

APRENDIZADO EM AGENTES ROBÓTICOS BASEADO NA TEORIA DE VYGOTSKY

ALZIRA F. DA SILVA*, LUIZ M. G. GONÇALVES*

* *Universidade Federal do Rio Grande do Norte
DCA-CT-UFRN, 59072-970, Natal, RN, Brasil*

Emails: arizla@dca.ufrn.br , lmarcos@dca.ufrn.br

Abstract— We propose a new learning approach for task execution by Robotics Agents based on Vygotsky theory. In the proposed model, the agents store knowledge in a database and take decisions through carried experiences that are acquired through the interaction with other agents or humans. The process of learning initiates with the search of information to execute a given task. At any development stage, the agent is considered self-capable of performing a task if it has enough knowledge to proceed without any help. Case negative, it analyses all other agents knowledge, by interacting through communication with these, in order to learn other actions that can solve the given task. The actions are then performed until the accomplishment of the current task. The set of actions are stored (learned) and it can thus act alone in the environment at next time the same task is presented. With this simple approach, knowledge has been passed through interactions from humans to robots and from robots to other ones. We formalize the proposed model including mathematical foundations and show practical experiments validating the model, carried out with small robots and children.

Keywords— Learning, Intelligent Agents, Interaction

Resumo— Propomos um novo método de aprendizagem baseado na teoria de Vygotsky para a execução de tarefas por agentes robóticos. No modelo proposto, os agentes guardam o conhecimento em uma base de dados e através das experiências adquiridas pela interação com outros agentes tomam decisões. O processo de aprendizagem inicia-se com a busca de informações para executar uma tarefa. Em todo o estágio do desenvolvimento, o agente é considerado capaz de executar uma tarefa se tiver conhecimento suficiente para prosseguir sem nenhuma ajuda. Em caso negativo, analisa o conhecimento dos outros agentes, interagindo através de um processo de comunicação, a fim de aprender outras ações que podem ajudá-lo a resolver a tarefa dada. As ações serão executadas até que a tarefa seja realizada. O conjunto de ações é armazenado (aprendido) e desta forma, na próxima vez em que a tarefa for apresentada, poderá agir sozinho para executá-la. Com esse método simples, o conhecimento foi construído através das interações de humanos com robôs e destes, uns com outros. Formalizamos matematicamente o modelo proposto e relatamos experiências práticas que validaram o modelo, realizadas com robôs e crianças.

Palavras-chave— Aprendizagem, Agentes Inteligentes, Interações

1 Introdução

Aprendizado é o processo de aquisição de mudanças relativamente permanentes no entendimento, atitudes, conhecimento, informação, habilidade e capacidade, através de experiência. Aprender é um evento cognitivo interno, sendo portanto uma mudança permanente na capacidade de execução de ações, adquirida pela experiência. Essa pode ocorrer usando uma interação aberta com o ambiente externo ou através de processos cognitivos internos.

A aprendizagem social é vista aqui como um processo causado ou favorecido pelos indivíduos que convivem em um mesmo ambiente. Neste sentido, o outro é percebido não somente como um critério para a comparação e o crescimento, mas também como uma fonte neutra de informação, que ajuda ou apressa o processo de aprendizagem. Esse fenômeno é ascendente, definido de maneira gradual, partindo de conceitos elementares (como crianças ensinando robôs, por exemplo).

Vários teóricos têm realizado estudos sobre o processo de aprendizagem nos seres humanos, e algumas teorias têm sido aplicadas para modelar o processo de aprendizagem de agentes intelligen-

tes, principalmente em agentes robóticos.

Neste trabalho, propomos uma metodologia formal para utilização da teoria sócio-interacionista de Lev Semenovitch Vygotsky visando modelar o processo de aprendizagem de agentes inteligentes em um sistema multi-robôs. Apresentamos um formalismo matemático da teoria e a implementação do modelo na prática, em pequenos robôs. A proposta foi então testada na prática em experimentos realizados envolvendo a interação entre crianças e os robôs. Nas seções seguintes, abordaremos a teoria de agentes e sistemas multiagentes, a teoria de Vygotsky e a relação entre essas teorias. Então, será apresentada a estrutura de dados necessária à implementação deste novo modelo e, nas seções finais, mostraremos como esta metodologia está sendo implementada dentro do software *RoboEduc*, com experimentos práticos.

