

Human- Computer Integration

 **Umer Farooq**, Microsoft Corporation
Jonathan Grudin, Microsoft Research

Insights

- Interaction can be described as stimulus-response, whereas integration implies partnership between the human and computer.
- There is a continuum from interaction to integration; i.e., integration extends but doesn't replace interaction.
- As designers, developers, researchers, product managers, entrepreneurs, and users, we can improve human-computer interaction by focusing on the larger context of integration.

The era of human-computer interaction is giving way to the era of human-computer integration—integration in the broad sense of a partnership or symbiotic relationship in which humans and software act with autonomy, giving rise to patterns of behavior that must be considered holistically. Cyborgs or brain-computer interfaces may come later, but integration is already well under way.

Even when generally aware of this profound change, we continue to observe the world through lenses acquired in the era of human-computer interaction. To most effectively design and evaluate software and hardware,

a new perspective is in order. Some engineers and fiction writers have envisioned aspects of a future that is now becoming the present. Some had utopian takes, others dystopian; both are evident in the intriguing benefits and cautionary challenges we face. Realizing the potential of pivoting to integration requires a conscious change of approach. Different research questions and design possibilities emerge when you shift from the familiar perspective of human-computer interaction to a view of human-computer integration that is still coalescing.

We will not stop interacting with computers and other digital devices.

The nature of our interaction has continuously evolved—from switches, cards, and tape to typing, mice, and styluses, adding speech and gesture. Skin sensors might someday become routine, or even brainwave interaction if hats make a fashion comeback. We can see these changes, but the most dramatic change affecting human-computer interaction was invisible: *what the computer does when we are not interacting with it.*

For decades, the relationship could be described as stimulus-response. A computer responded to our last input or command, then waited for the next. Our action could be to load in a program as a deck of cards; the computer then read them, returned a printout, and waited for the next deck of cards. We typed in a command name, the computer processed it, typed back a response, and waited. We clicked on an icon, the computer produced a menu or initiated an action, and then waited. Sometimes the control was reversed: An application issued commands and a human entered information. None of this describes a real partnership.

When the personal-computing era arrived, most computers were usually turned off or displayed a screen saver as they waited for a human to initiate an interaction. A few people installed SETI@home to devote unused cycles to exploring radio telescope data for evidence of extraterrestrial life, and some fell victim to a hacker who took over their computers to redirect large quantities of spam. But in general, little activity occurred until an owner returned.

Over time, slowly, background tasks began utilizing client or server cycles on a user's behalf. Background tasks range from programmed interactions to adaptive processing that proactively does tasks we need to perform or might overlook. Consider browser page predictions, where pages are pre-rendered in expectation

that one might navigate there next. Such unseen software activity shapes subsequent interactions.

Computers aren't like people in most respects, of course. But like our friends and colleagues, their autonomous activity can affect how we interact with them. Not only are they busy on our behalf, we don't even know what they get up to when we are asleep. Sound creepy? Not really, not anymore. Our timelines are independent, although they frequently intersect. As designers, developers, researchers, product managers, entrepreneurs, and users, we can improve human-computer interaction by focusing on the larger context of integration.

Integration implies partnership. Consider the following analogy. A professional working couple hires a home cook to prepare meals, specifying the menu for the week. The cook prepares the specified meals, leaves, and the couple later eats the meals. This is interaction. Now imagine that your partner, say, your husband, is cooking. You may crave burgers but he observes that you have ingredients for an Asian stir-fry that are approaching their use-by dates. He recruits you to chop vegetables while he makes sauces, and invites you to weigh in on the spice level. When the dinner is ready, you enjoy the meal together. This is integration.

The home cook functions more as an assistant than as a partner. In the other example, the couple negotiates over what to cook, which provides opportunities for compromise or information to emerge of which only one was cognizant. Partners construct meaning around each other's activities, in contrast to simply taking orders. They are codependent, drawing meaning from each other's presence.

A DAY IN THE LIFE OF ARYA

Humans are now both interacting and integrating with computers. A quarter century ago, Mark Weiser [1] envisioned

a future of ubiquitous digital devices. He imagined a day in the life of a designer named Sal, warning that “extrapolating from today's rudimentary fragments of embodied virtuality is like trying to predict the publication of *Finnegan's Wake* shortly after having inscribed the first clay tablets.” Weiser didn't predict the Web or wireless projectors, but he got a lot right. An optimist, he overlooked the privacy and security implications of maintaining visible digital trails of where neighbors had walked. Sal relied on paper instruction manuals and produced long documents instead of slide decks, but he saw weather forecasts and news delivered electronically, and a parking garage directed him to a specific empty space (we are getting there). Weiser's vision inspired many researchers, designers, and developers.

