Experiences With The Alternate Reality Kit

An Example of the Tension Between Literalism and Magic

Randall B. Smith

Xerox Palo Alto Research Center
3333 Coyote Hill Rd.
Palo Alto, CA 94304

ABSTRACT

This paper presents an overview of the Alternate Reality Kit (ARK), an animated environment for creating interactive simulations. ARK is built upon a physical-world metaphor: all objects have an image, a position, a velocity, and can experience forces. Users manipulate objects with a mouse-operated "hand" which enables them to carry and throw objects, to press buttons, and to operate sliders.

The interface features are discussed in light of a general user interface tension between *literalism* and *magic*. Literal features are defined to be those that are true to the interface's metaphor. Literal features enhance an interface's learnability. Magical features are defined to be those capabilities that deliberately violate the metaphor in order to provide enhanced functionality. Discussion of each ARK feature includes informal observations of early ARK users, an assessment of the feature's learnability, of its usefulness, and of its position on the magical-literal axis.

Even though ARK includes magical features, applications-level users have be trained in a few minutes. Although this paper is about ARK, the tension between literalism and magic raises some interesting questions on its own. Some of these questions are presented briefly in the conclusion.

RÉSUMÉ

Cette présentation donne une description générale d'une «trousse à outils pour une autre réalité» (ARK: Alternate Reality Kit), qui consiste en un environement animé pour faire de la simulation interactive. ARK est basé sur un modèle du monde physique: les objets y ont une image, une position, une vitesse, et y sont soumis à des forces. Les utilisateurs manipulent les objets à l'aide d'une «main» contrôlée par la souris, qui leur permet de porter et lancer des objects, d'appuyer sur des boutons et d'ajuster des leviers.

Les propriétés de l'interface sont décrites en fonction d'un conflit entre *réalisme* et *magie*. Les propriétés réalistes sont celles qui respectent le modèle physique sur lequel est basé l'interface et qui facilitent l'apprentissage de ce dernier. Les propriétés magiques sont celles qui violent les règles du modèle physique pour en accroître les fonctions. La discussion portant sur chaque propriété de ARK comprend les résultats d'observations des premiers utilisateurs, une évaluation de la facilité d'apprentissage de la propriété en question, de son utilité et de sa position le long de l'axe magie-réalisme.

Bien que ARK comprenne des propriétés magiques, ses utilisateurs, au niveau des applications, ont pu être entraînés en quelques minutes. Quoique cette présentation porte sur ARK, le conflit entre réalisme et magie soulève de son propre chef des questions, dont certaines sont brièvement présentées en conclusion.

Keywords: simulation, visual programming, Smalltalk, learnability, graphical interfaces

Introduction

The designer of a system for use by novices can gain great advantage by basing the interface on a known metaphor. If the computer behaves in a way analogous to a system already understood by the user, the learning time will be greatly reduced. Interface features that are true to the designer's metaphor might be called *literal*. The learnability of literalism makes it a good thing.

However, the designer can always provide the user with enhanced capabilities at the price of breaking out of the metaphor. These features might allow the user to do wonderful things that are far beyond the capabilities of literal features. Capabilities that violate the metaphor in order to provide enhanced functionality might be called *magical*. The power of magic makes it a good thing.

There is a tradeoff [11] between the learnability of literalism and the power of magic. I employ this tension as a way to present my experiences in designing and observing users of the Alternate Reality Kit, a metaphor-based system being developed for non-expert computer users.

The Alternate Reality Kit (ARK) is intended to allow users to play in their own simulated worlds and to create new ones. ARK is based on a strong analogy to the physical world. Many of the important capabilities of ARK are *literal*: they are transcriptions into the computer of physical world behavior. Even though the system is designed for use by inexperienced users, ARK has certain magical characteristics. Observations of ARK users suggest that novices are not significantly

hampered by a small amount of magic. Although each magical feature requires a brief explanation, ARK's basic functionality can still be taught in a few minutes.

This paper is about the Alternate Reality Kit: it is a brief introduction to the ARK user interface in terms of the magic-literalism dimension. A more complete description of the functionality and philosophy behind ARK is in reference [14]. I will not attempt much in the way of a general discussion of the tension between magic and literalism: I simply employ this tension as a useful way of analyzing the central features of ARK's interface. However, ARK does serve as an example of a magic vs. literalism tradeoff which I believe to be present in all user interfaces that are firmly grounded in a single metaphor. In the conclusion I will present some questions raised by the magical-literal view of this kind of user interface.

