Bernhard Rieder

Agent Technology and the Delegation-Paradigm in a Networked Society

Abstract

Software agents – programs that execute a defined set of tasks in an autonomous fashion – have been developed in computer science labs over the past three decades. They have been used to help administrate networks, filter newsgroups or simulate complex environments. With the success of the Web, the global Internet has become their favorite habitat and application patterns for agents technology is changing. Nowadays people use agents to haggle for them on eBay, to find and classify information using advanced data-mining techniques or to automate scheduling and communication tasks. New tasks are delegated to software agents by an increasingly disperse public – agents are leaving the inner circle of informatics, entering the domain of everyday technology.
The sheer quantity of computing power nowadays available to servers and clients and the need for assistance in the highly complex environment of the networked society pushed agent technology from university labs into the public sphere, helping users to cope with the challenges of "information at your fingertips". This process is far from over: mobile technologies (smartphones, PDAs, …) with reduced interfaces, limited stocking capacities and handicapped by relatively high communication costs will strongly rely on agent technology to filter, organize and restructure information before it is presented to the user.

This paper will introduce some of the technological possibilities and probabilities resulting from the coupling of mobile devices with agent technology before opening up into the broader field of possible social and cognitive consequences of technological progress in the area. Following Terry Winograd, I understand the design and development of software tools – and therefore agents – not as a mere challenge in engineering but a process of creating modes of perception and spaces of possible action. But agent technology goes even further by offering to lift some of the cognitive burdens off the user, e.g. performing information filtering and clustering using algorithms. The fact that agents once set loose, act in an autonomous fashion makes it desirable to theorize them not only as software programs, but also as "social actors" who exist in the vast information space that is the Internet.

Consequently, I will address three questions:

- Are agent technology and the paradigm-shift in human-computer interaction from "direct manipulation" to "delegation" a viable solution to the well documented problems of information overload and communicational stress? Will agents help making powerful mobile applications feasible and render information technology in general more user-friendly.

- What are the possible social consequences of the introduction of agent technology into ever wider parts of society? Are the new actors helping in decreasing fragmentation or are they fostering it?

- How can we conceive the act of programming as an active political act engaged in the structuring of perception? With agent technology going further into the human domain of decision making that "classical" software, is there a need to rethink the programmers position in society?

Like so many of today's buzzwords, the term "software agent"[1] refers to a multilayered patchwork concept made out of computer science research, industrial application development lore and marketing mantras. As in so many other cases, it is a new word for something not so new after all. A program constantly resident in a computers memory, autonomously making file backups, was written by Fernando J. Corbato on the IBM 7094 in 1963 – he called it a daemon.[2] From then on, it has been common for computer programmers to tackle certain kinds of problems[3] with programs that run in the background and wait for some event to occur to execute a specific set of instructions. On a first level of observation, an agent is nothing more than the application of this very basic principle to a different, or at least a wider set of jobs. Instead of backing up files, an agent would rather filter a newsgroup for articles of interest to his user or roam the web, hunting down the best price of the latest color TV. While this separation on the level of task is not a clean one, I would argue that while a daemon belongs to the depth of the machine world, an agent is a creature of the surface, its task being the negotiation between us humans and the logic of the machine. Put into current terms, an agent is a technical mediator; it negotiates and acts as intermediary.

While software agents have been developed throughout the eighties and nineties, it is the Internet revolution and the shifting embedding of digital technologies in western culture[4] that strengthen the rationale both of the term "agent" and the technical reality it refers to. It is the connected digital world where agents roam.

This article is located at the crossing point of computer sciences on the one hand side and humanities and social sciences on the other. The analysis of the technical features characterizing agent technology very quickly brings up issues that refer to questions of metaphor, production of culture and forms of social expression as much as of algorithms and code. In addition, agents evoke debates about autonomy of action and delegation of perception; debates not often found in the technical sciences. But it is particularly the logic of delegation, full of promises and hazards, which we must analyze with the critical tools of theory. But without basic knowledge of the technological context, such analysis is doomed to fail. In this paper I will identify some key issues of an analysis of agent technology in its connection to computer usage and cultural integration and production. The more technical parts of this text are a necessary prerequisite for a critical perspective.

The problem with agents

From a technical point of view, agents are the result of a fusion of different branches in computer science: they bring together research in databases, information theory, networking, interface design and – most of all – artificial intelligence (AI). In many ways different from classical software, agents embody much more than the sum of their parts. It is not what they do, but rather how they do it, that distinguishes them from "traditional" programs. Agents follow a different notion of what computing should be.

