Machine Graphics and Vision, Vol. 12, No. 1, 2003

Diagrams and Design Stories

Sidney C. Bailin
Knowledge Evolution, Inc.

1 Introduction

Engineering design, and especially software design, could profit by drawing on narrative techniques for communicating and documenting ideas [1]. In this paper, we advocate the notion of a *design story* as a form of documentation with certain structural and semantic features that distinguish it from conventional documentation. Stories work with the reader's imagination to evoke a *story world*. Through identification, the reader engages with and becomes immersed in the story world. Identification, engagement, and immersion give stories their power and make them memorable and capable of being assimilated [2]. They therefore offer a way to structure design rationale in an especially effective manner.

The central hypothesis of this article is that a good design diagram tells a story. As a vague intuition, this is not surprising; but we will argue that one can make the idea more precise by drawing on narrative theory. The results of the analysis are precise criteria for evaluating the effectiveness of a design diagram as a vehicle of communication.

We begin by clarifying what we mean by *story*. This helps to clarify our central hypothesis. We then discuss our argument methodology, identifying the criteria we use to adduce evidence for our case (Section 2). The main body of the article is contained in Section 3, in which a diagrammatic design story is presented and analyzed. Each “chapter” of the design story is first presented, then analyzed in terms of its narrative structure, then in terms of diagrammatics. At the end, we take a look back and assess whether we have illustrated, if not proven, our central hypothesis.

1.1 What is a Story?

The term “story” is fashionable at the moment, and it is often used rather loosely. Frequently, it simply means “a communication that tells you about something that happened.” The past tense is not crucial; the story might refer to something happening in the present, as in a news story, or to something that may happen in the future. Also, factuality is not essential: stories can be fiction, but when they are, they are read *as if* they communicated facts.

Unfortunately, this notion of story applies to almost any conceivable communication other than purely phatic or expletive expressions. In particular, any design diagram can be understood as communicating the way in which an artifact was (or should be) built; thus, can it be considered a design story? If so, the notion of story contributes nothing to our understanding of design diagrammatics.

There is one respect, however, in which this overly broad notion of story can be useful. It reminds us that within any such communication, although possibly hidden, there lies a story—perhaps many stories. Teasing out the stories can clarify the communication and enrich our insight. This principle has been developed at length by Foucault [3] and other critical theorists [4].

In this article we adopt a more precise notion of story. In our view, it is a text that exhibits three salient properties: world, forward movement, and shape. *World* in a story corresponds to *context* in design. It comprises information that allows the reader to immerse herself in the story—to place herself within the story world—by identifying with one or more characters or, as an observer, with the narrator’s *voice*. The narrative concepts of world and voice are closely related to the ideas of perspective and view in visual design.

Forward movement is the causal structure that drives the story. The main character of a story has a goal, just as in design. To achieve the goals, he takes certain actions, which cause a change of state for better or worse, perhaps causing other actions by other characters in response, necessitating further actions by the main character. The characters do not know the outcome in advance, and for the most part, neither does the reader. The storyteller, by controlling the release of information, using techniques of mystery, suspense, and surprise, keeps the reader engaged [5]. The resulting patterns of rising and falling tension, teasing vs. gratification, a preponderance of questions vs. a preponderance of answers, give the story its *shape*.

1.2 What is a Design Story?

A design story is a design document—a structured presentation of a design and the rationales behind it—that uses narrative techniques to facilitate understanding and encourage reflection on the part of the reader. The story records both the *design* process and the envisioned *use* process, and the interweaving and mutual influence of these two threads. The story, by virtue of the salient characteristics identified in Section 1.1, enables the reader to understand better the relationships between steps in the process, and to see how decisions made early in the process relate to subsequent decisions. Narrative techniques help to draw the reader into the story, so that he better appreciates the goals and challenges of the designers.

1.3 Diagrams and Stories

Story worlds are related to visualization. The evocation of a story world involves visual imagination as a major component, perhaps because of the primacy of visual processing in our cognitive makeup. One criterion for whether a design story is successful is whether the reader is left with an imagined picture of the story world. Good stories, in short, suggest pictures.

In this article, we argue the converse—that a good design picture tells a story. What does it mean for a diagram to tell a story? A design diagram typically says, “This is how the system operates,” or “These are the parts of the system, and this is how they are interrelated.” In light of the discussion in Section 1.1, such “stories” are hardly stories. They are the narrative analogue to the well-known caption below a picture of a horse, which says: *Horse*. Statements of the form, “This is so,” whether textual or pictorial, are not usually interesting. An obvious response is, “So what?”

A diagram tells a story when it preempts that question, offering at least partial answers in advance. The distinction between story and “horse diagram” does not rest on the presence or absence of temporal information. A diagram of a system’s dynamic behavior can be little more than a reductive statement that “This is so.” The distinction is, rather, one of motivation or rationale. The answer to “So what?” asked in response to “This is so,” is: “Because it might not have been so.” And: “Not only might it not have been so, but it is jolly good (or bad) that it is, in fact, so.” Put a different way, something must pique one’s interest—an interesting problem whose solution is not obvious.

2 Methodology

Is this science? It would be possible to perform experiments on the effectiveness of different forms of communication, controlling them to determine the factors in a discourse that produce desired cognitive effects. While such an avenue would be interesting, we question whether it is necessary. There is much empirical evidence on the effectiveness of stories, and a large literature to that effect [6]. Some of this literature attempts to identify those aspects of story that determine its effectiveness [7]. But again we must ask whether the research labs or, rather, the screenwriter workshops (more generally, the reflections of successful tellers of stories) are the more convincing source of insight into this question.

Just as, therefore, there is a nascent science of diagrammatics, so there can be a science of diagrammatic narrative, and in particular of a science of diagrammatic design narratives. But more engaging, perhaps—to some of us, at least—is the notion of diagrammatic design (indeed, design in general) as craft: part art, part engineering—and, to be sure, as required by engineering, part science. The criteria for success of our theory as an approach to engineering design must be found in practical experience. If it leads to better communications, as determined by the users of those communications, then the theory will be validated; and if it does not, then it will not be.

