Multi-tasking and Cost Structure: Implications for Design

David Kirsh (kirsh@ucsd.edu)
Department of Cognitive Science 0515
La Jolla, CA 92093-0515

Abstract

I argue that it is not possible to accurately represent our task settings as closed environments with a single well defined cost structure. Natural environments are places where many things are done, often at the same time, and often by many people. To appreciate the way such invariants of everyday life affect design I present a case study, a micro-analysis of espresso making at Starbucks to show the challenges facing a cost structure approach.

Introduction

Almost all natural environments are places where many things are done, often simultaneously. How shall we conceptualize the structure of such environments? Designers interested in analyzing and re-engineering artifacts, layouts and settings, expect cognitive science to deliver representations and theories about agent-environment interaction that will help them understand real environments. But interruption, multi-tasking, and workspace sharing, pose special problems for most theories of action.

The approach I shall evaluate is an extension of Herbert Simon's seminal ideas about a science of design (1997). Simon argued that if we assume that agents' make rational choices given their preferences and information, as well as cognitive and physical limitations, we can predict, at least in principle, how changes in an environment will lead to new equilibrium states of agent environment interaction. Although this is a radical extension of economic methods to non-economic environments – to activity spaces where agents respond to a universe of non-economic signals – it is one I personally find compelling and one which has driven a healthy research program in HCI. Card et al (1994), Russell (1994) and others since then have used this principle successfully to explain many of the regularities we find in the behavior of information seekers foraging for high value content in large information spaces, and to offer plausible accounts of why one application interface or one visualization of complex information is better than another (Kirsh 2005). It has also been used with success by ethologists and ecologists to explain regularities in the way animals behave given detailed descriptions of the niche they inhabit.

The heart of the idea is that the resources in a setting define a particular cost structure for an activity. When applied to distinctively human settings resources need to include technologies and artifacts, as well as layouts, surfaces and other humanly modified physical features. To compute a cost structure for an activity a cost function is applied to each task relevant action at each task relevant state. This focus on task relevance is necessary if there is to be a principled basis for a metric evaluating the costs and benefits of actions. Environments are conceptualized as spaces where an agent tries to perform a task, and all that really matters are the features or properties of that space that can support or hinder task performance (Newell & Simon 1972). The role of technology and design can then be understood as shaping the cost benefit structures of activity in that space. One design will be better than another if it enables completing a task faster or with reduced error. Better designs have better cost structures.

It may help to give an example of costing an environment before I critique the approach. My own favorite case of how technology has changed the cost structure of activity is the way remote control changed the activity of watching TV.

The obvious activities to cost in TV environments are the efforts associated with turning the TV on and off, changing channels and changing volume. Other costs, such as those associated with discovering the time and channel of programs, or with controlling aspects such as muting, titling or color intensity we can treat as of secondary importance.

Prior to the introduction of remote control the cost to change channels depended on the distance from seat to TV, the design of the channel switch on the television, and the speed a viewer can move. The greater the physical distance, slower the viewer, or the more channels, the greater the cost. A designer intent on improving the cost structure of watching TV would think about changing one or more parameters: moving the couch closer to the TV, adding random access to the channel dial. An even better solution would be to provide a hand sized controller that allows a viewer to control the TV from anywhere in the room without moving. In each case the objective of design is to lower the costs of some aspect of TV watching activity. Competing designs are evaluated by how much they lower costs.

Since adaptation to a new cost structure is an equilibrium process it takes time for agent environment interactions to stabilize. But once a new equilibrium is reached the design process can start over again. In TV watching the next cost to be reduced was the previously non existent routine of channel surfing. Remotes soon began offering 'last' buttons to allow for quick toggling between channels to allow 'timing' the advertisement. More radically, TV's themselves were changed by adding 'picture in picture' to support watching two channels at once. Again the force driving innovation was the desire to lower the cost of TV watching activities.

Kirsh, D. (2005). Multi-tasking and Cost Structure: Implications for Design. In, Bruno G. Bara, L. Barsalou, & M. Bucciarelli Mahwah (Eds.), Proceedings of the 27st Annual Meeting of the Cognitive Science Society. (pp. 1143-1148) Austin, TX: Cognitive Science Society.

Two Problems With Cost Structures

There are two problems with the cost structure approach that I will consider. The first concerns the metrics used to cost activity. In order to get a single number to use as the cost of an action at a state, Simon, as well as other theorists, suppose that all the factors that go into a cost can be compared. I argue that in Starbucks and other café environments we need several orthogonal cost metrics.