---

[1] Throughout this paper, the term "agent" always stands for "software agents". When I refer to human agents, I explicitly state so.

[2] Corbato chose the name in honor of Maxwell's daemon, which set out to defy the second law of thermodynamics only by its powers of observation. The refutation of Maxwell's thought experiment led to Shannon's theory of information.

[3] Daemons are used in situations where monitoring of environment is needed. Many system maintenance programs are daemons, but also web servers who patiently wait for clients to connect.

[4] A lot of new keywords seem to characterize this new phase in computing: Alan Kay called it the "third paradigm", Mark Weiser the "third wave"; the terms "ubiquitous computing", "pervasive computing" and "calm technology" along with "software agents" seem to be the adjunct vocabulary.

While we know very little about the consequences of the ongoing insertion of agent technology into the landscape of software and its subsequent everyday uses, we can be sure that it is happening. Today agents are developed in a large number of labs end enterprises. With very few exceptions[5], neither social sciences nor humanities have shown very much interest in software agents. I think that there are two main reasons for current reservations. The first source of trouble is the fact that the field of agent research is not only diverse, but extremely large and technically complex. There is no consensual definition, neither in computer sciences nor anywhere else, of what an agent really is; and the different designations and research directions make it hard for a non-expert audience to judge the possibilities and probabilities of actual design and use. A second reason for difficulties is the blurred line of judgment that seems to emerge when looking closely on any aspect of an agent. While technically not so new, their actual functioning is disturbing, in the sense that the common notions of what computers do and how they work seem to constantly slip. While we all have some idea of artificial intelligence, we lack categories when confronted with non-anthropomimetic programs. While we would surely refrain from attributing human qualities to a text-filtering agent, the task it performs seems somehow not very mechanistic or machine-like.[6] Agents seem to be in between; and very often, we even fail to see them.[7]

Changing computer use

The main reason for computer science's and the industry's new interest in agent development lies in the area of computer use and design[8] itself. PCs and home computers established microchips in the household, opening up the vast field of popular computer use: we rely on them not only for military, scientific or business purposes, but for entertainment and creative expression, learning and exploration. Nowadays, PCs are media of every kind, looking glasses and doors to virtual worlds. The consequential diversity and complexity of potential ways of doing things with computers increases the difficulty of knowing what to do and how to do it. And people concerned are not only researchers in lab coats anymore but virtually anybody; in western societies that is.

The Internet, with its millions of servers and still growing numbers of participants, repositioned the computer as a means of interaction with the world; access to information has never been as fast and powerful – for those who make the step inside. The proliferation of possibilities made available by the Internet creates complexities – and the resulting difficulties build up fears that prevent many people from going online. The digital divide is a reality inside western civilization. Many of those who plug in are quickly plagued by information overload and disorientation; but by heavily reducing

---

[5] To mention two: Wagner, Dirk Nicolas: Software Agents take the Internet as a Shortcut to Enter Society: A Survey of New Actors to Study for Social Theory. First Monday 5.7, 2000 (http://www.firstmonday.dk/issues/issue5_7/wagner/index.html) as well as Rammert, Werner: Giddens und die Gesellschaft der Heinzelmännchen. Zur Soziologie technischer Agenten und der Multi-Agenten-Systeme. In: T. Malsch (ed.): Sozionik. Soziologische Ansichten über künstliche Sozialität. Berlin: Sigma, 1998 p.91-128
Some of the issues linked to agents are closely related to questions of artificial intelligence in general. Among the most interesting works are: Dennett, Daniel C.: Brainchildren: A Collection of Essays 1984-1996. Cambridge MA: MIT Press, 1998, Dreyfus, Hubert: What Computers Still Can't Do: A Critique of Artificial Reason. Cambridge MA: MIT Press, 1992 and Weizenbaum, Joseph: Computer Power and Human Reason. San Francisco: W. H. Freeman, 1976

[6] Whereas a lot of people attributed human qualities to Joseph Weizenbaum's ELIZA (1966), a computer program that mimics human conversation. In technical terms, most of today's text-filtering algorithms are much more complex than the simple chatterbot that is ELIZA.

[7] How many people notice the existence of GroupLens, the collaborative filtering agent technology behind amazon.com's recommendation system?

