Metrics for Evaluating Human-Robot Interactions

Dan R. Olsen, Michael A. Goodrich
Computer Science Department
Brigham Young University
Provo, Utah 84602
{olsen, mike}@cs.byu.edu

ABSTRACT

Metrics for evaluating the quality of a human-robot interface are introduced. The autonomy of a robot is measured by its neglect time. The robot attention demand metric measures how much of the user's attention is involved with instructing a robot. The free-time and fan-out metrics are two ways to measure this demand. Each of them leads to estimates of the interaction effort. Reducing interaction effort without diminishing task effectiveness is the goal of human-robot interaction design.

INTRODUCTION

Autonomous robots that can perform a variety of tasks with no human intervention are an interesting but ultimately marginal goal. What we really want are robots that can do what we want when we want it, not whatever they want whenever they want it. We are not interested in producing alternate life-forms. We are interested in effective servants. We want devices that will leverage human attention and human ability. In this paper we ignore the leveraging of human physical abilities and focus on the leverage of human attention.

In this paper we present a series of metrics for measuring the effectiveness of robots as servants of their human masters. In particular we are looking for measures of interface effectiveness that capture our desires to leverage human attention

The first metrics are those that measure task effectiveness (TE). Task effectiveness is some measure of how well a task is actually performed. At the end of the day we care mostly about getting some task done. In driving or navigation scenarios we might measure effectiveness as the time required to get from point A to point B. In search tasks we could measure the time to find all targets or the number of targets found in a given amount of time. In an assault task we might measure targets destroyed and losses taken.

Ultimately task effectiveness measures are key to successfully designing and evaluating human-robot teams. However, task effectiveness measures do not shed any insight on how to improve the human-robot interface or how that interface might be modified to increase the effectiveness. We believe that metrics must be based in a framework that guides design. We are looking for an engineering approach that leads us through a space of design alternatives to a human-robot interface that enhances the task effectiveness of the team.

In this paper we will discuss six interrelated metrics that can guide the design of human-robot interaction. They are task effectiveness (TE), neglect tolerance(NT), robot attention demand(RAD), free time(FT), fan out (FO) and interaction effort (IE). These metrics are somewhat generic and are instantiated differently for different robot tasks. However, together they provide a framework for thinking about interaction design.

TASK EFFECTIVENESS

As mentioned earlier, task effectiveness is a measure of how well a human-robot team accomplishes some task. There are a variety of such metrics and for the purpose of our framework we do not care what metrics are chosen. There are time-based metrics that attempt to maximize the speed of performance, error metrics that attempt to minimize mistakes or damage, coverage metrics that measure how much of some larger goal is achieved, as well as other possible metrics. The overarching goal is that effectiveness is maximized,but the details are task specific.

In some of the scenarios presented below, we will need to differentiate between overall task effectiveness and current task effectiveness. Overall task effectiveness is best measured after the task is complete. An example would be the time required to accomplish the task. In many situations we need a measure of current task effectiveness which is the effectiveness of the robot right now. Such a measure might be the speed with which the robot is closing the distance to a goal. The problem with measures of current task effectiveness is that they can be very wrong. A robot might be getting closer to the target very rapidly and yet be wandering into a cul-de-sac from which it will need to back out. It currently appears to be effective but on the overall goal it is making negative progress.

NEGLECT TOLERANCE

A very important metric in measuring the autonomy of a robot with respect to some task (and corresponding task effectiveness metric) is the robot's neglect tolerance (NT). Neglect tolerance is a measure of how the robot's current task effectiveness declines over time when the robot is neglected by the user. We hypothesize that for a given robot and a given problem space there is a characteristic neglect curve such as that shown in figure 1.