Literalism vs. Magic in ARK

The Alternate Reality Kit (ARK) [14] is a system for creating interactive animated simulations. ARK simulations are intended to enable the development of intuitive understanding of the simulation's interaction rules by making these rules appear as accessible physical objects called *interactors*. ARK also supports the modification and creation of new simulations from within the animated ARK environment.

The interface is quite faithful to a physical-world metaphor: all objects have a visual image, a position, a velocity, and can experience forces. One of the objects is a hand, which the user controls with a mouse. With the hand, the user can carry objects, throw them, press buttons and operate slider controls (see Figure 1). As in the real world, many things are happening simultaneously: a pendulum can swing while numbers change on a control box in response to the operation of a slider. The intent is to have the user conclude very quickly that the screen depicts a physical world, and that the user is directly manipulating physical objects. This is the advantage of *literalism* -- interfaces strongly based on a well-known metaphor require very little explanation to users [4, 5, 6].

However, sticking completely to a metaphor can cripple a system's functionality [7, 10]. For example, an ARK user may wish to connect a simulated pushbutton to some ARK device, perhaps for turning the device on and off. Both the button and the device are depicted as physical objects that can be directly manipulated with the hand. Should the user be required to connect the button by drilling a hole in the device and cutting into metaphorical electrical work? Something like this would be required if ARK were perfectly analogous to the everyday physical world. In the design of ARK, I considered perfectly literal ways of connecting buttons to be too tedious. Instead, the ARK user connects the button simply by dropping it onto the device. Buttons have the message they send stamped on the surface - if the device does not understand the button's message, the button will fall right through the object (see Figure 2). If the button's message is meaningful, it will stick to the surface of the object. An invisible connection is established automatically, and the button is immediately functional. Furthermore, buttons can be created that cause non-physical effects such as doubling an object's size and mass, or causing the object to vanish. Features like these are called *magical* because they enable the user to do powerful things that are outside of the possibilities of the metaphor.

Admittedly, literalism and magic are not part of conventional computer science parlance. However, I find them to be particularly appropriate for discussing ARK, where even the name "Alternate Reality Kit" suggests both the real (literal) and the ability to choose between or modify realities (magic).

Although ARK is more literal than most systems, it does contain certain magical features which are useful where literalism would be limiting. But one of the lessons of ARK is that the literal aspects of the interface are often obvious while magical capabilities are harder to learn. In ARK, the time to explain the basics is actually measured in seconds.

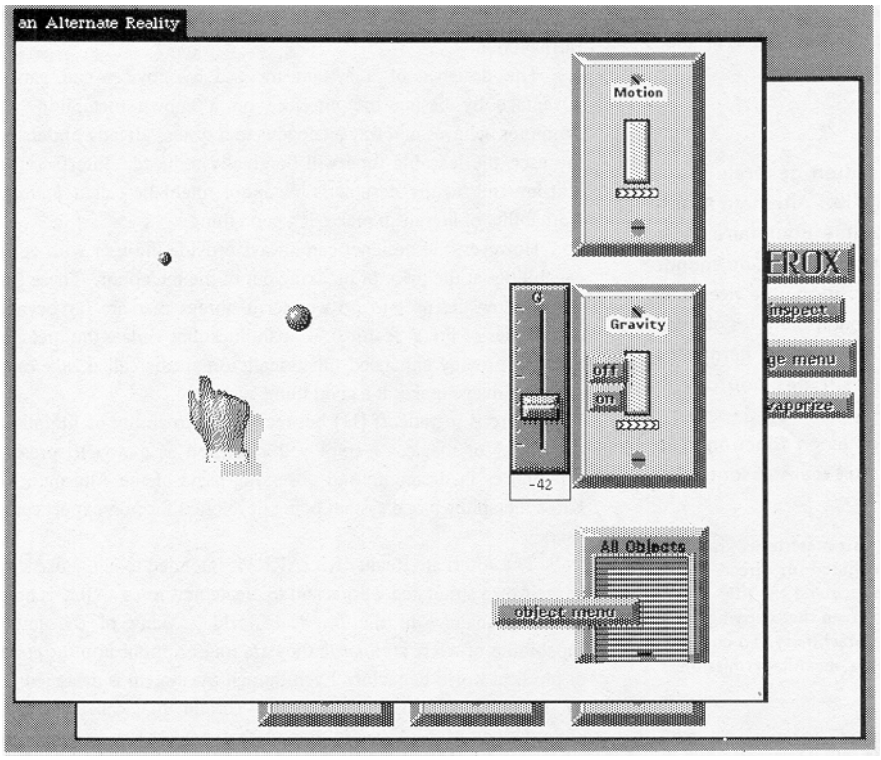
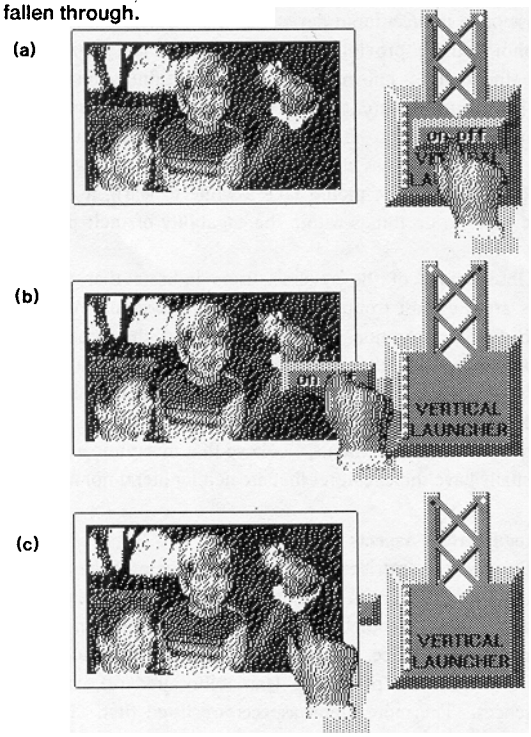


