultural standoff within systems engineering, yet this is far from a new phenomenon: What is happening today reflects the same tension as historically evidenced. The two communities which we saw emerge in the cultural standoff – the core and the broader groups – reflects the same sociological structure as when systems engineering first entered aerospace in the 1960s. But this time, instead of SEs in opposition to pilots, it is the SE community engaged in a cultural clash with itself. A new generation of SEs with a new education and a new

kind of thinking about the system (i.e., the core group) is injecting a new approach (i.e., MBSE) into the workflow of the traditional and broader community. These new SEs are disrupting the broader community's way of working, and furthermore, the broader community sees their injection as undermining their autonomy and expertise. Thus, MBSE adoption is slow, and accompanied by mistrust. The three main reasons for the mistrust are as follows.

First, members of the broader community recognize the importance of telling the story of how the system design evolved. Some remember the days of the paper memo, when a long, thoughtful narrative was the primary focus of the document, and figures and tables accompanied the narrative. Moreover, memos were very good at providing cross-references, and encouraging engineers to examine, cross-reference, and think critically about the system design. The current tools and practices are not capturing the story and are not enabling the kinds of work that engineers see as essential to their job.

Second, the tool and its extensions are weighty, cumbersome, and unsupportive of the current work practices. The broader community cannot use the tools to contribute to the model, and have difficulty using them to perform their own analyses. The core group is also frustrated by the simultaneous lack of flexibility and structure the tools offer, yet they work around their frustration by building additional extensions as needed.

Third, In a pattern identified by Grudin [14], the group that is doing the work is different from the group that is deriving the benefits. MBSE injects a new workflow and a new kind of process of systems engineering for all SEs. The core group is asking the broader community to adopt and use MBSE practices, either in addition to or instead of their current ones, thereby changing their workflow. In fact, nobody is seeing the benefits of the professional-looking documents with complete, referenced, and reviewed descriptions that traditional SEs and especially those in the broader community associate with prideful good work. The lack of benefit put undue burden on the broader group, and sets MBSE adoption up for unintended failure.

Thus, there is a miscommunication. The core group believes that the broader community is rejecting MBSE, when in fact, the broader community is ready to embrace MBSE but is distrustful. Considering the criticality of MBSE acceptance, we will use this our findings and results to create a technosocial intervention. We believe if we can remove the barriers to acceptance, we can spur forward cultural change. We now present the design hypothesis and artifact we developed to achieve that end.

A Design Intervention

By focusing on modeling as diagram authoring punctuated by occasional descriptions, the movement to MBSE is capturing the digital artifacts, but not capturing the decisions, rationale, trade-offs, and story underlying the design. Thus, the broader community is hesitant to invest time in a practice that fails to recognize their efforts.

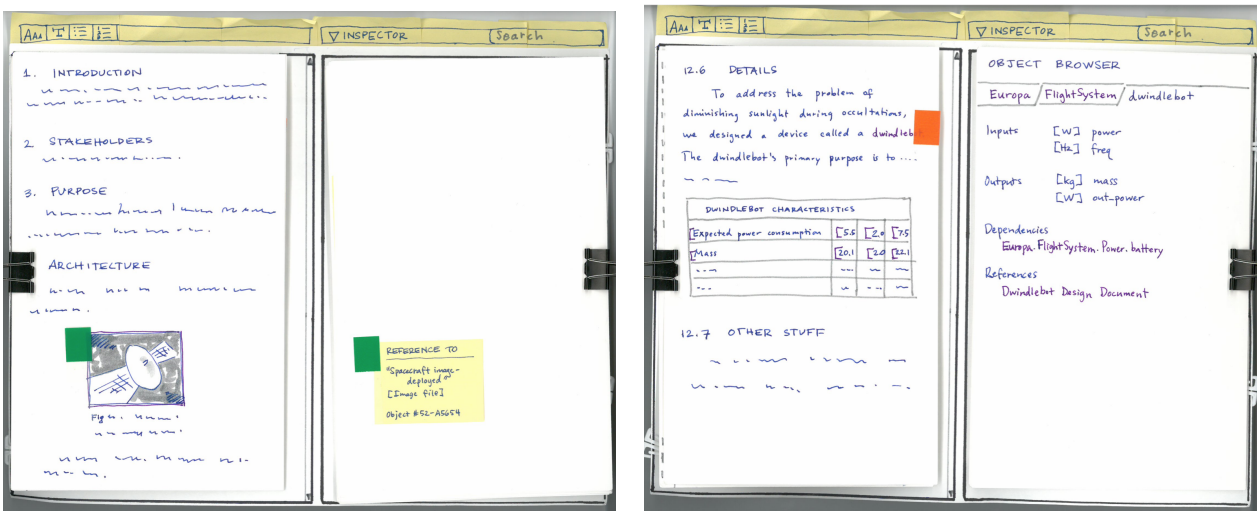
Figure 5. Pilot participant interacting with the low-fidelity prototype, a narrative-first design inspired by the paper memo, with rich object data