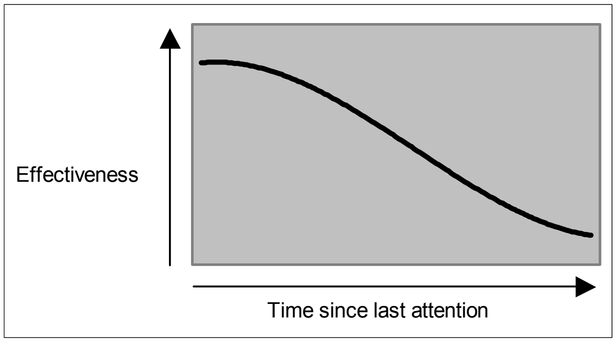

Figure 1. – A Characteristic Neglect Curve

This curve shows that the current task effectiveness of the robot reduces as a function of the time since the user last paid attention to the robot. For a simple navigation problem we can define current task effectiveness as the speed with which the robot is making progress towards a goal. We can establish an acceptable minimum effectiveness threshold and using the characteristic neglect curve we can define the neglect tolerance as the time that can expire before the robot's effectiveness drops below the acceptable minimum. This is shown in figure 2.

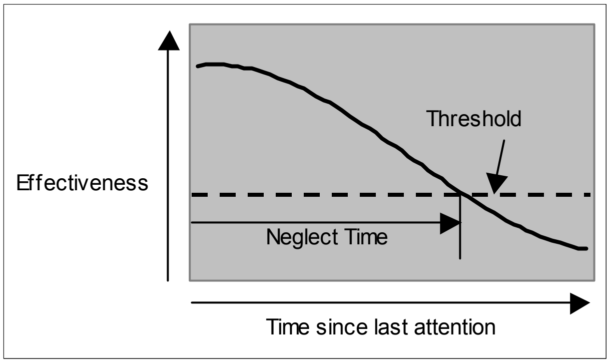

Figure 2 – Neglect Time

An Artificial Robot World

A simple robot world is helpful in illustrating the nature of neglect. Consider the world shown in Figure 3. There is a robot (upper left), a target (lower right) and trees and rocks that form obstacles to movement.

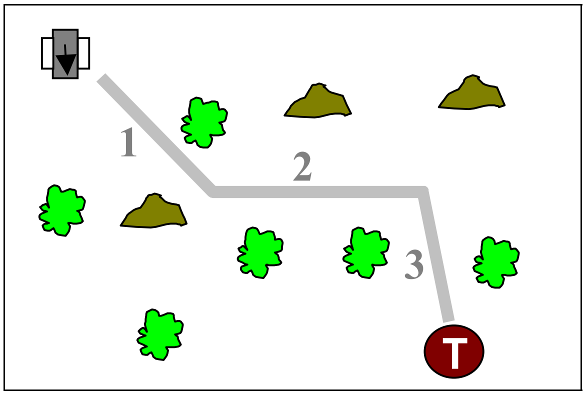

Figure 3 – Simple Robot World

A very simple robot accepts a compass direction from the user and will travel in that direction until it reaches an obstacle in which case it stops and waits. We can also equip our robot with an odometer so that the user can set a desired distance to travel before pausing for further instruction. Our current task effectiveness measure is the speed of travel from start to target. This robot has some degree of autonomy in that it can move without instruction and can sense when it can go no farther and has a limited sense of distance. Designing robots with the ability to diagnose when they need assistance is an area of current and ongoing research.

The gray line shows one possible path to the target with three numbered segments. For each segment the user can set a direction and then neglect the robot and go do other things. The neglect time depends upon the speed of the robot and the distance to the turn to segment 2. On path segment 1, however, it is possible that the robot might drift to its left and encounter the tree in which case it will stop much earlier. It is also likely that the robot odometer is inaccurate and the segment will end early or late.

In the case of this simple robot the ideal neglect time is the time to reach the next turning point. The actual neglect time may vary depending upon encountered obstacles. For any given segment the neglect curve is as shown in Figure 4.

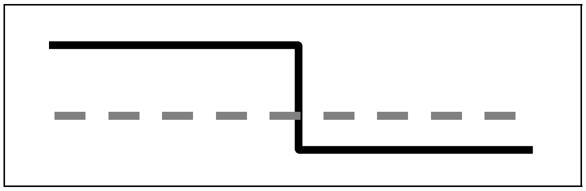

Figure 4 – Single Segment Neglect Curve

If we assume that the distance to an obstacle has a Gaussian distribution about some mean distance then the average neglect curve over a number of segments will be similar to that shown in figure 1.

