

## Clarisse Sieckenius de Souza (SERG/PUC-RJ): Letras, Computação - interfaces e impactos

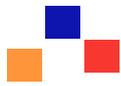
Por Vinícius Carvalho Pereira (UFMT)



Convidamos a Profa. Dra. Clarisse Sieckenius de Souza (Semiotic Engineering Research Group - SERG/PUC-RJ) para uma conversa sobre as interfaces entre Letras e Computação nas mais diferentes esferas e como esse diálogo pode impactar ambas as áreas. Ela é Bacharel em Tradução e Interpretação de Conferências - Inglês, Francês (1979), mestre em Língua Portuguesa (1982) e doutora em Linguística Aplicada (1987) pelo Departamento de Letras da Pontifícia Universidade Católica do Rio de Janeiro. É professora titular do Departamento de Informática da PUC-Rio. Sua pesquisa concentra-se na área de Interação Humano-Computador (IHC), na qual foi uma das pioneiras na Ciência da Computação brasileira. Seus principais interesses na área são: abordagens teóricas e científicas de IHC, métodos de avaliação de interfaces e interação com sistemas,

comunicação mediada por computador, *end-user programming*<sup>1</sup> e sistemas de explicação. Em 2010, voltou-se para o ensino de raciocínio computacional para crianças e jovens, trazendo seus interesses de pesquisa para este campo e propondo que programação - neste contexto - é uma forma de autoexpressão através de software. Clarisse é a criadora da Engenharia Semiótica, uma teoria de base semiótica para IHC, e (co)autora de artigos e livros internacionalmente publicados. Foi, por cinco anos, membro do Comitê Assessor de Ciência da Computação do CNPq e detentora de bolsas de "Cientista do Nosso Estado", concedidas pela FAPERJ, de 2009 a 2015. Em 2010, foi agraciada com o ACM SIGDOC Rigo Award e em 2013 tornou-se membro da ACM SIGCHI CHI Academy. Em 2014, recebeu o título de HCI Pioneer, outorgado pelo Comitê Técnico de Interação Humano-Computador (TC13) da IFIP.

1 Técnicas e tecnologias que permitem a usuários leigos criarem ou modificarem programas de computador.

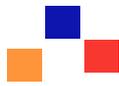


*Vinícius – Clarisse, gostaríamos de iniciar essa entrevista pedindo que nos contasse, ao cabo de mais de 30 anos de uma brilhante carreira, como você hoje entende sua trajetória acadêmica transdisciplinar, iniciada com Bacharelado, Mestrado e Doutorado em Letras, e hoje coroada com uma série de pesquisas, publicações, orientações e prêmios internacionais na área de Informática. O que a levou a esse movimento entre áreas do saber tradicionalmente apartadas por comitês de apoio à pesquisa e mesmo pela estrutura departamental das universidades?*

Clarisse – Eu já recebi várias vezes essa pergunta e a resposta sempre vai mudando ao longo dos anos, refletindo uma reinterpretação de vários *turning points* que aconteceram ao longo de minha história de vida. Há alguns episódios que contribuíram para isso e outras coisas constitutivas. Acho que a primeira coisa constitutiva já é minha própria origem familiar. Sou filha de um engenheiro matemático e uma psicóloga, isto é, minha mãe é formada em Psicologia e Filosofia, e meu pai é formado em Engenharia Elétrica e Matemática. Creio que esse tipo de dupla perspectiva já nasceu comigo: desde pequena, tive uma formação com dois tipos de olhar. Mas muita gente tem esse tipo de experiência na vida e não necessariamente faz o trajeto profissional que eu fiz depois.

Continuando na linha histórica, acho que há um episódio durante a minha juventude, quando eu fui morar na França, na época em que meus pais estavam fazendo pós-graduação. Esse período coincidiu com o do meu segundo grau (hoje, seria o Ensino Médio). Na ocasião, tanto na França quanto no Brasil, havia especializações desde o ensino médio: um ensino médio que ia para a área de Artes e Humanidades; um que ia para a área de Ciências Sociais; e um que ia para área de Exatas e da Vida. Naquela ocasião, eu já falava fluentemente francês, por estar morando na França, mas falava também fluentemente inglês, porque tinha estudado desde pequena. Eu queria ser intérprete de conferências, o que me levaria naturalmente a seguir a carreira de Humanidades, ir para Letras, Línguas etc. Mas a França daquela época não me deixou escolher; a escola em que eu estudava orientava os alunos para o segundo grau, de acordo com o que eles chamavam de *débouché* (as oportunidades de mercado). Os alunos que tinham possibilidade de entrar para certas fatias de mercado mais interessantes eram colocados nessas trilhas – mais interessantes à revelia deles próprios.

Então, o que aconteceu? Colocaram-me na trilha de Ciências Exatas, porque disseram que eu tinha uma cabeça de Engenheiro. E eu odiava, porque não queria fazer aquilo, mas o fato é que eu tive que cursar um ano desse Ensino Médio preparatório para Ciências Exatas na França. É interessante pensar retrospectivamente nisso, porque o currículo de Ensino Médio na ocasião, na França, não era o mesmo que o do Brasil. Lembro que, quando cheguei à França já no final do Ensino Fundamental, eu sabia fazer equação de segundo grau e várias coisas; talvez por isso tenha sido colocada nessa trilha de Ciências Exatas. Mas os franceses estudavam primeiro lógica, então havia um investimento desde o Ensino Médio na lógica e na álgebra. Nesse um ano que eu fiz de

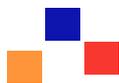


preparação para a carreira de Exatas, estudei bastante álgebra e bastante lógica, o que me deu um certo fascínio. Eu gostava da lógica e da álgebra, não a ponto de fazer disso uma carreira, mas acho que aquilo plantou uma semente que, mais à frente, vai explicar o que estou fazendo na computação.

Continuando, voltei para o Brasil, feliz da vida por poder retomar o meu sonho de carreira. Fui fazer o que na ocasião era o ensino clássico aqui, preparando-me para seguir a carreira de intérprete. Fiz um vestibular que, na ocasião, era unificado: fazia-se um único vestibular para todas as carreiras. Novamente eu obtive pontuação para ir para vários tipos de carreira que não eram as que me interessavam, mas eu insisti e fui fazer o curso de tradução e interpretação de conferências na PUC, que, na ocasião, era a única instituição aqui Rio de Janeiro que oferecia esse bacharelado. Fiquei muito feliz de poder fazer o curso e fui aluna, desde a graduação, de várias pessoas muito fortes na área de Linguística. Naquela ocasião, Marisa Pimenta Bueno estava voltando do Doutorado em Stanford e Regina Bustamante e uma porção de pessoas estavam dando aula de sintaxe formalista. Margarida Basílio foi minha orientadora depois de Doutorado. Eram pessoas muito voltadas para as teorias que, na época, tinham muito prestígio na área. Como aluna de graduação, fiz mais ou menos a mesma coisa que já tinha acontecido no meu Ensino Médio: ao mesmo tempo em que me preparava tecnicamente para uma carreira de tradutora e intérprete de conferências, adorava as questões teóricas que estava conhecendo em Teorias Linguísticas. Nas cadeiras de Linguística propriamente ditas, eu estudava o Gerativismo, Teorias Semânticas, e eu achava o máximo!