[8] In this paper, "design" refers to both hardware and software. Both are intrinsically intertwined, but with platform independent languages as Sun's Java, software becomes increasingly detached from its actual implementation in hardware.

possibilities and creating an Internet "light" as AOL does it, the Network's potential stays largely untapped. Both realities are of course not very satisfactory.

The usability debate[9] has yielded some encouraging results and heightened the awareness for the issue at hand. User-friendly applications are becoming more widespread as screen design education is being institutionalized and professionalized. But the principle problem remains: functionalities and raw information are still increasing in a rapid pace, and *direct manipulation* interfaces seem to work fine only for those who are willing to invest considerable amounts of time into the necessary permanent learning process. Those who can or will not invest this time and effort are excluded from (parts of) the most complete informational representation of western society, the Internet.

If it is true that "our societies are increasingly structured around a bipolar opposition between the Net and the self"[10] as Castells argues, agents might be aiding in reconciling the two poles, traveling from one to the other, translating, connecting and structuring. The promise is unambiguous: easier and more powerful use of information technology without reduction of potential and choice. We are well advised to be cautious: things rarely come without a price.

Connecting – translating – filtering

While the Internet is a gigantic and complex construct, it is based on the common binary code. It is run on and accessed by computers. Any data on the Internet is stored on and consequently in principle readable by a computer, in contrast to the analog world outside. That does not mean that the digital world is homogenous, on the contrary. The different types of data syntax create zoning and every sector has to be accessed with the right tool. Microsoft Word for .doc files, Acrobat Reader for .pdf, a Browser for .html and so forth. Syntax aside, the semantic information stored in a file is only sometimes meant to be understood by a computer (binary executables: programs); most often it concerns the user. The machine only brings its symbols on screen, without further knowledge of their nature or meaning. That is the bases agents start from. Relying on today's ample supply of processing power, programmers attempt to bring some of that meaning in their zone of influence.

An agent's activities can be abstracted to three interrelated dimensions that somewhat structure the entirely heterogeneous world of existing and future software agents. Social sciences and humanities can connect on this level of analysis, as the three layers of functioning are not at all confined to the world of computers.

Connecting

While from an ecological point of view, it may be true that "everything is connected to everything else" [11] and it may also be true that most computers today are in one way or another (cables or wireless) connected to the Internet, in the digital world connection is not just here, is something that has to be established. Agents are in the connection business. First of all, they propose their services in connecting man and machine by supplementing or even replacing an interface. Also, agents are being developed that link by teaching; others help teams working together by coordination workflow and schedules. But most of all, agents will be used to find information. On the Web, search engines and directories have taken over the task of giving some degree of order to the rhizomatic structure of the net, helping with finding what the user needs. Information Agents take the search for records and

[9] In the usability debate, designers argue about the balance between graphical appeal and functionality. It was brought to new heights by the work of Jakob Nielsen (http://www.useit.com)
[10] Castells, Manuel: The Rise of the Network Society. Oxford and Malden: Blackwell, 2000 (second edition) p.3
[11] See: Barabási, Albert-László: Linked. The New Science of Networks. Cambridge MA: Perseus Publishing, 2002

documents to the next level. Software like Copernic[12] searches various information sources at once, tracks changes on sites and alerts users of those changes. Shop bots like Aristocart[13] help users find the best price for a product. Ubiquitous computing and mobile technologies pose the problem of connection in a new way. The administration of the user's attention, the connecting to what is important and the blocking of what is not, will gain much more importance than it has today. Push-technologies in general and proximity services in particular will give agents the opportunity to decide when to connect to a flux in the user's place. This is what could be called the delegation of connection – or the secretary function.

Translating

The translation function is on the more obvious level a translation between the realm of human concepts and machine concepts. Translation is vital to connection. While people are used to phrase their information needs in natural languages, computers administer information on the basis of symbolic logic. Interfaces in general are means of bridging this gap via metaphorical representation, and agents offer their help by making a further step into the direction of the user, hiding their machine origin always a little better. Information agents use different techniques to enable users to phrase queries in natural language instead of the usual combination of keywords and Boolean operators (or even SQL statements). Even rather complex non-informational responsibilities may be described in ways accessible to the non-expert user. Following the logic of conversation instead of command, agents can split up complex operations into step-by-step dialogs, proposing options and offering context-specific help.