Task Complexity

This simple robot world also illustrates the role of task complexity. If we take our simple world and scale it up to thousands of rocks and trees spread over a larger area, the neglect curves would remain the same as long as the density of obstacles (obstacles per unit area) remains the same. If, however, we increased or decreased the density of obstacles, then the neglect curves will change as the distribution of time to stopping changes. Neglect curves are also a function of task complexity, as shown in figure 5. That neglect curves follow this hypothesized shape has been validated in [1] and used by [2].

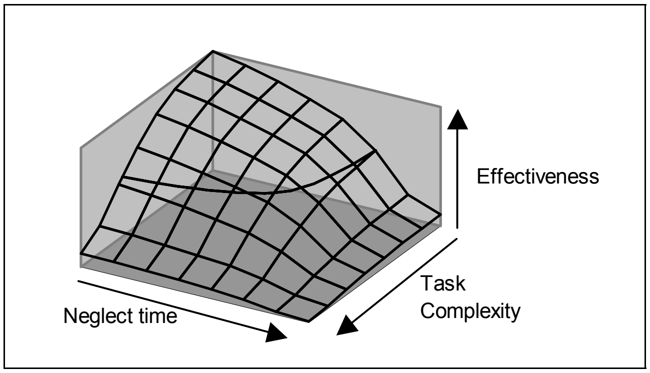

Figure 5 – Neglect and Task Complexity

In our simple robot world complexity is a function of obstacle density. In other worlds this may be much more complicated. Sensor error, active obstacles such as other vehicles, and uneven terrain that modifies vehicle speed can all contribute to the complexity of the task.

Measuring neglect tolerance

Neglect tolerance is our basic mechanism for measuring the autonomy of a robot. The amount of time that a human can ignore a robot has a lot to do with the attention leverage that the robot can provide. This attention leverage is important for two reasons. First, attention leverage allows an operator to manage multiple tasks; this is important for such typical tasks as simultaneously guiding the robot through a world and looking for some target (such as a victim in a search-and-rescue task [3]. Second, attention leverage allows an operator to manage multiple robots which is an important special case of managing multiple tasks.

We have identified two ways to measure neglect tolerance. The first is a premeasured average neglect time. We can measure this by placing a robot at some random location in a problem world and giving it a random goal to achieve. We then can measure the amount time that the robot is effective, that is, the elapsed time during which the robot makes progress towards that goal before dropping below the effectiveness threshold. In our simple robot world this is equivalent to placing the robot and target at random locations and measuring the time before the robot stops. The nice thing about this approach is the neglect tolerance is a simple measure of the robot capability and the task complexity.

Our experiments, however, have shown that neglect tolerance is not quite so simple. There is an interaction between neglect tolerance, the user interface and the global problem space. Frequently the users will detect global problems, such as the robot wandering into a cul-de-sac, and will intervene before the robot itself detects a problem. This problem is partly caused by the use of estimates of current task effectiveness that differ from the human's perception of task progress. An alternative neglect tolerance measure that relies on the human's estimate of task progress is to measure actual active usage of the robot by a user. In this case neglect tolerance is measured as the time between some user instruction and either dropping below effectiveness threshold or some new user instruction. This leads to more accurate neglect tolerance, but now is no longer independent of the user. For example the user's trust in the robot's autonomous abilities has a lot to do with such neglect measures. If the user does not trust the robot they will intervene much sooner. The impact of trust on neglect tolerance needs further study.

Increasing NT

An obvious goal is to increase the neglect tolerance of a robot. One way to do this is to increase its intelligence and autonomy. If our simple robot had some rudimentary vision capability, it might easily see its way around a tree and thus keep making progress without human intervention. Thus neglect tolerance is increased. As we will show later, increasing neglect tolerance can increase the leverage of human attention, but not necessarily so.

Fortunately, much work has been done, albeit indirectly, in the robotics community on designing neglect tolerant robots. This work has been necessary for designing robots that work under conditions of high communications latency. Since communication latency is analogous to attentional neglect, techniques such as safe-guarding [4,5], waypoint-navigation, and shared control [6,7] are important.