The scenario below depicts behaviors and capabilities that exist today, some of them envisioned by Weiser and others, that illustrate aspects of human-computer interaction and integration.

Half asleep, Arya rolls over to turn off her alarm. It's still dark. 6:44 a.m., 16 minutes before the time she had set. The alarm weather app indicates that inclement weather requires her to get moving earlier to be sure of making her 8:30 a.m. meeting.

Once on the freeway, Arya puts the car on autodrive and skims a newspaper. When sunlight breaks through the clouds, line markers on the wet road are obscured and the dashboard pings her to take control.

In her office, Arya turns to email. It is prioritized based on her relationship and past email interaction with the sender and whether it was sent to her or to a list. When a calendar alert signals that the 8:30 meeting is five minutes off, she clicks on it to bring up the agenda.

As meeting attendees walk in, the room connects their devices. Four attendees are in the room and four are teleconferencing. Three of the latter appear as still images; one, near the end of his workday in Paris, has turned on his camera. A room camera enables remote viewers to see those in the room, with the current speaker's image larger than the others'.

While eating lunch, information junkie Arya skims her social feeds. “Why have I started getting ads in French?” she mutters. She doesn't understand French. Then she realizes this began when she

Integration implies partnership...Partners construct meaning around each other's activities, in contrast to simply taking orders. They are codependent, drawing meaning from each other's presence.

Scenario	Example of interaction	Example of integration
Waking up	Turning the alarm off	Alarm poses a consequential choice
Driving to work	Car signals return of control	Car is semi-autonomous
Starting the workday	Reading email	Email app prioritizes messages
Prior to meeting	Reading notification	App tracks meetings, sends alert
Meeting start	Arriving connected, agenda up	Room software tracks calendar
In meeting	Watching speaker	Teleconference app magnifies speaker
Reading news	Noticing advertisements	Service selects ads based on history
Watching cartoons	Setting educational goal	Tablet negotiates with the child
Surfing social media	Scrolling news feed, seeing ads	App suggests video, selects ads

→ Table 1. Scenario activities revealing the computer as a partner to the human.

started working—in English—with a French team, several of whose members she connected with on social media. “Who rattled me out to advertisers?” she wonders crossly. “And I’ll bet the uptick in ads for weight loss and knee surgery is related to my recent 40th birthday!”

In the evening, Arya’s three-year-old, Zain, cries and refuses to eat dinner, demanding his tablet. Too tired to argue, Arya hands it over. The toddler tries to view Daniel Tiger, but the UI won’t let him because he is two minutes short of finishing an interactive educational exercise. After thinking about the dilemma for a moment, Zain finishes the exercise.

With Zain busy with Daniel Tiger, Arya scans social media. Scrolling news feeds, Arya “likes” a short video of making a simple yet luscious-looking molten chocolate cake from scratch. Arya tries one or two recipes each weekend. As she stares at the chocolate-cake image, a cloud crosses her face. Those weight-loss ads! “Could my fitness watch be communicating with advertisers?” she muses aloud.

In this scenario, computers are continually working on Arya’s behalf, even when she is asleep. Table 1 lists examples of interaction and integration between the human and computer that are possible today.

This scenario also illustrates two key points that we will discuss later. First, integration focuses us on new possibilities and opportunities for design and evaluation. For example, the autodrive system responds to other vehicles with deceleration, acceleration, and lane changes. An integration approach brings Arya into the picture. Detection of droopy eyelid movements that signal fatigue could trigger an alert and notify her of the

distance to a rest area or a coffee shop. A designer might consider what human passengers do. With Arya commuting south on I-405 on a sunny day, the car might ask, “Isn’t that a nice view of Mt. Rainier?” based on weather reports, sunlight, GPS, visual recognition, and the knowledge that being able to see the snow-capped peak is a rarity in typically overcast Seattle.