What criteria, then, have we brought to the development of our theory? What evidence can we offer? In short, a combination of anecdotal evidence—our own, and that of others found in the literature—and narrative theory [2]. We have tried to specialize, apply, and adapt principles of narratology to the content (subject matter) and context (stakeholders, processes) of engineering design. At the same time, we have applied and adapted principles of the relatively new and still somewhat amorphous field of diagrammatics to the content and context of engineering design. We have attempted to fuse the insights of these two avenues of inquiry. The result is a theory of diagrammatic design stories, which we present in this article.

3 A Design Story

We begin with a design diagram in which the story is not very explicit. We will use this diagram and variations of it as a running example throughout the article.

The diagram describes a design developed by a group of scientists and engineers at NASA's Goddard Space Flight Center. The author of this paper is a member of that group. We draw on this example because the group's struggle to produce a cogent design, and to express it in diagrammatic form, manifested the principles of design narrative that we want to convey.

3.1 What's Wrong with this Picture?

If the reader finds the diagram in Figure 1 to be crystal clear—its meaning, role, and rationale abundantly apparent—then we beg her indulgence and ask that she imagine someone for whom the meaning might not be so clear. This was certainly the case for members of our design group, despite their participation in the design activity.

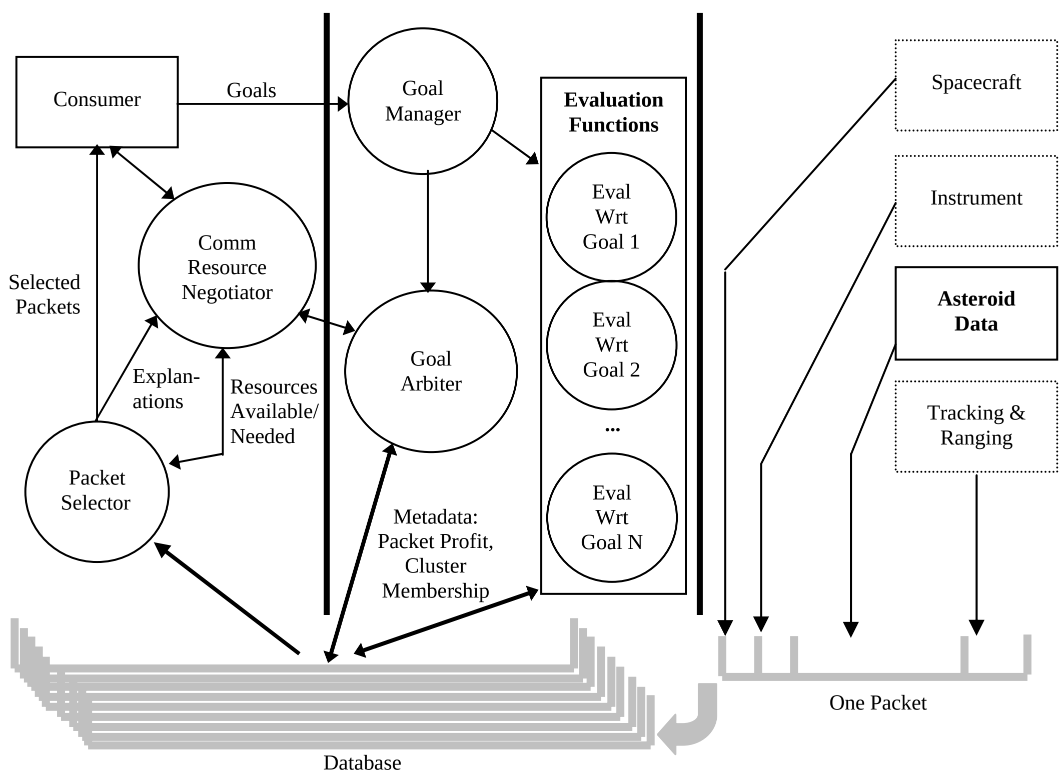

Figure 1. Our case study concerns a Science Data Filtering Architecture.

Let's be generous. The title of the diagram tells us that it is an architecture diagram, and that the architecture described is of something called filtering. So (being generous) we can infer that this is a system architecture, and that the system is performing a function called “filtering.” Furthermore, since there is something called a database in the diagram, it is reasonable to infer that this is a data processing system. That, in turn, suggests that the boxes represent processes or functions, or perhaps subsystems, and that the arrows indicate some sort of flow, whether of control or data or both. Moreover, if this is a data processing system, then it is likely that the "thing" being filtered is information.

A host of questions arise. What information is being filtered? Why? Where does the unfiltered information come from, and where does the filtered product go?

The arrows in the diagram provide some information. There seems to be an overall flow from right to left. On the right there are boxes from which arrows leave, but to which none points; on the left is a box—labeled Consumer—to which one arrow arrives, and from which two leave. The Consumer—consistent with its label—appears to be a sink, the destination of the filtered information, while the boxes on the right-hand side of the diagram appear to be sources of—presumably unfiltered—information.

This interpretation is highly structural. We still have no sense of what is being filtered, or why—let alone how. Actually we can glean some idea of how, in the sense that whatever the process is, it seems to involve goals. That is still quite abstract, however. The overarching question is, "What is going on here? What is this all about?"

3.1.1 Stories as Problem Resolutions

The diagram in Figure 1 fails as a story, in the first place, because it does not set the stage. In narrative terms, a story first establishes a setting (context), and then presents a problem. It does this is a particular way: included in the setting are one or more characters, and the problem that arises is not a theoretical problem, but rather a problem for one or more of the characters.

The story begins, therefore, with a character in a predicament. As we observe this happen—through watching, reading, or listening—we are invited by the story to identify with one or more characters—at least to recognize them and find them interesting—and thereby to hold a stake in the predicament and its outcome. We are thus drawn into the story.

3.2 Filtering Architecture: The Setting

There is a belt of asteroids flying through space between Mars and Jupiter. These are relatively small bodies (compared to the planets) about which scientists know very little. But scientists suspect that the asteroids are very old, that they have not changed much since their formation, and that they may, therefore, provide exciting insights into the early stages of the universe. If only we could take a close look at them.

3.2.1 Story Structure: the Setting and the Inciting Incident

Here we have the elements of a setting. The initial scene is the asteroid belt. We have a group of characters: the scientists, here on Earth. The characters face a problem, of the form *if only…* (a challenge) *then…* (something really good will happen). That problem is the inciting incident.