Solely focusing on NT has other problems. In our simple robot world we can increase NT just by slowing down the robot. If it goes slower, it will take more time to reach a stopping point and thus can be neglected longer. However, in our task effectiveness measure of speed to target, this approach is very poor. If, however, TE was measured as number of rocks and trees studied along the way, slowing down the robot might be a very effective solution. Although measuring neglect tolerance is an important step to improving a human-robot team, other metrics are also necessary for creating successful designs.

ROBOT ATTENTION DEMAND

Since we are trying to increase the leverage that a robot offers to a human-robot team, we should measure how much attention a robot is demanding. We call this robot attention demand or RAD. This is a measure of the fraction of total task time that a user must attend to a given robot.

We define RAD as a relationship between NT and something we call interaction effort (IE). Interaction effort is a key component in our attempts to improve the human-robot interaction. A simplistic view of IE is the amount of time required to interact with the robot. We will discuss the nature of IE in more detail later. The relationship between these three measures is defined as follows:

$$RAD = \frac{IE}{IE + NT}.$$

RAD is a unitless quantity that represents the fraction of a human's time that is consumed by interacting with a robot. The numerator is the amount of effort that the user must expend interacting with the robot and the denominator is the total amount of effective time of the robot. If IE is small relative to NT then the RAD will be quite small. In the case of teleoperated robots or simple driving a car, NT is very small and thus RAD approaches 1. The goal of a good human-robot interface is to reduce RAD so that the user can focus on other things besides interacting with the robot. Reducing RAD can be done by increasing NT or decreasing IE.

Increasing NT will not always decrease RAD because NT and IE are not independent. For example, we could create a robot that can accept predicate logic descriptions of a world and similar predicate logic statements of a desired behavior. Such a robot might reason independently and function quite well for an extended period of time (higher NT). However, in many scenarios the effort required to formulate robot instructions as predicate logic would increase IE to the point where the NT gains are irrelevant and RAD is actually much worse.

Another example of how increasing NT does not always decrease RAD is one that is experienced by many roboticists. Creating an autonomous robots requires extensive engineering, programming, re-engineering, and reprogramming. The result is that the robot may be fairly autonomous --- it may have a high NT --- but to improve the robot's performance, the designer must re-engineer and reprogram the robot. Such re-engineering is a form of interaction that takes a tremendous amount of effort. As a result, the "up time" where the robots operate autonomously and can be neglected is a small fraction of the time spent by the operator on the robot.

Free time

A metric related to RAD is the user's free time (FT). This is the fraction of the task time that the user does not need to pay attention to the robot. We define free time as:

$$FT = 1.0 - RAD.$$

Free time is interesting not only because it is a measure of the attention leverage that a robot provides, but it also gives us a mechanism to measure RAD. If the user has free time, then that free time can be used on some alternate task. One way to measure free time is to give the user a robotic task and some other secondary task. In our simple robot world we can give the user the task of guiding the robot from start to target. However, because the robot wheelbase can only travel so fast, the time to target will not change with most improvements of the human-robot interface. If, however, we asked the user to count the number of purple-tailed, bullfinches nesting in the trees along the way we could measure how much of the user's attention was demanded by the robot. Finding more bullfinches without increasing the time to target would mean that RAD had been reduced.

For the kind of environments addressed in this paper, there is usually another task that is of importance upon which the user should spend their time. This might include surveillance, finding victims of a disaster [3], threat detection, or surveying the terrain. What we would like, however, is a means for understanding the RAD of our human-robot team in a task independent way. A human-robot solution with a low RAD can perform many secondary tasks. This assertion has been validated in work presented in [6].