Figure 1. Appearance of the screen as seen by a user of the Alternate Reality Kit. Each window and the objects it contains represent an *alternate reality*. The mouse-operated hand (near the center) is casting a shadow that indicates it is "above" the alternate realities. The hand is the user's means of interacting with the system. It is used for picking up and carrying objects, and for pressing buttons. Two *interactors* are present in the alternate reality under the hand. One, labeled "Gravity," causes a gravitational force field to be present; the other, labeled "Motion," causes objects to change their position according to their velocity. (The windows shown here are considerably smaller than those typically used in ARK.)

Figure 2. Manipulation of buttons. In (a), the user is preparing to pick up the "on off" button from the surface of the Vertical Launcher. In (b), the button is carried over to the simulated photograph. In (c), the button has been released. Since the photograph does not understand how to turn on or off, the button has fallen through.



Every piece of added magic is relatively "expensive" because it requires its own explanation: it does not "come for free" as it does when the user realizes there is a physical metaphor. In designing ARK, I am therefore faced with a tension between the *limitations* imposed by literalism and the *obscurity* of magic. Or, in positive terms, between the *power* of magic, and the *learnability* of literalism.

Overview of ARK

ARK is a project under development in the System Concepts Laboratory of the Xerox Palo Alto Research Center: it is being implemented in the Smalltalk-80* programming environment [8]. The system described here has already evolved under the influence of user feedback, and will continue to do so. The six kinds of objects mentioned in this paper are shown in Table 1.

The system consists of a collection of "physical" objects that can be manipulated with a simulated hand. Except for rare use of the keyboard for typing text, the hand is the user's sole means of interacting with the system. The ARK user can do three kinds of things with the hand: directly change an object's position or velocity (by carrying or throwing), send an object a message (by pressing a simulated button), and introduce a new object or button into the environment (by selecting from "pop-up" menus).

The user can change the state or query objects by sending them messages: messages are represented by buttons. Examples of button messages are *velocity, set mass to:<some parameter>,* and *describe yourself.* One special button, the *message menu* button, causes a very magical effect: attaching it to an object and pressing it will cause the object to create a menu that lists all of the messages the object understands. Selecting a message from this menu will cause a corresponding button to be created, attached to the object and ready for use.

Buttons that send an unparameterized message with no response are called *simple* buttons and are illustrated in Table 1. Buttons requiring parameters or representing an object's reply to a message are called *non-simple* and are shown in Figures 3 and 4.

The warehouse object contains one copy of every kind of object in the system, including one of each type from the underlying Smalltalk environment. By pressing the appropriate button, the user can cause the warehouse to display a menu listing all of the objects the warehouse contains. Selecting from the menu causes a copy of the named object to be introduced into the alternate reality.