But translation can also be understood in more technical way: while the Internet and computers in general provide a much more homogenous environment than the "real world" does, the basic communication protocols[14] are far from being sufficiently structured for consistent interrelation of information items. Even with information that is solely represented using the limited space of the ASCII[15] character set, there is no common syntax: text-files, HTML-files and XML-data, office documents and mail messages, all use different structuring elements. Common semantics, e.g. for description of information items, are of course even further away. Agents are often able to interface heterogeneous data types by either learning the structure of a data format[16] or working on the more abstract level of direct language analysis[17].

Structuring - Filtering

The computers ability not only to calculate and create data, but to filter and structure information has been a major interest of civil computer science from its start. While search engines like Google[18] already use high-level cross-referencing to determine result rankings, clustering techniques like Vivísimo[19] use AI to generate subcategories and structure a search result. This way, not only the latent problem of homonymy is tackled, but in the same step, imprecise queries or one-word searches become a lot more powerful and enable the untrained user to tap a database's full potential. In

[12] http://www.copernic.com/desktop/products/agent/index.html
[13] http://www.aristocart.com
[14] TCP/IP is the protocol that administers packet transport on the Internet. It tells computers how to send data so that other machines can receive it.
[15] The ASCII character set defines 128 characters. Many file formats are built on its base.
[16] Or getting help from another agent that already knows the format in question.
[17] An agent can be taught to ignore the syntactic elements of a file and only pay attention to the natural language information in a file, extracting it or relating it to other texts.
[18] http://www.google.com
[19] http://vivisimo.com

clustering results, the software enters a conversational mode by providing "suggestions" for refinement.

Synonymy, the second big problem of information mining, is already being addressed by various approaches[20] – but existing solutions are still somewhat limited by processor speed. We can expect advances in this area as microchips get faster and cheaper.

Clustering techniques are used to filter, classify and rate information. Google's news agent[21] monitors 4.000 different news sources and attempts to group articles by topic. Topics with high occurrence count are rated as important; those with only few occurrences are left out. Personal information agents will couple these functions with high personalization and user specific learning, becoming a membrane between people and the information storm going on in the outside world. I will discuss possible problems arising from the delegation of information filtering later on.

What all those information processing techniques have in common is that they make use of the structure of a document and the context it is found in, to make an educated assumption about its meaning. What they offer is not only a more comprehensive way of working with huge amounts of information, but a "view" on the data. This view is of course not based on the complex reasoning done by a human agent but on the algorithmic creation of relation and difference in human artifacts.

Code of conduct

Agents seem to reflect the change of the computer's inscription in our culture, its ongoing shift from pure machine to something else. To render this process more visible and facilitate a critical analysis from a social sciences and humanities point of view, I will try to get a closer look on software agents by establishing four different axes on which certain shifts can be emphasized. In some way, these changes mark not only the slowly emerging agent paradigm, but the recent developments in computing on the whole.

The following axes are rather a set of interrelated indications than closed categories, as we do not yet know in which directions agents will truly evolve. But the logic of delegation echoes through all four of them, since an agent performs its tasks either for the user or in the user's place.

Autonomy

The regular computer is nowadays equipped with a standard window-based desktop[22], where the user makes an instruction by clicking on the appropriate button. Most of the time, the computer will perform the action associated with it. Additionally, one may schedule tasks to be performed at a given time, e.g. a backup during the night. Advanced users will even be able to script small rule sets that trigger an operation when a certain condition is met, e.g. an email-sorting rule in a standard mail client software.

Agents go one step further: constantly resident in the computers memory (which may be local or on a server on the internet), they are able to constantly monitor their assigned environment (e.g. a newsgroup or the user's mailbox) and their own inner state. Actions will be performed in an autonomous fashion, based on the gathered data, the available knowledge of the assignment and its context and the agent's internal reasoning. The user only provides rough outlines for general behavior.

[20] An industrial example is Sensoria's "Influo" technology (http://www.influo.com).
[21] http://news.google.com
[22] Window based operating systems include Microsoft Windows, Mac OS and the different flavors of UNIX.