Actually measuring free time can be hard because we don't actually know when the user is doing nothing. However, we can produce surrogate measures for FT that will allow us to detect when RAD has been reduced. In many cases we do not actually care what the free time measure is. We only care that some change in our human-robot interface has increased FT and reduced RAD. From the psychometric world we can import a number of attention consuming tasks that we can measure as a surrogate for FT such as performing mental arithmetic [8], carrying on a fabricated cell-phone conversation [9], classifying objects [10], and reading email [11]. We can use any of these as a secondary task and measure increases in their performance or frequency as an indicator of reduction of RAD. For example, Crandall had subjects perform mental arithmetic while driving a robot under two teleopeartion schemes [6]. In experiments, the more autonomous teleoperation scheme allowed users to perform many more secondary tasks (a statistically significant difference with very few subjects), and with marginally higher performance.

FAN-OUT

One way to leverage human attention is to allow a user to operate multiple robots simultaneously. This generally should allow the human to accomplish some tasks more quickly and effectively. For example, on tasks such as surveillance or exploration, multiple robots can cover a space more effectively than a single robot.

We propose to measure the effectiveness of a human-robots team using what we call fan-out. Fan-out is an estimate of the number of robots that a user and effectively operate at once. The fan-out metric is defined in terms of RAD as

$$FO = \frac{1.0}{RAD} = \frac{IE + NT}{IE}.$$

From the FO equation we see that FO increases as neglect tolerance becomes large relative to interaction effort. The more neglect tolerant a robot becomes, the more robots a single user can operate. This equation, however, does not tell the whole story. As fan-out increases, interaction effort also increases, as will be discussed in the next section.

This means that if a person wants to be able to control multiple robots with given capabilities, they should spend their design effort in making IE low One of the attractive things about fan-out is that it can be measured. There are two ways of measuring fan-out that yield similar results, but on different scales. The first approach measures the performance plateau. If we consider the graph in figure 6 we see that task effectiveness should increase as more robots are added to the task. However, at some point the user becomes overloaded and adding another robot does not improve the performance.

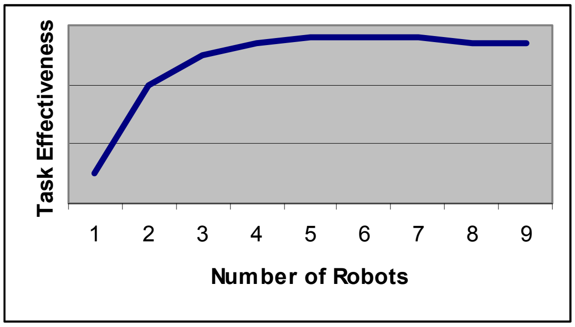

Figure 6 – Fan-out performance plateau

One of the problems with the performance-plateau method of measuring fan-out is that it requires a large number of trials. To get a good fan-out estimate, it is necessary to run multiple task trials for each potential number of robots. However, it does give a very realistic estimate of fan-out.

A second approach to measuring fan-out is the average robot activity. In this approach, the user is given more robots than they can realistically use. While the task is progressing we periodically count the number of robots operating above the effectiveness threshold. We take the average of these counts as a measure of fan-out.

There are several physical and cognitive constraints that limit how well a system can achieve the theoretical fan-out limit.   The first constraint we call *task saturation*. This is when the task, not the user becomes saturated. The measured fan-out is lower than the actual RAD would indicate because it is not possible to bring more robots to bear on the current task.

Task saturation can occur for two reasons. First, it can occur when the task is so simple that dedicating a lot of robots to it will not improve performance. Consider for example our sample robot world where there are only 2 targets. No matter how effective our interface or high our neglect tolerance, no more that 2 robots are required to get the job done. Sending multiple robots after the same target is pointless in this case.   The second cause of task saturation occurs when the task space is too crowded. If all of the robots start in the upper left corner of our world, it is hard to get many of them moving because they run into each other. In search tasks, the search perimeter imposes a limit on the number of robots that can be applied to the task. The task saturation limits are important in understanding how to apply a human-robots team to a task, but they get in the way of understanding the human-robots interface.

The second constraint that limits fan-out is caused by limitations of human cognition, primarily memory. In controlling multiple robots, the human must remember robot state information, interface modes, robot abilities, etc. This places demands on working memory since only a limited number of pieces of information can be stored in short-term memory and since only a limited number of mental models can be active in long-term memory at a time. We will discuss how these limitations affect FO via interaction effort in subsequent sections.