Mesmo antes de terminar minha graduação, já comecei a trabalhar como intérprete de conferência, o que fiz por bastante tempo. Aí vem outra questão episódica: na época, a gente estava sob a Lei de Informática no Brasil, de modo que havia proteção para o setor. Muitos consultores estrangeiros vinham dar cursos de treinamento para profissionais brasileiros na área de informática, então havia muita oportunidade de trabalhar como intérprete de conferência, justamente nesses cursos de treinamento. E foi o que aconteceu: eu terminei trabalhando com muita frequência nesse setor e fazendo tradução de grandes nomes da época que vieram ao Brasil. Assim, me interessei pela informática: inicialmente, pelo assunto do que eu estava traduzindo.

Quando terminei minha graduação, por uma questão realmente de pendor pessoal, achei que parar de estudar não era nem questão, embora não tivesse a menor necessidade de fazer o mestrado, porque eu já trabalhava como intérprete. No mestrado, estudei que tipo de treinamento os intérpretes deveriam receber na sua língua *target* para terem mais flexibilidade para fazer interpretação de línguas que exigem maior rapidez ao traduzir. Explicando em mais detalhes, eu estudava que tipo de estruturas sintáticas intérpretes deveriam desenvolver ou dominar, de tal maneira que, se tivessem que traduzir simultaneamente para o Português o que outra pessoa estivesse falando em uma *língua A*, dispusessem de estruturas sintáticas mais abreviadas (ou mais enxutas), não perdendo o *timing* atrás da pessoa. Eu achava aquilo divertido de estudar. Mais tarde, acabei descobrindo que depois minha dissertação foi utilizada na formação de intérprete na PUC.

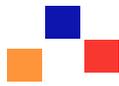


Ao final desse curso de mestrado, fui fazer um estágio como intérprete e tradutora na Universidade de Georgetown, uma das grandes escolas de tradução e interpretação da época, nos Estados Unidos. Durante o estágio, visitei a Organização Pan-americana de Saúde, onde já estava instalado e em funcionamento um sistema de tradução automática. Quando vi aquilo, falei: “Nossa, tenho que trabalhar neste sistema!”, pois achei fascinante alguém desenvolver um sistema de computador capaz de produzir texto numa outra língua.

Quando defendi a dissertação de mestrado, cheguei à conclusão de que seria muito legal continuar fazendo doutorado para estudar tradução automática. Então, fiz o mesmo trajeto de nove em cada dez pessoas que estudavam tradução automática naquela época: desviar da tradução automática para a compreensão automática de linguagens, o que na ocasião chamávamos de processamento de linguagem natural, isto é, modelos computacionais para interpretação de línguas. Para tanto, lancei mão dos recursos formais, algébricos, lógicos etc. que eu tinha desenvolvido ao longo da vida até ali. Trabalhei com formalismos de representação gramatical, sob orientação de Margarida Basílio, sobretudo em questões metodológicas. Ela não trabalhava nessa área, mas teve a generosidade de me orientar, ajudando-me a desenvolver um tipo de gramática semântica que conseguia dar suporte para certo tipo de uso de linguagem natural, em sistema de consultas a banco de dados. Eu desenvolvi um formalismo específico, mas bastante ameno e amigável para a parte de computação, embora não fosse ameno nem amigável para a parte de descrição linguística.

Olhando retrospectivamente, acho que aí eu já estava começando a trabalhar com esses decaimentos tão interessantes na computação, os quais pegam uma teoria que se propõe a descrever o fenômeno humano de produção de linguagem natural e cortam diversas características para apropriação prática num determinado contexto de desenvolvimento tecnológico. Com isso, pode-se decair, isto é, abrir mão de uma série de minúcias e exigências imprescindíveis para a teoria da linguagem natural; esse decaimento é possível no contexto de apropriação tecnológica justamente porque o processamento que um computador pode fazer de linguagem é muito menos rico do que o que fazemos mentalmente. Desse modo, as gramáticas podem ser menos perspicuas, mas sem prejudicar o resultado de uma interpretação automática de texto em determinado contexto.

Na minha banca de doutorado, estavam o Carlos Lucena, hoje meu colega de departamento, e o Roberto Lins Carvalho, que trabalhava na época com linguagens formais e processamento de linguagem natural. Ao final da defesa, eles me convidaram para ser professora visitante do Departamento de Informática, a fim de ver se os alunos se interessavam pelo que eu estava propondo. Para minha surpresa, a primeira turma que abri como professora visitante tinha vinte e cinco alunos, o que, na ocasião, para uma pós-graduação em processamento de linguagem natural, era um recorde absoluto. Depois desse período como professora visitante, eles me fizeram um convite para prestar o concurso para a PUC do Rio de Janeiro.



Fiquei me perguntando como prestaria concurso, porque na realidade eu não era uma pessoa de computação, mas o Departamento de Informática, seguindo a sua estrela de inovação e liderança, tinha um concurso de proposta temática, no qual se avaliava o que o candidato se propunha a fazer no departamento, o que ele gostaria de fazer, como ele via a área naquele momento e nos próximos dez anos. Isso eu conseguiria responder, mesmo sem ter formação em computação.

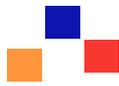
Havia seis vagas e vários candidatos, e eu fui aprovada nesse concurso, tornando-me, na ocasião, a única pessoa formada em Letras que era Professora Assistente de um departamento de Informática no Brasil. Aliás, até hoje acho que eu sou uma das poucas pessoas com esse perfil e talvez eu ainda seja a única Professora Titular de algum departamento de Informática que não é formada em Informática.

*Vinícius – Apesar de um discurso do senso comum (às vezes apropriado pela academia) de que as áreas de Letras e Ciência da Computação são completamente distintas, seguindo uma divisão Ciências Humanas versus Ciências Exatas, como você vê a dialética similaridade/distinção entre essas áreas? Quais são os pontos de tangência e quais são os limites dessa aproximação?*

Clarisse – Você comentou na pergunta anterior que as disciplinas são tradicionalmente apartadas, mas, na verdade, elas não são nada apartadas: essas duas disciplinas são incrivelmente vizinhas em termos conceituais, a despeito do que possa parecer. Por outro lado, do ponto de vista da política científica e disciplinar no Brasil, realmente a compartimentalização é horrível, mas eu não fiz esse trajeto entre as duas áreas por causa da política, e, sim, por causa da paixão e da oportunidade.

A razão pela qual eu consegui fazer a migração entre essas áreas foi justamente meu pendor pelo formalismo e pela algebrização, que fez com que eu segurasse na mão de Noam Chomsky e fizesse a transição muito confortavelmente de tudo o que eu tinha estudado em Linguística e sintaxe da linguagem natural para uma aplicação na informática.