2 Teoria de Vygotsky

Vygotsky elaborou um modelo em psicologia aliado na teoria marxista do funcionamento intelectual humano (Vygotsky, 1993). Segundo o teórico, o desenvolvimento cognitivo ocorre pelo pro-

cesso de internalização das interações sociais. De acordo com Oliveira (Oliveira, 1997), a interação social, seja diretamente com os outros membros da cultura, seja através dos diversos elementos do ambiente culturalmente estruturado, fornece a matéria prima para o desenvolvimento intelectual do indivíduo. Com o aprendizado cultural, mediado pela linguagem, parte de funções cognitivas elementares transformam-se em funções cognitivas superiores (Vygotsky, 1998).

2.1 Processos de Desenvolvimento

A aprendizagem se dá quando ao ocorrer uma situação inesperada, um colapso, o indivíduo procura ajuda, direcionando suas ações para o grupo e as condições sociais externas em busca de apoio. Esse processo se dá de acordo com três conceitos elementares: a internalização, a mediação e o controle (Frawley, 2000).

A internalização refere-se ao processo através do qual sugestões ou conteúdos externos ao indivíduo apresentados por um “outro social” são trazidos para o domínio intrapsicológico (do pensar e do sentir subjetivos), passando a incorporar-se à subjetividade do indivíduo. A mediação é o processo de intervenção de um elemento numa relação, que deixa de ser direta e passa a ser mediada por esse elemento. A internalização e a mediação são os meios para se obter o controle sobre o pensamento e a ação (Frawley, 2000).

O controle possui três características importantes que possibilitam a aquisição de novos conhecimentos, que são o planejamento (antecipar e regular as ações), a inibição (que constitui um filtro cognitivo que limita as opções feitas durante o planejamento) e o local de controle (onde o indivíduo obtém as informações para regular o pensamento). O pensamento superior busca o controle através da mediação e da internalização, podendo ele ser regulado pelo objeto, regulado pelo outro ou regulado pelo self.

Vygotsky (Vygotsky, 1998) identificou dois níveis de desenvolvimento, um real, que determina o que uma pessoa é capaz de fazer por si própria, e um potencial, ou seja, a capacidade de aprender com outra pessoa. A aprendizagem evolui com o desenvolvimento, produzindo abertura nas Zonas de Desenvolvimento Proximal (ZDP), que é a distância entre o nível de desenvolvimento real e o potencial, nas quais as interações sociais são centrais. Essas zonas correspondem às funções que estão em maturação no indivíduo.

3 Agentes Inteligentes

Um agente é definido como um sistema computacional que está situado em um ambiente e é capaz de agir autonomamente para atingir seus objetivos de projeto (Wooldridge, 1993). A autonomia

é uma propriedade que um agente inteligente deve ter para tomar decisões importantes para a conclusão de uma tarefa ou objetivo, sem a necessidade da interferência do ser humano ou qualquer outra entidade. Um agente robótico pode se prevalecter perfeitamente desta definição, incluindo, claro, movimentos físicos nas respostas ao ambiente.

Um sistema multi-agentes é definido como um sistema composto de vários agentes que interagem de forma autônoma, trabalhando juntos para resolver um determinado problema ou objetivo. Os objetivos podem ser comuns a todos os agentes, a um sub-grupo ou a apenas um deles.

Neste contexto, para termos o aprendizado, um agente deve possuir capacidade para executar uma tarefa com maior eficiência do que em execuções anteriores, ou no mínimo igual, quando um limiar de aprendizado da tarefa é atingido. Sem esta capacidade básica, o agente reagirá sempre da mesma maneira para um mesmo ambiente e uma mesma situação. Existem vários métodos de aprendizagem que podem ser aplicados como, por exemplo, métodos indutivos e aprendizado por reforço.

4 Aprendizagem de Agentes e a Teoria de Vygotsky

A idéia central desse artigo é propor o uso da teoria de Vygotsky para implementar agentes cognitivos. Definimos um agente como uma entidade computacional que possui motivos e significados pessoais e que aprende através da interação com outros agentes no mesmo ambiente. O ambiente é um local onde os agentes se comunicam com o objetivo de administrar a diferença entre suas próprias habilidades e a dos seus pares, eventualmente mais capazes. Isso significa que o ambiente é dinâmico, pois o conhecimento está no grupo e surge na proporção que os indivíduos tornam-se socializados. Os agentes podem mudar os motivos, significados, planos, podendo, até, ser substituído por outros agentes. As relações são mediadas por um objetivo e trabalho comum. Com base no ferramental teórico acima descrito, podemos enunciar a nossa proposta.