INTERACTION EFFORT

As can be seen from the free-time and fan-out equations the human-robot team can be improved by either increasing the robot's neglect tolerance or by reducing the interaction effort (IE). Neglect tolerance is primarily a function of robot ability. Therefore, reducing interaction effort (IE) is the key problem in improving the human-robot interface. Being able to measure interaction effort and particularly to determine when that effort has been reduced by a new interface design is critical to the development of the types of human-robot systems that serve our needs.

In most cases interaction effort is directly related to the time necessary to interact with a given robot. However, the difficulty lies in identifying exactly when a user is interacting. Interaction effort is more than just the time required to manipulate input devices. In most scenarios, interaction effort is dominated by cognitive rather than physical effort. Without "mind probe" technology we cannot tell if the user is day-dreaming or focused on robot control. Eye-tracker experiments have demonstrated significant differences in gaze patterns between various behavioral states [12]. However, eye-tracking is hard to deploy in many situations where robots are useful.

We resolve the problem of measuring interaction effort in two ways The first is to focus not on interaction time, but on interaction effort. This is a unitless measure of how much effort a user must put into interacting with their robots. What we are interested in is relative values of the interaction effort. How much less effort is required using interface B instead of interface A. Though we cannot pin down the units, we do have a comparative tool for measuring progress.

The second component of our approach is to measure IE indirectly using the free-time and fan-out measures along with their corresponding equations. As we have shown we can measure neglect tolerance and, using secondary task performance, we can get a measure that is related to free-time. Using NT and FT, and the free-time equation we can compute an estimate of interaction time:

$$IT = \frac{NT(1-FT)}{FT}.$$

Note that we do not actually have a measure of free-time, we only have a measure of secondary task performance (STP). What we really have then is an estimate of interaction effort using a similar equation.

$$IE = \frac{NT(1-STP)}{STP}.$$

This estimate of IE can now be used to compare various interfaces.

We can also use our fan-out measures, neglect tolerance and the fan-out equation to produce an estimate of interaction effort by solving for IE

$$IE = \frac{NT}{FO-1}.$$

These now give us two indirect means for measuring interaction effort that we can use in evaluating human-robots interfaces. Note that the various measures of IE are not directly comparable because they depend on other measures that have differing characteristics.

Components of Interaction Effort

When designing human-robots interactions our key problems are to increase neglect tolerance and reduce interaction effort. Interaction effort is not monolithic. We have identified at least four components to the interaction effort. They are subtask selection, context acquisition, solution planning and expression of robot directives. We will discuss each of these components in turn. These components exist for the general case of an arbitrary secondary task as well as for the special case of managing multiple robots.

Subtask Selection

Task selection is most important when working with multiple robots. Having completed an interaction with a robot the user must next decide which robot will receive assistance. There are several approaches to the problem of subtask selection that can reduce this effort. One simple approach is to have an automatic round-robin selection mechanism. This is where the system automatically chooses each robot in turn and presents that robot to the user. The interactive effort from subtask selection goes to zero, but the task effectiveness and fan-out may suffer because the robots in most need of human attention may not get that attention when they need it. This is like the building security system that sequentially presents security camera images to the guard. The cameras all get equal time but there is a strong likelihood that a fast intrusion will escape the guard's attention.

A second approach is to show the data (or a summary) on all of the robots to the user and let the user select. An interface that supports such interaction has been developed by Scholtz [13]. User selection of the next robot can produce better selections, but will increase interaction effort. Preliminary experiments strongly suggest that interaction effort increases with fan-out. Obviously searching for the right robot to service will be on the order of log(FO) or FO. Getting the best robot to service could be FOlog(FO).

A third approach is to provide an automatically computed measure of attention need. The user is then directed to the robot with the most perceived need. This would be like showing the security guard the images that have detected the most movement in the recent past. This can bring selection effort back to near zero, but can also have problems. If the attention-need metric is not a good one, then it may actually be worse than round-robin. If for example the attention metric is lack of progress and one robot has a bad wheel, the best approach may be to abandon the robot, but the attention metric will constantly show that poor robot to the user. Similarly if there is a wind storm, the motion-based camera attention algorithm will constantly images of waving trees in the parking lot to the security guard.