Aliás, Chomsky é um herói para os dois lados: ele é o herói da teoria da computação e era, na ocasião em que eu estudei, o herói da teoria linguística. A partir desse fato, percebe-se que, dentro do paradigma que então vigorava na Linguística, havia um impulso muito grande para produção de uma descrição linguística bastante algébrica e formal, de modo que o paradigma computacional estava totalmente instalado. Do lado da Computação, também é interessante ver que hoje o processamento de linguagem natural se faz de formas bem mais probabilísticas, através de redes neurais etc., em um processo mais subsimbólico, ao passo que, naquela época, estávamos totalmente assentados na perspectiva de fazer um modelo de processamento de linguagem natural que fosse simbólico. Para tanto, queríamos identificar camadas de descrição sintática, camadas de descrição semântica, a formação de léxico; então, dada essa similaridade, havia uma conversa extremamente intensa entre as duas disciplinas, tanto na América do Norte, quanto na Europa e também no Japão.



Quando penso na similaridade hoje, talvez tenhamos que colocar essa questão uma oitava acima: afinal, o que é a computação? Na realidade, a computação é um exercício linguístico, pois, para poder trabalhar com computadores, é preciso expressar uma intenção de trabalho conforme um código linguístico. Não se trata de uma linguagem natural, e, sim, de uma linguagem computacional (e, portanto, artificial), mas, mesmo assim, a atividade linguística é o coração da computação: se tiramos a linguística da computação, esta fica parada no ar, pois não há nada que se possa fazer com ela.

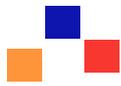
Quando eu comento que, para o processamento de linguagem natural, usam-se hoje em dia modelos probabilísticos que extraem o significado de alguma coisa com base na ocorrência de palavras, na linha do que se vê hoje nesses *bots* que fazem mineração de texto na web, é preciso deixar claro que não se está trabalhando com um sistema simbólico, isto é, com uma linguagem. Mas, para expressar que se quer que este cálculo seja feito pela máquina, usa-se uma linguagem de programação.

Então, tudo na computação é linguagem. Talvez não estejamos, hoje em dia, no que diz respeito à linguagem natural, utilizando um modelo em que simplesmente se transferem determinadas teorias linguísticas ou simbólicas para a computação. No entanto, ainda se precisa da velha estrutura simbólica de computação para haver o vocabulário, a sintaxe e a semântica de uma linguagem de programação, a fim de manejar essa linguagem para construir programas. Logo, se olharmos bem, estamos falando sobre léxico, morfossintaxe, semântica e pragmática, só que de um sistema de representações que não é a linguagem natural.

Por outro lado, acho que uma das grandes diferenças entre as duas áreas é que se necessita de uma cabeça de algebrista ou de lógico para trabalhar com computação. Eu até não tenho formação algébrica ou lógica suficiente – esta não foi a minha formação universitária –, mas acho que tenho uma maneira de pensar algébrica e lógica. E eu mencionei, ao longo da minha história de vida pessoal, momentos em que fui exposta a esse tipo de conhecimento e como aproveitei isso ao máximo e curti estudar essas coisas. Creio que essa seja uma linha divisória importante, mas que eu não chamaria de uma linha divisória disciplinar. Não acho que você precisa dizer que as disciplinas são diferentes, mas há nichos ou maneiras de abordar essas disciplinas que são diferentes.

É indispensável para uma pessoa que faz computação uma familiaridade com a álgebra e a lógica, pois ela tem que pensar em algoritmos como uma moeda corrente. Ela tem que conseguir olhar uma coisa e enunciar o que está vendo de uma forma computacional. Acho que isso é difícil para um aluno que seja formado como eu fui, com um trajeto dentro da Faculdade de Letras, dentro de um sistema de créditos que permite passar ao largo ou muito distante das questões de lógica e de álgebra dentro da Linguística. Pode-se fazer um trajeto mais humanista, que vai manter o aluno longe da Computação.

Mas eu também gostaria de comentar o lado da Computação, porque este é o curso em que sou professora de graduação. Já ministrei diferentes disciplinas, mas habitualmente sou professora de Interação Humano-Computador (IHC), curso que, na PUC, oferecemos



no primeiro ano na universidade, para que o aluno saiba desde o início que há gente do outro lado da computação.

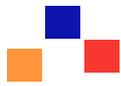
Também já dei aula de linguagens formais e de paradigmas de linguagens de programação (cadeira que nem existe mais), o que gostei muito de fazer. Os alunos olhavam para mim resabiados, talvez pensando: "Nossa senhora, como ela está contando essa história?" Isso porque eu contava essa história de uma maneira um pouco diferente, como no tipo de pesquisa que faço hoje, que se volta para o olhar da computação como uma forma de expressão. A ideia é olhar para os sistemas expressivos que estão à disposição e ver que tipo de objeto significativo se pode construir com esses sistemas. Esta é uma operação semiótica, mas passa pela linguística porque envolve linguagens de programação perfeitamente definidas, através das quais atingimos nossos objetivos.

A esse respeito, acredito que o aluno de Computação também precisa ter a oportunidade de enxergar aquilo que ele está fazendo como uma forma de comunicação e expressão; e aí, se ele puder fazer isso, também se aproxima de Letras. Da mesma maneira como comentei as condições de aproximação de Letras para Informática, querer chegar perto de algebrização e de lógicas, eu gostaria de comentar que, para um aluno de Informática chegar perto de Linguística, de Comunicação, ou de Semiótica, ele tem que querer olhar aquilo que está fazendo como ato de comunicação e de expressão.

Sobre os limites dessa aproximação, o que se pode fazer nas duas graduações é mostrar aos alunos o sabor desse processo, dando a oportunidade aos alunos de uma área de chegarem perto da outra, e vice-versa. Dá a oportunidade para essas pessoas continuarem estudando tal relação entre áreas e se aprofundarem nessa aproximação talvez não seja algo para a graduação, mas, sim, para um projeto de extensão ou de pós-graduação, em que se comecem a sistematizar um pouco mais alguns conceitos de uma ou de outra área, a fim de usá-los de maneira mais instrumental. Esta é a condição *sine qua non* para poder praticar a interdisciplinaridade. Uma coisa é ser curioso da interdisciplinaridade, como eu fui durante muito tempo; outra coisa é ser praticante da interdisciplinaridade (que eu também fui durante bastante tempo).

*Vinícius – Entre as suas principais contribuições epistemológicas e metodológicas à área de Informática, está a Engenharia Semiótica, pela qual seus estudos se tornaram um dos principais pilares para entendimento da Interação Humano-Computador segundo um paradigma comunicacional. Como podemos entender a Engenharia Semiótica dentro do quadro mais geral das Teorias Semióticas? E o que ela traz de mais específico à comunicação que se dá por meio de uma interface computacional?*

Clarisse – Responder a essa pergunta tem uma premissa, que é a minha capacidade de falar sobre as teorias semióticas em geral. Eu não me acho competente para isso e, às vezes, as pessoas não entendem quando digo: "Eu não sou semioticista, eu estudo semiótica". Assim, eu gostaria de fazer uma preliminar à resposta, dizendo que, na



realidade, a Engenharia Semiótica é uma coisa mais modesta do que parece. Às vezes, ela parece ser uma coisa mais importante, a despeito do que eu tento explicar.