We believe the tools would not have the same adoption challenges if they focused on narrative first, and used an underlying data model to allow rich diagram authoring, tying the elements of the narrative and diagrams into the model. We look to the original capabilities of the memo as a design metaphor for a new kind of tool that we believe will be accepted by the broader community. However, rather than adhering to the linearity, unsearchability, and brittle nature of paper documents, the capabilities of today's technology can provide the tools needed to create a new way to communicate with the process of systems engineering.

More than simply a tool, the broader community requires a practice, incorporating the technology with process and cultural change. We recognize the technosocial implications of a paradigm shift, especially in light of the field's rocky history, and the cultural stand-off which has been in place for the duration of the rise of MBSE.

We believe following the cultural place of the traditional memo and add current technological abilities such as relational object management for explicit data references, allowing explicit links to solid data rather than implicit references through narrative; document control, branching, and explicit document versioning, with the top of the document a living, dynamic, editable, and collaborative endeavor and the underlying versions as static artifacts; and provenance at the data level, exposing where the document and all of its elements came from, including history, ownership, permissions, authority, and rationale to create a new software artifact that we believe will result in acceptance by the broader community.

Our vision for the future of MBSE and model-based design is **a system of communication-driven design using a new kind of memo that is a tracked living document, providing data continuity and seamlessly capturing and exposing explicit relationships, provenance, and review.**



(*a*) (*b*)

Figure 6. Low-fidelity prototype, with authoring pane on the left and model view on the right. (*a*) The image is a model element; model view pane shows its element reference. (*b*) Text in narrative (authoring pane) is a reference to a model reference, which is seen in the object browser.

We created a low-fidelity prototype (Figure 5, Figure 6) reminiscent of a collaborative authoring environment yet showcasing a rich data layer, and making explicit relationships between text and source data and cross-wise between data. In order to simplify the low-fidelity prototyping process, we made assumptions about versioning, revision history, and notifications of changes. We tested the low-fidelity prototype with a subset of the participants for qualitative feedback, and proceeded with designing and building the system when we saw that our understanding of the domain was thorough. Initial reactions to the solution from both the core group and the broader community have been positive, yet as the prototype is in its early stages, we cannot provide a controlled study at this time.

CONCLUSION

In this paper, we investigated obstacles to the adoption of Model Based Systems Engineering at the NASA Jet Propulsion Laboratory. Over six months, we conducted an ethnographic field study with 19 SEs, 18 of whom were actively working on the early stages of a mission using model-based techniques for system specification.

Our investigation points towards a cultural misunderstanding. Because of their concern that the broader community was rejecting modeling on principle, the core community of modeling early adopters were providing tools that focused on making models. The broader community was rejecting not modeling, but the tools that were provided: the tools generated models but none of the critical details on rationale and tradeoff that accompany the documentation.

Given the significant consequences of organizational failure, we were compelled to present this information to the broader community. We chose to create a design intervention, and asserted that a tool that could rejuvenate MBSE tooling with a rich design rationale and accompanying narrative might be able to provide the necessary catalyst to spur organizational adoption.

Our investigation pointed to the memo as the artifact that historically captured the design rationale. The historical use of the memo primed the workplace culture to be available to a software tool that looked, acted and felt like the memos of old. By harnessing these social affordances of the old paper memo, we offered the JPL community a tool that brings the narrative to the foreground and provides explicit references to the underlying data, exposing the model and encouraging model-based practices.

While preliminary results are positive, much study remains. Though deep, our investigation reports only a small window onto a significant transition underway in the engineering workforce worldwide. We expect a rich comparison between effects found at multiple work sites will provide even more insight into the causes of this cultural standoff.

REFERENCES

1. A. Ahmed, J. Alexander, D. Boussalis, and et al. *Cassini Control Analysis Book*. Jet Propulsion Laboratory, 1998.
2. T. Bayer, S. Chung, B. Cole, B. Cooke, F. Dekens, C. Delp, I. Gontijo, K. Lewis, M. Moshir, R. Rasmussen, et al. Early formulation model-centric engineering on NASA's Europa mission concept study. In *INCOSE International Symposium*, 2012.
3. T. J. Bayer, M. Bennett, C. L. Delp, D. Dvorak, J. S. Jenkins, and S. Mandutianu. Update-concept of operations for integrated model-centric

engineering at JPL. In *Aerospace Conference, 2011 IEEE*, pages 1–15. IEEE, 2011.