There are buttons that allow the user to create new kinds of objects and store them in the warehouse. Ways to combine existing buttons into new kinds of buttons are still being explored.

Types of ARK User

ARK, like some other visual programming environments, [2, 9], is intended to have more than one kind of user. The *applications-level* user might typically be a student carrying out a simulated lab. At a lower level, the *simulation builder* is the creator of a particular application. There may be a role for another layer below that, populated by individuals who create tools for use by simulation builders.

So far, I have observed about 50 applications-level users and two simulation builders. Most applications-level users are part of an experiment studying people's reactions to the violation of physical laws [12]. These users do not have to create new objects, or new kinds of objects. The accumulated empirical evidence enlightens only certain

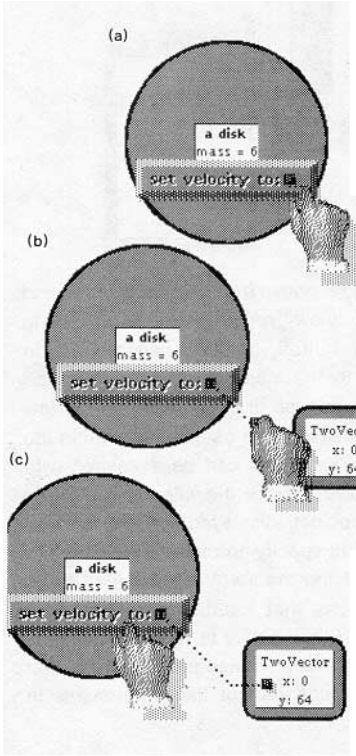


Figure 3. Buttons that require a parameter (such as the *set velocity to:* button shown here) have a small "plug" used for specifying the parameter. In (a), the user reaches for the plug. In (b) the user has grabbed the plug and is carrying it over to the parameter object. In (c) the user has dropped the plug onto the object, and is preparing to activate the button. Once the parameter is specified, the small dots start moving along the path connecting the plug to the button. (Only simple buttons, shown in table 1, have been used in the experiments in which users have been observed.)

* Smalltalk-80 is a trademark of the Xerox Corporation. In this paper the term "Smalltalk" refers to "the Smalltalk-80 language."

portions of the interface: the applications-level users made no use of menus and the non-simple button types illustrated in Figures 3 and 4.

Every user comes to the computer with expectations about what will be encountered. Some sophisticated computer users take slightly more time to learn the literal features of the interface -- they appear to expect more magic than ARK contains. On the other hand, extremely naive users (young children) have sometimes expected the interface to be more literal than it is, apparently expecting *less* magic. ARK's balance between literalism and magic seems about right for computer novices above the age of ten. However, the effect of user sophistication is not great: misunderstandings have always been correctable with one or two sentences.

Limitation of the Magic-Literalism Analogy: External Factors in ARK

This presentation of the Alternate Reality Kit depicts the designer as only violating a metaphor in order to provide enhanced functionality (magic). But sometimes designers face factors beyond their control. Input devices, computer performance limitations, or other constraints can cause the metaphor to be violated in a way that does not necessarily enhance functionality. These fixed requirements are called *external* factors because they are imposed upon the designer. That is not to say that external factors are unimportant - successfully presenting external factors to the user can be absolutely crucial.

I have found external factors difficult to present as either literal features or magical enhancements to the ARK interface. In ARK, external factors typically degrade learnability *without* enhancing functionality: I consider such features to be neither literal nor magical.

For example, when an alternate reality contains a very large number of strongly interacting objects, the animation rate (frames per second) drops and it can become noticeably harder to grab and throw objects. The "jerky" motion of objects makes them look less like real-world things, and makes them harder to interact with. This external factor has broken the metaphor in a way that degrades functionality. This behavior is clearly not literal. And even though it is outside the real world's behavior, I prefer not to use a sense-of-wonder term like "magic" for this odd and annoying visual phenomenon.

A second example is the mouse, which is used to operate the hand. The use of an indirect input device like the mouse breaks the real world metaphor without providing enhanced functionality. As a pointing device, the mouse is known to take a small but finite amount of time to learn [3]. Furthermore, one mouse button is used to make the hand grab objects, another to make the hand activate the simulated buttons. Every observed user has at some point confused these two functions [1, 13]. Not only does the mouse take a while to learn, it does not even enable users to do things within the capability of their physical world hand.