From the point of view of design, the problem—the inciting incident—is the overall goal. What makes it interesting is the fact that it is not obvious how to achieve the goal. Design stories are about solving non-trivial design problems.

There are three types of inciting incident: promise, injury, and threat. Our example illustrates a promise: if the challenge can be met, rewards will follow. A story can also begin with an injury (literal or metaphorical) to one or more characters, or, more generally, a disruption of a satisfactory state of affairs. In that case, the characters attempt to overcome the damage by restoring or creating a new stable state. Alternatively, the story may begin not with an injury itself but rather with the threat of injury: a problem looms on the horizon, and it is up to the protagonist to fend it off.

Figure 2. Scientists think that the Asteroid Belt could provide insight into the early stages of the universe.

All of these types of inciting incidents can occur in design stories. Our example illustrates promise. A typical example of injury would be a system failure that sets in motion a repair effort. Warding off an expected hacker attack would be a story that begins with a threat.

Scene setting in a story is the way in which a *story world* is created. A sense of realism of the story world—whether it is, in fact, realistic or fanciful—allows the reader to immerse herself in the story. From the point of view of design, setting the scene is the same as establishing context. The problem with context, of course, is that there is always too much of it. Narrative principles offer a way to focus on the context that is most important in communicating the design. Specifically: what needs to be told in order for the reader to care about the characters and to appreciate the significance of the problem?

3.2.2 Diagrammatics: How to set the Scene?

Scenes being (literally or metaphorically) visual, setting the scene ought to be the easiest part of telling a story through pictures. Suppose we include a picture of the asteroid belt, and a scientist pondering it. We could pose the challenge by summarizing it in a caption—perhaps in a thought bubble emanating from the scientist's head. Or we might try a more thoroughly visual tack: a notional picture of the Big Bang, inset on that of the asteroid belt, and some connective graphics to show how the latter provides insights into the former for the observing scientist.

Unfortunately, the Filtering Architecture diagram is not intended to tell the story of how scientists will tackle this challenge. It tells, rather, a more focused story within that larger story. So we are not yet finished setting the scene.

3.3 Filtering Architecture: The Back Story

The problem is that the asteroids are far away, and there are many of them. If we were to send up a deep space probe, how would we direct it to the most interesting asteroids, which we can only identify once we start receiving data from the probe? Communication with the ground (Earth) entails long delays. We would be expending copious resources on a vehicle that may, in fact, miss the most interesting targets of observation.

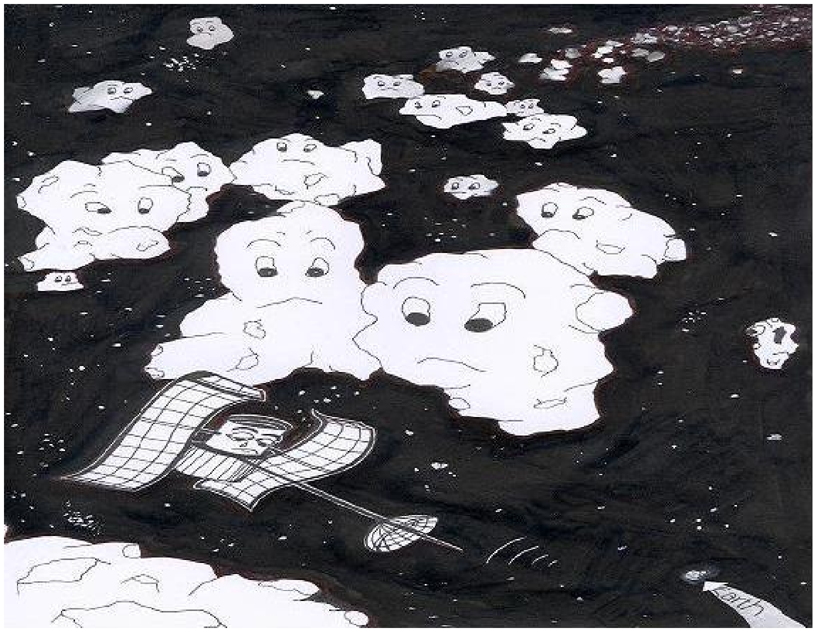

Figure 3. The are just too many asteroids for a single probe to explore.

3.3.1 Story Structure: First Steps

In response to the inciting incident, the characters attempt to solve the problem. Their initial steps towards this end set in motion a chain of events—action and reaction—that coalesce into a story.

Why a chain of events? Because the first response does not solve the problem. Perhaps it partially solves the problem, leaving some of it still to be solved. Or perhaps it solves the problem but only at the cost of introducing a new problem.

In our example, the first response is the obvious one: send up a space probe. Let's say that one of the characters—one of our scientists perhaps—proposes it. It gets shot down by his peers because it won't work. The problem remains, but it is different now because we have already considered one solution, played out its likely consequences, and rejected it. So the problem is, in fact, more difficult than we might have thought at first.

3.3.2 Diagrammatics: Representing Problem and Response

The primary challenge in telling a story through diagrams is finding a way to represent this sequence of

problem→action→new problem→…

Frequently, the new problem is created by a character reacting to the previous action. This is the classical structure of protagonist and antagonist. In our example, one scientist proposes an obvious solution ("send up a probe") and his peers shoot down the idea. If there were no reaction—if there were no new problem created by the initial response to the inciting incident—then we could simply show the problem and the solution, and be done with it. That would not be much of a story, but only because the design problem itself was not very difficult.

The diagrammatic challenge is to show a *sequence* of problems and solutions, each emerging from the preceding. This diagrammatic challenge is not restricted to design diagrams: it arises as well in diagrammatic theorem proving, as we discuss in Section 3.8.1.

One way to show the sequence of problems and responses is to use a sequence of diagrams. Using a sequence of pictures to tell a story is the time-tested mode of comic books[1]—a medium that should not automatically imply superficiality or lack of serious intent [8, 9]. Essentially the same medium is used by film directors in their *storyboards*, which present the sequence of planned shots in cartoon form, each shot annotated with the dialogue that will occur during the shot [10].

Another approach is to build sequence cues into a single diagram. These are visual indicators that say, in effect, "Look here before you look there." Sequence cues may take the form of numbered annotations; alternatively, they may piggyback on data or control flow implied by arrows in the diagram. At their best, sequence cues utilize their own visual grammar in which the eyes see, the mind raises questions, the eyes seek the answer, and so on.