4. D. Bindschadler, C. L. Delp, and M. McCullar. Principles to Products: Toward Realizing MOS 2.0. In *Proc. Space Operations*, 2012.
5. E. Bradner, W. A. Kellogg, and T. Erickson. The adoption and use of 'Babble': A field study of chat in the workplace. In *ECSCW'99*, pages 139–158. Springer, 1999.
6. A. Collins, D. Joseph, and K. Bielaczyc. Design research: Theoretical and methodological issues. *Journal of the learning sciences*, 13(1):15–42, 2004.
7. S. Davidoff, M. K. Lee, A. K. Dey, and J. Zimmerman. Rapidly exploring application design through speed dating. In *UbiComp 2007: Ubiquitous Computing*, pages 429–446. Springer, 2007.
8. C. Delp, D. Lam, E. Fosse, and C.-Y. Lee. Model based document and report generation for systems engineering. *Jet Propulsion Laboratory, National Aeronautics and Space Administration*, 2013.
9. J. A. Estefan et al. Survey of model-based systems engineering (MBSE) methodologies. *Incose MBSE Focus Group*, 25, 2007.
10. E. Fosse and C. Delp. Systems engineering interfaces: A model based approach. *Jet Propulsion Laboratory, National Aeronautics and Space Administration*, 2013.
11. S. Friedenthal, R. Griego, and M. Sampson. INCOSE Model Based Systems Engineering (MBSE) Initiative. *INCOSE MBSE Track*, 2007.
12. S. Friedenthal, A. Moore, and R. Steiner. *A practical guide to SysML: the systems modeling language*. Access Online via Elsevier, 2011.
13. W. W. Gaver. Technology affordances. In *Proc. CHI*, pages 79–84. ACM, 1991.
14. J. Grudin. Why CSCW applications fail: problems in the design and evaluationof organizational interfaces. In *Proc. CSCW*, pages 85–93. ACM, 1988.
15. C. Hill, C. Corbett, and A. St Rose. *Why So Few? Women in Science, Technology, Engineering, and Mathematics.* ERIC, 2010.
16. E. Hornecker. *Tangible user interfaces als kooperationsunterstützendes Medium.* PhD thesis, University of Bremen, 2004.
17. E. Hornecker and J. Buur. Getting a grip on tangible interaction: a framework on physical space and social interaction. In *Proc. CHI*, pages 437–446. ACM, 2006.
18. T. P. Hughes. *Rescuing Prometheus: Four monumental projects that changed our world.* Random House Digital, Inc., 2011.
19. A. J. Lopes, R. Lezama, and R. Pineda. Model based systems engineering for smart grids as systems of systems. *Procedia Computer Science*, 6:441–450, 2011.
20. B. W. Mar and B. G. Morais. Frat–a basic framework for systems engineering. In *Proc. ICSE*, 2002.
21. D. A. Mindell. *Digital Apollo*. MIT Press Cambridge, MA, 2008.
22. D. A. Norman. *The Psychology of Everyday Things*. Basic Books, 1988.
23. C. Paredis, Y. Bernard, R. M. Burkhart, H.-P. de Koning, S. Friedenthal, P. Fritzson, N. F. Rouquette, and W. Schamai. An overview of the SysML-Modelica transformation specification. In *2010 INCOSE International Symposium*, 2010.
24. D. Robey and M.-C. Boudreau. Accounting for the contradictory organizational consequences of information technology: Theoretical directions and methodological implications. *Information systems research*, 10(2):167–185, 1999.
25. E. H. Schein. Culture: The missing concept in organization studies. *Administrative science quarterly*, pages 229–240, 1996.
26. A. J. Sellen and R. Harper. *The Myth of the Paperless Office*. The MIT Press, 2003.
27. S. C. Spangelo, D. Kaslow, C. Delp, B. Cole, L. Anderson, E. Fosse, B. S. Gilbert, L. Hartman, T. Kahn, and J. Cutler. Applying model based systems engineering (MBSE) to a standard cubesat. In *Aerospace Conference*, pages 1–20. IEEE, 2012.
28. J. P. Spradley. *The Ethnographic Interview*. Harcourt, Brace, Janovich, 1979.
29. T. M. Wheeler and M. D. Brooks. Experiences in applying architecture-centric model based system engineering to large-scale, distributed, real-time systems. *The Mitre Corporation Technical Papers Case*, pages 07–0838, 2007.
30. N. Wiener. *Cybernetics; or control and communication in the animal and the machine.* John Wiley, 1948.
31. W. E. Woodson, B. Tillman, and P. Tillman. *Human factors design handbook: Information and guidelines for the design of systems, facilities, equipment, and products for human use.* 1992.