Breaking the Fidelity Barrier:

An Examination of our Current Characterization of Prototypes and an Example of a Mixed-Fidelity Success.

Michael McCurdy, Christopher Connors[1], Guy Pyrzak[1], Bob Kanefsky[1], and Alonso Vera[1]
NASA Ames Research Center
M/S 262-4
Moffett Field, CA 94035
Michael.McCurdy@nasa.gov

ABSTRACT

This paper presents a summary of the space of commonly-used HCI prototyping methods (low-fidelity to high-fidelity) and asserts that with a better understanding of this space, HCI practitioners will be better equipped to direct scarce prototyping resources toward an effort likely to yield specific results. It presents a set of five dimensions along which prototypes can be planned and characterized. The paper then describes an analysis of this space performed by members of the NASA Ames Human-Computer Interaction Group when considering prototyping approaches for a new set of tools for Mars mission planning and scheduling tools. A description is presented of a prototype that demonstrates design solutions that would have been particularly difficult to test given conventional low- or mid- fidelity prototyping methods. The prototype created was "mixed-fidelity," that is, high-fidelity on some dimensions and low-fidelity on others. The prototype is compared to a preexisting tool being redesigned and to a tool that has been developed using the prototype. Experimental data are presented that show the prototype to be a good predictor of eventual user performance with the final application. Given the relative cost of developing prototypes, it is critical to better characterize the space of fidelity in order to more precisely allocate design and development resources.

Author Keywords
Prototyping, fidelity, planning, methods.

ACM Classification Keywords
H.5.2. Information interfaces and presentation (e.g., HCI): Prototyping.

INTRODUCTION

Prototyping methods are widely recognized as an important component of the HCI process. When correctly applied, the ability of a prototype to identify and correct potential problems early in the development cycle can pay for the cost of the prototype many times over [2].

The current range of prototyping methodologies are generally described within a spectrum of fidelity with low-fidelity, ostensibly low-cost methods such as whiteboard or paper sketches at one end, and highly developed, highly interactive artifacts (often developed as interactive web-based applications or in high level programming environments such as Visual Basic) at the other. Many examples of "medium-" or "mid-fidelity" prototypes also exist, but it is important to distinguish between these characterizations and the "mixed-fidelity" concept described here. The term "mid-fidelity" is often used to describe prototypes which are neither low- nor high-fidelity and therefore lie somewhere in the middle along that axis. "Mixed-fidelity" refers to a prototype which is high fidelity in some respects and low fidelity in others.

Low-fidelity methods have received a great deal of recognition in the field for their ability to validate designs and predict large problems at an extremely low cost [2,13,16,18,19]. Indeed it may seem counterintuitive, especially to software engineers and developers, that a paper sketch could provide such valuable insight, but several years of experience and sound methodology has validated the approach.

High-fidelity methods, on the other hand, have received passing recognition for their ability to convince management or other stakeholders that due diligence has been given to the product design and that the real thing is

[1] Additional author affiliation and contact information can be found at the end of this paper.

indeed on its way [3,16,19]. However, beyond this grudging recognition, high-resolution methods are often dismissed as being too cumbersome, too expensive to develop, or too likely to set unrealistic expectations in the minds of users and customers [3,11,16,19].

This paper will argue first that this single low-fidelity/high-fidelity continuum is not sufficient to capture the variety of prototyping approaches in use today. In particular, it will describe a tradeoff analysis made by the HCI Group and software developers at NASA Ames that led to the development of a prototype that is difficult to characterize using these traditional means. This "mixed-fidelity" prototype is high-fidelity in some ways but low-fidelity in others. Finally, some experimental data will be presented from the NASA case study that show the mixed-fidelity prototype to be a good predictor of eventual user performance on data-rich and highly interactive tasks using the fully-developed application.

THE SPACE OF PROTOTYPES IN HCI

This section explains why the concept of "low-" vs. "high-fidelity" is insufficient to cover the space of possible prototypes. It then presents a framework for characterizing prototypes along five dimensions: Level of Visual Refinement, Breadth of Functionality, Depth of Functionality, Richness of Interactivity, and Richness of Data Model. Also, some examples are presented of prototypes produced via common methods that are of "mixed" fidelity – that is, they are high-fidelity along some dimensions and low-fidelity along others.

The Fidelity Barrier

Although the terms "low-fidelity" and "high-fidelity" are often used to characterize different prototyping approaches, the concept of "fidelity" has a tendency to conflate several orthogonal aspects of the artifact. For example, it is unclear whether "fidelity" refers primarily to level of functionality, level of visual polish, or level of interactivity (among others). This distinction is especially difficult to make when an artifact is particularly well developed in one area but not in others. For example, it is easy to characterize a hand-drawn storyboard as "low-fidelity" and a fully interactive, fully-conceived, running artifact as "high-fidelity." However, consider an artifact that consists of only one or two non-interactive screens but is visually accurate to the pixel. Or consider a prototype, such as the one later described here, that uses the same input data and similar back-end logic as the delivered application, but whose visual look-and-feel is intentionally kept "low resolution." In each of these cases it would be difficult to apply conventional low- or high-fidelity classification.

Early attempts at further characterization in the literature tended to rely on a post-hoc example-driven approach. For example, in 1984 Floyd [6] recognized the need to understand the range of prototypes in practice and their relative utility, pointed out several existing examples, and attempted to characterize them in terms of their expected utility. However, by the author's own admission, the possible space of options was limited by the technology of the day. For example, lack of abstract GUI programming languages limited the ability to rapidly iterate in that area.