The second problem concerns the assumption that all the relevant actions which agents perform in a setting can be assigned values relative to one task environment or another. In realistic environments where people do more than one task, are not always by themselves, and occasionally have other things on their minds some of their actions are external to all task environments. They are 'inter-task' actions.

For instance, the best remotes should accommodate the realities of use, such as, being interrupted, multi-tasking, sharing the remote, operating in the dark and other aspects of real world environments where errors creep into performance because of 'extra-task' factors. The need to accommodate these additional elements puts a limit on the empirical adequacy of all cost structure accounts as currently practiced.

Ordering & Preparing Drinks: Starbucks

To study how the problems of multiple costs, interruption, multi-tasking and multi-person coordination affect design, I have used various ethnographic methods, to observe routine activity in cafés, specifically the interactions during ordering, communicating and preparing espresso-based drinks at Starbucks. Starbucks does not permit filming so we brought baristas into the lab to interview them as they drew the layout and activity flows of their workplace on a large whiteboard.

In any café there are five main tasks baristas must perform: (1) interact with clients to specify and record their order; (2) take cash and make change, offer a receipt (step 2 may occur after step 3); (3) communicate the order; (4) prepare the order; (5) announce completion of order and queue drinks for clients to collect.

In addition to these obvious steps there are six or seven support activities such as maintaining supplies of cups, milk, coffee beans, grinding, cleaning up and so on, that are not always visible to clients.

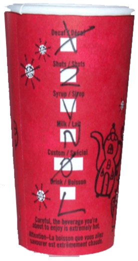
There are good practices for all these activities and each step can be viewed as a task with an associated task environment and activity space in which skilled agents have learned routines for efficient performance. There are routines and standard operating procedures for taking orders, making cash, passing the order to a barista; of course there are routines for preparing orders, for queuing drinks, and for maintaining the requisite resources and the general environment in which all these activities take place. Each step corresponds to a functional role that someone has to learn or be trained to fill.

The complication with this attractive and theoretically tractable view of a collection of distinct tasks and task environments is that in cafés the staff has to complete their tasks in the same small physical space behind the counter, and usually at the same time. Each person has several tasks and often they multi-task. In typical shifts during busy moments there are four or five people behind the counter, two to take orders, two to make the drinks, and occasionally a floater to help whomever is pressed. The counter space serves multiple functions. It is not uncommon for one person to reach over another, or in spare moments to offer help. For instance, maintaining the milk temperature in frothing pitchers, cleaning a portafilter or the counter space, or restocking beans are all tasks that anyone with the requisite skills and a free minute can perform. Because of this dense sharing of physical space an individual agent working on his or her own task will often change the state of a surface also being used by another, and change things in a way that impacts on the other's task. Sometimes this is anticipated, sometimes not.

Starbucks is an intense environment. But it is not unusual. Interruptions occur because people talk and listen to each other all the time. Baristas at the bar – the name for the region immediately encompassing the espresso machine, its top, the surface directly in front, the immediate surface to the right and left where the frothing pitchers are kept – always listen to the next order being taken at the register. They want to be prepared. They ask each other for help or divide tasks: "you pull the shots while I steam the milk". They remind each other of forgettable specifics: "Don't forget it's Soya milk". These sorts of reminders also may come from the register person: "That's sugar free vanilla." Baristas reach over each other: bringing in clean shot glasses or cups; they share the same space, occasionally taking a shot (1 – 1.5 oz of espresso) just pulled by someone else. When one makes an error another may notice it, or help recover. For example, a shot that has been sitting too long – sometimes as little as 40 seconds – may be thrown away and another pulled to take its place.

Work in such a confined space where so much is going on and where efficiency is so highly valued demands tight coordination. In some modern cafés coordination occurs by means of a monitor above the espresso machine displaying drinks and queue structure. But at Starbucks the fundamental coordinating mechanism is their paper cup.

The Starbucks paper cup has a printed form on one side whose 6 fields each support a fixed vocabulary of symbols. Fields are filled to indicate the specific ways in which an order deviates from the default. A standard Grande Cappuccino would have only one symbol – C – marked in the bottom field to indicate drink type. It would receive the default values of two shots of caffeinated


Decaf / Décaf
X
Shots / Shots
2
Syrup / Sirop
V
Milk / Lait
2
Custom / Spécial
0
Drink / Boisson
L
Careful, the beverage you're about to enjoy is extremely hot.
Attention–La boisson que vous allez savourer est extrêmement chaude.

espresso, full fat milk and a normal amount of froth.