Observations of the system's users indicate that these external factors are the most troubling aspects of the system. As indicated in Figure 5, the operation of the mouse and the occasionally slow animation rate are neither enhancements to the functionality nor aids to ARK's learnability. The ideal system would be one in which everything, including external factors, fit along the magical-literal spectrum. But due to unfortunate constraints, I believe that metaphor-based interfaces will usually have some features that are neither literal nor magical.

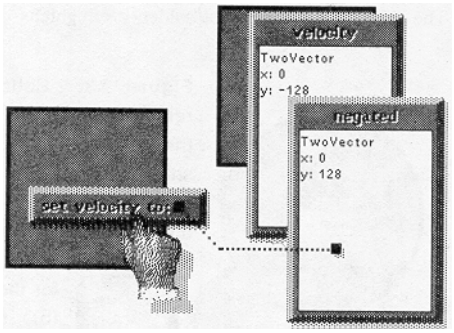


Figure 4. Buttons that elicit a response from their message send are large objects that have a "view" rectangle for representing the result. Both the *velocity* button, and the *negated* button shown here receive answers to the messages they send. The answer is displayed as some text or graphics within the view rectangle. The answer is accessible as an object living within the view. Objects within the view rectangle can be accessed with buttons just like any other object. Here the *negated* button is attached to the TwoVector object displayed in the *velocity* button's view. Similarly, the plug specifying the parameter to the *set velocity to* button is specifying the TwoVector object in the *negated* button's view. Buttons that capture responses are somewhat magical, especially in their ability to represent objects that are "really there" in the view rectangle. (Buttons that represent responses have not been part of the experiments in which users have been observed.)

Selected Interface Aspects

This section describes in greater detail some of the aspects of the interface which have been used by applications-level users. This is not intended to be a complete list, but a sample of the more important aspects of the interface. For each feature listed, I evaluate the magic content, discuss the power vs. learnability tradeoff, and note user experiences. The more literal aspects are listed first. This section is summarized in Figure 5.

Use of the hand: The user can pick up any object with the simulated hand. As the grasped object is carried about, it casts a shadow on the alternate reality beneath it. When a grasped object is released, it falls back into the alternate reality and maintains any velocity imparted by the hand's motion. In this way the user playing in the gravity simulation depicted in Figure 1 can throw the moon into orbit around the planet.

Magic vs. Literalism: Literal

Power vs. Learnability: The hand has only limited abilities. It enables users to change position and velocity of objects, and establish physical contact relationships between objects. However, it contributes quite a bit to the user's understanding of the system's basics. Users need only be told "This mouse moves the hand on the screen. The left mouse-button enables you to grab hold of an object," and "Try throwing something." Some of this is about the mouse. The remainder of this brief explanation is about the use of the hand, from which most users infer the following:

- The objects on the screen are physical entities.
- Physical proximity has semantic content. (Only one subject, an experienced mouse user, has asked if it was necessary for the hand to be over an object in order to pick it up.)
- The hand can carry an object and drop it at a new location, thus changing its position.
- The hand can change an object's velocity. (After being invited to throw an object, some users ask how throwing is done. When asked to go ahead and guess, most of these users have guessed correctly: start the hand moving and release the held object. Experienced mouse users have

sometimes asked if throwing is accomplished by pressing a special mouse-button.)

- The use of shadow to indicate that an object is "above" the reality has only been moderately successful. For a few users, an additional sentence or two of explanation is required. ("See the shadow? That indicates the object in the hand is hovering *over* the objects in the window.")

Activation of Simple Buttons: A user playing in the planetary orbit simulation of Figure 1 may wish to suspend gravity temporarily. On the right side of the figure there is a kind of controller device labeled "Gravity" with an *off* button on its surface. All buttons have the name of the message they send stamped on them. The user activates a simulated button by positioning the hand over it and pressing the middle mouse button. Only *simple* buttons (as shown in Table 1) have been used by applications-level users: buttons which require parameters (Figure 3) or which both send messages and represent responses (Figure 4) have not yet been used in ARK experiments.