This oversimplification of the prototype space, while convenient for some applications, has two effects. First, it makes it more difficult to choose a prototyping approach, and second, once an approach is chosen, it makes the application of methods more difficult. For example, when deciding what kind of prototype to build, it is reasonable to consider the end goals. John and Salvucci [10] identify three potential high-level goals for prototypes: to help sell software, to collect usability data via user testing, and to feed cognitive models for expert performance prediction. It would not be possible to decide which kind of prototype to build using a single axis: fidelity. Instead, they identify several aspects of a prototype required for each eventual use. For example, they assert that "best-guess visuals" are required for user testing, but not for feeding to a cognitive model.

Virzi et al [19] recognize that prototypes can vary along several orthogonal dimensions, including some of those listed here, but then revert to the low- to high-fidelity characterization in making the post-hoc argument for low-fidelity late in the design process. Bryan-Kinns and Hamilton in [4] and Houde and Hill in [9] and Hall in [8] also recognize several orthogonal dimensions, but they focus on eventual use of the prototype (e.g. when in the development stage it is to be employed, and who the target audience is) in order to form a more precise characterization of the artifact. In particular, Houde and Hill consider a broader definition of a prototype to include one used to validate an implementation concept or a products role in a larger work practice. While this type of characterization is useful for determining the role of prototyping in a product development cycle, and indeed these analyses make some assertions about the fidelity required to support particular uses, they do not appear to offer the HCI practitioner any more insight into the concept.

The authors of this paper faced a similar dilemma when deciding how to construct early prototypes of a software tool for planning Mars rover activity. Given the end goals for the prototype: identifying problems in presentation and interaction with a complex set of data, and evaluating efficiency gains from an interactive plan visualization, it was difficult to determine a target "fidelity." There were certain aspects of the tool that needed to be extremely high fidelity, such as the richness of the data and the underlying application logic, but there were others that did not, such as the aesthetics. As a result, the designers derived a system for first characterizing different prototypes and then selecting from those characterizations based on the eventual goals for the prototype.

The Five Dimensions

The authors identified five dimensions along which a prototype can be characterized. Each dimension has a "low-fidelity" and a "high-fidelity" equivalent, but importantly they can be manipulated independently. The following are descriptions of the five dimensions considered by the authors as well as some common wisdom and rules of thumb for each:

- **Level of Visual Refinement:** How refined should the prototype be from a visual standpoint? Artifacts on the low end of this scale include hand-drawn sketches and box-and-line wireframes [13,16]. Artifacts on the high end include fully resolved, pixel-accurate mockups as described in [16]. A high level of aesthetic refinement is not always desirable: when user tested, prototypes that are highly refined tend to elicit more commentary on visual attributes [11,13]. This is often desirable late in the design cycle, but it is important to address higher-level issues early.
- **Breadth of Functionality:** How broadly is the functionality represented within the prototype? [14] For example, if one were to develop a prototype for a banking kiosk, a broadly functional prototype would include approximations for most of the various functionality (withdrawals, deposits, balance checking, bill paying, etc.) requirements. A broadly functional prototype gives users a better understanding of the range of capabilities [16] that the interface will ultimately provide, and offers the opportunity to challenge system-wide issues (such as navigation) utilizing methods such as Heuristic Evaluation.
- **Depth of Functionality:** To what level of detail is any one feature or sequence represented? [14] Again considering the banking kiosk, one could imagine having a single path through the interface - a withdrawal - modeled in the prototype all the way though to its conclusion. Having a task modeled to its conclusion allows designers to interrogate the interface's capabilities with task-centric user evaluations like think-aloud studies and cognitive walkthrough.
- **Richness of Interactivity:** How are the interactive elements (transitions, system responses to user inputs, etc.) captured and represented to the user by the prototype? Paper prototypes and sketches have traditionally represented the lowest fidelity in terms of interactivity, although efforts such as SILK [11] and DENIM [12] have been explicitly designed to increase the interactive richness of hand drawn interfaces. Higher levels of interactivity have historically come at the cost of development expense, time, and inflexibility.
- **Richness of Data Model:** How representative of the actual domain data is the data employed by the prototype? For example, if a design team wanted to develop a prototype for a television program listing service, will the prototype utilize a small set of imaginary channels and programs, or will it utilize an actual channel lineup of potentially hundreds of channels and programs? The former may be expedient, but might not provide a good example of the scale of the data space the user will eventually have to manage through the interface. For example, it is common for designers to overlook the possibility that a half-hour program may have a long title that will not fit without being truncated, whereas a large set of actual data will quickly reveal this case.

Prototypes can be designed and implemented to low or high fidelity on any of these five dimensions depending on the type of data designers hope to gather. By using these dimensions to inform prototype development, and recognizing that each is fully independent and can be manipulated separately, it is possible to create mixed-fidelity prototypes that more precisely apply prototyping resources in support of specific end goals.

Examples

Paper or Wizard of Oz prototypes are often comprised of hand-rendered drawings on sheets of paper and generally have low levels of visual refinement. It is possible that they may be broad or deep, but somewhat uncommon that they be both. They tend to have a very low richness of interactivity, and low richness of data model. Because of these attributes, it is very difficult to use them for timed tasks, to test interactive features within an interface, or to represent the scale of the actual data space of the domain.

PowerPoint or HTML prototypes are often slideshows of screens, sometimes linked together using hotspots to simulate interactivity. These artifacts can range from low to high in visual refinement, are likely to be either broad or deep, have low to medium richness of interactivity, and low to high (with considerable effort) richness of data. These prototypes are nearly as facile as paper prototypes in terms of rapid generation and modification, and offer at least some interactivity, allowing users to get some sense of flow through the artifact.

Prototypes rendered in Flash or static HTML can range from low to high in terms of visual resolution, low to high in terms of both breadth and depth (although again, time constraints usually dictate that only one of the two be high). They can range from low to high in both richness of interactivity and richness of data. Prototypes built using these technologies often require higher levels of effort and expertise, but these artifacts can be designed to leverage any of the five dimensions depending on the goals of the implementers.