The Starbucks cup is a remarkable technology. It tightens the interactions between workers by reducing the number of places in the system where things can go wrong. Because the barista at the register selects the right sized cup, one parameter – size – can be ignored. Because the barista at the register places each marked cup in the queue building up beside the espresso maker, a second parameter – queue position – can also be left implicit. And because the cup encodes its specification and often the customer's name, baristas have a mechanism for checking that the right drink is picked up once it is placed in the collection zone. So many functions for such a simple technology.

Increased Robustness Not Reduced Effort

Does the Starbucks cup actually lower cost structure? That depends on how cost is measured. The greatest virtue of the cup is the way it increases the robustness of production. This focus on robustness is not the same as a focus on performance time or effort. Order takers at the register still need to encode orders and pass them on to baristas at the bar. Arguably, it takes longer to write an order than to call it out. So filling in a cup does not save effort or time at the register. Similarly, it takes longer to read a specification on a cup than to fill an order from memory. So reading a cup does not save time.

But of course the cup more than pays for itself because of the errors it prevents and the time saved in recovering from errors. Given the prevalence of interruption, distraction and multi-tasking in cafés, process modifications that insulate baristas from the cognitive and physical consequences of disruption are highly valued. Insulation is even more important when a barista is relatively inexperienced and has yet to develop the internal strategies and expert memory of veterans.

To capture the virtues of the cup let us distinguish errors that affect quality from errors that affect time. For instance, drinks suffer when espresso is pulled longer than 28 seconds, milk is heated above 170° F, or a shot of decaf espresso is used when the customer ordered it caffeinated. These errors affect quality. A well considered cost or cost benefit function should include factors that affect quality. When a barista knocks over a drink, however, or bangs her thumb, or runs out of milk in the nearby fridge, output is delayed but quality is unaffected.

Because the cup changes the probability of meeting specifications it improves aspects of the production process which affect quality. If we treat loss of quality as variance from the ideal, then the goal of reducing variance is a different cost dimension than speed accuracy. The two are independent because quality can rise while speed accuracy remains constant.

Another function of the cup is to improve error recovery. The most damaging effect on production time occurs when one of the baristas is burned by scalding coffee. The cup he or she has been working on is dropped and spills, others stop what they are doing and offer help; generally the whole system spasms and breaks down. It is a huge interruption for everyone. Had requests been given orally most of the orders would likely be lost at this point. Had they been written down on paper the ink might smear, papers get lost or confused with others. But with the Starbucks cup simply by picking up the cup(s) that were dropped a new barista can identify the order or orders that were ruined. Each cup still has its specification, so the new barista can restart the process using fresh cups of the right size and maintain the queue order intact. The cup is the great coordinator of the Starbucks espresso making process, moving along with production, rather than being just another resource to use up and throw away.

Is recovery time a dimension of cost that can be merged with speed accuracy – the major measure of efficiency? One reason to keep them apart is that they operate on different parts of the production process. Techniques and artifacts that increase the speed of production when all goes well are quite different than those that help recovery once accidents occur.

Another important factor to consider when designing environments concerns vigilance: how easy is it to monitor the process? A process that offers greater opportunity for vigilance, greater chance of detecting when things seem to be going off track is to be preferred. Being vigilance friendly is an attribute that fits well into speed accuracy, but is conceptually distinct.

In economic models, and in current HCI models, costs are not differentiated. Typically everything is reduced to a single dimension of expected time or effort. Yet distinguishing cost types is of real concern to technologists because each has important design implications. For instance, a well designed environment will reduce error, improve quality and increase speed by:

- making it easy to track the current state of a process – easy to monitor the key parameters that matter to success – quality control, speed, error;
- allowing users to back off or abort a process up to the last second without harmful consequences, thus stopping serious error before it goes too far;
- making it easy to prepare for difficult moments as when the café is crowded and everyone is working at their fastest;
- having checks in place so when errors do occur the recovery costs are low – a safe fail design.

Such concrete maxims for design follow from a better specification of cost dimensions. Discovering such dimensions requires micro-analysis of activity.