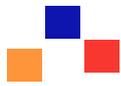
Eu nunca quis que a Engenharia Semiótica fosse algo pomposo, mas, às vezes, ela parece requintada porque, para a computação, a ideia de semiótica é meio arcana, meio difícil, meio complexa, então parece que quem consegue sequer entender o que é semiótica já deve ter algum tipo de pendor especial. Que dirá propor uma Engenharia Semiótica! Mas não é nada disso. A Engenharia Semiótica é interdisciplinar porque conversa com outra disciplina, porém não pretensiosamente. Ela conversa modestamente; todavia, às vezes, é ouvida pretensiosamente.

Embora tenha se tornado uma teoria de um fenômeno computacional, a Engenharia Semiótica é uma simples apropriação de determinados conceitos das semióticas de Peirce e de Umberto Eco para explicar como a computação é entregue ao seu destinatário. O objeto da Engenharia Semiótica, aquilo que ela quer descrever, é essa última parte do processo computacional, quando um programa chega ao seu usuário, seja ele leigo ou profissional. Desse modo, ela versa sobre o fenômeno da passagem do meio computacional ao meio social. Nunca pensei na Engenharia Semiótica como tendo uma contribuição para a própria Semiótica, até porque sempre achei – e ainda acho – que não sei semiótica o suficiente.

Ao longo dos anos, tive muitas oportunidades de discutir com semioticistas de bom calibre a respeito da Engenharia Semiótica e recebi também críticas deles de que não utilizei o conceito tal como ele é, portanto estaria fazendo um reducionismo. Muito honestamente, sempre comentei: “É, estou mesmo fazendo isso”; esse crachá eu uso na lapela. Sempre digo que a Engenharia Semiótica não é uma teoria semiótica concorrente com outras teorias de semiótica. Ela é uma teoria para descrever um fenômeno dentro da computação.

Mas o que a Engenharia Semiótica traz para a comunicação que se dá por meio de uma interface computacional? Talvez, para responder a essa pergunta, eu possa transformá-la em: Por que eu criei a Engenharia Semiótica, para começo de conversa? Afinal, ela não é a única teoria semiótica para a computação ou muito menos para a IHC. Tive precursores, como o dinamarquês Peter Bøgh Andersen, que publicou o primeiro livro sobre semiótica e computação; o Mihai Nadin e o Winfried Nöth também têm na área de semiótica muitos trabalhos sobre a computação; e por aí vai.

Agora, por que eu arregacei as mangas e resolvi que não queria nenhuma dessas teorias, e, sim, fazer outra? Para explicar isso, vou pegar o caso do Peter Bøgh Andersen, que eu acho que tem a teoria mais completa. Andersen utilizou uma semiótica do também escandinavo Hjelmslev, com uma perspectiva estruturalista, para analisar a computação. Assim, descreve-a como uma coisa que está lá do outro lado do rio, falando sobre um fenômeno da computação para quem está sentado no território da semiótica. Desse modo, parece-me que ficaram faltando na teoria dele alguns elementos fundamentais para uma possível apropriação das contribuições semióticas por quem desenvolve programas.



Um desenvolvedor de software que se depare com a teoria do Andersen pergunta-se: “Mas o que eu faço com isso? Andersen deu explicações lindas sobre várias coisas, mas o que eu entendi delas? Como é que elas mudam a minha vida?” Talvez por incompetência minha, não tenha encontrado a resposta a essas perguntas. Então, propus-me a estudar essas questões e começar a trazer os conceitos semióticos de uma maneira que falasse mais perto do coração da Computação. Fiz isso pautada na minha própria história, nos meus momentos de perplexidade, surpresa, inspiração, ou estímulo para estudar mais. Enfim, o que eu achava tão incrível na computação e como é que a teoria semiótica poderia alavancar o pensamento sobre essas questões que eu julgava interessantes?

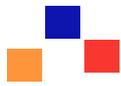
Hoje em dia, acho que posso contar *in a nutshell* aquilo que eu acho mais interessante na computação: o fato de que o tempo e o espaço da comunicação mediada por computador não têm comensurabilidade com a experiência psicológica humana de tempo e espaço. Por exemplo, quando alguém faz um programa de computador, diz assim: “Este programa está falando o que eu quero dizer, dando um recado meu para outra pessoa”. Só que esse programa vai se comportar infinitas vezes da mesma forma, o que o torna infinito. Ele pode rodar infinitas vezes, mas o infinito dele é bem diferente daquele de seu desenvolvedor, porque, se este tivesse tempo infinito, sempre construiria no seu momento  $T+1$  uma realidade diferente da que foi vivida na realidade  $T$ . Ela sempre seria diferente por causa do processo semiótico, pois sempre se amplia o espaço de significação a cada momento da vida. A ampliação aqui é no sentido qualitativo, não no quantitativo, pois não quer dizer que se esteja angariando mais; pode-se até ter um momento de retração porque se aprendeu que se deveria retrair.

Essa dimensão qualitativa de ir ganhando *insights* que não são consequências (deriváveis, no sentido algébrico) do seu estado interior faz com que, na experiência humana, se tivéssemos um tempo infinito, nunca faríamos duas vezes exatamente a mesma coisa. Eis-nos de volta à história de não nos banharmos duas vezes no mesmo rio.

Na computação, é o contrario: um programa de computador está condenado a fazer infinitamente a mesma coisa. Isso é muito interessante porque o tempo na computação é infinito, mas é infinitamente o mesmo, que é uma experiência que não se pode ter humanamente. Acho isso uma coisa incrível do ponto de vista de comunicação.

Aí é que começou essa minha inspiração de que a interface, na realidade, inspeciona um momento seu de semiose e o repete infinitamente, que é uma coisa que você mesmo não consegue fazer, relatando aquele seu momento. Mas quando você codifica a sua interpretação do mundo em um programa de computador, essa interpretação está então fixada e pode se repetir infinitas vezes, em um processo de extensão temporal.

E também temos a extensão espacial, no sentido de que esse programa pode rodar em infinitas máquinas, ou infinitos lugares, hoje em dia com a internet. Dessa maneira, é interessante como a computação quebra a noção de espaço-tempo porque abre uma porta para um infinito, tanto no espaço quanto no tempo. Mas esse é um infinito que é só mais do mesmo. É um infinito engraçado, poderoso.



É fascinante estudar como eu posso, através de algoritmos, criar essa visitação do infinito, multiplicar-me infinitas vezes no tempo e no espaço (quando digo “multiplicar”, refiro-me a um pedaço de mim que só existiu naquele momento e já não existe mais). Essa é uma questão muito interessante de ser observada no processo de comunicação por meios digitais; não é só como a digitalização vai mudar o caráter estético do que se faz, ou permitir que se criem virtualidades e que se transgridam princípios da realidade. Tudo isso é, sem dúvida, muito legal, mas não é o que me fascina pessoalmente. O que me fascina pessoalmente é esse embate com um infinito que é, mas não é.

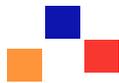
Essa é uma questão incrivelmente bacana de estudar, inclusive pelas consequências socioculturais desse fenômeno. Quando se faz uma tecnologia, ela tem valores fixados, que se replicarão infinitamente no tempo e no espaço, mas aqueles valores são parados. Então, como é que a sociedade se apropria disso? Qual é o impacto social de repetir infinitas vezes a mesma coisa?