In general, recent advances in prototyping tools have made it increasingly easy to create "mixed-fidelity" prototypes that are high fidelity on some of these dimensions and low fidelity on others. These include not only specialized toolkits such as SILK and DENIM but also commercial off-the-shelf software such as Macromedia Flash, visual programming environments such as Visual Basic and Real Basic, and evolving web standards ranging from

DHTML/DOM, CSS and JavaScript to recent techniques such as AJAX. [7]

THE SPIFE PROTOTYPE: A MIXED-FIDELITY CASE STUDY

This section describes three artifacts involved in the case study. All three artifacts are activity-planning tools for Mars surface operations. The three artifacts are, the current tool (MAPGEN), the prototype (SPIFe), and the future tool under development (Ensemble).

First, we will discuss some key usability problems with MAPGEN observed during the Mars Exploration Rover missions in 2004. Although there are many aspects to the redesign, this paper focuses on a specific set of problems – plan inspection inefficiencies – the solutions for which would have been difficult to test using traditional "low-fidelity" prototyping methods.

Next, the HCI Group's analysis in preparation for building the SPIFe prototype is discussed in light of the five dimensions outlined above, along with the technical approach to building the artifact. We will also introduce Ensemble, the future tool under development and discuss how it differs from the SPIFe prototype with respect to fidelity and development costs.

Finally, experimental data is presented that supports the theory that the SPIFe mixed-fidelity prototype is a good predictor for Ensemble in terms of user performance as compared to the current tasks in MAPGEN. We also discuss how the mixed-fidelity concept enabled the team to collect the type of data required at a minimal cost.

The "Before" Case: MAPGEN and Motivations for Redesign

During the Mars Exploration Rover (MER) mission, operations personnel used a planning tool called the Mixed-initiative[2] Activity Plan GENerator, or MAPGEN, to schedule rover activity (see Figure 1). The fundamental task of planning consists of selecting activities requested by mission scientists (e.g. "capture an image of rock *x*" or "drive to location *y*") and assigning them execution times. This assignment requires users to pay close attention to rover resources and can get orders of magnitude more complex for every activity added to the plan. On average, an activity plan contains 40 activities (n=775). MAPGEN displays these activities on "legends" according to the subsystem that will execute the activity. For example, in Figure 1, the legend labeled "Navcam" represents images to be acquired by the Navigation Camera (or Navcam). A single person operated MAPGEN: the Tactical Activity Planner (TAP). The time allotted for generating a schedule using MAPGEN was approximately three hours, during which time refinements to individual activities were also taking place. This time allocation was seen as being very ambitious – non-surface missions typically plan spacecraft activity on a timeline measured in weeks or years. As such, activity planning time was a scarce resource and so became a key metric for tool performance.

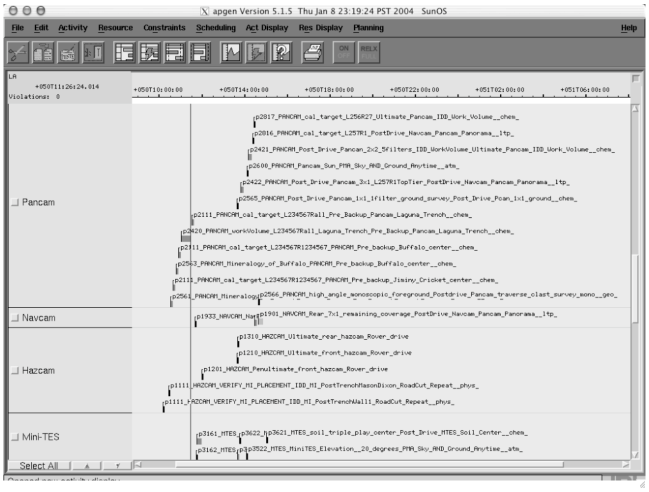

apgen Version 5.1.5 Thu Jan 8 23:19:24 PST 2004 SunOS
File
Edit
Activity
Resource
Constraints
Scheduling
Act Display
Res Display
Planning
Help
ON OFF
RELX FULL
LA
+050T11:26:24.014
Violations: 0
+050T10:00:00
+050T14:00:00
+050T18:00:00
+050T22:00:00
+051T02:00:00
+051T06:00:00
Pancam
p2817_PANCAM_cal_target_L256R27_Ultimate_Pancam_IDD_Work_Volume__chem_
p2816_PANCAM_cal_target_L257R1_PostDrive_Navcam_Pancam_Panorama__ltp_
p2421_PANCAM_Post_Drive_Pancan_2x2_5filters_IDD_WorkVolume_Ultimate_Pancam_IDD_Work_Volume__chem_
p2600_PANCAM_Pancam_Sun_PMA_Sky_AND_Ground_Anytime__atm_
p2422_PANCAM_Post_Drive_Pancam_3x1_L257R1TopTier_PostDrive_Navcam_Pancam_Panorama__ltp_
p2565_PANCAM_Post_Drive_Pancam_1x1_1filter_ground_survey_Post_Drive_Pcan_1x1_ground__chem_
p2111_PANCAM_cal_target_L234567Rall_Pre_Backup_Pancam_Laguna_Trench__chem_
p2420_PANCAM_workVolume_L234567Rall_Laguna_Trench_Pre_Backup_Pancam_Laguna_Trench__chem_
p2111_PANCAM_cal_target_L234567R1234567_PANCAM_Pre_backup_Buffalo_center__chem_
p2563_PANCAM_Mineralogy_of_Buffalo_PANCAM_Pre_backup_Buffalo_center__chem_
p2111_PANCAM_cal_target_L234567R1234567_PANCAM_Pre_backup_Jiminy_Cricket_center__chem_
p2561_PANCAM_Mineralogy
p2566_PANCAM_high_angle_monoscopic_foreground_Postdrive_Pancam_traverse_clast_survey_mono__geo_
Navcam
p1933_NAVCAM_Na
p1901_NAVCAM_Rear_7x1_remaining_coverage_PostDrive_Navcam_Pancam_Panorama__ltp_
Hazcam
p1310_HAZCAM_Ultimate_rear_hazcam_Rover_drive
p1210_HAZCAM_Ultimate_front_hazcam_Rover_drive
p1201_HAZCAM_Penultimate_front_hazcam_Rover_drive
p1111_HAZCAM_VERIFY_MI_PLACEMENT_IDD_MI_PostTrenchMasonDixon_RoadCut_Repeat__phys_
p1111_HAZCAM_VERIFY_MI_PLACEMENT_IDD_MI_PostTrenchWall1_RoadCut_Repeat__phys_
Mini-TES
p3161_MTES
p3622_M
p3621_MTES_soil_triple_play_center_Post_Drive_MTES_Soil_Center__chem_
p3162_MTES
p3
p3522_MTES_MiniTES_Elevation__20_degrees_PMA_Sky_AND_Ground_Anytime__atm_
Select All