Seja $Ag_t = \alpha_1, \alpha_2, \dots, \alpha_m$ um conjunto de agentes no tempo t e um ambiente que pode ser definido como um conjunto enumerável de estados: $\Phi = \phi_0, \phi_1, \phi_2, \phi_3, \dots, \phi_k$. Seja $T_t = t_1, t_2, \dots, t_n$, um conjunto finito de tarefas para serem realizadas por um agente α_i . A cada tarefa, associamos um grau da capacidade do agente para realizá-la. Esse grau é uma função $g_{\alpha_i} t_n \rightarrow [0, 1]$. A função g informa o quão perto de realizar a tarefa sozinho o agente i está, sendo monotônica no intervalo $[0, 1]$. A partir desta medida, pode ser concebida como a ZDP de cada agente. É evidente que, se $g(t_n) = 0$, então o agente não poderá realizar a

tarefa mesmo com ajuda de outros agentes, ou seja, ele não tem ainda funcionalidades elementares para poder aprender a efetuar a tarefa. O processo de aprendizagem ocorre da seguinte maneira:

- Dada uma tarefa t_i , o agente recorre a sua base de dados e verifica o grau de capacidade para realizá-la, $g(t_i)$.
- Se $g(t_i) > 0$ De acordo com a informação $g(t_i)$, o agente poderá buscar informação:
 - em si mesmo, realizando a tarefa sozinho ou em conjunto, mas tendo toda a informação necessária para poder realizá-la sem a ajuda de outros;
 - em outro agente.
- Realizar a tarefa e armazenar as informações necessárias que possibilitarão a realização de tarefas similares, diminuindo a distância entre os níveis potencial e real.
- Se $g(t_i) = 0$, observar a comunicação entre os outros agentes, buscando extrair informações redundantes que possibilitarão o aumento do valor da sua ZDP com relação à tarefa t_i .

4.1 Sociedade de Agentes Socio-Interacionistas

Definição 4.1 *Seja Ω uma sociedade de agentes constituída dos seguintes elementos:*

- $Ag_t = \alpha_1, \alpha_2, \dots, \alpha_m$, um conjunto finito de agentes, num dado instante t ;
- $\Phi_t = \phi_0, \phi_1, \phi_2, \dots, \phi_k$ um conjunto enumerável de estados do ambiente no instante t ;
- $T_t = t_1, t_2, \dots, t_n$, um conjunto finito de operações para serem realizadas por cada agente α_i sobre o ambiente.
- $RS = r_1, r_2, \dots, r_s$, um conjunto de regras sociais.

O processo de aprendizagem consiste em, dado um evento causador de desequilíbrio à comunidade, os agentes irão encontrar uma solução obedecendo as regras sociais. Ou seja, se existir na comunidade um agente que tenha a capacidade de resolvê-lo, o evento será entregue a este. Caso contrário, o grupo *tentará*, em conjunto, encontrar a melhor solução. O agente que tem a maior capacidade deverá usar seus conjuntos de operações e ações para tentar resolver a tarefa. Os agentes com graus de capacidade menor que este poderão adicionar ou não as ações executadas ao seu conjunto de ações possíveis, podendo empregá-las em outra situação e, de posse dessas, transformá-las em meras operações.

O desafio está em propor uma heurística para que um dado agente não faça uma mera cópia do

que outro agente realizou ao executar uma tarefa. Uma saída é guardar o evento causador de desequilíbrio e apresentá-lo novamente à comunidade. Pode-se então selecionar outro agente para resolver e assim novas informações serão incorporadas à solução. Outra maneira, seria preparar os agentes para resolver situações previstas em determinados ambientes. Para isso, pode-se construir um algoritmo de ensino-aprendizagem que possibilite a um agente ensinar a outros como se comportar em determinadas situações.

Os agentes são constituídos de um sistema de controle (fator de inibição que possibilita a execução das ações e operações), do nível de desenvolvimento potencial, P , e da ZDP, conforme descrito na seção anterior. O nível Potencial e a ZDP são representados formalmente pelas funções, ρ e ZDP respectivamente, como definidos a seguir.