Só para dar um exemplo, se, hoje em dia, alguém coloca uma mensagem errada na internet, intencionalmente ou não, e nunca mais a recolhe, aquilo vira um fato. Não há como “descriar” aquele fato, mesmo que o desmintamos (supondo uma boa intenção de, em seguida, fazer um esclarecimento), pois, como o fato e o desmentido não foram gerados conjuntamente, serão recuperados independentemente nos seus caminhos infinitos.

Assim, cria-se uma realidade muitíssimo diferente da que se tinha com os meios de comunicação de massa até aqui. Você pode até objetar: “Os livros poderiam espalhar mentiras e o desmentido poderia vir num próximo livro que talvez eu não lesse”. É verdade, mas é interessante pensar que o livro tem a representação do texto, enquanto a computação tem a representação da ação. A ação do livro está num espaço do perlocutório, da pragmática: é o impacto fora do livro. Mas a computação tem um impacto dentro do próprio programa. Há programas que mudam a realidade; se eles mudarem um valor num banco de dados, a realidade mudou.

Isso é bem diferente de um livro, pois o efeito perlocutório dessa comunicação que traz para dentro da mensagem a capacidade física ou virtual de atuação já está sob controle de efetuação da própria mensagem, de uma maneira bem diferente do que as palavras nos atos de fala declarativa. Nestes, dependendo da sua posição (se você for um juiz, por exemplo), você pode mudar a realidade pela sua fala, mas a computação faz isso o tempo inteiro. Todos os atos de fala de uma interface computacional são declarativos, pois criam essa virtualidade do mundo.

*Vinícius - Em alguns dos trabalhos mais recentes do SERG, seus integrantes vêm enfatizando a necessidade de a Ciência da Computação, na medida em que fala de fenômenos de linguagem, atravessar também um pragmatic turn, tal qual o experimentado pela Linguística na década de 80. Em que consistiria uma virada dessa natureza para a área de Informática? E como um trabalho mais próximo à área de Letras poderia contribuir para essa mudança de perspectiva?*



Clarisse – Essa pergunta é bem interessante. Uma das coisas que me chamam muita atenção é o quanto eu já não entendo de linguística, tendo em vista que minha migração para computação se fez justamente por volta desse *pragmatic turn*, que estava começando quando eu entrei para a informática. Eu conheço uma velha teoria linguística, mas consegui ler, acompanhar e entender, ainda que de longe, essa grande mudança do ponto de vista de teorias da linguagem, que eu achei realmente fascinante.

A computação ainda está na linguística da década de 60, o que é um pouco chocante. A teoria da computação sustenta tudo o que a gente faz em tecnologias de informação. Hoje, em termos de tecnologia, estamos *pervasivos*: a tecnologia afeta a vida de todo mundo o tempo inteiro, imbricada no tecido sociocultural e na psicologia humana. Apesar disso, a teoria da computação que está por baixo não tem nem pragmática, parando no mundo semântico.

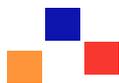
Então, causa certo incômodo ver teóricos da computação que ainda estão discutindo semântica. Eu entendo por que eles estão discutindo isso, afinal a semântica que eles discutem é operacional. Nesse sentido, aquilo que um programa faz é entendido como semântica, não havendo uma distinção clara entre essa dimensão e a pragmática.

No entanto, uma coisa é não ter a distinção clara e reconhecer que as duas dimensões existem; outra é não ter distinção clara e chamar tudo de semântica, achando que se trata de uma coisa só. Nisso eu vejo o quanto a teoria da computação está defasada da teoria linguística e o quanto, por isso, tem perdido poder de explicação teórica sobre alguns dos fenômenos que vemos acontecer, em termos de tecnologias de informação e comunicação.

Dentro da computação, as áreas que se preocupam em tratar desses fenômenos são esta em que trabalho, a Interação Humano-Computador, e a computação móvel, computação ubíqua, sistemas multimídia, ou até mesmo a área de inteligência artificial. Mas é interessante que elas entram nessa discussão tendo que se valer de teorias outras que a teoria da computação, da mesma maneira como eu abri mão da teoria da computação e me vali de uma teoria semiótica, ou como outros que prescindem da teoria da computação e pegam uma teoria cognitiva. Enfim, são escolhas.

A pergunta que me faço é bem mais central: por que a teoria da computação parou no tempo desse jeito? Será que é por que as áreas como a Interação Humano-Computador (para falar da que eu mais conheço) estão fazendo um trabalho bastante intenso e dando conta de complementar aquilo que a própria teoria da computação não consegue prover? Acho esta uma explicação ruim, para falar a verdade, inclusive porque alimenta um preconceito dentro da computação em relação às áreas periféricas, ou áreas *fringe*.

IHC é uma coisa meio *soft*, meio periférica, não é *core computer science*, não é *hard computer science*. Já a *hard computer science*, francamente, não dá conta do recado e deveria se ocupar muito mais seriamente das questões humanas, estudando sua extensão pragmática de uma maneira muito mais efetiva. Se isso acontecesse, teríamos um movimento bacana, em que áreas como a Interação Humano-Computador poderiam ser a ponte para que Letras, Linguística e Psicologia viessem a trabalhar conosco no coração da Computação. Mas a gente não tem visto isso.



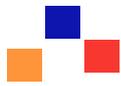
Fazendo um comentário aqui, quase como uma nota de pé de página, mas bastante relevante, vemos hoje as carreiras de Computação se tornando cada vez menos atraentes para os jovens. No mundo inteiro, temos visto, há uma década, o fato de que os jovens não estão mais interessados por essas carreiras, apesar do mercado de trabalho extremamente pujante. Acho que uma das possíveis razões – como conjectura minha, professora de Computação – é que, quando se começa a estudar a teoria da computação, essa apresentação se dá de uma maneira completamente divorciada da realidade sociocultural. Ora, uma teoria que não tem nada a falar sobre a realidade em que se vive não é interessante para o perfil dos jovens de hoje, tão conectados com a realidade do seu entorno (inclusive, durante a aula, eles continuam conectados com os amigos). Uma teoria que não se junta com nada, que não se liga com nada, está fadada a ser desinteressante.

Talvez um aporte de várias disciplinas, como Letras, Linguística, entre outras, pudesse ajudar a computação a se abrir um pouco para sua inserção na realidade sociocultural. Assim, teríamos uma versão mais completa da teoria da computação. É possível haver uma versão teórica extremamente formal e a Linguística está aí para mostrar exatamente isso: pode-se fazer uma descrição linguística rigorosa e obter, com a adição da camada pragmática, um fenômeno mil vezes mais interessante para caracterizar.

Além disso, há outros pontos mais aplicados em que esse *pragmatic turn* na Computação poderia ser amparado nas discussões da área de Letras. Por exemplo, hoje em dia a *storytelling* é muito importante para a computação.