Figure 1 : The MAPGEN Activity Planning Tool

In addition to actually assembling the day's schedule, the TAP was also required at several points to explain his progress or finished plan to interested parties. This explanation sometimes took the form of a plan walkthrough, but it often consisted of answering direct questions about the plan such as *"what's happening at 13:00?"* or *"how big is the gap between activity x and activity y?"*

Members of the HCI Group, including two of the authors of this paper, were staffed in the role of MAPGEN support during the MER primary mission (90 days starting in January of 2004). In this capacity, they were able to closely observe the use of and work practice surrounding the MAPGEN tool. Although many observations were clearly positive – the team and tools far exceeded expectations in never once failing to generate a valid command load by their deadline – the HCI Group also observed some key opportunities for improvement. Among those observations was a set of issues pertaining to efficiency when interrogating the plan in order to answer questions like those posed above.

TAPs used various strategies to questions while planning and explaining the plan to interested parties. For example, to answer the question ***"how big is the gap between activity x and activity y?"*** – TAPs used two strategies, a quicker but less accurate strategy, and a more time consuming but extremely accurate strategy.

[2] The "Mixed-initiative" part of the name refers to the integration of user-directed planning with Ames's automated planner [1]. Improving the new mixed-initiative features has been one of the goals of the SPIFe prototype, but is beyond the scope of the experiment discussed in this paper.

The quicker but less accurate strategy involved placing the mouse at the end of activity *x* and noting the approximate time as listed at the top of the screen, then placing the mouse at the beginning of activity *y* and noting the time, then (mentally) subtracting the first time from the second. MAPGEN included a constantly updated display of the approximate time pointed to by the mouse cursor. In order to aid visibility and help co-register activities near the top and bottom of the display, MAPGEN also had the ability to draw a "Vertical Cursor" line (see Figure 1). So if a user were content with an approximate time (accurate to within about two minutes, depending on zoom factor) he could simply point at a location on the timeline (the end of activity x, for example) and note the approximate time.

The accurate strategy involved invoking an "editor" dialog to obtain the detailed start time and duration details about each activity and then (again mentally) adding to determine the end time of activity *x* and subtracting that time from the start time of activity *y*.

It was not uncommon for plan interrogation tasks to require significant time and mental resources and attention to complete. For example, using the accurate strategy, the user was required to perform at least five mouse clicks, complete several mental math operations, and store and retrieve several chunks of data in working memory. Even using the quick method, the task requires the user to perform mental math and accept a somewhat large margin of error.

MAPGEN presented one additional impediment to answering the gap question. Each activity bar on the timeline was annotated with a label that contained the name of the activity. On MER, these activity names often grew to be over 100 characters long. This led to considerable visual clutter and in particular consumed a great deal of vertical real estate. Plans in MAPGEN were often many screens "tall" requiring, a user to "page down" multiple times in order to see all the categories. This property made it difficult to determine the very *existence* of a gap between activities, let alone its duration. In order to be confident that an apparent gap was not in fact caused by an activity hiding off-screen, users would scroll up and down visually searching for activities occurring at a particular time. It was not uncommon for a user to place a finger on the screen while scrolling in order to keep his place. This visual search was time consuming at best and often led to errors.

The example chosen here – ***"how big is the gap…"*** – is just one instance of a commonly asked question during operations. This question was asked both by personnel external to the TAP, as well as by the TAP himself over the course of planning. Anecdotally, the process described above took tens of seconds to complete, sometimes taking up to a minute to derive a precise answer. This observation is supported by data collected in the course of the experiment described later: answering this question took 43.3 seconds on average in the MAPGEN case. Considering the fact that this question, and questions like it, were asked many times per day, this amounted to a significant time investment on the part of the TAP. Taking together this set of plan inspection questions, it is not inconceivable that reducing time to answer to a few seconds could save 30 minutes to an hour over the course of planning.

The SPIFe Prototype: Conception

The HCI Group identified goals that a prototype would achieve. First, it would be necessary to convince mission managers at JPL that the project had merit and deserved mission funding. To convince the managers of this the team had to show the new design would cause a significant reduction in the time needed during plan inspection. After all, a proposed tool would at least partially replace one tried-and-true tool in an organization that tends to value heritage. The prototype would also be necessary for testing concepts with actual mission personnel, for exploring a variety of design solutions, and as a tool to educate potential users and stakeholders about the importance of activity planning.