Second, there is a continuum from interaction to integration. For instance, ordering from a menu in a restaurant is a command-response interaction. When a waiter says, “The trout is especially good today,” there is a bit of mixed initiative, as when Arya’s alarm clock acted on the implications of bad weather. In both cases, the information recipient decides. It is a partnership when you say, in a favorite restaurant, “Ask [owner-chef] Edward to choose something good for us today.” *Software has traveled along this continuum and now takes the initiative often enough to think of it as a partnership.* Fully embracing the shift is more than a cosmetic change in labels. Integration inspires a significantly different mindset and approach to design and use, potentially extending the practical and intellectual aspects of HCI.

Before exploring this new approach, we note that we have reached a point envisioned long ago by HCI pioneers.

EARLY VISIONS OF SYMBIOSIS

Until transistors supplanted vacuum tubes, a weak computer cost millions of dollars and filled an entire room. Only in science fiction did technology have equal footing with people, as in E.M. Forster’s eerily prescient *The Machine Stops* (1909) and Charlie Chaplin’s comical assembly-line feeding machine

running amok in *Modern Times* (1936). Transistor-based commercial computers arrived at the end of the 1950s and freed the imaginations of scientists and engineers. Some wrote about aspects of a future that are largely realized now.

Marvin Minsky envisioned the functions of the human brain being replicated by a computer. He and others went even further to suggest that ultra-intelligent machines would surpass human intelligence and capability, predicting this would happen by 1980. Knowing this clarifies the analysis of an influential and prescient psychologist and engineer, J.C.R. Licklider, who managed research funding at ARPA. He wrote in 1960 of a coming “symbiosis” of people and machines that “will involve very close coupling between the human and the electronic members of the partnership.” He continued:

It seems worthwhile to avoid argument with (other) enthusiasts for artificial intelligence by conceding dominance in the distant future of cerebration to machines alone. There will nevertheless be a fairly long interim during which the main intellectual advances will be made by men and computers working together in intimate association . . . say, five years to develop man-computer symbiosis and 15 years to use it. The 15 may be 10 or 500, but those years should be intellectually the most creative and exciting in the history of mankind [2]. (emphasis added)

Licklider went on to specify what needed to be done in the transitional “five years,” laying out a framework for the field of human-computer interaction that included “desk-surface display and control,” “computer-posted wall display,” and “automatic speech production and recognition.” Other 1960s visions of a symbiotic future

COMMUNICATIONS APPS

Access the latest issue, past issues, **BLOG@CACM**, News, and more.



Available for iPad, iPhone, and Android



Available for iOS, Android, and Windows

<http://cacm.acm.org/about-communications/mobile-apps>



Association for Computing Machinery

included Engelbart's "augmented human intellect," Ted Nelson's dynamic hypermedia, and Alan Kay's Dynabook.

To satisfy his colleagues who predicted super intelligence in 20 years, Licklider gave the era of human-computer interaction only five years. The effort to develop the necessary interface features and interaction began with an artifact-centered human factors approach. Licklider's paper was published in a human factors journal long before CHI existed.

It took us 50 years to reach symbiosis or integration—we are now there. We developed the input and display technologies Licklider called for. Speech production is not unaccented but it is comprehensible, and speech and image recognition has made great strides due to advances in machine learning and the availability of big data. Engelbart's prototypes have developed into office applications, the dynamic Web supports the heart of Ted Nelson's hypermedia vision, and capable tablets and e-readers realize the promise of Dynabook.

HCI ALONG THE INTERACTION-INTEGRATION CONTINUUM WITH TURBOTAX

Let's move from Arya to a real product that exemplifies how the era of human-computer interaction is giving way to the era of human-computer integration.

TurboTax, built by Intuit for online tax preparation in the U.S., nicely illustrates possibilities and complexities that arise in a partnership in which control is shared. TurboTax can be seen as an assistant that helps with a specific annual task, yet it is designed to allow the locus of control to vary. A tax filer can shift between decision maker and an assistant tracking down information for the software.

Upon being launched, TurboTax for the 2015 tax year asked, "How are you feeling about doing your taxes?" (Figure 1). Unexpected, but not annoying. Was it designed to reduce angst as we confront a complicated and consequential activity? Does the software use the response to adjust its behavior, or does it aggregate responses to measure the overall outlook of the customer base this year? Or is it gauging from the answer our openness to paying for additional services? We don't know, but such possibilities make sense for a

designer who steps back from the basic interaction to consider the engagement as an ongoing partnership.