Fala-se muito em jogos (sérios ou para entretenimento) e já há bastante gente interessada em estudar enredos e histórias. Também tenho trabalhado com meus alunos em cima de histórias, mas a partir de dados, não só a partir de jogos. Eu me preocupo com uma coisa chamada jornalismo de dados e a inocência que abre espaço para o uso mal-intencionado em relação ao tratamento de dados e o que estes querem dizer. Ainda há muito pouca curiosidade sobre a proveniência dos dados, o modo como foram levantados, que tipo de significado eles, de fato, podem expressar, quantas histórias se podem contar a respeito da mesma configuração de dados... Acho esse exercício de pensamento crítico muitíssimo importante e tenho tentado trabalhar com meus alunos desde a graduação em cima disso. Agora estou tentando uma reunião com o Departamento de Educação e de Letras da PUC e alguns professores que estão interessados em trabalhar *Storytelling* a partir de dados tabulados.

Nessa época de *big data*, *e-science* etc., cria-se uma certa glorificação do dado quantitativo, das tabelas: "Ah eu levantei esses números", "Os números estão na internet", "Os números não mentem"... É claro que os números mentem o tempo inteiro! A questão é essa ingenuidade que as pessoas têm na aproximação dos números, e a disciplina de Letras enriqueceria muito o pensamento crítico da Computação nesse âmbito. Lembro-me de um livro da Martha Nussbaum, intitulado *Not for Profit*, o qual foi publicado há alguns anos e discutia as consequências dos cortes dos orçamentos para pesquisa nos Estados Unidos, em que se reduz o apoio ao desenvolvimento das áreas humanas e mantém-se o aporte para as áreas tecnológicas.



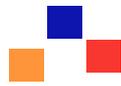
A esse respeito, é interessante o que ela diz sobre a função da ficção e da Literatura em desenvolver a empatia. A leitura literária desde a infância desenvolve a capacidade de se colocar no lugar de outros personagens, diversificando a experiência emocional. À medida que se desprestigia esse tipo de saber, contribui-se para o desenvolvimento de uma sociedade em que as pessoas cada vez têm menos capacidade de empatizar com o outro. Assim, não é surpreendente que, em tempos recentes, estejamos assistindo a sociedades completamente radicalizadas e impossibilitadas de tolerar outra sociedade. Basta olharmos para o que aconteceu durante décadas, que foi o progressivo desprestígio do saber ou da prática da ficção e da arte de contar histórias e de imaginar, visto que o fomento migrou para áreas mais técnicas, atendendo a uma demanda do poder econômico.

Nesse contexto, acho interessante também abordar esse papel da área de Letras, sobretudo num momento em que a importância dos dados cresce tanto, seja do ponto de vista do jornalismo de dados, seja da ciência. A glorificação excessiva do dado e da pesquisa quantitativa cria uma situação fantasmagórica. Como há sistemas computacionais que alavancam esse tipo de pesquisa, acelerando sua obtenção de resultados, a pesquisa qualitativa vai ficando cada vez mais devagar e discriminada.

*Vinícius – Por sua vez, se a abordagem que a Linguística faz dos fenômenos semióticos e pragmáticos pode ajudar a pesquisa em Ciência da Computação, certamente o efeito retroativo também é verdadeiro. Assim, como você entende que o que se faz hoje em termos de Linguística Computacional, Pragmática Computacional e Engenharia Semiótica pode nos ajudar a entender melhor os fenômenos da linguagem humana? Você acha que essas contribuições podem se dar tanto em nível mais “laboratorial”, permitindo diferentes análises e testes de fenômenos das línguas naturais, quanto sob um viés mais teórico e filosófico, a fim de compreender o que é, em sentido mais amplo, a linguagem?*

Clarisse – Tenho que invocar aqui a minha ignorância em relação à Linguística Computacional, que não estudo há bastante tempo. Nessa área, se você fica dois anos sem estudar, perde o pé. Só posso falar sobre um nicho bem específico, que seria a Engenharia Semiótica, e talvez a partir desses meus pequenos ensaios sobre Pragmática Computacional, em que se discutem linguagens computacionais (inclusive a linguagem de programação) como sendo passíveis de um tratamento pragmático.

Acho que, de certa maneira, eu já endereci um pouco essa questão quando falei sobre o espaço-tempo. O discurso tecnológico que se faz com a linguagem artificial produzida pelos humanos para realizar certos efeitos nas sociedades tem algumas características interessantes de espaço-tempo que, por contraste, podem sublinhar talvez as condições de produção do discurso natural. Quando eu falei que, por meio da computação, criam-se infinitos, mas que são mais do mesmo, essa fixação de um processo é algo muito interessante. Cito aqui um conto do Umberto Eco que foi publicado no livro *Meaning and Mental Representations*, do qual ele é um dos editores. Publicado em 1988, o livro é uma



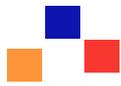
coletânea de artigos sobre significado e representações mentais, contendo um capítulo que é um conto, intitulado “On Truth a Fiction”, bem no estilo do Umberto Eco. Aliás, esse conto aparece no capítulo final do meu livro de 2005, em que faço uma consideração de natureza um pouco mais filosófica sobre o que eu estava propondo ali.

Muito brevemente, o que esse conto faz é argumentar, através de um enredo de ficção, que a forma como os computadores funcionam é uma representação do que nós achamos que seja a forma como nós funcionamos. Mas isso não quer dizer que seja uma representação de como nós funcionamos; é uma representação da nossa crença sobre como nós funcionamos. Assim, temos representações que são mais abstratas, mais concretas, mais funcionais, representações mais lógicas, mais declarativas, mais proposicionais... Temos todo tipo de representação e é interessante como essa proposição do conto do Umberto Eco teve uma imensa influência sobre mim. Ele conseguiu transmitir exemplarmente o que julgo a onda de maior impacto da computação sobre estudos de semiótica e, talvez, estudos de linguagens: a possibilidade de representar de uma maneira tão rica em detalhes o que achamos que está acontecendo num processo de uso de linguagem.

Isso exige a criação de uma hermenêutica específica, pois, quando se abre um programa de computador e se quer enxergar ali rastros da crença de seus criadores sobre seus próprios funcionamentos linguísticos e cognitivos, é preciso ter muita sofisticação. Uma hermenêutica dessa natureza não pode ser ingênua; ela tem que estar atenta às condições de engenharia de software. Vou citar um exemplo bastante banal: hoje em dia, a programação se faz em cima de componentes. Logo, um programador que recebeu a incumbência de fazer um programa que realiza a tarefa X vai pegar partes que foram programadas por várias outras pessoas. Então, o que ele está fazendo é uma espécie de colagem de fragmentos de programas de outros. Nesse contexto, é interessante até pensar na noção de composicionalidade de sentido: como, por exemplo, um programa que vai marcar eventos na minha agenda é a somatória de pequeníssimos programas de que cada um fez um pedacinho? Trata-se de uma reedição de teorias composicionais de significados, teorias semânticas em cima de traços, só que o gozado é que esses traços são construídos por pessoas. Não são traços ideais, e, sim, traços práticos que emergiram dessa pragmática da computação.

Acho essas questões muito interessantes e eu talvez tenha algumas dicas e perguntas a esse respeito, mas passo muito longe de ter competência para executar isso. Tenho alguns anos de prática e de convívio com programadores, e eu mesma já coordenei projetos de desenvolvimento de software conceitualmente. Não sou uma boa programadora; aliás, eu não sou uma programadora. Faço isso por diletantismo, e mal, faço para meu próprio consumo, mas já coordenei o projeto conceitual de sistemas, o que acabei aprendendo a fazer. Para esse tipo de atividade, é preciso ter a sensibilidade de como e de onde se extrai o significado de um programa final.