In order to accomplish these goals and to make it possible to collect the performance data, the HCI Group used the five dimensions to provide insight into the requirements for a mixed-resolution prototype that would be most effective. The group wanted answers as quickly and cheaply as possible to questions like "what is the impact of design changes on overall task performance, and accuracy?" The following is the result of the analysis for each dimension:

- **Level of Visual Refinement**. Since the primary objective of the prototype was to collect time accurate user performance data, the level of visual refinement was set low. While the interface had to be clear, visual aesthetics and widget consistency were not critical. Color and other visual aids were used only where essential for understanding of the plan inspection task.
- **Breadth of Functionality**. The prototype was designed to test a relatively small aspect of the interface – the visual feedback when inspecting a timeline display – so little emphasis was given to breadth of functionality.
- **Depth of Functionality**. Within the timeline component, some depth was required. Although some aspects of the timeline could be left unimplemented some functionality would need to be fully implemented in order to provide performance data related to plan inspection. A medium level of emphasis was given to depth of functionality.
- **Level of Interactivity**. It was critical that the interactivity of the tool be as close to the actual implementation as possible. Accurate evaluation of the new designs required the user to inspect many

aspects of the plan very quickly with minimal impediments to the task. It was important that performance of the plan inspection task was similar in the prototype as it would be in the final tool. A high level of emphasis was given to the level of interactivity.

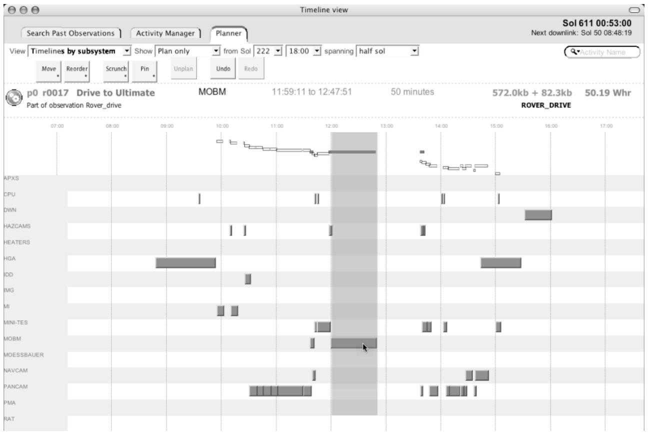
- **Depth of Data Model**. Because plan interrogation tasks are significantly easier when a simplified plan is used, getting accurate data from the prototype required it to scale to plans of realistic complexity. A typical MER plan has characteristics that present challenges to timeline layout. For example, it was common for plans to contain hundreds of activities, ranging from nearly instantaneous hazard-camera snapshots to twelve-hour spectrometry data collection. As such, it was important that realistic data was used in testing the prototype, and a high level of emphasis was given to the depth of the data model.

This tradeoff analysis places the desired prototype in a somewhat underrepresented segment – prototypes exhibiting high interactivity and data fidelity coupled with low visual refinement and fairly narrow functionality are not often discussed in the literature. However, there are many tasks that share some of the same characteristics as ours, such as project planning, scheduling of patients, and job shop scheduling. Prototypes in this segment are likely to be beneficial to designers of applications supporting those tasks.

The SPIFe Prototype: Redesign

Having identified several areas for improvement, the HCI Group is in the process of designing several mission tools from the ground up, including MAPGEN. Although the primary motivation for the redesign is to address many higher-level concerns such as consistency across tools or error-prone aspects of existing tools, smaller inefficiencies of the sort described above are also a driver. The prototype described in this paper focuses on some fairly specific solutions to the efficiency of plan interrogation.

One proposed redesign in support of this task introduces the concept of a "smart vertical cursor," which makes available a more rich set of information given a particular point of interest on the timeline. As the user sweeps the mouse over the timeline, the current "region of interest" is highlighted, and some selected metadata is displayed in a persistent status bar. For example, if a user mouses over the bar representation of an activity, the region of interest is set to the span of the activity. The entire region is highlighted, and some additional information about the activity is presented in the status bar (see). Some of the information is taken directly from the parameters on the activity (*Name*, *Type*, and *Priority*, for example). However, some of the information is calculated based on the activity's current position in the plan: the status bar presents the precise start time, end time, and duration of the moused-over activity.



Timeline view

Sol 611 00:53:00
Next downlink: Sol 50 08:48:19

View Timelines by subsystem Show Plan only from Sol 222 18:00 spanning half sol

p0 r0017 Drive to Ultimate MOBM 11:59:11 to 12:47:51 50 minutes 572.0kb + 82.3kb 50.19 Whr
Part of observation Rover_drive ROVER_DRIVE

Figure 2: The SPIFe Prototype Timeline

The new design also allows the user to mouse over other areas of the plan in order to define the region of interest. In direct response to the "*how big is the gap?*" scenario described above, if the user mouses over a gap between activities, the region of interest is set to the gap. The status bar again displays additional metadata, although now it pertains to the gap: the word "gap" appears, and the precise start time, end time, and duration of the gap are displayed. If the user mouses a blank area directly above or below two or more overlapping activities, the overlap duration is displayed and the overlapping activities are identified.

An additional proposed design change is the elimination of labels from the individual activity bars on the timeline. This makes it possible to fit an entire plan on a single screen, eliminating the vertical search required by MAPGEN. As the experimental data will show in the next section, this significantly reduces time to complete the task.

Using the new design, answering the question "*how big is the gap between activity x and activity y?*" should be as easy as locating the gap in question, placing the mouse pointer over it, and reading the duration information from the status bar.

These design solutions, although somewhat minor, are not without potential tradeoffs. It was somewhat unclear what impact the elimination of labels would have on various plan interrogation tasks. For example, it is ostensibly easier to locate gaps, but the elimination of labels from activity bars could have an effect on the efficiency of another plan interrogation task such as "*locate activity x.*" In that task, the user may need to sweep his or her mouse over a significant subset of the plan in order to answer the question. Answering the same question in MAPGEN, where the names are displayed at all times, might be expected to involve a more straightforward visual search. Data on time to complete this and other plan interrogation tasks using the new interface was desired.