Definição 4.2 (Função Potencial) . *Seja $Ag_t \subset \Omega$ um conjunto de agentes e $\rho \in T$ uma operação tal que*

$$\begin{aligned} \rho : Ag_t &\rightarrow [0, 1] \\ \rho(\alpha_i)_t &= x \end{aligned} \quad (1)$$

A função Potencial, ρ , fornece o estágio atual das potencialidades do agentes, isto é, o que ele pode fazer com ajuda de outros, agentes ou humanos.

Dado o conjunto Ag_t , obtemos o conjunto P que fornece as potencialidades de cada agente sobre determinado conhecimento (c), no tempo t , isto é, $P(Ag_t)_{c_t} = P_1(\alpha_1), P_2(\alpha_2), \dots, P_m(\alpha_m)$, sendo $P_1(\alpha_1) \leq P_2(\alpha_2) \leq \dots \leq P_m(\alpha_m)$. $P_m(\alpha_m)$ fornece o agente que tem a maior capacidade para executar uma determinada tarefa. A relação ρ é uma relação de ordem, pois é reflexiva, simétrica e transitiva.

É importante ressaltar que:

- i) quando $\rho(\alpha_i) = 0$ isso significa que o agente não consegue resolver o problema, mesmo com a ajuda de outro agente (humano ou não).
- ii) quando $\rho(\alpha_i) \in (0, 1)$ significa que o agente precisa da ajuda de outro agente mais capacitado para resolver o problema.
- iii) quando $\rho(\alpha_i) = 1$ significa que o agente sabe resolver o problema sem a ajuda de outro agente. O valor 1 representa o nível de desenvolvimento *real*.

Definição 4.3 Operação ZDP *Seja $Ag_t \subset \Omega$ um conjunto de agentes, $ZDP \in T$ uma operação tal que:*

$$\begin{aligned} ZDP_c : Ag_t &\rightarrow [0, 1] \\ ZDP_{c_{t+1}}(\alpha_i) &= 1 - \rho_{c_t}(\alpha_i) \end{aligned} \quad (2)$$

A função ZDP fornece o quão perto de realizar a tarefa, sozinho, o agente α_i está.

Observamos que:

- i) Se $ZDP_{ct}(\alpha_i) = 0$ isso significa que o agente consegue realizar a tarefa sozinho;
- ii) Se $ZDP_{ct}(\alpha_i) = 1$ isso significa que o agente não consegue realizar a tarefa sozinho ou com ajuda de outros. Nesse caso, pode-se optar em *criar* uma ZDP para esse agente;
- iii) $ZDP_{ct}(\alpha_1, \alpha_2, \dots, \alpha_m) = (ZDP_{ct}(\alpha_1), ZDP_{ct}(\alpha_2), \dots, ZDP_{ct}(\alpha_m))$.

Definição 4.4 (Função Controle) . Seja $\alpha_i \in Ag_t$ e T o conjunto de tarefas que devem ser aprendidas pelos agentes em Ω , sendo $t_i \in T$ uma tarefa específica. A função controle é definida como:

$$controle : Ag_t \times C \rightarrow Ag_t$$

$$controle(\alpha_i, c) = \begin{cases} \alpha_i & \text{se } ZDP(\alpha_i, c) \geq x \\ \alpha_j & \text{caso contrário} \end{cases} \quad (3)$$

A função controle fornece o identificador de um agente ao qual o agente α_i deve pedir ajuda para executar a tarefa t_i .

5 Implementações

Para validação da teoria proposta, implementamos algumas classes no software *Roboeduc* (Aranibar et al., 2006), desenvolvido inicialmente para o *Projeto de Inclusão Digital com Robôs*, atualmente em andamento na Escola Municipal Professor Ascendino de Almeida, na cidade do Natal, RN. Após essas implementações, foram realizadas oficinas de robótica pedagógica, nas quais foram testadas a operação remota e programação de protótipos, bem como o desenvolvimento e testes das ferramentas criadas para o aprendizado.

O *RoboEduc* dá suporte ao ensino de conceitos de robótica e programação de robôs, assistindo na construção, controle e programação de diversos modelos de protótipos de robôs, e permite o aprendizado de comportamentos usados para os robôs construídos. Encontra-se implementado um banco de dados de protótipos agrupados pelo nível de complexidade na construção dos mesmos e das tarefas que podem ser realizadas para cada robô.