Interaction

Studies in interface design have been going strong trough the eighties and nineties but the window-based desktop and the command line interface are still the usual environment for communication with the machine. As show above, the standard man-machine interaction is based on the *direct manipulation* metaphor: a single command will trigger a single action. The metaphor herein follows the common logic of mechanics and the handling of analog machinery – the laws of cause and effect. In many cases it works rather well because we have acquired the cognitive capabilities to recognize and make use of *direct manipulation* in everyday live from early age on.[23]

Nevertheless, the mechanistic approach is coming under attack from (particularly but not only) agents, who are in general built on the *delegation* paradigm. When an environment becomes too complex for a given user and there is no knowledge of how to get to a specific result, the task[24] may be delegated to an agent who will perform in an autonomous fashion, sometimes asking the user for his or her consent or further details on the desired results. The mode of interaction thus becomes *conversational*.

While direct manipulation concentrates on the single command (message), a conversation follows a much more complex dramaturgy. An agent using a conversational interface not only considers the current command but takes into account the history of the ongoing conversation, as well as its knowledge about the user and the context the task is performed in. It therefore becomes possible to split up a complex assignment into several conversational items, at each step only considering a limited set of options, the interface thus adapting to the users level of expertise. The ability to help in defining and precising a goal follows out of this. Sometimes we only have a vague idea of what we want to do and interaction can help clear things up. In the extreme interpretation, one could say that the process of defining goals is in part delegated to the machine.

Learning

Standard Software normally comes with a fixed set of functions that perform the same way in an identical situation but at different points in time. Most often, certain functions are configurable, but once the basic parameters are set, any further adaptation has to come from the user – who most of the time ignores the vast majority of options software provides. Classic computer programs are therefore in principle highly predictable and the fact that we very often are surprised by a computers behavior is due to the high number of potential actions – and the combination of such actions. Programming errors are the second source of constant surprise.

Agents are not confined to this very complex form of inertia, which differs only slightly from the nature of mechanical devices. Agents have the ability to learn, to adapt their configuration and in some cases even their program structure. MIT's Pattie Maes gives four basic methods of agent learning[25]:

- Agents may learn by getting direct or indirect user feedback. The user can explicitly approve an agent's action (direct feedback) or just ignore a given suggestion (indirect feedback). Many agents will ask if not being able to decide.

- The user may also provide examples for actions in hypothetical situations and train the agent to behave the way he or she wants.

---

[23] Lakoff, George / Johnson, Mark: Metaphors We Live By. Chicago, London: University of Chicago Press: 1980 p.68

[24] For the moment only few tasks can be delegated to an agent, but there are several areas where agents have proven to yield encouraging results. See Green et al.: Software Agents. A Review. Intelligent Agents Group: 5, 1997 (http://www.cs.tcd.ie/research_groups/aig/iag/toplevel2.html)

[25] Maes, Pattie: Agents that Reduce Work and Information Overload. (http://pattie.www.media.mit.edu/people/pattie/CACM-94/CACM-94.p1.html)

- Agents may learn by just looking over the user's shoulder, monitoring, analyzing and then imitating.

- Sometimes and agent may ask another agent with already more experience how to handle a certain task.

An agent might thus at different points in time react differently in an identical situation. As agents learn, their behavior becomes less predictable. Creating knowledge about the user and the work environment helps to reduce the set of possible actions and allows once again the split up of rather complex tasks into smaller steps. In the best case, agent and user learn together, the process being in part delegated to software.

Reasoning

Agents make use of artificial intelligence techniques in most areas of their functioning. Sometimes their job can even be defined as facilitating the application of artificial intelligence algorithms, which would be too complex for the average user to calibrate. Agents hide the vast number of parameters present in every AI algorithm and configure themselves based on their perception of their inner state, their environment and the interactions with the user. All of the above dimensions require some degree of artificial intelligence to make sense.

The different approaches have brought about a wide variety of concepts for reasoning. Lately, we have seen a shift from standard binary logic to more "humane" ways of handling reasoning. Fuzzy Logic[26] allows for truth values – and therefore logical operations on the whole – to be handled gradually instead of a dualistic true/false; calculating the concept of "partial truth", which supposedly fits the human way of thinking, becomes possible. Perception, reasoning and generation of Information in the world we live in every day are not based on the clear cut data types that are at the base of Shannon's theory of information.[27] Artificial intelligence has reacted to this observation and nowadays works primarily on such more "social" forms of reasoning.

The delegation of perception and reasoning are by far the most striking issue when thinking about agents and their possible uses. While much of the actual industrial use of elaborated AI algorithms is still years, perhaps decades away, the principal possibility of partially handing over human core functions to computer programs seems disturbing.