TurboTax then asked for prior tax records and accounting data that could be imported, and a few other background questions, while always attempting to engage users (Figure 2). It took the initiative, sounding much like a human tax consultant. It then posed an option that the human consultant would not—we could let it continue to interrogate or jump to filling in forms directly.

With a traditional HCI lens of interaction, a human tax filer is "the user," who is learning what to do. Using an integration lens, the software can assume the position of learner. In fact, the value proposition of TurboTax is explicitly that the software is learning about the human: "You answer simple questions about your life. We do all the math."

Software once came with instruction manuals. Many products still have online manuals or FAQs. Potential human partners—a tax expert or contributors to social media—don't come with instruction manuals, and neither does TurboTax. The software is open to questions, requests for definitions of terms, and explanations for processes. At times, it offers to turn over control: "Would you like to enter another 1099 or move on and come back later?" Maybe when I hesitated about clicking on the button to start a new one, it surmised from population averages or my past returns that I probably have more.

TurboTax developers have honed the software for years. A tedious series of questions about rarely occurring conditions are now grouped for quick resolution. TurboTax designs for error recognition and recovery, relying on both human intelligence and its own computational logic. For instance, a dependent's age must be entered; the user is asked to manually enter a birthday, but the software can validate it based on the social security number.

From an interaction perspective, TurboTax became flat and modern to be accessible to a broad audience. It largely meets that goal. From an integration perspective, TurboTax can now look beyond functional capability and aesthetic design to improve the conversational language—for example, minimizing obscure tax terminology