5.1 Implementação do Aprendizado

Para a implementação das ferramentas de aprendizado no software *RoboEduc*, utilizamos a linguagem de programação C++, juntamente com a biblioteca QT4 e a linguagem XML *eXtensible Markup Language* para a construção das interfaces com o usuário e as bibliotecas do BrickOS (SO e compilador) para as interfaces com os protótipos propriamente ditas (comandos, controle remoto,

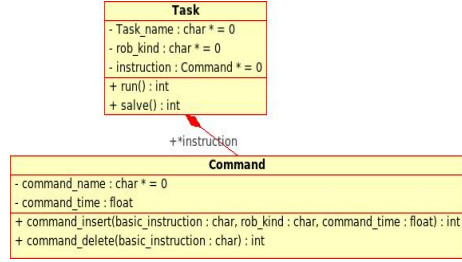
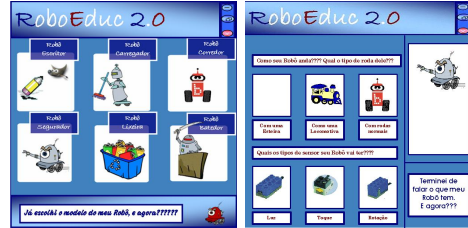
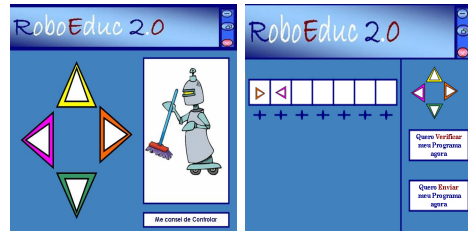


Figura 1: Diagrama da classe Task



(a) Escolha de modelos (b) Escolha de componentes



(c) Controle Remoto (d) Programação Nível

Figura 2: Telas do Software de Robótica Educacional *RoboEduc*

comunicação, etc). O XML é usado para dar ao software desenvolvido a capacidade de expansão automática.

A classe mais importante é a classe *Task*. Esta classe implementa um conceito importante na teoria de Vygotsky que é a representação de uma atividade ou tarefa, como também o que uma criança ensina a um robô. Uma atividade é constituída pelo seu nome, o tipo de robô que a executa e os passos necessários para a sua execução (vide figura 1).

As interfaces com o programador foram construídas utilizando C++ e a biblioteca de elementos visuais QT. A figura 2 mostra algumas das telas da última versão do software construído. Os elementos visuais na figura 2 que representam os modelos, os componentes dos protótipos, os protótipos construídos e as suas funcionalidades são carregados em tempo de execução, levando em consideração a informação contida nos arquivos XML do software.

5.2 Comunicação entre agentes

O protocolo de comunicação implementado é bem simples, baseado em troca de mensagens, sobre

TCP-IP. Cada agente é implementado como um processo que pode (e deve) executar em computadores diferentes. O hardware de comunicação testado consiste de um roteador sem fio (*wireless*) que fornece endereços IPs dinâmicos aos computadores de uma sub-rede, na qual três computadores, cada um controlando um robô, estão conectados. Assim, cada agente robótico é traduzido em um processo que pode enviar e receber mensagens usando esta rede local. Um protocolo em linguagem *C* usando *sockets* faz a parte de comunicação. Basicamente, numa requisição de ajuda, as mensagens são compostas da identificação do agente robótico, identificador da tarefa, e nível potencial para realização da mesma. Uma primeira mensagem de retorno também é composta do identificador do robô, a tarefa e o nível potencial para realização da mesma, apenas dos agentes que possuam nível potencial maior que o do requerente. O requerente, ao receber cada mensagem, verifica qual o robô que possui maior nível potencial, armazenando as informações desse robô. Usando essas informações, o robô requerente solicita então ao vencedor que repasse as informações (identificador e seqüência das ações) referentes à tarefa a ser realizada. A troca do plano (tarefa) é então realizada entre esses dois robôs, sendo interpretado diretamente pelo requerente e suas informações salvas em seu banco de dados. Convém ressaltar que após a realização da tarefa, a função de atualização da ZDP, acima descrita, é efetuada.