Techniques for assisting the user in making the subtask selection will be important to reducing the interaction effort. Most approaches to this problem will involve increasing the salience of robots that most need attention. It is interesting to note that the techniques for automating subtask selection are analogous to Sheridan's 10 levels of sharing responsibility between a human and an automated system in supervisory control [14]. Additionally, work on management policies has direct bearing on this problem [15].

Context acquisition

Context acquisition comes when the user must switch from one subtask to another. This arises both when operating many robots as well as when operating a single robot while performing other tasks. When the user's attention is switched, the user must take a moment to understand the situation of the new robot that has received attention. Although part of this understanding is required for proper subtask selection, many aspects of context acquisition must be obtained after selection occurs. For example, when the interface draws attention to a particular robot, the human must still acquire context (e.g., diagnose the problem) before controlling the robot.

There are multiple issues in context acquisition. A key approach is the externalization of memory. For example, when driving robots through their front-mounted camera, switching to a new robot will cause memory problems for the user. The user sees what the camera sees, but they must remember, or search again (via, for example, range sensors), for what is left, right or behind the robot. Making such information visible in the interface should reduce context acquisition time.

There is a serious problem with heterogeneous robots because not only must the user reaquire knowledge of a robot's situation, but must also mentally adjust to the different abilities of the current robot. This mental adjustment includes loading relevant state information into short-term memory and activating relevant mental models from long-term memory.

We have only scratched the surface of the context acquisition issues and how they affect interaction effort. It is clear the context acquisition effort will go up as fan-out goes up. Automatic selection will not make the context switch go away. There is a possibility that automatic subtask selection may actually increase context acquisition time because the user has no understanding of why the task was selected. This problem has been identified as a key factor in the failure of some automation systems [16]. The context acquisition problem is probably the largest contributor to an upper bound on fan-out regardless of robot capability.

Planning

Once a user has selected a robot, understood the robot's situation, the user must plan what instructions the robot must be given. This depends very little on the number of robots but rather on the complexity of the task, the intelligence of the robot and the user's understanding and trust in that intelligence.

As the complexity of a task increases, the amount of effort required for the user to come up with a robot's next direction is increased. This interaction effort can be decreased, however, if the robot or the interface can supply some of the planning information. In our simple robot world the user interface could show a possible path to the target that was automatically calculated. The user's planning problem is now greatly simplified. However, if the interface is capable of completely solving the problem, then the human is not required at all. Usually there are issues that the software or the robot's sensor processing cannot resolve. If the software-supplied solution is not appropriate then the user must plan on their own or may even be distracted by the erroneous plan presented. Such issues have been identified in other assisted planning domains [17].

In addition to the increase in planning difficulty caused by increasing task complexity, planning can also be made more difficult when robots become more sophisticated. This occurs because communication may be more involved, trust and expectations may be misplaced, and developing a correct mental model of possible robot behaviors may become prohibitive.

Not only must the problem be solved (find a path to the target) but the user must also understand the robot's capability. We have found that neglect tolerance goes down when the users have less trust in the intelligence of the robot. It also makes a difference if the user clearly understands the nature of the robot's abilities. The robot may be powerful but if the capabilities are obscure then the users will ignore them and planning will still be done by the user. The user will also set more conservative goals for the robot and the neglect tolerance will be reduced.

Expression

Having selected a robot, acquired the context, and planned a solution, the user must express intent to the robot. Even at this phase, the physical effort is rarely the dominant factor. As a result of planning the user has conceived of some action that the robot should take in the physical world. The user must now translate that desired physical behavior into inputs to the human-robots interface software. Don Norman has characterized this translation from a planned solution to actual control inputs by the user as the "Gulf of Execution"[18]. This translation is what requires most of the effort.

We believe that a basic problem in many human-robot interfaces is that user intent must be expressed in terms of robot control values rather than in terms of intended action in the physical world. This requires the user to map physical world intent backwards to the control values that will produce the desired result. This mapping from problem space to control space is a key source of interaction effort. We believe that the human-robot interface should mitigate or automatically perform such mappings and thus reduce the effort required.