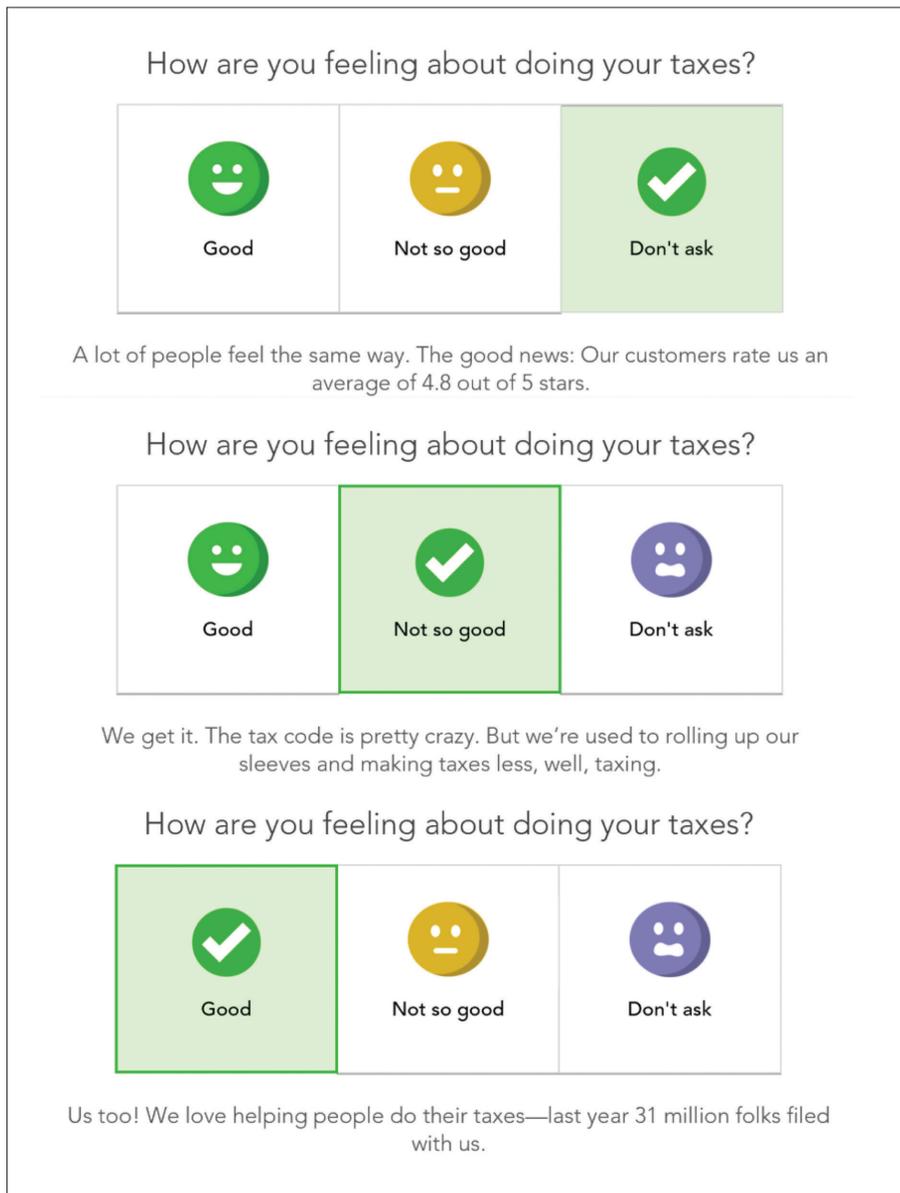


Figure 1. TurboTax's conversational user interface adapts to a user's response, much like a partner.

that, however accurate, risks confusing users in a context of possible financial risk. The software benefits from establishing common ground with tax filers before getting to the core activity of working on taxes.

We have no direct insight into Intuit's internal engineering processes and culture, but as long-time users we have seen the software evolve from an often awkward product to a more graceful tool that exhibits activities along the interaction-integration continuum.

IMPLICATIONS FOR DESIGN, EVALUATION, AND THEORY

Changing the focus from interaction to integration may seem subtle, but it can have profound implications for

practice. Interaction remains part of the picture, but there is more. We benefit by consciously considering: Does our research question or design decision focus on integration or solely on interaction? Consider the familiar example of error messages or software updates that are signaled as technical alerts. These once seemed useful computer-to-human alerts, but the abrupt, opaque, and often annoying messages are inadequate as friendly assists to a colleague. No partner wants to be spoken to that way.

For a particular application or scenario, consider whether integration represents a sharp pivot or a minor extension. As practitioners, how do we design, build, and evaluate products around integration? As

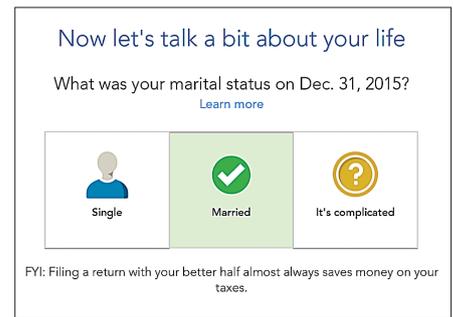


Figure 2. TurboTax engages the user continuously, even during mundane background questions.

designers, how do we mesh integration with the growing emphasis on visual design and branding? Which theories are strengthened, which require epistemological revision, which have we moved past?

Designing for integration is designing for at least two entities. Good basic interaction design is a precursor to integration design, and designing for dyads or groups is different from and generally more difficult than designing for individual users. All design aimed at supporting integration takes on this additional complexity.

The simplicity of Google's interaction design was a factor in its rise above search competitors. Today, the command-response interaction style of Internet search is largely resolved from the perspective of basic relevance and retrieval. With integration, interaction designers can shift their focus beyond command-response interaction and draw on users' daily activities. Imagine that your search engine is replaced by your colleague who has a heightened sense of social and activity awareness, for instance. In responding to a query, your colleague takes into account where you are, your work or personal interests, and your past interactions. Search engines now have access to contextual information that your colleague would, and much more. Search engines have an opportunity to evolve as partners. They may be doing so already—our point is, they *must be doing this* to remain competitive, and all designers will benefit by consciously taking this approach.

A change in the role of software along the interaction-to-integration continuum also changes our approach to evaluation. An early discovery was that evaluating group support applications is more difficult [3]; lab studies are less

useful for group dynamics that emerge over time. The groups studied then comprised people. Now, any application can face the same challenges, involving human and digital agents. An ostensibly single-user application such as TurboTax benefits from the consideration of affective and motivational effects that once were considered only for group support applications.

When recast as a team effort or partnership, new considerations and approaches arise. To assess how the software affects the taxpayer's feelings of psychological safety and comfort, naturalistic and longitudinal studies must complement task-based analyses. Once the relationship is viewed as a partnership, TurboTax might consider additional steps such as alerting a tax filer that a refund should have been received or a payment cleared. It could ask to be alerted if an audit materializes.