6 Experimentos e resultados

O modelo proposto neste trabalho remete a um novo paradigma de aprendizado em agentes robóticos e o assunto robótica também é novo para as crianças. Assim, para resolver este duplo desafio, em cada oficina os alunos construíram o robô utilizando o kit Lego e depois o manipularam a fim de resolver o problema de programação proposto. Nas seis primeiras oficinas os alunos só utilizaram o nível 0 do *RoboEduc*. Depois de familiarizados com o computador, com o software, e com os robôs, começaram a utilizar o nível 1. Cada vez que uma tarefa é concluída, esta é armazenada no banco de tarefas, para posterior utilização. Um conjunto de arquivos simples, em modo texto, cujos identificadores lembram o nome da tarefa proposta, representam este banco.

6.1 Escolha da tarefa e treinamento

Para analisar o processo de aprendizagem entre crianças e robôs foi escolhido a tarefa de viajar entre sete cidades do Rio Grande no Norte: Mossoró, Caicó, Touros, Santa Cruz, Macau, Martins e Natal (representadas por círculos contendo ligações entre eles, em um grande painel no piso). Os alunos calcularam a distância de uma cidade

a outra e também o menor caminho que envolvia todas as cidades. Os alunos montaram em grupo um robô móvel com o kit Lego, com três motores, um para cada par de rodas e para uma caneta demarcatória da trilha seguida. Essa tarefa foi escolhida porque a partir dela começou-se a preparar os alunos para as mudanças necessárias ao uso do controle por si só, visando a programação dos protótipos.

Cada aluno fez o robô navegar pelo mapa, saindo de uma posição inicial, com uma certa orientação (requisitos de cada tarefa). A cada passo, as funções implementadas no *RoboEduc* registraram o comando utilizado (direita, esquerda, frente, ré), por quanto tempo é executado e um peso foi atribuído à performance de cada ação. Estes componentes básicos que definiram um tarefa foram armazenados com um identificador (exemplo: *Viajar_Caico_Mossoro* no próprio nome do arquivo) no computador que controla o protótipo. O peso para cada comando (ação) variou entre 0 e 1, sendo o 0 atribuído a uma péssima ação e 1 a ação média-ótima (o monitor de cada grupo é o crítico da tarefa). Desta forma, ao final, obtivemos um banco de dados de programas (ou planos) para realizar cada tarefa proposta. Note que cada tarefa pode possuir vários planos diferentes.

Depois dessa etapa começamos a treinar 3 robôs com os programas armazenados. Cada robô pode acessar o banco de planos e escolher aleatoriamente um a ser executado. A idéia é que, ao encontrar em um plano um passo cujo peso esteja abaixo de 0.5, um canal de comunicação com outro agente seja estabelecido, à procura de um plano melhor. Se o plano executado por outro agente for melhor, esse procedimento é armazenado e executado. Senão, a comunicação com outro agente é estabelecida ou então o plano corrente é executado, claro, prevendo um resultado ruim. Note que, de qualquer forma, a ZDP de cada robô é modificada. Depois desta fase de treinamento, cada robô possui um plano para as tarefas a ele associadas, com um grau de capacidade de execução. O objetivo a cada passo do treinamento é melhorar a performance de cada robô.

6.2 Testando o Aprendizado

Após o treinamento de cada robô, com seu respectivo conjunto de tarefas, esses são instruídos a realizar uma dada tarefa pelo usuário (por exemplo tarefa a realizar é *Viajar_Caico_Mossoro*). Caso ele não possua esta habilidade, uma requisição é feita aos outros robôs, enviando o identificador da tarefa. No caso, outro robô que possua a tarefa em questão retorna ao primeiro, repassando os comando constantes da tarefa. O primeiro robô executa então a tarefa. Ao final desse tipo de experimento, verificamos que os robôs conseguiram executar a tarefa de modo satisfatório. Al-

gumas vezes, por problemas de disposição inicial dos robôs ou de erros devido ao arrasto, ocorreram falhas da posição final. Para obtenção desse resultado foram levados em consideração a quantidade de comandos e o tempo total gasto para realizar a tarefa. Quando uma tarefa envolve muitos comandos (acima de 8, na média), isso causa imprecisão no resultado final.