While the three categories of functioning (connection, translation and structuring) were focusing on what agents do, the four axes refer to how they do it. And while I willingly accentuated the scope of those dimensions, following them bring us closer to the understanding of a shift that occurs in the design of computer software in general and in agent technology in particular. This process is of course related to changes in how people use computers, and consequently to changes in computers' status within society.

---

[26] Zadeh, Lotfi: Outline of a New Approach to the Analysis of Complex Systems. In: IEEE Trans. on Systems, Man and Cybernetics. 3, 1973

[27] Chicoisne, G. / Pesty, S.: Modèle de conversation et agents rationnels socialement corrects. In : Atelier Thématique TALN 1999: La langue dans l`interaction homme-machine. Paris, 1999 p.91-104

A new paradigm

In his work on the coming age of ubiquitous computing, Mark Weiser establishes a timeline[28] that reflects the changing relationship of people and computers. He distinguishes three "waves" of computing that sum up very well the fundamental difference between the early age of automatic calculation and its current form. I will follow his periodization, emphasizing on changes in use.

Fist Wave of Computing: Many users work on a big computer (mainframe), CPU time is shared. The computer is essentially used as number cruncher – as a giant powerful calculation and data administration machine.

Second Wave of Computing: Every user has his or her own machine (PC) which is small in size and affordable to most in the western world. The use of these new microprocessors expands largely: computer games, encyclopedias, image manipulation and sound applications are examples for the multimedia PC as a creative tool.

Transition: The Internet connects big computers (servers) and PCs, establishing computers as media in the classic sense and beyond: windows to the world, means of interaction with others and infinite information warehouse.

Third Wave of Computing: Ubiquitous computing means that no place is chip-free; users are surrounded by a multitude of very small processing units. Like writing and electricity, computing is everywhere and the digital becomes a layer on top of analog reality. With the third wave still ahead of us, we can only say that computing is set to become a part of nearly every cultural activity.

In Weiser's vision, computers become something fluid, adaptable and flexible – they are miniature and cost next to nothing; "smart dust" seems to be the fitting buzzword. From the monolithic and few mainframes to the billions of small chips in ubiquitous computing, what began as pure automated calculation for military applications has come a long way. To stress the metaphor, computers will be small enough to fill the spaces they could not squeeze in before, and will find uses that can be realized by reduced size and production cost.

While Weiser explicitly opposes[29] his vision on "calm" ubiquitous computing to the "loud" agent paradigm, there are two reasons why such an opposition is in fact not necessary at all.

First: Agents do not have to be intrusive. While it is true that many agents have been designed to be rather visible in their attempt to get the user's attention[30], using anthropomorphism and screaming graphics, "calm" agents are not only possible but probable. There is no necessity for agents not to populate the periphery and only come to the center when something very important comes up. And while it is true that the anthropomorphic design approach is very prominent in industrial

---

[28] Weiser, Mark and Brown, John Seely: The Coming Age of Calm Technology. Xerox PARC, October 5, 1996 (http://www.ubiq.com/hypertext/weiser/acmfuture2endnote.htm)

[29] Weiser, Mark: Open House. Xerox PARC, March 1996 (http://www.ubiq.com/hypertext/weiser/wholehouse.doc)

[30] The set of cartoon agents delivered with Microsoft Word is a good example for how annoying agents can be.

development[31], only the future will tell whether the user wants this kind of representation. Negroponte's idea of the information butler[32] might be not the way to go. Microsoft's Bob (1995) – a graphically elaborate desktop replacement heavily relying on pet-like agents – was perhaps the biggest flop in the company's history. Non-anthropomorphic agents that are limited in scope but highly specialized and effective in what they do might be much more adequate. Calm technology and the agent paradigm are in many ways a powerful combination.

Second: The Internet Revolution – and its current extension to the explosion of wireless devices and services – puts an emphasis on communication, integration and networking. It will be agents that help assimilate ubiquitous technology into the currently existing software sphere. It will be them who make the connection and translations necessary to navigate a heterogeneous environment. Even with XML[33] making its big entry, it is highly improbable that a single set of data definitions will unify the patchwork carpet of the digital world. The need for connection, translation and filtering is only growing.

There is no need to think ubiquitous computing and agents as opposed principles but rather as colleagues in the same quest: to establish computing in every corner of our lives, at work, at home and on the street. It is in fact when coupled with mobile technologies, which out of physical necessity use reduced interfaces and where traffic cost is very high, that agents are the most powerful. Filtering and pre-structuring help reduce traffic and conversational interfaces shrink the need for feature-packed interfaces.