The SPIFe Prototype: Implementation

It was decided at an early stage to use actual mission data – the richest data available – to populate this prototype in order to provide a realistic environment and scale. Also, since upcoming missions are largely staffed with people with experience on MER, populating the prototype with MER data tended to make it not only realistic but also familiar to user-test subjects and mission management. Admittedly, this decision was easy in this case; it was not a prohibitively costly approach. Due to their involvement with MER operations, the designers and prototype developer were fortunate in having access to this data set and an understanding of it. They were able to reuse plan-reading software from the Constraint Editor, an operational tool that is in use on MER.

The SPIFe prototype is web-based, and thus takes advantage of some of the interactive display capabilities available in Firefox and other browsers compliant with the DOM and CSS standards. This also simplifies installation for demos, since Firefox runs on three ubiquitous computer platforms and is widely installed. The client-side scripts are implemented in JavaScript. All of the HTML, and the variable data for JavaScript, are generated dynamically from actual MER data by a server-side Lisp program. This approach allows designers to choose from well over a thousand actual plans for a one-time programming cost. In contrast, with a handcrafted paper or PowerPoint prototype, the cost would be proportional to the depth of the data model. Maintaining state, such as the effect of moving activities around the time timeline, was tricky in a web application, but that problem was solved early on, and the prototype can be used for end-to-end tasks; that is, it provides unlimited depth of functionality at no further development cost, once again in contrast to a hand-crafted prototype. Due to reuse of existing code and the decision to focus on particular axes of fidelity, the SPIFe prototype was created by a single developer in approximately one month.

Although it would be possible to simulate some aspects of the interface using traditional low-fidelity means, for example using acetate overlays to indicate highlighting, the cost of building such a prototype to handle diverse scenarios would have been large. Additionally, due to physical limitations on how quickly such an prototype can be manipulated, it would not have been possible to collect the timing data desired.

The Final Application: Ensemble

While this prototyping effort was underway, tool developers at NASA Ames and JPL realized that a unified, component-based approach to software development could yield an integrated suite of ground based mission tools for deployment on upcoming Mars surface missions. Many of the design solutions tested as part of the SPIFe prototype have been incorporated in to the new application, called Ensemble (see Figure 3).

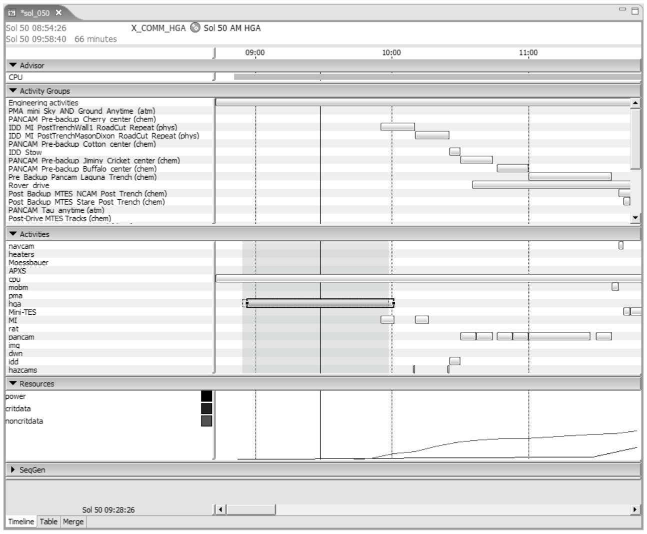

*sol_050
Sol 50 08:54:26
Sol 50 09:58:40 66 minutes
X_COMM_HGA
Sol 50 AM HGA
09:00
10:00
11:00
Advisor
CPU
Activity Groups
Engineering activities
PMA mini Sky AND Ground Anytime (atm)
PANCAM Pre-backup Cherry center (chem)
IDD MI PostTrenchWall1 RoadCut Repeat (phys)
IDD MI PostTrenchMasonDixon RoadCut Repeat (phys)
PANCAM Pre-backup Cotton center (chem)
IDD Stow
PANCAM Pre-backup Jiminy Cricket center (chem)
PANCAM Pre-backup Buffalo center (chem)
Pre Backup Pancam Laguna Trench (chem)
Rover drive
Post Backup MTES NCAM Post Trench (chem)
Post Backup MTES Stare Post Trench (chem)
PANCAM Tau anytime (atm)
Post-Drive MTES Tracks (chem)
Activities
navcam
heaters
Moessbauer
APXS
cpu
mobm
pma
hga
Mini-TES
MI
rat
pancam
imq
dwn
idd
hazcams
Resources
power
critdata
noncritdata
SeqGen
Sol 50 09:28:26
Timeline
Table
Merge

Figure 3: The Ensemble Timeline

Ensemble is being developed using the Java Eclipse development framework [5], which has allowed the team to rapidly realize concepts into working software components.

The Ensemble application is necessarily "high fidelity" in all five dimensions. It is the final product of the development process, and as such it sets a practical upper bound on each of the five dimensions. For example, the Eclipse framework provides a highly refined, platform appropriate look-and-feel, giving Ensemble a high level of visual refinement. Additionally, Ensemble developers have already incorporated many of the interactive behaviors from the SPIFe prototype, lending it a high level of interactivity.

Experiment Setup and Results

With three artifacts in hand – the MAPGEN baseline, the SPIFe prototype, and the Ensemble application, the authors were able to compare user performance across all three artifacts. Of primary concern to the designers was the effect of the design changes made in the SPIFe prototype, on the performance of the plan inspection task in light of current task performance in MAPGEN. Specifically would the proposed changes improve performance with the gap finding but reduce performance on other tasks? However, for the purposes of this paper, it is more interesting to examine how well the mixed-fidelity prototype is able to predict final application performance. The following is a description of the experiment followed by a brief discussion of the results in light of the mixed-fidelity concept.