Magic vs. Literalism: Literal

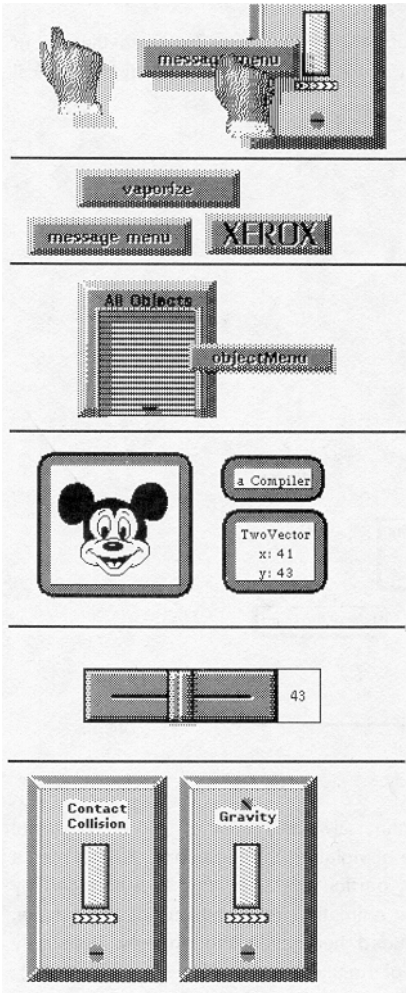
Power vs. Learnability: Buttons are moderately useful. They enable users to send a message to an object. They are fairly easy to explain: users seem to understand immediately what buttons are for. The only training time is spent in pointing out to the user the characteristic visual presence of a button. The idea that an object responds to a button press in a way suggested by the name stamped on the button is simply assumed. Thus, without explicit instruction, the user adopts a model consistent with the picture of a button as a thing that "sends a message." The button builds on the importance of physical contact in establishing relationships between objects.

Manipulating buttons: A simulated button can be picked up from the surface of an object and put down anywhere, even on top of certain other objects. A button will stick to the surface of any object that can respond to its message. If a button is dropped on an object that does not understand its message, the button will "fall through" the object, failing to stick to its surface. Many buttons can be simultaneously attached to an object. Sometimes a button will be larger than the object upon which it rests. It is not uncommon for an object to have a button "hanging off the edge," or even completely covering the object upon which it rests.

Magic vs. Literalism: Moderately Magical

Power vs. Learnability: Buttons are easily connected and removed, enabling the user to communicate with objects in a flexible way. It is not uncommon to have several generally useful buttons laying about. The "selective sticking" of buttons prevents a certain class of semantic errors. (As an example, sending some text the message *cube root* would result in such an error.)

Table 1



The Hand The hand is the user's primary means of interacting with the system. It is used to activate buttons and to carry objects. At left, the hand is shown in its normal "relaxed" position. At right, the hand is shown carrying a button. The shadow being cast indicates that the hand and its held object are in *Meta Reality*.

Simple Buttons User inputs originate through buttons. Buttons can be picked up and dropped on objects, then activated by the hand. Buttons bear the name of the message they send (e.g., the Xerox button is for making copies). Non-simple buttons (which require parameters, or which capture a response to a message) are depicted in Figures 3 and 4.

The Warehouse The Warehouse icon "contains" one of every kind of object in the system, including those in the Smalltalk environment in which ARK is implemented. The *objectMenu* button creates a list of all the kinds of objects from which the user can select. The selected object is emitted into the alternate reality from the warehouse.

Representatives Any Smalltalk object can be made to appear within ARK. If it is not itself a kind of ARK object, it will appear inside a representative object. A representative has an image or some text describing the object being represented.

Slider Controls Sliders are a convenient way to specify numbers. The hand can grasp the tab in the center and move it along the groove.

Interactors Interactors are the physical manifestation of the normally intangible rules of nature. The physical manifestation provides a place for interaction with the rule, as well as a tactile presence that denies the mystical character normally attributed to these universal laws. These highly magical objects are of central importance in the ARK strategy for enabling intuitive understanding of a simulation's interaction rules.

While the manipulability of the buttons is useful, these aspects require a few sentences of explanation. The fact that a button will "stick to" or "fall through" an object depending on the object's ability to respond to the button's message is a bit of a tale. Uninformed users have sometimes discovered accidentally that buttons can be picked up from an object; they are slightly startled. Some of these users wonder if a button will still work when removed from the surface of an object and dropped off to the side. Some have dropped the button onto the surface of a large, non-understanding object, only to have the button "disappear." (Actually, the button is lying underneath the object, having fallen through.)

Interactors: In ARK, an interaction law of the simulated universe (Newton's law of gravity, for example) is represented by a "physical" object called an *interactor.* Interactors form an interface between the user and the fundamental laws of the simulation.