Complexity And Cognitive Costs

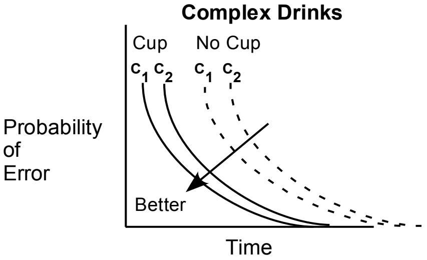
I have been arguing that to save the cost function approach to environment design it is necessary to find the right parameters to assign costs to. Determining these costs requires a micro-analysis of the activity involved in using the technology, at least in simulation. Errors have to be anticipated and analyzed, recovery methods observed, and attention has to be given to the way team members coordinate their activity and respond to changes in load. A second lesson the Starbucks cup offers is that the key cost to lower may have more to do with cognitive costs than physical costs. This is nicely illustrated by the increase in drink complexity that the Starbucks cup system facilitates.

At modern cafés drink complexity has risen so dramatically that it is no longer expected that an order will be as simple as 'One grande latte'. For example, a client may now request a large cappuccino made from non-fat milk with an extra shot of decaf espresso, more froth, a standard dose of sugar free hazelnut syrup, a drop of vanilla syrup, and a request that the drink be served at a cooler temperature. The customer himself may then garnish the drink with a few shavings of chocolate or powdered sugar. For the attendant on cash to call out this order to a barista – the tradition in classical European cafés – takes an unacceptably long time, and puts an unacceptable cognitive burden on the barista, who may well be in the midst of making another drink. Any number of errors can creep into this oral process: a miscall by the order taker, the barista forgetting specifics, or confusing some parts of the next order with the present order and ruining the drink currently being prepared. Once confusion has occurred, moreover, there is no easy way of recovering the details of the order because there is no persistent record to review.

As argued earlier the cup does not increase production speed per se. The time it takes to produce a generic grande latte is more or less the same whether the order is requested using the cup, orally, or on a display board. Shots still have to be pulled, milk frothed, poured and so on. The division of labor is somewhat improved by having the register person select the cup, and as we mentioned earlier, the probability of confusing orders and making errors during hectic moments is decreased, which is a major cost saver. But time per drink, the basic speed accuracy curve for preparing a generic drink exclusive of error recovery time, remains about the same.

There is a clear cost reduction on the cognitive side, however, when we consider the cognitive effort vs. drink complexity tradeoff. When a drink's specification is on the cup the cognitive cost of reading three extra specifications is not much greater than reading two or one. If a barista cannot remember whether the third shot is supposed to be decaf then she can consult the cup. The same goes for whether the syrup was sugar free. Since production can proceed incrementally – read the cup, execute the operation, read, execute – the memory load on the barista remains constant. An interruption, therefore, only affects the current ingredient. Even if the barista loses her place, the combination of specification on the cup, layout and disposition of equipment, and visible state of the cup, provide enough situational information for the baristas to 'see' where they are and pick up the process. This means that product quality will rise because now drinks of greater complexity can be delivered in acceptable time and within quality tolerance.

Graphically, the effect of the cup can be imagined as shifting toward the origin the speed accuracy curves for producing drinks of greater complexity. The more effective a complexity reducing technology is the more the cost profile of complex drinks will resemble generic drinks. The cup technology lowers the cost structure of more complex drinks. Since making such drinks are among the tasks a barista must perform, an adequate cost function would assign a cost structure to making drinks as a function of their complexity and weight different drinks by their importance to the user.



What Is The Environment Of Activity?

So far, I have considered ways to enhance cost functions in order to preserve the core idea that effective technologies reshape the cost structure of activity. To save the cost structure approach I argued that the range of costs that must be measured is far greater than standard cost accounts mention and it is not obvious that all these different dimensions of cost can be meaningfully reduced to a single number such as overall cost, goodness, or fitness. One environment may be better than another along some dimensions worse along others: costs need to be traded off.

One reason to be skeptical of the entire approach, though, is that the assumptions needed to define cost functions seem to contradict most of the insights derived from situated, distributed, and interactive cognition. This argument comes in two forms. The first is easy to dismiss. In studying situated activity, we are told, the devil is in the details. Cost functions, with their coarse, quantitative and objectivist approach, neglect practices and other 'cognitive' factors that partially determine how users think and behave. There is no such thing as an objective cost function definable in abstraction from users' cognitive practices and processes in concrete settings.

This is not compelling. In my accounts of recovery cost, complexity and cognitive load, variance and monitoring I have tried to show how qualitative and cognitive factors can be incorporated into cost functions. Clearly more needs to be done, but in showing the diversity of cost dimensions we have at least dulled the first argument.