Task

Using each of the interfaces in a randomly selected order, participants were instructed to complete a series of four tasks in the given order. The tasks were selected based their similarity to common plan inspection tasks observed during mission operations. The four tasks were:

1. Find a gap larger then 5 minutes long in the given time interval, e.g. *For Spirit Sol 50, find one gap*

more than five minutes long between times 12:00:00 and 14:00:00, near 13:00:00. Determine its start time and end time.

2. Find the start and duration of the activity by the following name, e.g. *For Spirit Sol 50, where is Activity Mineralogy of Jiminy_cricket of Type PANCAM? Determine its start time and duration*
3. List the activities happening at the given time, e.g. *For Spirit Sol 50, name the activities that are happening at time 14:45:00*.
4. Find the longest activity in the given time interval,e.g. *For Spirit Sol 52, name the longest activity between time 08:00:00 and time 10:00:00.*

Each task consisted of 3 randomly selected questions to investigate a variety of times and sol plans. The participants were instructed to answer each question as quickly as possible while still maintaining a reasonable level of accuracy. These questions involved the mouse-based investigation of a timeline for the given interface. In total, each participant answered 36 questions, 12 per interface.

Procedure

We conducted a series of timed tests of 4 tasks-types consisting of 3 similar questions each. Each of the 4 tasks-types was run using 3 artifacts described in this paper, MAPGEN, SPIFe, and Ensemble. Participants were trained with each artifact and prior to the tasks. The participants consisted of 10 individuals with at least a college level education; both male and female of age ranges 24 to 40. Each participant investigated all 3 artifacts in a randomized order.

Results

After completing the user tests the time results of the interfaces were paired and analyzed with the Wilcoxon Signed Ranks Test with an $\alpha = 0.05$. The results of the tests showed a significant difference between SPIFe and MAPGEN as well as between Ensemble and MAPGEN in three of the four tasks-types. However, there was no significant difference with the Ensemble/SPIFe pairing in any of the task-types. This shows that, for the four task-types, there was no statistical difference between SPIFe and Ensemble and that in these cases SPIFe was an accurate predictor of user performance for the given tasks. Figure 4 shows the average time of the 10 participants for all 3 artifacts, grouped by the task-types.

Performance with the MAPGEN interface was significantly slower than that that with SPIFe or Ensemble in three of the four tasks. For one of the task-types (Determine the start time of an activity), there was no statistical difference between the three artifacts. Our expectation, as mentioned previously, was that SPIFe and Ensemble might actually do worse that MAPGEN for this particular task— after all, that task involved ask participants to give the name of the activity, information that SPIFe and Ensemble display only in response to a mouseover.

Based on the results from the user test and subsequent analysis, it was concluded, that for all tasks categories, SPIFe (prototype) and Ensemble (application) show no significant difference. This indicates that for all four tasks SPIFe (prototype) was an accurate predictor of the performance for Ensemble (application).

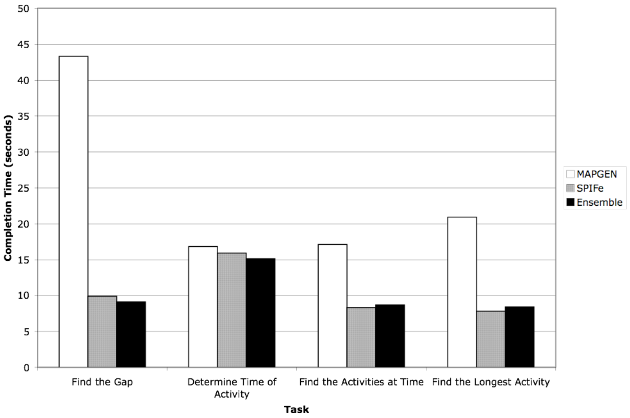


Figure 4: Comparison of Task Times by Task

The mixed-fidelity prototype has a few properties that made this experiment possible. First, since the level of interactivity was set high, the designers can be assured that the timing data is accurate to within a few milliseconds – plenty of precision given the multiple-second differences between the tools. Second, since the prototype uses a very high-fidelity data model – in fact it reads the exact same plan files that the other two tools do – it was possible to ask the same set of questions (randomized between subject) in all three conditions. Since the other dimensions of the prototype were left low-fidelity, it remains lightweight and easy to change based on ongoing experiment or user test data.

CONCLUSION

It is generally accepted that prototyping is a valuable tool for answering questions about proposed design changes without investing in the development of finished tools. However, existing methods of characterizing prototypes, generally along a spectrum of low- to high-fidelity, is too coarse to ensure that prototyping resources are well spent, and that the artifact will yield the desired data. Instead, it is useful to conceive of prototypes along five orthogonal axes: level of visual refinement, depth of functionality, breadth of functionality, level of interactivity, and depth of data model. Modern prototyping tools make it easier to manipulate these axes independently in order to create more targeted prototypes and avoid spending resources on areas that don't directly influence the type of data desired.

The SPIFe prototype for mission planning tools is one example of a mixed-fidelity prototype that incorporates a high-fidelity data model and level of interactivity, but a low level of visual refinement and fairly narrow functionality. By creating the prototype this way, it was possible to save resources on visual design and breadth of functionality and apply them to the areas most likely to yield data of interest to the designers. In particular, performance data was collected on several types of plan interrogation task. The experimental evidence shows that this prototype, while considerably less costly to develop than the final application, gives good predictions of eventual user performance. Further, the accuracy of this performance data relies on aspects of the prototype that would not have been possible to recreate using traditional low-resolution methods, such as realtime feedback on mouse over events.