Evaluation metrics reveal what and when users did something, but rarely do they reveal *why* they did it. The essence of a good partnership is understanding why the other person acts as they do. Understanding the intricate dance of a person with a software agent requires longitudinal or ethnographic approaches to produce realistic scenarios and personas. We expect that much successful development in an era of human-computer integration will be more sophisticated, with agile efforts drawing on longer-term investments that proceed in parallel. Rapidly collected metric data could be supplemented with more slowly obtained qualitative research over time to get the understanding needed to design for integration.

Theory will also evolve as we focus on integration. For example, the 1980s' Actor-Network Theory [4] was an approach to modeling based on identifying networks of actors that included people and artifacts. At the time, it seemed more of a stretch than it does now. Some theories are vindicated and can retire, such as "minimalism" [5], the human-computer interaction theory that posited "less is better" in instruction manuals and stand-alone computer-based training. A good digital partner comes with no instruction manual at all. Other theoretical and methodological

approaches, such as ethnography, contextual design, and persona use, may find new uses as we support activities in ever-finer granularity.

We have emphasized the benefits of adopting the lens of integration; we should also note the potential risks. For example, software that appears thoughtful and considerate in one place could lead users to assume that it is generally thoughtful, as humans often are; but elsewhere the software might not be. Also, an integration lens that recognizes the computer as a partner to the human may have to take sides when computer and human goals are in conflict. For example, a website's goal may be to keep customers engaged, whereas site users may want to leave after a quick transaction. Investigating computer initiative and transparency with respect to human agency is a pressing area for research.

LOOKING AHEAD

Integration extends but doesn't replace interaction. Not all technologies are in the same stage of maturation. Interaction may take precedence. Robots, virtual assistants, and connected devices à la the Internet of Things, for instance, are largely focused on interaction but will eventually graduate to the other end of the interaction-integration continuum. For a radically novel or nascent technology, getting interaction right can be good enough and serve as a foundation for subsequent integration. For such a technology, a central focus on basic interaction design can be crucial in attracting customers and fending off competitors. We don't expect social graces from infants; we love them for who they are. But as they age, our expectations rise.

We are not suggesting that the magazine be renamed *ACM Integrations*—at least not yet. Integration can create and amplify the risk of software that appears thoughtful and knowledgeable but that is largely ignorant about related issues in a way a person would not be. Certainly, no one wants a partner with whom interaction is awkward. At the same time, it is incumbent upon us to identify when a focus on specific interaction scenarios yields

diminishing returns. In those cases, time could be better spent researching and designing for integration.

Humans and computers are integrating. As researchers and practitioners, we can get ahead of the curve and increase our disciplinary impact by embodying the mind shift and becoming even more valued bastions of HCI expertise.

Licklider envisioned the era of man-computer symbiosis lasting 10 to 500 years. Then the Singularity would arrive and humans would presumably become at best assistants and at worst extinct. The good news is that he forecast, with some measure of optimism, that the era of human-computer interaction would last five years, and it ended up taking 50. The Singularity has receded ever further into the future. Licklider speculated that the human-computer integration partnership could last as long as 500 years, by which time the resistance may unfreeze us from our cryogenic state and bring back the good old days of interaction. In any case, we can all agree with Licklider: An intellectual adventure lies ahead.

ACKNOWLEDGMENTS

We both are grateful to Erik Stolterman, Jack Carroll, and Mike Bortnick for providing insights that significantly improved the article.

ENDNOTES

1. Weiser, M. The computer for the 21st century. *Scientific American* 265, 3 (1991), 94–104.
2. Licklider, J.C.R. Man-computer symbiosis. *Transactions on Human Factors in Electronics Vol. HFE-1*, (1960), 4–11.
3. Grudin, J. Why CSCW applications fail: Problems in the design and evaluation of organizational interfaces. *Proc. CSCW '88*, 85–93.
4. Law, J. and Lodge, P. *Science for Social Scientists*. Macmillan Press, London, 1984.
5. Carroll, J.M. *The Nurnberg Funnel: Designing Minimalist Instruction for Practical Computer Skill*. The MIT Press, 1991.

📍 **Umer Farooq** is a principal research manager in the product group at Microsoft Corporation. He has shipped SQL and Windows Azure, Xbox One, and Windows 10.
→ umfarooq@microsoft.com

📍 **Jonathan Grudin** is a principal researcher in the natural interaction group at Microsoft Research. He is a pioneer of the field of CSCW and one of its most prolific contributors.
→ jgrudin@microsoft.com