A second and better reason to be skeptical of cost function analyses, however, is that they are applied to single tasks and to scale up to multi-tasking they make an assumption about the additivity of costs that rests on a false assumption about the linear separability of tasks and task environments. Since the real world rarely poses tasks singly this is a strong argument. Indeed it is even a challenge to single tasks because any task involving sub-goals or conjunctive goals can be interpreted as a multi-task context. Multi-tasking does not require the multiple tasks to be unconnected. In Starbucks, when a barista reaches for a second shot glass with her right hand while using her left to fill up the first shot glass she is multi-tasking among connected tasks. The second shot glass is needed to complete the current drink, though it need not be fetched while the first glass is being poured. Another example arises when a barista reads the symbols on the cup s/he is currently filling. The two activities, reading and pouring are sub-tasks of the drink making process but may be worked on at the same time. If there is something about multi-tasking that ruins cost analyses the whole approach of using cost structure to inform design will be jeopardized.

The strongest grounds for challenging cost function analyses then is that they rest on a false assumption about multi-tasking. In classical cost analyses when one or more agents multi-task in the same space, it is assumed to be acceptable to suppose that each task occurs in its own task environment, and that multiple task environments can be superposed on the same physical space. Since superposition implies additivity (and homogeneity) it is permissible to add the costs and benefits of achieving outcomes in each task and talk about the total costs involved in performing multiple tasks. It is this assumption that is false.

Superposition of Task Environments

Here is a quick argument to show why task environments have to be superposed. Begin by assuming two tasks can be performed in the same space only if they can be completed successfully in that space. This implies that it is possible to complete each task without making it impossible to complete the other. If progress in task A disrupts progress in task B, it must be possible to recover one's state in A without disrupting the state in B. If this were false, the tasks could enter a self-destructive loop and attempting both tasks could be self-defeating. For example, attempting to cook pizza and peach pie in the same oven, even using separate pans, is self-defeating because the temperature needed to cook pizza burns pie, and the temperature needed to cook pie undercooks pizza. The goals of one task clobber the goals of the other.

The simplest way to deal with goal or sub-goal interaction is to define a task broadly enough to include consideration of all goals and then search for a viable path in that more encompassing task environment. Multi-tasking in a single task environment satisfies the superposition condition when agents correctly order their sub-goals to support simultaneous pursuit of those goals.

But if the tasks remain separate then coordination between tasks becomes problematic because there are no opportunities inside core tasks to protect other tasks from negative side effects. Coordination between tasks is not a goal of any task. Accordingly, agents who do perform coordinating actions are stepping outside their task environments and performing extra-task actions. Since extra-task actions, by definition, lie outside the scope of cost functions it follows that a cost function approach will yield unreliable costs when coordination is required.

By assuming tasks can be superposed we guarantee that costs can be added. Formally, two tasks are superposable if any trajectory T_A of states created by applying operators in task A in a vector space (defined by property values at points in an environment) never intersects a trajectory T_B of states created by applying operators in task B. This implies that if A and B are performed in the same space and time changes caused by A can be distinguished from changes caused by B. If the states of tasks can't be so distinguished agents might lose their place in each task, or the tasks might destructively interfere with each other.

The assumption of superposition and hence linear separability also implies that the outcomes of the two tasks can be separated as if they had been completed at different times. For instance, if the dual task is to prepare spaghetti for a first course, and chicken with rice for the second course, then if we cook the rice and spaghetti in the same pot of water, it is not feasible in any reasonable sense, to separate them afterward. Hence we cannot simultaneously perform those tasks in the same space.

An easy way to visualize the failure of superposition here is to reflect on the way waves interact when they move through the same region of space. Waves are superposable because they combine in a well-defined manner, maintaining their integrity when they overlap. They can pass through each other without being permanently changed. When cooking pasta and rice, the two processes do not 'pass through' each other. Given the operators for extracting pasta – a strainer or pasta claw – and the operators for extracting rice – a spoon – it is not possible to separate the two outcomes in the task environments defined by each set of operators..

Let us call tasks which can be physically separated and accomplished in the same space or with the same resources, physically separable. If two tasks are physically separable, they are in a sense, physically modular, and we expect them to pass the linear separability requirement. They can be analyzed in their respective task environments and their costs added.

Physical interaction between sub-goal states is not the only way one task can produce unwanted side effects in another. A second method of interfering with a task is to disturb the cues, affordances, constraints or symbols relied on. For instance, when an item in task A serves to remind an agent to perform a certain action, or when the position of an item is a landmark which the agent relies on to know where s/he is in a task, then disruption or displacement of that item by activity in task B affects the 'informational state' of A. Assuming the placement of the item is not an actual sub-goal of task A but rather a useful consequence of a goal directed action, then no sub-goal in A, no task state, has been physically obstructed by task B. Both tasks A and B may still be successfully completed providing the agent has enough knowledge to overcome the absence of situational cues. But because important informational elements of the state have been lost, the agent cannot complete task A the way s/he would have were A performed in isolation. In such cases, A and B fail to be informationally separable.