Bricolage and Metaphor

While Weiser's perspective on the future of computing in relation to society seems in many ways highly pertinent – and not only because it is perfectly compatible with marketing rationale and consumerism – I believe that it leaves out something. Already today we see a multitude of different types of relating to digital technology, from libidinous to indifferent, from subcultural to military and from empowering to depending. It seems obvious that technological progress found its place at the heart of capitalist logic, and the propagation of microchips into the pores of society is a viable way of fueling the production-consumption spiral. But the focus on "waves" in computing leaves out all the contradictions, inequalities and struggles that have become so connected to technology nowadays. The computer has become a means for expression, resistance and individuation and no single paradigm, no unique metaphor does it justice. The target in the ubiquitous computing discussion is of course, first of all, the question of actual application design; but as Winograd and Flores put it, "We encounter the deep question of design when we recognize that in designing tools we are designing ways of being."[34] The focus on the technological side of things – that I have so far been guilty of as well – leaves out that living in the information age does not only mean work and recreation (which sometimes seem to be the only objectives of design), but also politics, privacy, empowerment, expression, critique and

[31] Especially telecom companies like France Telecom are working on graphically complex anthropomorphic agents, trying to underline the benefits of broadband access and hoping to reduce cost of product support and training.
[32] "The best metaphor I can conceive of for a human-computer interface is that of a well-trained English butler." Negroponte, Nicholas: Being Digital. New York: Alfred A. Knopf, 1995 p.148
[33] XML (eXtensible Markup Language) is a W3C standard that proposes a general, ASCII-based syntax for the structuring of content (http://www.w3.org/XML/).
[34] Winograd, Terry / Flores, Fernando: Understanding Computers and Cognition. A New Foundation for Design. Reading: Addison-Wesley, 1986 p.XI

relation. The metaphors people will choose to interact with the machine are going to be in accordance with those dimensions as well as others. When software design is more than the production of (virtual) objects, when we perceive it as production of culture, the question of "quality" not only implies a person with hypothetical needs and a task to be done and the straightest line between the two. Competence and trust, the two main factors of success in agent design[35], quality indicators so to speak, are mirrored on the level of society, and the stakes are not the same. Metaphors shape minds.[36]

Delegation of control

Following Jacques Perriault[37], communication machines – and agents can be seen as elaborate communication machines – are introduced into society to counter a latent disequilibrium. Western civilization's current technology related dilemma seems to be information overflow and the resulting stress and orientation loss. As I hope to have shown, agents are fit to provide a more comforting view on data reality, reducing and filtering, inserting yet another layer between man and his environment, mediating to heighten quality of interaction. The illusion of control is restored by delegation to an algorithm that even molds to models of people's needs. Orientation is regained and individuality maintained, manifest in a database of personal parameters. Another problem solved.

While we do not yet know whether agents will have their big breakthrough or how they will be used, we can see that the potential for a rather hard impact on our culture is there. The difficulty on the level of design is defined as the search for ways to reduce complexity without reducing potential for empowerment. This balance will be very hard to keep. In difference to other technical gatekeepers such as search engines, agents are most often programmed to place the user in the center of attention, thus giving him the illusion of being in charge[38]; but delegation always is a loss of control, a tradeoff for a bigger sphere of influence. And while agents can be used to initiate and train rather than simplify and hide, it is very unclear which alternative is going to dominate. Both are present in current research and development.

More than in the case of any other technology before, the agent's functioning disappears thoroughly in the black box. Not knowing the nature of the underlying algorithms, which are too complex for even the expert to adequately evaluate from the outside, we are tempted to put agents in a position of authority. An agent's view on information might consequently be taken as an exclusive or "better" view on information.

But agents provide interpretations which are in a sense very powerful, in another very poor. They are very good with statistic analysis of large amounts of data and rather bad with actual meaning; it is always a perspective, and information allows for the generation of infinite numbers of such interpretations. And every single one is like a map, a dimensional reduction performed to give a picture of a whole; but as in cartography there is no "correct" projection, just different traditions.

Günther Anders introduced in 1942, long before computers were parts of our homes, the notion of "promethean shame", which he defined as the "shame in the face of the 'humiliating' high quality of

---

[35] Maes, Pattie: op. cit.

[36] Lakoff and Johnson (op. cit.) see metaphors as the basic fabric that perception and action is built on. Interfaces are metaphors.