Magic vs. Literalism: Highly Magical

Power vs. Learnability: Interactors are the physical embodiment of normally intangible abstractions, and are therefore important actors in the story of how the design of ARK attempts to provide intuition by making things concrete. In enabling the user to change physical laws, interactors provide capabilities fundamentally beyond those suggested by the real-world metaphor.

Most users have no trouble accepting the idea of a "control center" for gravity, for example. However, they do need to be explicitly introduced to the idea that an interactor object represents some abstraction. On occasion, users have had difficulty in understanding exactly what abstraction is being controlled. These users may require a minute or two of discussion to clarify the interactor's role in the simulation. While this is a short time in absolute terms, it is much longer than is required by any literal feature.

Multiple Realities: When regarding the computer screen, the ARK user sees one or more possibly overlapping rectangular windows. Each window represents a separate *alternate reality.* To move between alternate realities, the user simply moves the hand over the exposed portion of the window, and that window will automatically become completely exposed, moving to the "top of the stack" of overlapping worlds. The user can carry objects between the various alternate realities.

Magic vs. Literalism: Highly Magical

Power vs. Learnability: The use of overlapping windows brings the usual advantage of enlarging the virtual screen area. The provision of multiple realities enables users to organize their ARK tasks -- for example, one world can be used for building new objects, another for trying them out. For example, two side by side windows can facilitate comparison of separate worlds, each with its own speed of light, for example.

Users understand the idea quite readily. A brief description and one or two trials are required before most users are comfortable with their ability to go "reality hopping." However, users who accidentally bring a buried alternate reality to the top are startled. In certain applications, objects may drift off the window, disappearing under the edge. Users are sometimes concerned or amused at this -- it is not always clear what has happened to the object. Can it be retrieved? Does it still exist? (In fact, the coordinate system for each window world is indefinitely large, and an object can go as far as is allowed by the computer's ability to allocate new words for storing its growing x and y coordinates.)

As with the interactors, the absolute time to teach the use of overlapping worlds is short, but is longer than time taken by a typical literal interface feature.

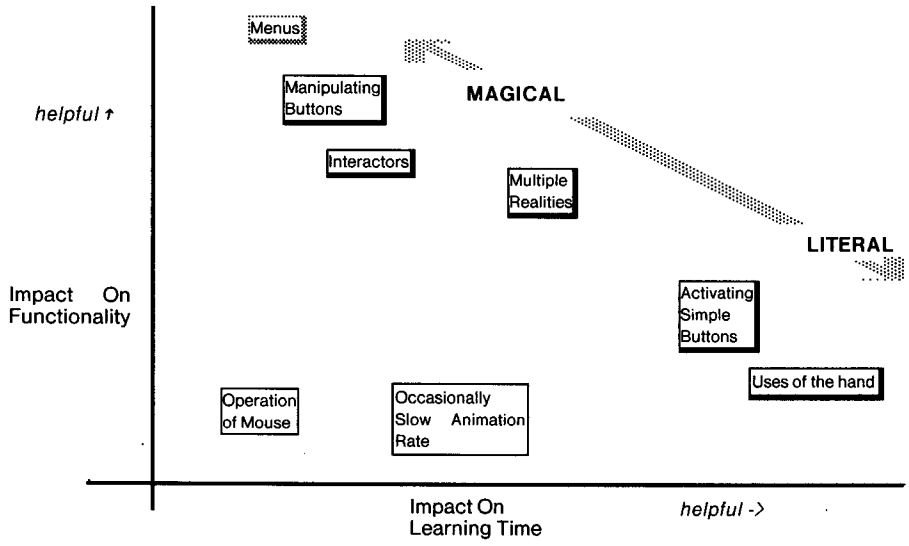


Figure 5. Various aspects of the ARK interface are represented on a graph in which easier to learn things are to the right, and very useful things are toward the top. Interface aspects tend to lie on a line with magical features in the upper left, and literal features in the lower right. The existence of a void in the upper right is a sign of the fundamental tension between literalism and magic: interface designers can always provide more powerful functionality at the price of violating the metaphor. Sometimes a broken metaphor is not particularly enabling, as evidenced by ARK's occasionally slow animation rate and use of the mouse. (Menus have been included here even though they have only rarely been used as part of the user experiments with the system.)

The preceding list demonstrates the inverse correlation between power and learnability. Interface aspects that are literal are easily understood -- in fact, some important parts of the literal functions are simply assumed by the user. Those aspects that are magical are quite useful, but require the majority of the training time. ARK's limited use of magic does not prove too confusing for novices, and the total teaching time remains quite short. After a few minutes of explanation, most novices are able to use the capabilities outlined above.