3.4 Filtering Architecture: A Plan

NASA is investigating the feasibility of using nano-satellites and swarm technology to provide scientists with information about the asteroid belt. The Autonomous Nano-Technology Swarm (ANTS) mission will send a large number of very small satellites into the asteroid belt. Equipped with software that provides them with a high degree of autonomy, the "ANTS" will be able to coordinate opportunities and plans among themselves without much direction from the ground.

3.4.1 Story Structure: Pressure and Resistance

We are getting closer to the story we really want to tell, the story within the *scientists-ponder-the-asteroid-belt* story, the story of our Filtering Architecture. The back-story posed a challenge: if only we could observe the asteroids up close. Our characters' first move to meet this challenge was met by resistance: it won't work. They produce a new plan: nano-satellites and swarm intelligence [11]. But nobody has done this before. Will it work? The risks are manifest, but the rewards are great.

The draw of a story—the quality that keeps us reading until the end—is this quality of push-pull. The designers push the solution forward; the laws of physics—or, sometimes, Murphy's Law[2] or, for that matter, the Dilbert Principle [12]—push back, doing all they can to thwart the success of the design effort.

In conventional design documentation, writing about such obstacles and the way they have been overcome is often a burden. But this is the life-blood of stories. According to Henry James and others, the author should put his main character in the most crucial situation of his life, and then throw everything and the kitchen sink at him in order to foil his attempt to extricate himself [13]. The harder the design problem—the more reality pushes back—the better the story.

3.4.2 Diagrammatics: Representing Action and Reaction

The irony of our approach lies in the use of design diagrams to express dysfunction. In order to drive the story forward—in order to justify the eventual design solution—we must show why other designs do not work (or do not work as well). For the sake of a convincing story, we want to *show* this rather than *tell* it, which means conveying the dysfunction visually.

The practice of simulation, which is a standard way of testing out ideas in space system engineering, suggests a way to do this: draw a picture of the proposed design *in operation*. Let the reader see what happens, so that the problem becomes manifest. Figure 1 does show something about our filtering system's operation. It shows packets arriving and being either filtered or forwarded. But it does not show enough of what happens during the filtering process to illustrate any problems. In order to show the problem, we have to decompose the process into a finer level of detail. This illustrates the relationship between design stories and hierarchical design processes, whether top-down, bottom-up, or hybrid.

---

[1] Thanks to Zenon Kulpa for this observation.
[2] Anything that can go wrong, will go wrong.

Diagrammatically and narratively, when we test out a candidate design—delving into the inner workings of a process, putting (in our imagination) the single probe up in space to see how it performs—that is a change of scene. Whether the change involves zooming into a more focused and detailed view, or changing from a structural view to a behavioral view, the picture changes. This should not be surprising, since the candidate design is proposed as a solution—a successful solution. Discovering problems with it, then, most likely requires a change in perspective. This is consistent with the widespread use of multiple views in visual design. It is another argument for the storyboard approach.

3.5 Filtering Architecture: Finally, the Filter!

The swarm of ANTS will be able to visit on the order of a thousand asteroids. This is both good news and bad news. It is good news because it provides the kind of coverage that scientists want, offering lots of information. It is bad news because—it is really *lots of information: too much information to send back over such distances when contact times with the Earth are so limited.*

To make this work, the ANTS must have the intelligence to decide what information to send back to Earth, and what information to filter out. Perhaps not all of the ANTS need this intelligence, but at least some of them do: those that are responsible for such decisions. How do we build this intelligence into the ANTS?

Figure 4. Sending up a swarm to explore the asteroids means that the information they collect must be filtered before being downloaded to Earth.

3.5.1 Story Structure: Who are the Characters?

We have arrived at the design story proper: designing a filtering capability for the ANTS. The challenge of the design story proper is a functional goal. The ANTS must be able to perform a certain function, namely intelligent filtering of science data, and it is not obvious how they are going to do it.

From a narrative point of view, the ANTS have become characters in our story. This is typical of a design narrative. Human stakeholders (such as customers, users, the designers themselves) may be characters in the story, but so are the design elements and, in fact, the design as a whole. They solve problems, they create problems; therefore they are characters.

3.5.2 Diagrammatics: Showing Roles

Characters in a story play different roles. In design, we also speak about roles, meaning the purpose that an element or substructure plays within the overall design. Regarding design elements as characters is consistent with this viewpoint.

A formal theory of design roles has been developed by Chandrasekaran and Josephson in their work on Functional Representation [14]. An element (or substructure) is introduced into a design in order to serve a certain purpose—its function. This notion of *function* is more general than the computational notion of transforming inputs to outputs; it corresponds to purpose within a design. Similarly, *element* should not necessarily be understood as a physical object or computational function: it might be an aspect of the design, such as the curvature of a surface, which serves some purpose—its function—within the design.

How should roles be shown in a diagram? As Chandrasekaran's work shows, this is a fundamental problem of design, whether or not a narrative approach is followed. The narrative approach, however, introduces a curious ambiguity. Design elements are introduced in response to problems. In that sense, they are plot developments in the story. But as solutions within the design—as in Functional Representation—they have a role. That makes them like characters. They also potentially cause, or participate in, new problems. In that sense, too, they are characters (this too is consistent with Functional Representation). The ambiguity stems from the fact that we are conflating the introduction of a design element—an action carried out by the designer, who is a different character—with the design element itself, which as the story proceeds becomes a character in its own right.

Iconography—using shapes suggestive of roles—is an effective way of suggesting roles in a design diagram. Visual design languages embody this technique by using different icons for different types of elements; however, roles in a particular design are more specific than this. A designer may not have the tools or resources to develop design-specific—even domain-specific—icons for different roles. Thus, while iconography may be a desirable method of suggesting roles, more often a designer will resort to suggestive naming of design elements, and to relational information conveyed through positioning, connectors, or other visual cues.

Annotations can help to clarify the role of a design element. In our Filtering Architecture design group, despite a careful choice of names and the specification of flows between elements, we encountered recurring confusion among ourselves about the roles of certain elements (see Section 3.7). Eventually the confusions were resolved through textual annotations.