SUMMARY

The obvious goal of any human-robots interface is to increase the effectiveness of the team in accomplishing some task. We believe that the keys to this effectiveness are increasing the neglect tolerance of the robots and reducing the interaction effort of the interface. We have captured this in the free-time and fan-out metrics. We have shown how these two metrics along with neglect tolerance can be measured and then used to produce estimates of interaction effort that can be used to chart the progress of improvement in human-robots interface design. Lastly we have broken down interaction effort to identify where and how it can be reduced.

REFERENCES

1. J. W. Crandall and M. A. Goodrich, "Principles of Adjustable Interactions." In proceedings of the *2002 AAAI Fall Symposium Human-Robot Interaction Workshop*, North Falmouth, MA, USA, November 15-17, 2002.

2. P. Scerri,, D. Pynadath and M. Tambe, "Towards adjustable autonomy for the real world." *Journal of AI Research*, 2002, v17, pp. 171-228.

3. J. Casper and R. Murphy "Workflow study on human-robot interaction in USA." *ICRA 2002.* pp 1997-2003.

4. T. Fong, C. Thorpe and C. Baur, "Safeguarded teleoperation controller." *IEE International Conference on Advanced Robotics*, August 2001, Budapest, Hungary.

5. E. Krotkov, R. Simmons, F. Cozman and S. Koenig, "Safeguarded teleoperation for lunar rovers: from human factors to field trials." *IEEE Planetary Rover Technology and Systems Workshop*, 1996, Minneapolis, MN.

6. J. W. Crandall and M. A. Goodrich, "Characterizing efficiency of human robot Interaction: A case study of shared-control teleoperation." In proceedings of the *2002 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2002, Luasanne, Switzerland.

7. T. Röfer and A. Lankenau, "Ensuring Safe Obstacle Avoidance in a Shared-Control System." *Proceedings of the 7th International Conference on Emergent Technologies and Factory Automation*, pp, 1405-1414, 1999, ed. J. M. Fuertes.

8. C. D. Wickens and J. G. Hollands, *Engineering Psychology and Human Performance*, Prentice Hall, 2000, 3rd edition.

9. N. Ward. "Driving Performance During Cell Phone Use and Common Secondary Tasks Combined Under Conditions of Driver Impairment." This is an active research project. See the project website at http://www.its.umn.edu/research/projects/2003014.html

10. H. E. Pashler, *The Psychology of Attention*, The MIT Press, 1997

11. T. Brown, B. Caven, S. Haake and J. Lee, "Speech-based interaction with in-vehicle computers: The effect of speech-based E-mail on drivers' attention to the roadway." *Human Factors*, 2001, 43, pp. 631-640.

12. D. Salvucci and A. Liu, "The Time Course of a Lane Change: Driver Control and Eye Movement Behavior." *Transportation Research, Part F*, 2002, v.5, n2, pp. 123-132.

13. [Scholtz03]. Personal communication and MARS 2020 presentation. This is an ongoing project; see http://www.itl.nist.gov/iad/vvrg/staff/jean.html

14. [14] T. B. Sheridan (1992*). Telerobotics, Automation, and Human Supervisory Control*. MIT Press, 1992, Cambridge, MA.

15. C. E. Billings. *Aviation Automation: The Search for a Human-Centered Approach.* Lawrence Erlbaum Associates, 1997, Mahwah, NJ.

16. L. Bainbridge, "Ironies of Automation." *Automatica*, 1983, v19, n6, pp. 775-779.

17. A. Baba, J. Bresina, L. Charest, W. Edgington, A. Jonsson, A. Ko, B. Kanefsky, P. Maldague, P. Morris, N. Muscettola, and K. Rajan, "MAPGEN: Mixed Initiative Planning and Scheduling for the Mars03 MER Mission." In *Proceedings of the 2003 AAAI Spring Symposium on Human Interaction with Autonomous Systems in Complex Environments*. 2003, Stanford, CA.

18. D. A. Norman. *The Design of Everyday Things*. Basic Books; (September 17, 2002).