[37] Perriault, Jacques: La Logique de l'Usage. Essai sur les machines à communiquer. Paris: Flammarion, 1989

[38] "If an agent seems smart, it might really mean that people have dumbed themselves down to make their lives more easily representable by their agents' simple database design." Lanier, Jaron: My Problem with Agents. Wired 4.11, 1996 (http://www.wired.com/wired/archive/4.11/myprob.html)

manmade things"[39] providing us with a very powerful concept for analyzing modern technology. Especially with computers, people have the tendency to blame themselves and not the machine when something goes wrong. Because of the immense complexities involved with microchips and their software, we often attribute abilities and a degree of perfection to the machine that it just does not have. The process of interpretation and the creating of meaning in the dealings with agents may become a co-construction between user and software, where it is the agents that leads that the way – highly skilled in conversation as they are.

We often overestimate the machine and take it to be neutral and objective; but an agent is far from that. The shift in computing I described on the last couple of pages is in fact an attempt to make the machine more subjective, giving it a situated base in accordance to user, task and environment. And it has been designed by people with values and assumptions.

Following Virilio, I would argue that an agent is a highly elaborate "vision machine"[40] – it produces a view on reality. In doing so, it is based on its own perception of reality (ontology) and follows a programmed method of reasoning.

With perception and generation of perspective and interpretation partially delegated to a program, we might not feel any more need to dig in deeper and find alternative views on the infosphere. Slavoj Žižek's notion of interpassivity[41], where parts of the realization of the self are delegated to the external symbolic dimension[42], seems to strangely fit into a rather critical perspective on agents. With our agents constantly monitoring the information world around us, will we feel the need to go look for ourselves?

Choices

The cognitive and subsequent social consequences of a wide integration of agents into the fabric of digital culture are not yet to be judged. But some of the implications of handing over a part of our perceptional praxis to a software construct may well be identified. The software developer, embedded in the context he or she works and lives in, becomes even more than today an active actor in creating perceptional realities – vision machines. And while computers are powerful tools, they do nothing without instruction. And agent is an artifact created by programmers, and does not belong to the realm of magic. It is linguistics and computer science that enable such technologies. In the lack of a better comparison, one might say that the programmer enters the business of gate keeping, which in some way brings him or her close to journalists. But in writing software, the possibilities for exerting influence lay on a much more basic and subtle level. As metamedia[43], they are specialized in creating context and not content. They are not creating messages but conversations. And while journalism is ruled by an ethical code of conduct and special sets of laws, software production seems only to be inserted in the laws of capitalist production.

Whatever the possible use of agent technology, whether it will create nations of couch potatoes or informed citizens, it will be the existence of alternatives to choose from that will increase any positive effects and decrease the negative ones. Unfortunately, legislation and policy makers are right now doing just about anything they can to put the commercial interest of large companies above the greater

---

[39] Anders, Günther: Die Antiquiertheit des Menschen. Über die Seele im Zeitalter der zweiten industriellen Recolution. München: Beck, 1992 p.23, my translation
[40] Virilio, Paul: La machine de vision. Paris: Galilée, 1988
[41] Slavoj Zizek, The Plague of Fantasies, London and New York: Verso 1997.
[42] In this line of interpretation, the agent would take over the role of the lacanian "great other".
[43] Winkler, Hartmut: Suchmaschinen. Metamedien im Internet? Telepolis, 1997 (http://www.heise.de/tp/deutsch/inhalt/te/1135/1.html)

good of the community. Software patent legislation in its current form[44] considers software development as technical innovation and offers legal protection that eliminates the little freedom left by compilers.[45] The progress bar indicating the load state of a program is protected by United States Patent No. 5301348 and the European Patent No. 0394160. What can not be patented?

Related to agents, this means that one patent holder might soon exclusively manage the parameters of one type of agent. Perspectives on reality will be administered solely by companies big enough to recheck every line of code with an attorney. That's when the black box is not only closed but our hands are tied and we are unable to open it.

But a user-friendly society can only be an open society.

[44] See the "Proposal for a Directive of the European Parliament and of the Council on the patentability of computer-implemented inventions" (http://register.consilium.eu.int/pdf/en/02/st14/14017en2.pdf).

[45] Compilers translate human readable code into machine code. Even when somebody possesses software, he or she ignores the details of its functioning. Software is from the start much better protected against plagiarism than other technical devices.