3.6 Filtering Architecture: A Different Point of View

Each ANT will carry a single instrument, which will collect spectral data in a particular frequency range from the asteroid it is currently observing. Data arrive at the filtering system from the instrument and from other subsystems of the spacecraft (such as Power, Thermal, Guidance and Navigation) in the form of packets, which are assembled periodically. A packet contains spectral data from the instrument, as well as engineering data pertaining to both the instrument and the spacecraft, and tracking and ranging data to assist in the interpretation of the spectral data.

As each packet arrives, it is passed to the Filter, which decides whether or not it should be forwarded to the Consumer. The Consumer of the filtered information might be the scientists here on Earth, or it might be a supervisory ANT responsible for collecting information from multiple worker ANTS before forwarding it to Earth (perhaps with additional filtering). Rejected packets are sent to a Recycling Bin, while packets to be forwarded are placed in a Downlink Queue, to be sent to the Consumer during the next contact time (known as a pass*).*

3.6.1 Story Structure: Narrative Voice

At this point in the story, engineering takes over from the scientists. The concern now is to design a specific capability within the ANTS mission. As we noted in Section 3.4.2, when we drill down to a more detailed level, and we narrow the scope of attention accordingly, we are shifting scenes. In this case, not only are we shifting scenes, but we are also changing voice, or perspective, from that of the scientists—or of someone peering into the laboratory observing the scientists—to that of the software designers.

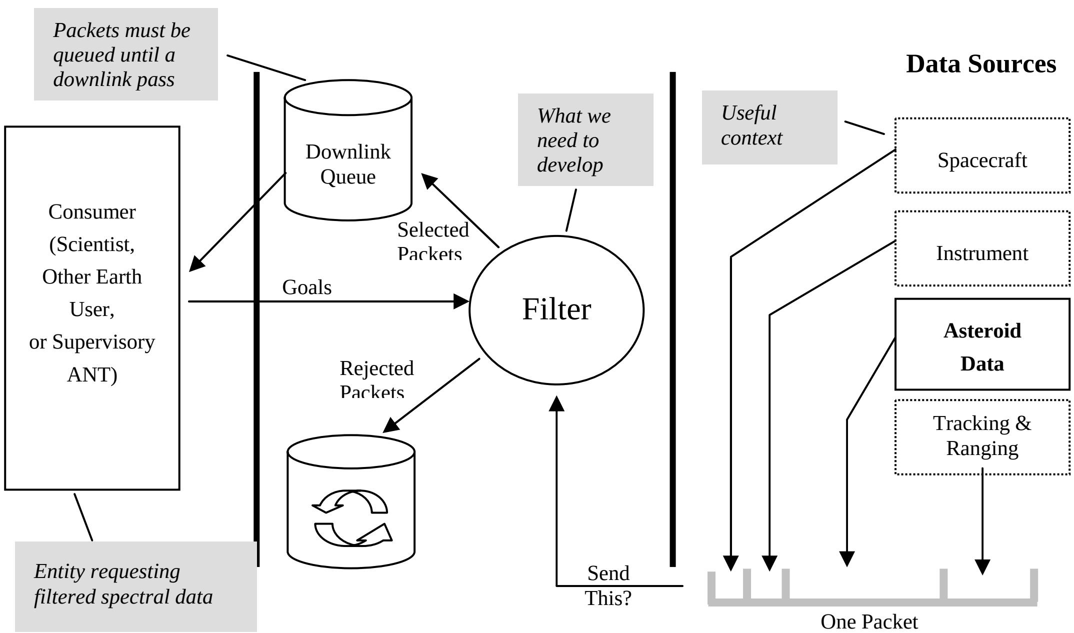

Figure 5. The first draft of a filtering architecture is a simple data flow.

3.6.2 Diagrammatics: Perspectives

As the voice and perspective change, so do the diagrammatics. Now we are in the realm of system design diagrams—boxes and arrows. But this is a very simple box-and-arrow diagram. There is only one data processing entity, the filter itself. A packet is assembled from the four indicated sources, fed into the filter, and either forwarded to the Consumer (via a Downlink Queue) or discarded. This is data flow at its simplest. The flow from right to left is complicated only by the fact that the Consumer specifies goals to the Filter. These are the current mission goals, which enable the Filter to determine what data will be most useful to the Consumer.

Despite the diagram's simplicity, we found it helpful to annotate it with virtual PostIt® notes (sometimes known as "stickies") to explain the roles of various entities. The annotations explain:

- The secondary sources of information (those other than the asteroid itself) as providing useful context
- The meaning of the abstraction *Consumer* (either scientist on Earth or supervisory ANT)
- The role of the downlink queue

Amidst all this information, a further annotation draws the reader's attention to the focus of the diagram—the thing that must be designed, namely the Filter itself. Thus, even the simplest box-and-arrow diagram may be enhanced through a narrative that walks the reader through the roles and the processing steps.

3.7 Filtering Architecture: On with the Story!

There is a problem with Figure 5: it presents the Filter as a function, which inputs a packet and routes it either to the Downlink Queue or to the Recycling Bin. This implies that the decision—to forward or discard the packet—is based solely on the goals and on the content of the packet. From the scientists' perspective, this is unreasonable. The value of a packet of spectral data can only be assessed in the context *of what has come before. The Filter needs a memory. Figure 6 shows the result. The Filter can now examine clusters of packets rather than individual packets. The Database, in which packets are "remembered," also plays the role previously played by the Downlink Queue. The Filter stamps each packet with an* Accept *or* Reject *tag, on the basis of which it is forwarded or discarded during the next downlink pass.*

But there's another problem with both Figure 5 and Figure 6: goals may conflict with each other. How does the Filter handle such conflicts? To address this issue, we decompose the filtering process. In Figure 7, each goal has its own Evaluation Function, which assigns a profit value to a packet or cluster with respect to that specific goal. We then introduce a Goal Arbiter function, which examines the potentially conflicting goal-specific profit values and assigns an arbitrated *profit value. The arbitrated profit value determines whether the packet or cluster will be forwarded.*