Durante muito tempo, trabalhei numa equipe que dizia que eu era a melhor pessoa para testar software porque eu conseguia derrubar o sistema em dois tempos. Eu dava



gargalhada porque no fundo era verdade: de fato, eu derrubava muito rapidamente o sistema; então, comecei a me perguntar por que outras pessoas não identificavam rapidamente aquelas falhas. A questão que eu estava testando e que fazia o sistema cair era algo com que eu me importava; logo, como é que não tinha passado pela cabeça do programador? Durante muito tempo, fiquei naquela história de que o programador estaria tão comprometido com o que fez que se tornava incapaz de olhar de fora. Então, eu achava que era simplesmente o fato de eu estar de fora que fazia a diferença.

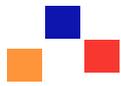
Hoje em dia, ainda acho que estar de fora seja o fator mais importante, mas eles se surpreendiam com a minha velocidade ao me comparar a outras pessoas que também estavam de fora e testavam o sistema. Então, o que fazia a diferença? Creio que fosse essa suspeita meio paradigmática, que consistia em olhar como o programa foi construído e tentar colocar o dedo na falha do processo de construção, especialmente na costura entre as partes diferentes de onde supostamente emergiriam seus significados. O que eu fazia era cutucar a costura das partes, perguntar pelo significado de cada parte e explorar a ignorância sobre o que as partes queriam dizer.

Recentemente, orientei uma tese de doutorado de um aluno excelente, em que nós trabalhamos sobre linguagens de programação, na forma de pacotes de programação que se chamam APIs (*application programming interfaces*), como se fossem exatamente esses componentes de que falei mais acima. Essas camadas de programação já trazem funções prontinhas para desenvolver com mais rapidez algumas aplicações computacionais. Por exemplo, temos API do Google Maps, que acelera como você *plota* dados georreferenciados sobre um mapa. Nesse contexto, fizemos um trabalho interessante de nos perguntarmos quais são os pressupostos que quem faz uma API imagina em termos de suas condições de uso.

Se tivéssemos uma teoria semântica em cima de componentes de significados e estes não viessem abstratamente de algum modelo ideal de linguagem, mas fossem construídos – que é o que acontece na computação –, faríamos a seguinte pergunta: “Quem construiu esse componente o fez com que intenção?” Na computação, a intencionalidade do construtor equivale à função desse componente, mas o interessante é que essa intenção é lançada dentro de uma teia de pressupostos e valores, os quais a computação não questiona em termos pragmáticos. O impacto de não discutir o que quem fez aquele pacote imaginava sobre os contextos de uso é que, volta e meia, os pacotes falham, normalmente na mão dos usuários.

Eu orientei esta tese porque grande parte dos problemas de IHC, acho eu, se originam nas camadas mais profundas da Engenharia de Software. Normalmente, se aborda muito a questão de design da interface, que é importante, mas eu acho que não vamos resolver os problemas de IHC parando por aí. Temos que entrar na forma como o software é codificado e como os valores do desenvolvedor são inscritos em cada linha de código que ele faz.

Como meus alunos mesmo dizem sobre a sua formação, o desenvolvedor é treinado para construir software uma maneira impessoal. No entanto, o software só existe



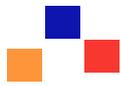
porque existem pessoas. É como se eles estivessem fazendo uma coisa completamente objetiva e impessoal, mas eles são pessoas e estão inseridos dentro de uma sociedade, de uma cultura, de uma prática profissional, em que um conjunto de pressupostos vigora e outro, claramente, não.

O problema é que a computação não tem nenhum instrumento para começar a sondar a natureza desses pressupostos e a extensão em que eles determinam a composicionalidade de significado quando são usados por outros programadores. Voltando à questão do Umberto Eco, quando olhamos a maneira como os programadores compõem os seus programas, vemos todas essas idealizações, tudo o que eles acham que eles estão fazendo; enfim, vemos um discurso implícito deles sobre a comunicação que aquele software está realizando, seja em nome deles, seja em nome de terceiros para quem estão trabalhando. Temos aí um campo de pesquisa maravilhoso, que pode ser conduzido por um conjunto de pessoas que, neste momento, têm um pendor e uma curiosidade, mas ainda são praticamente autodidatas sobre essas temáticas.

Esta minha crença me dá a ousadia de dizer que isso é algo que eu gostaria de estudar, mas ainda não tenho ferramental para isso. Você falou em questões epistemológicas e metodológicas, mas eu não sei os limites epistemológicos disso, nem as metodologias adequadas para fazê-lo. Seria um trabalho científico muito interessante construir essa epistemologia em torno de um problema concreto. Como Umberto Eco diz, pode-se abrir o código e inspecioná-lo, pois ele tem em si uma concretude muito interessante. No entanto, quando se põe aquele código para funcionar, ele começa a criar uma extensão para uma virtualidade, não exatamente uma concretude.

Seria muito bacana haver um grupo de pessoas interessadas em pensar sobre essas coisas, pois a nossa sociedade está sendo determinada por acontecimentos que não são bem entendidos, em uma velocidade espantosa. Quanto mais tarde viermos a entender essas coisas, pior será para revertermos os efeitos negativos que elas, por acaso, possam acarretar. Nesse sentido, o atraso na criação dos mecanismos para que tal interdisciplinaridade seja alimentada, nutrida e estimulada pode ter um custo bastante elevado.

*Vinícius – Boa parte das produções artísticas e literárias do final do século XX e do início do século XXI se dá em suporte digital, o que impacta diretamente nos sistemas de produção, circulação e recepção desses textos e, portanto, nos processos semióticos que engendram sua interpretação. Um dilema hoje da Teoria da Literatura para compreender os regimes de significação desse tipo de texto é que seu ferramental teórico tradicional não dá conta de fenômenos discursivos que emergiram com as tecnologias digitais de informação e comunicação. Como exemplo de objetos que desafiam a crítica e a teoria literária por esses motivos, podemos citar as narrativas multimodais e ficcionais dos videogames; softwares programados por designers-poetas para gerar randomicamente poemas; textos literários produzidos coletivamente em sistemas colaborativos ou em redes sociais. Você acredita que os estudos na área de Engenharia Semiótica, ou na Ciência da Computação, de forma*



*mais ampla, podem ajudar a entender esses processos comunicacionais em que o texto é construído enquanto polissemia, estranhamento, obra aberta, por meio de ferramentas computacionais? Indo um pouco mais além, será que um eventual ramo de desenvolvimento da Engenharia Semiótica poderia ser uma Engenharia Semiótica da cyber-arte?*

Clarisse – Duas das primeiras pessoas a aderirem positivamente à Engenharia Semiótica, no sentido de acolher suas ideias, são Ernest Edmonds (professor da Universidade de Loughborough e da Universidade de Tecnologia de Sidney, que trabalhou na IHC muito tempo, mas hoje se dedica, sobretudo, à arte digital) e Frieder Naker (um dos pioneiros de arte digital, professor da Universidade de Bremen). No mês passado<sup>2</sup>, aqui no Rio de Janeiro, foi inaugurada na Oi Futuro uma exposição de cyber-arte com obras dos dois, entre outros artistas.