The application of this characterization method more broadly would likely yield prototypes with the same general fidelity profile as SPIFe (that is, data rich, highly interactive, functionally deep but narrow, and visually unrefined) for tasks requiring complex reasoning about and interaction with large sets of data. Activity planning is one such application, but others might include spreadsheet manipulation, fault tree or failure data analysis, multivariate simulation and modeling tasks, and more.

Likewise, the application of this characterization method for tasks not requiring complex reasoning, or to serve different end goals, would yield mixed-fidelity prototypes of a different character. For example, highly visually refined and functionally broad prototypes can serve well for conducting marketing-oriented studies or guiding focus group discussion. Similarly, however, prototyping resources can be saved by deemphasizing richness of data or depth of functionality.

ACKNOWLEDGMENTS

The authors would like to thank our colleagues at JPL and Ames who made it possible to collect usage data, build the prototype, and test against real users. Funding for this work was provided by the Intelligent Systems program, the Collaborative Decision Systems program.

ADDITIONAL AUTHOR CONTACT INFORMATION

Christopher Connors, Apple Computer, Inc. 1 Infinite Loop, MS 10-EC, Cupertino, CA 95014. Chris.Connors@apple.com.

Guy Pyrzak, San Jose State University Foundation. NASA Ames Research Center, M/S 262-4, Moffett Field, CA 94035. gpyrzak@mail.arc.nasa.gov.

Bob Kanefsky, University of California, Santa Cruz. NASA Ames Research Center, M/S 269-2, Moffett Field, CA 94035. kanef@email.arc.nasa.gov.

Alonso Vera, Carnegie Mellon University. NASA Ames Research Center, M/S 262-4, Moffett Field, CA 94035. Alonso.Vera@nasa.gov.

REFERENCES

1. Ai-Chang, M., Bresina, J., Charest, L., Chase, A., Hsu, J., Jonsson, A., Kanefsky, B., Morris, P., Rajan, K., Yglesias, J., Chafin, G., Dias, W., and Maldague, P., "MAPGEN: Mixed-Initiative Planning and Scheduling for the Mars Exploration Rover Mission," IEEE Intelligent Systems, vol. 19, no. 1, pp. 8-12, January/February, 2004.
2. Bailey, B. P. , Konstan, J. A., Are informal tools better?: comparing DEMAIS, pencil and paper, and authorware for early multimedia design, Proceedings of the conference on Human factors in computing systems, April 05-10, 2003, Ft. Lauderdale, Florida, USA
3. Berghel, H., New Wave Prototyping: Use and Abuse of Vacuous Prototypes, interactions, v.1 n.2 p.49-54, Apr. 1994
4. Bryan-Kinns, N., Hamilton, F., One for all and all for one?: case studies of using prototypes in commercial projects, Proceedings of the second Nordic conference on Human-computer interaction, October 19-23, 2002, Aarhus, Denmark
5. Eclipse Technical Platform Overview http://eclipse.org/whitepapers/eclipse-overview.pdf
6. Floyd, Christiane. A Systematic Look at Prototyping. In: Budde, R., Kuhlenkamp, K., Mathiassen, Lars, Zullighoven, H. (Eds.) "Approaches to Prototyping." Springer Verlag. 1984.
7. Garrett, J. J. , Ajax: A New Approach to Web Applications. http://www.adaptivepath.com/publications/essays/archives/000385.php, 2005
8. Hall, R. Prototyping for usability of new technology. Int. J. Human-Computer Studies 55, 485-501. 2001.
9. Houde, S., and Hill, C. What do prototypes prototype? In Helander, M.G., Landauer, T.K. and Prabhu, P. (eds), Handbook of Human-Computer Interaction, 2nd edition. Amsterdam, The Netherlands: Elsevier Science, 367-381. 1997.
10. John, B.E., Salvucci, D.D. (in press) Multi Purpose Prototypes for Assessing User Interfaces in Pervasive Computing Systems. To appear in *Pervasive Computing*.
11. Landay, J.A., Myers, B.A. Sketching Interfaces: Toward More Human Interface Design, Computer, v.34 n.3, p.56-64, March 2001
12. Lin, J. , Newman, M. W. , Hong, J. I. , Landay, J. A. , DENIM: finding a tighter fit between tools and practice for Web site design, Proceedings of the SIGCHI conference on Human factors in computing systems, p.510-517, April 01-06, 2000, The Hague, The Netherlands
13. Maldague, P., Ko, A. Y., Page, D. N., and Starbird, T. W., APGEN: A Multi-Mission Semi-Automated

Planning Tool. 1st International Workshop on Planning and Scheduling for Space, Oct. 1997, Oxnard, California

14. Olsen, H., Balancing fidelity in prototyping: Choosing the right level of graphic detail, interactivity, breadth and depth. The Interaction Designer's Coffee Break, Issue 13 http://www.guuui.com/issues/03_05.php, 2005

15. Rettig, M. Prototyping for tiny fingers, Communications of the ACM, v.37 n.4, p.21-27, April 1994

16. Rudd, J., Stern, K., Isensee, S. Low vs. high-fidelity prototyping debate, interactions, v.3 n.1, p.76-85, Jan. 1996

17. Rudd, J., Isensee, S. Twenty-two tips for a happier, healthier prototype, interactions, v.1 n.1, p.35-40, Jan. 1994

18. Sefelin, R., Tscheligi, M., Giller, V., Paper prototyping - what is it good for?: a comparison of paper- and computer-based low-fidelity prototyping, CHI '03 extended abstracts on Human factors in computing systems, April 05-10, 2003, Ft. Lauderdale, Florida, USA

19. Virzi, R.A., Sokolov, J.L., Karis, D. Usability problem identification using both low- and high-fidelity prototypes, Proceedings of the SIGCHI conference on Human factors in computing systems: common ground, p.236-243, April 13-18, 1996, Vancouver, British Columbia, Canada