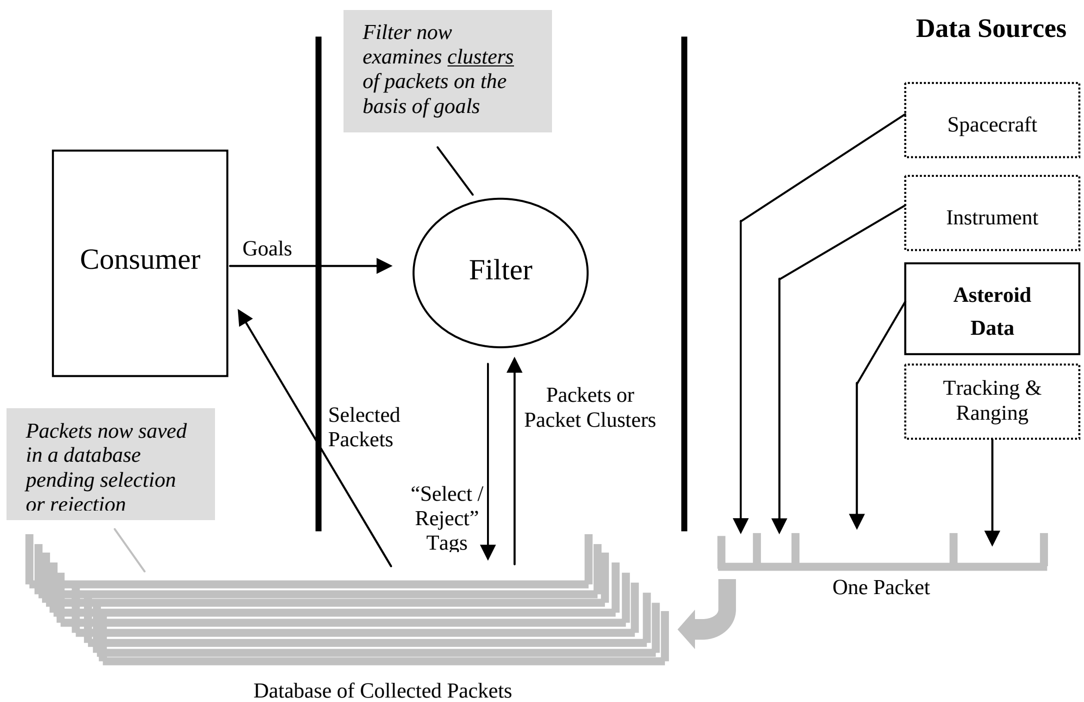

Figure 6. To enable more intelligent selection, we give the filter a memory in the form of a database.

There is a problem with this design too, however. Though the selection process is obviously becoming more refined—more intelligent—we are still faced with having to trade off the intrinsic value of a packet against its size. A packet may be quite valuable, scientifically, but it may also be very large. Other fairly valuable packets might be much smaller. Since there is only a limited bandwidth to the downlink, sending the larger packet may entail having to discard the smaller ones. Will the scientist be happy losing all the fairly useful small packets for the sake of the very useful large one?

3.7.1 Story Structure: Controlled Release of Information

The story is now heating up, with the scientists and the engineers not always seeing things the same way. Among the scientists themselves, there is disagreement about the answer to this last question. Some say that good data should never be deliberately discarded. Others point out that, one way or another, some data will be lost, so they might as well try to control the process as intelligently as possible.

The issues and alternatives proliferate. There is more information to keep track of. The design is becoming more complex. During the design discussions, all of these issues—and others—arose more or less simultaneously. We have presented them in sequence, with a solution to each in turn, to help the reader comprehend the eventual design.

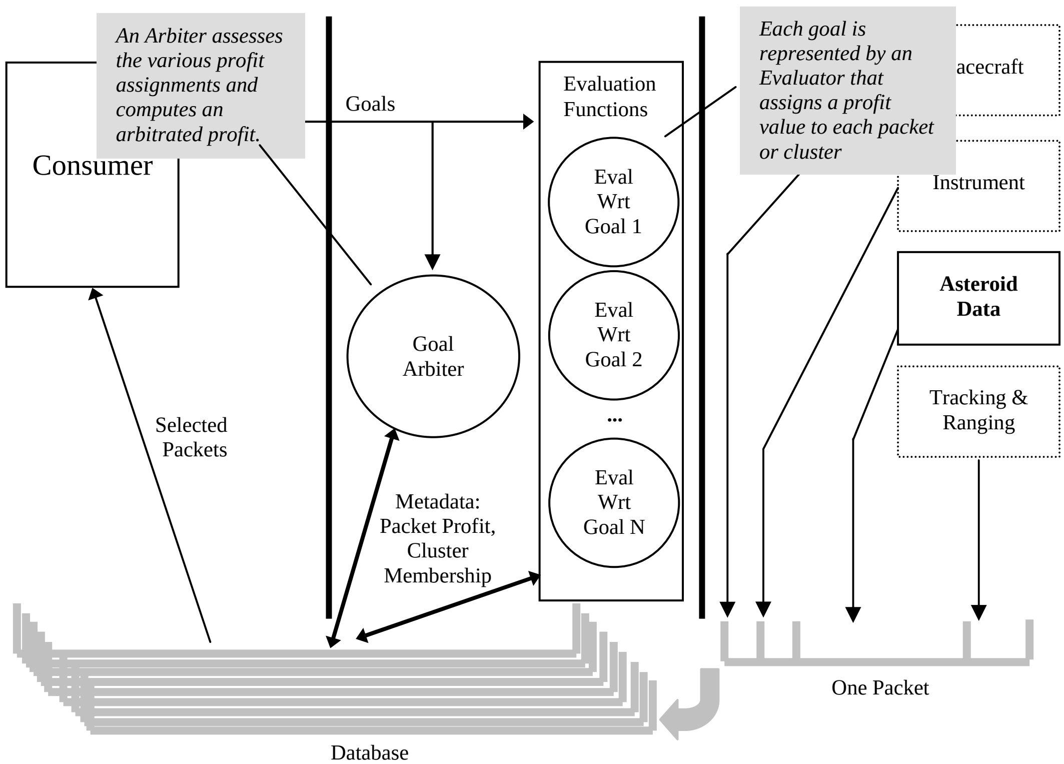

Figure 7. The problem of filtering becomes a problem of arbitrating among conflicting goals.

3.7.2 Diagrammatics: Hide and Seek

We noted, in Sections 1.2 and 3.5.2, that there are two major strands in a design story: the story of the design process itself, and the story of the envisioned use of the designed artifact. In the story of the design process, design elements are introduced in response to problems encountered in previous designs. From a functional point of view, the role of the design element is to solve or alleviate that problem. In the story of the designed artifact's use, the design element plays that functional role as a character.