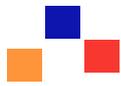
Eu os conheço há muitos anos e me dou muito bem com eles. Foi um grande prazer reencontrá-los e conversamos muito sobre a exposição, sobre arte digital, sobre semiótica... Os dois são grandes fãs da semiótica para essa questão da arte digital. Frieder Naker tem, inclusive, vários textos em alemão sobre essa temática (infelizmente a gente perde um pouco em inglês). E Ernest Edmonds tem alguns textos também sobre semiótica, mas a maior parte do nosso debate se dá 'ao vivo e em cores' quando nos encontramos em algum lugar no mundo. Então, sua pergunta, além de interessante, é oportuna, fazendo tão pouco tempo que estive em contato com eles dois.

Acho que grande parte das questões que eu já levantei durante esta entrevista mostra que a Engenharia Semiótica, a teoria que construímos para dar conta do fenômeno de metacomunicação no contexto da comunicação humana mediada por computador, é muito específica. Nesse ponto, insisto no que eu falei lá no início: a necessidade de adotar uma perspectiva mais modesta e menos arrogante sobre a Engenharia Semiótica, enquanto uma teoria de como caracterizar a metacomunicação (essa mensagem que, por sua vez, é um agente comunicante), ou de como as pessoas constroem a metamensagem (se dispõem de explicitação das estratégias envolvidas).

Tenho a impressão de que a Engenharia Semiótica é um instrumento de aproximação bastante interessante para várias das esferas que você menciona na sua pergunta. Por exemplo, a questão das narrativas multimodais e ficcionais no videogame: a Engenharia Semiótica consegue dar conta de um componente que é difícil de apreender com teorias literárias mais tradicionais, penso eu. Estas fariam sobre a narrativa, sobre o texto, mas não sobre a interatividade; e o 'negócio' da Engenharia Semiótica é justamente a interatividade. Não diria que ela pode dar conta de tudo, mas acho que ela pode compor uma suíte interessante de ferramentas para analisar esses fenômenos que você está levantando, por exemplo, a construção coletiva de enredo em redes sociais. Se esses enredos têm interatividade no processo de criação, a

---

2 Julho de 2015



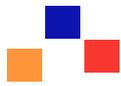
Engenharia Semiótica pode caracterizar as condições dessa interatividade, o que será um trampolim para depois estudar o produto que foi gerado por essa colaboração em rede social. Se a própria ficção que é montada na rede social traz a interatividade embutida no seu objeto, tal como no caso do videogame, ela pode ajudar a entender as questões de interatividade.

Eu diria que pode contribuir, mas jamais pensei em fazer uma Engenharia Semiótica da cyber-arte. Acho que Freider Nake e Ernest Edmonds talvez pudessem responder a essa pergunta, porque eles têm interesse na Engenharia Semiótica um pouco sob esse viés. Eu lembro que, nesta última vez em que nos encontramos aqui no Rio, eles diziam, ao apresentarem algumas das suas obras: “Fiz esta obra aqui [em ambos os casos eram obras visuais] através deste programa de computador”. Nesse momento, alguém da plateia perguntou: “Quando se faz um programa de computador, não se sabem todos os resultados que ele vai dar; então, você se surpreende com o resultado? Melhor dizendo, a surpresa também faz parte da sua criação quando você mesmo se surpreende?” Foi uma pergunta muito esperta, e eles enveredaram pela resposta a esta pessoa dizendo que justamente esse é um dos ingredientes mais especiais da arte digital: a surpresa que o infinito, através da computação, faculta, de modo que o próprio autor tem uma surpresa diante da sua criação. Sobretudo, se você põe uma semente randômica, por exemplo, no algoritmo de geração, como eles fizeram em várias obras. Nesse caso, você tem uma noção de *emergence*, de uma coisa que emerge e que surpreende o próprio autor, de uma natureza bem diferente.

Mas o que eu estava conversado com eles em particular, não durante a conferência, era numa outra linha. Eu comentava que todo programador erra, logo todo programa tem algum *bug*; então, é interessante teorizar sobre uma coisa que pode estar atuando errado.

É interessante pensar que, se eu sei escrever Português e você sabe escrever Português, e se nós dois escrevermos uma peça de ficção, o que a peça está dizendo ela está dizendo. O texto não vai nos trair no sentido de que a leitura que nós estamos fazendo dele está errada. Talvez haja subtextos de que a gente não está consciente, mas isso não vem ao caso; a minha leitura é sempre autorizada. A questão da computação é que eu posso ter dado um falso passo de programação, sobretudo nessas condições de que estou falando, quando se usam programas de terceiros para fazer o seu próprio programa. O programa de terceiro pode ter um erro que se desconhecia, então tem-se a alegação de que certa coisa está funcionando assim por determinada razão (o que é uma maneira de tentar dar conta do que ela significa), quando, na realidade, aquilo é efeito colateral de uma coisa completamente diferente.

Então, a surpresa que se tem não necessariamente é uma surpresa pelo seu ato positivo de criação – por aquilo que se acha que programou. Ela pode ser simplesmente uma surpresa por causa do *bug* de alguma parte do programa. Ernest Edmonds até virou para mim e disse: “Lá vem você com essas suas perguntas difíceis!” Mas nós discutimos que isso é uma coisa interessante porque existe uma idealização na cyber-arte de que o artista que produz a arte virtual é isento de erros; porém, isso não é verdade.



Um programador cometer um erro na computação é muito diferente de um escultor cometer um erro em cima do seu molde, ou de um poeta cometer um erro em cima de sua linguagem. A natureza do erro computacional é muito, muito, muito, muito diferente, justamente porque ela se evidencia num espaço infinito, ao passo que o erro de um escultor ou de um poeta se evidencia no espaço finito. Assim, esse erro está fora do nosso alcance cognitivo.

A relação do artista com a obra, a significação do artista para a obra, começam a mudar muito. Muita gente insiste que essa mudança é por causa do grão de aleatoriedade, e eu insisto que não é só isso. A aleatoriedade é a versão popular e mais bonitinha, porque eu posso dizer que quis colocar uma semente randômica, para fazer a minha obra ou a minha instalação ter um comportamento imprevisível. Sim, só que às vezes acha-se que se fez uma coisa, mas fez-se outra. Aí, aquele comportamento invisível, ou imprevisível, está descolado do seu discurso: não se tem segurança sobre seu discurso de significação. A posse do significado na computação é sempre questionável: esse é um ponto que a Engenharia Semiótica poderia perseguir e ajudar a explorar.

*Vinícius – Clarisse, ouvir você é sempre muito esclarecedor. Suas contribuições nesta entrevista, permeadas pela costumeira modéstia de suas falas e textos escritos, encerram aprendizados fundamentais para a área de Letras e tocam em questões com as quais nos deparamos o tempo todo quando nos pomos a analisar práticas de linguagem, artísticas ou não, no universo computacional. Muito obrigado por compartilhar conosco esses saberes!*

Clarisse – Eu é que agradeço a oportunidade de ter esta conversa, este momento de reflexão, e de aprender com o meu entrevistador.