7 Trabalhos Correlatos

Nosso trabalho apresenta pontos em comum com alguns trabalhos pesquisados. No trabalho apresentado por Correia e Costa (Correia e da Costa, 2006) destacamos a importância do meio social. Já no discutido por Chu e Hong (Chu e Hong, 2000), também nos preocupamos com a tomada de decisões nos baseando em informações locais, que no nosso caso, são fornecidos pelas as zonas de desenvolvimento proximal de cada agente. Assim como descrito por McLurkin e Yamins (McLurkin e Yamins, 2005), nos preocupamos com a distribuição dinâmica das tarefas.

8 Conclusão

O processo de aprendizagem social foi testado na prática, e os resultados iniciais mostram sua efetividade. O aprendizado do robô a partir das instruções dadas pelas crianças resultou num conjunto de tarefas individuais que cada robô conseguiu executar. Então, eles puderam trocar informações, entre si, na execução de qualquer uma das tarefas aprendidas. A formalização dos agentes e do processo de aprendizagem, bem como de sua arquitetura interna mostra que a utilização de uma teoria para dar suporte à interação em ambientes em que a cooperação é o foco principal é de grande valia para ampliação do uso de sistemas multi-agentes. Isso foi comprovado nos experimentos práticos, tendo os robôs cooperado entre si na execução das tarefas.

Um dos problemas detectados é como um robô deve escolher entre os comandos dados pelas crianças. Uma solução pensada e em fase de desenvolvimento é a utilização de um grafo que contém os comandos básicos e um peso para cada ação. Assim, o robô deve busca o caminho ótimo neste grafo, visando executar uma dada tarefa. Outro problema é que um robô transmite o seu comando para outro, no entanto, não implementamos um sistema em que haja um retorno do agente robótico, visando que o processo não se configure uma mera troca, mas sim um processo de imitação inicial, onde aprendiz e professor colaboram visando uma melhor execução da tarefa em foco. Isso pode influenciar na atualização da ZDP, não implementado ainda atualmente.

Para trabalhos futuros pretendemos desenvolver uma métrica formal (não apenas a dada pelo

crítico ou monitor) que forneça o grau de aprendizagem de cada agente. Cada agente, após obter uma aprendizagem satisfatória, do ponto de um avaliador, poderá ensinar outro agente. Uma nova versão do *RoboEduc* já esta sendo elaborada, com uma estrutura de dados para armazenar cada ação. Um algoritmo de busca nessa estrutura está sendo implementado e testado. Com isso, esperamos solucionar os problemas detectados, podendo então utilizar o processo de aprendizagem em problemas mais complexos. Assim, futuramente, pretendemos formalizar melhor o ambiente, outros agentes robóticos (com outras habilidades), melhorar o processo de comunicação e a maneira como os aspectos sociais interferem no processo de aprendizagem.

Referências

- Aranibar, D. B., Gurguel, V., Santos, M., Araujo, G. R., Roza, V. C., Nascimento, R. A., da Silva, A. F., Silva, A. e Nascimento, L. M. G. (2006). Roboeduc: A software for teaching robotics to technological excluded children using lego prototypes, *3rd IEEE Latin American Robotics Symposium*, Santiago, Chile.
- Chu, H.-T. e Hong, B.-R. (2000). Cooperative behavior acquisition in multi robots environment by reinforcement learning basead on action selection level, *proceedings of the 2000 IEEE/RSJ*, pp. 1397–1402.
- Correia, L. C. e da Costa, A. (2006). Socialbots: Um mecanismo de aprendizado multiagente para sistemas multi-robôs, *XXVI Congresso da Sociedade Brasileira de Computação - Encontro de Robótica Inteligente (EnR)* 1.
- Frawley, W. (2000). *Vygotsky e a Ciência Cognitiva: linguagem e integração das mentes social e computacional*, Artmed.
- McLurkin, J. e Yamins, D. (2005). Dynamic task assignment in robot swarms, *Robotics: Science and Systems Conference*.
- Oliveira, M. K. (1997). *Vygotsky: aprendizado e desenvolvimento um processo sócio-histórico*, Scipione.
- Vygotsky, L. S. (1993). *Pensamento e Linguagem*, Martins Fontes.
- Vygotsky, L. S. (1998). *A Formação Social da Mente*.
- Wooldridge, M. (1993). *An Introduction to Multiagent System*, John Wiley e Sons Ltd.