A story does not usually introduce all of its characters at once; nor does it present all the action at once. Instead, the story presents a thread of events, or several concurrent threads, alternating attention from one thread to another, so that the rationale for each event is either apparent (thereby fulfilling expectations, answering questions, diminishing the story's tension) or unclear (thereby surprising us, raising questions, increasing tension).

A design diagram, or sequence of diagrams, must do the same thing in order to present a coherent rationale and not to overwhelm the reader. Serializing the reasons for introducing the major design elements is one way to achieve this goal. The sequence need not be structured as

one problem→one answer→one problem→one answer→…

as we have done in this article. Real design processes are not always like that; design stories need not be either. Clustering a group of problems together can increase the level the tension in the story, thereby keeping the reader interested. Too many problems surfaced together will, however, clutter the field and cause confusion. A good story strikes a balance.

We can use this principle to decide how much information to put in a single diagram. To illustrate this point, the next chapter of our design story presents several problems at once, the solution to which yields the final architecture diagram, which we first viewed as Figure 1.

3.8 Filtering Architecture: Climax and Denouement

The introduction of a Goal Arbiter raises as many questions as it answers. Is the Arbiter actually choosing which packets to forward, or just offering an opinion based on an impartial analysis of the potentially conflicting goals? If it is just offering an (albeit impartial) opinion, then who is doing the actual selection? If the Arbiter is *doing the selection, doesn't it have to worry about the profit/size tradeoff mentioned in Section 3.7? Should that be considered as part of the arbitration process, or it is an entirely different sub-problem?*

Some of the physical scientists are a little suspicious of throwing computer science algorithms at this tradeoff—essentially, the Knapsack Problem—possibly at the expense of good science data. They wonder whether it would better to try to negotiate increased communications bandwidth during those periods when the Evaluation Functions are finding an abundance of useful spectral data. After all, bandwidth is allocated between several different sources, even on a single ANT. Perhaps some of the spacecraft subsystems can relinquish some of their bandwidth for a limited period of time, if the state of the system indicates it is safe to do so.

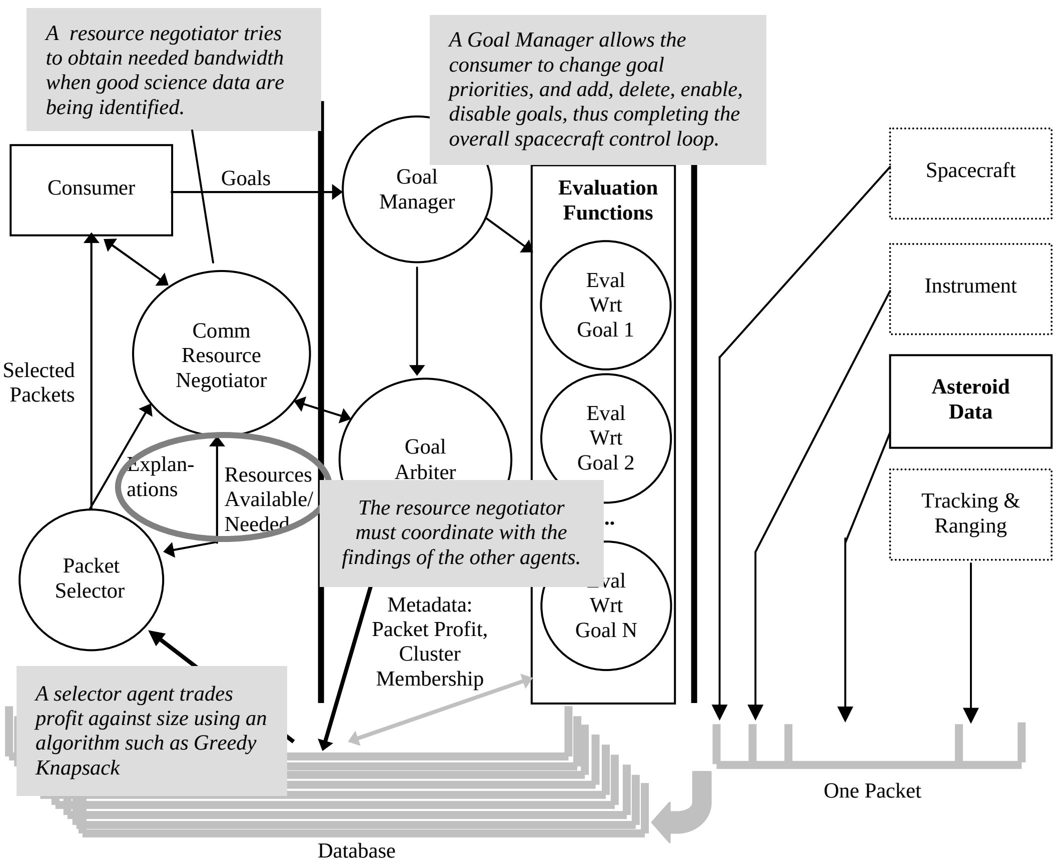

Figure 8. Several additional design elements address the remaining concerns of the group.

The mission engineers see this suggestion as an opportunity to raise a concern they had all along: where is the control loop? The software people have been treating this system as a functional pipe: stuff comes in, stuff goes out. But spacecraft systems are not like that. They require continual monitoring of state and resource allocations, and command channels through which problem recovery and resource reallocation directives can be issued to the subsystems, including the data filter. This might include, for example, allocating processor or memory resources to certain goals (and their evaluation functions) rather than others.

The software designers counter that, while all this is true, the design group's charter is to design a filtering capability for the ANTS, not to design the ANT's entire software system. The control loop is part of the system-level context in which the filter must operate; it is not part of the filtering problem itself. They concede, however, that a managerial function should be added to the Filtering Architecture to serve as an interface for such control functions.