Conclusions and Questions

I have used the magic-literalism tension to discuss some of the central features of the Alternate Reality Kit because it serves as a useful way to analyze some of ARK's design issues and user experiences. In particular, because a large portion of ARK is at the literalism end of the spectrum, many of the important aspects do not need to be explicitly explained. Furthermore, although each magical aspect requires its own explanation, limited use of magic in ARK keeps the total teaching time quite low. User experiences indicate that applications-level functionality can be taught in a few minutes. Features that are neither magical enhancements nor literal adherents to the metaphor are the most troubling in ARK.

The magic vs. literalism tradeoff may be an interesting perspective on other systems whose interfaces are centered about a single metaphor (such as other graphical programming environments or desktop-like window systems). Some questions are raised by this way of viewing metaphorical user interfaces. How does the designer decide when to implement a capability magically instead of literally? Literalism can be carried too far. When does an interface become so literal that it surprises even novice users? Building systems with both literal and magical ways of doing the same task may enable users to move smoothly into wizard status -- how is this best done? What is the minimum set of magical capabilities that allow users to build their own magic spells? Is it possible to find a metaphor that puts all external factors along the literalism-magic spectrum? Questions like these can lead to interesting discussions of ways to enable controlled release of the magic latent in computers.

Acknowledgements

The following individuals were sources of stimulating discussion and thoughtful comment: Sara Bly, Adele Goldberg, George Goodman, Laura Gould, Jane Laursen, Tim O'Shey, Dave Robson, and Frank Zdybel.

References

1 Bewly, W.L., Roberts, T.L., Schroit, D., and Verplank, W.L."Human Factors Testing in the design of Xerox's 8010 'Star' Office Workstation." *Proceedings of the Conference on Human Factors in Computer Systems.* (Boston, December 1983) pp. 72-77.

2 Borning, A.H. The Programming Language Aspects of ThingLab, a Constraint-Oriented Simulation Laboratory. *ACM Transactions on Programming Languages and Systems* 3(4):353-387, October 1982.

3 Card, Moran, and Newell *The Psychology of Human-Computer Interaction.* Lawrence Erlbaum Associates, Hillsdale, New Jersey, 1983.

4 Carroll, J.M. and Mack, R.L. Metaphor, computing systems, and active learning. *Int. J. Man-Machine Studies* (1985) 22, pp. 39-57.

5 Carroll, J.M. and Thomas, J.C. "Metaphor and the cognitive representation of computing systems." *Yorktown Heights, N.Y.: IBM Watson Research Center technical report no. RC 8302.* (1980).

6 Gentner, D. "The structure of analogical models in science." *Report 4451* (Bolt, Beranek, and Newman, July 1980)

7 Gittins, D. "Icon-based human-computer interaction." *Int. J. Man-Machine Studies* (1986) 24, pp. 519-543.

8 Goldberg, A.J., and Robson, D. *Smalltalk-80: The Language and its Implementation.* Addison-Wesley, Reading, MA, 1983.

9 Gould, L., and Finzer, W. "Programming by Rehearsal." *Technical Report SCL-84-1* (Xerox Palo Alto Research Center, May, 1984). A short version appears in Byte Magazine, Vol. 9 No. 6, June 1984.

10 Halasz, F., and Moran, T.P. "Analogy Considered Harmful." *Proceedings of the Conference on Human Factors in Computer Systems.* (Gaithersburg, Maryland). March 1982.

11 Norman, D.A. "Design Principles for Human-Computer Interfaces." *Proceedings of the Conference on Human Factors in Computer Systems.* (Boston, December 1983) pp. 1-10.

12 O'Shea, T. and Smith, R.B. "Violating the Laws of Nature: Experiments in Understanding Physics by Exploring Alternate Realities." To appear in *Proceedings of the 1987 Computer Assisted Learning Conference.* (Glasgow, 1987).

13 Price, L.A., and Cordova, C.A. "Use of Mouse Buttons." *Proceedings of the Conference on Human Factors in Computer Systems.* (Boston, December, 1983) pp. 262-266.

14 Smith, R.B. "The Alternate Reality Kit: An Animated Environment for Creating Interative Simulations." *Proceedings of the 1986 IEEE Computer Society Workshop on Visual Languages.* (Dallas, June 1986) pp. 99-106.