Given that the perception and extraction of information about a state is partly independent of the operators determining the state space in a task environment, information separability does not imply physical separability (.e.g. rice and noodles are informationally but not physically separable) and physical separability does not imply informational separability (e.g. cup position).

Co-located multi-tasking requires physical separability and to be robust also requires informational separability. If two tasks are not separable, then the actions taken in one may lead to incorrect decisions in the other.

What do people usually do when collocated environments become inseparable – when they fail the superposition test? They add structure! They make them superposable. At its simplest this just involves recall, projecting 'mental' structure onto a scene: 'I remember I put down two clean cups over there, I see one of them is still there.' More broadly, though, there are hundreds of mechanisms for adding structure physically. In chess if one player forgets whose turn it is, and so the game becomes inseparable for him, he can ask 'whose move is it?' This linguistic move was not a defined move in the chess task environment which is solely concerned with movement of pieces. Similar extra-task moves are found in almost every task-oriented activity. In managing a desk an agent can break the confusion that arises from multi-tasking by adding marks or annotations to documents. Intentional placement of resources can help disambiguate task elements, or specific cues or reminders can be laid out to indicate what still needs to be done.

The significant element in all these actions is that they are task external actions that fall outside the cost function of the task. They are inter-task or coordinating actions., performed to enable tasks to be kept separate. Since these actions are task external but can significantly affect costs and benefits, cost functions are fundamentally incomplete.

Conclusion

Throughout this paper I have been considering an intuitive account of technology that treats it as a force which shapes and reshapes the cost structure of activity. These costs can be represented by curves showing trade-offs in speed-accuracy, variance, recovery time, complexity and so on. Cost functions formally depend on treating the setting of a task as a task environment, with sparse choice points, circumscribed option sets and so on. I argued that task environments are not realistic containers for tasks if we recognize the universality of such events as interruption, disruption, multi-person activity, and multi-tasking in the same physical space. Task environments are unrealistic because extra-task events intrude on behavior and inevitably have an impact on how easily, how reliably, how precisely an agent can do a task. Some of these extra-task factors can be accommodated indirectly in cost functions. But there are times when multi-tasking causes side effects in each task in such a way that the states of the tasks are inseparable, agents become confused. If states cannot be separated then decision errors can be expected to creep into task performance. To insulate themselves from these unwelcome intrusions human agents keep tasks separate by performing a wide range of non-pragmatic actions, actions that are external to the option sets formally defined in each task. Since these actions are important to task completion they should be part of the task, but they are not. Hence they cannot be considered in cost functions. This puts an upper limit on the usefulness of cost functions.

This limitation on cost functions should not be a surprise. Given the importance which micro-analyses of activity have for designers the concept of a cost function is too coarse to serve as more than a rough explanation for the success of certain technologies. Perhaps a more ecological notion of cost structure will work. In that case, the only real cost structure is the one that accommodates all tasks, including coordination tasks. This more holistic approach saves the idea of viewing design as an evolutionary process but violates the assumption of task superposition. This may be an improvement but we still face the daunting problem of reducing the many dimensions of cost to a single metric of overall cost.

Acknowledgments

I am grateful for support by the ONR under grant N00014-01-1-0551, and for helpful conversations: Rick Alterman, Aaron Cicourel, Peter Gärdenfors.

References

Card, S., et al., (1994) The Cost-of-Knowledge Characteristic Function: Display Evaluation for Direct-Walk Dynamic Information Visualizations. *Proceedings of ACM Conference Human Factors in Computing Systems, CHI*

Kirsh, D. (2005) Metacognition, Distributed Cognition and Visual Design, *Cognition, Education and Communication Technology (eds.) P. Gärdenfors & P., Johansson, Lawrence Erlbaum*.

Newell, A., & Simon, H.A. (1972). *Human Problem Solving.* Englewood Cliffs, NJ: Prentice-Hall.

Russell, D. M., et al., (1993). The cost structure of sensemaking. INTERCHI '93, *ACM Conference on Human Factors in Computing Systems*.

Simon, H.A. (1997). *The Sciences of the Artificial* (3rd ed.). Cambridge, MA: The MIT Press.