As a result of these discussions, three new elements are added to the architecture:

- *A Selector, responsible for deciding which packets to forward and which to discard, performing any necessary profit/size tradeoffs in order to make these decisions*
- *A Communications Resource Negotiator, which tries to obtain the necessary bandwidth from the Consumer, using information provided by the Selector in support of the request*
- *A Goal Manager, responsible for prioritizing goals and allocating resources to them in response to directives from the Consumer and/or the spacecraft executive system*

The group is, overall, happy with the new design. In exploring the details, however (as the story continues beyond the focus of this article), they from time to time revisit the issues just raised: What exactly is the difference between the Goal Arbiter and the Goal Manager? What is the difference between the Goal Arbiter and the Selector? Where are the control functions? As these and other questions are revisited, it becomes apparent that the answers must be explicitly documented because even the authors of the answers are forgetting them! Hence this story.

3.8.1 Story Structure: Sub-Plots and Sub-Goals

Our narrative has told of successive design problems, each solved by introducing a new design element. We discussed, in Section 3.5.2, the relationship between this type of design story and the Functional Representation method. An interesting relationship also exists between this type of design story and the use of diagrams to suggest proof strategies in mathematics. In our work with Dave Barker-Plummer on the Grover (Graphical Prover) system, we found that mathematical diagrams can often be parsed as suggesting a series of existence proofs [15, 16]. Elements in the diagram represent objects whose existence—that is, the existence of some object standing in relation to other objects shown in the diagram—must be proven. Grover applies a set of heuristic rules to determine the relevant properties of each such object, and in what order the existence of the objects should be proven. These heuristics are, in effect, inferring the dynamics of the intended proof from a static diagram.

Formal approaches to design, i.e., approaches backed by theorem proving, frequently translate design problems into theorems by treating the desired solutions as existence proofs to be found. Thus, the correspondence between our role-based design story and an existence-proof approach to diagrammatic theorem proving is more than just an analogy. It would be interesting to see whether design diagrams could be parsed in a manner similar to the way Grover parses mathematical diagrams. The resulting proof of correctness would, in effect, be a formalization of the underlying design story.

Design solutions are not, however, always found through a simple process of adding elements. There are usually alternative solutions, and tradeoffs between them that must be investigated. We view these investigations as sub-plots of the design story. This approach is consistent with [17], which proposes a representation of sub-plots as stacked context-switches.

3.8.2 Diagrammatics: That's the Story?

Did we succeed? Have we transformed Figure 1 from a jumble of boxes and arrows (and some other shapes as well) into the visual representation of a story? Is the path from Figure 1 to Figure 8 just a matter of sticking some PostIt® notes onto the diagram? To be sure, Figure 8 differs from Figure 1 only in that respect. But Figure 8 is not the whole story—it is the culmination of the story. There were intermediate diagrams along the way, and a lot of text. In what sense, then, can we speak of a diagram—either Figure 8 or any of the preceding diagrams—telling a story?

One answer: taken together, the diagrams do tell a story. But this is not entirely true because the succession of diagrams was narrated in text. Could the text have been dispensed with? Probably not. Why would we want to dispense with it?

It is undoubtedly true that *more* of the story could be represented in purely visual terms, thereby reducing the need for textual narration. Clever iconography might allow us to communicate the succession of design *problems*—not just the solutions—visually rather than in text. We face a cost-benefit tradeoff: determining the visual vocabulary and grammar needed to tell the story in pictures takes a lot of work—both thinking and experimentation. The reward is in the reading, which will be more compelling, the story more easily comprehended and retained. Whether the effort is worth it must be judged for each situation in its own right.

The message that this article hopes to convey is that where there is a design, there is a design story. One way or another, the story must be told. Diagrams should help.

4 Acknowledgements

Many thanks to the participants in the design group that developed the Filtering Architecture: Pamela Clark, Jay Karlin, Jagan Iyengar, Sanda Mandutianu, Tim McClanahan, Fred Mills, Mike Rilee, Chris Rouff, Walt Truszkowski, and Victoria Yoon. The design insights expressed through the story are theirs.

References

[1] J. Orr (1986) Narratives at work: Story telling as cooperative diagnostic activity. *Proceedings of the Conference on Computer-Supported Cooperative Work*, Austin, Texas. Association for Computing Machinery (ACM) Press. Pages 62-72.

[2 S. Chatman (1978) *Story and Discourse: Narrative Structure in Fiction and Film*. Cornell University Press.

[3] M. Foucault (1980) *Power/Knowledge: Selected Interviews and Other Writings, 1972-1977*. Pantheon Books.

[4] D. Haraway (1997) *Modest_Witness@Second_Millenium. FemaleMan©_Meets_OncoMouse™: Feminism and Technoscience*. Routledge.

[5] D. Spoto (1992) *The Art of Alfred Hitchcock*. Anchor.

[6] S. Denning (2001) *The Springboard: How Storytelling Ignites Action in Knowledge-Era Organizations*. Butterworth Heinemann.

[7] M. Mateas and P. Sengers, ed. (1999) *Narrative Intelligence: Papers from the 1999 Fall Symposium*. Technical Report FS-99-01, American Association for Artificial Intelligence (AAAI) Press.

[8] A. Spiegelman (1993) *Maus: A Survivor's Tale*. Pantheon.

[9] L. Gonick and W. Smith (1993) *The Cartoon Guide to Statistics*. Harper Perennial.

[10] S. Katz (1991) *Film Directing Shot by Shot: Visualizing from Concept to Screen*. Butterworth.

[11] J. Kennedy and R. Eberhart (2001) *Swarm Intelligence*. Morgan Kaufman.

[12] S. Adams (1997) *The Dilbert Principle: A Cubicle's-Eye View of Bosses, Meetings, Management Fads & Other Workplace Afflictions*. HarperBusiness.

[13] C. Keane (1998) *How to Write a Selling Screenplay*. Broadway Books.

[14] B. Chandrasekaran and J. Josephson (2000) Function in device representation, *Engineering with Computers*, Vol. 16, Special Issue on Computer Aided Engineering, pages 162-177, 2000.

[15] D. Barker-Plummer and S. Bailin (1997) The role of diagrams in mathematical proofs. *Machine Graphics and Vision*, Vol. 6, No. 1.

[16] D. Barker-Plummer and S. Bailin (2001) On the semantics of mathematical diagrams. In *Diagrammatic Reasoning*, ed. M. Anderson, Springer Verlag.

[17] M-L. Ryan (1991) *Possible Worlds, Artificial Intelligence, and Narrative Theory*. Indiana University Press.