

UCL : Uma Linguagem de Comunicação para Agentes de Software baseada em Ontologias

Carlos Alberto Estombelo-Montesco¹, Dilvan de Abreu Moreira¹

¹ Instituto de Ciências Matemáticas e de Computação - SCE-ICMC-USP Cx. Postal 668, Av. Trabalhador São-Carlense, 400 – Centro.
São Carlos-SP - Brasil CEP 13560-970
estombelo@universia.edu.pe, dilvan@icmc.usp.br

Abstract. Language is very important to humans. If software agents are going to communicate with them, they should be able to do it using natural language. However, dealing with human language is very difficult. To solve this problem we propose a simplified form of a natural language, called Universal Communication Language (UCL). UCL can fulfill the role of an ACL (Agents Communication Language) and, at the same time, be convertible to and from a natural language. UCL design is concerned with the description of message structures, their underlining semantic context and the support for protocols for agent interaction. The key point about UCL is that the language can be used not only for communication among software agents but with humans too. This is possible because UCL is derived from the Universal Network Language (UNL), a language created to allow communication among people using different languages. UCL was defined using the Extended Markup Language (XML) to make it easier to integrate into the Internet. In addition, an enconverter-deconverter software prototype was written to serve as a tool for testing and experimenting with the language specification.

1. Introdução.

A tecnologia de agentes de software desperta interesse como ferramenta para criação de um novo modelo para sistemas de software complexos. Nestes sistemas, existe a cooperação entre agentes para realizar tarefas com maior eficácia. Os agentes que operam nestes sistemas têm que comunicar dados, informação, conhecimento, etc. e desta forma permitir interação (Dignum, 2000). Para que a comunicação entre os agentes seja possível é necessário utilizar uma Linguagem de Comunicação entre Agentes (*Agent Communication Language* - ACL), que expresse a forma como as mensagens são comunicadas de um agente para outro. No contexto deste trabalho, o atributo mais importante é a capacidade social do agente a ser explorada.

O objetivo deste trabalho é a especificação de uma nova ACL, chamada UCL – *Universal Communication Language* (Montesco & Moreira, 2001), que se preocupa com a descrição da estrutura das mensagens, com seu modelo semântico e com o suporte a protocolos da Internet para interação entre agentes (de software ou humanos).

No contexto da ciência da computação, um agente de software pode ser utilizado para facilitar a criação de softwares capazes de interoperar, ou seja, trocar informações e serviços com outros programas e, dessa forma, resolver problemas complexos (Moreira & Walczowski, 1997).

2. Ontologias para Agentes e outros recursos para UCL

‘Ontologia’ é um termo usado para se referir ao senso comum de algum domínio de interesse. Uma ontologia necessariamente vincula ou inclui algum tipo de “visão geral”

referente a um domínio. Esta “visão geral” é frequentemente concebida como um conjunto de conceitos (entidades, atributos, processos), suas definições e suas inter-relações, o que é chamado de conceitualização (*conceptualisation*).

Enumeram-se brevemente tecnologias que podem ser aplicadas na comunicação entre agentes de software, especificamente para auxiliar a descrever a implementação da linguagem *Universal Communication Language* – UCL, que será vista com maior detalhe nas seguintes seções. Estas tecnologias são: a linguagem *Universal Networking Language* (UNL) (Uchida, 1999), a Linguagem XML (metalinguagem) (Connolly, 2000), a ferramenta de processamento de linguagem natural (ferramenta de código aberto), desenvolvida por Erik T. Mueller.(1998), Neste último item a ferramenta TT é formada por uma base de dados de 25,000 conceitos organizados hierarquicamente.

3. A LINGUAGEM UCL PARA A COMUNICAÇÃO ENTRE AGENTES

3.1 Metodologia utilizada

O desenvolvimento da linguagem foi dividido em duas fases (Montesco & Moreira, 2001): definição e implementação.

- a) Definição da linguagem: Nesta fase procurou-se especificar a linguagem dando ênfase aos aspectos conceituais de como representar o significado das mensagens. Para esta representação, utilizou-se a base teórica da linguagem *Universal Networking Language* (UNL).
- b) Implementação do Protótipo: Utilizando a especificação inicial da linguagem que se baseia nas características e critérios mostradas anteriormente, passou-se a preencher os aspectos gramaticais necessários para a implementação da linguagem. Para isso foi necessário:

3.2 Características da linguagem UCL

A linguagem UCL representa uma sentença (mensagem) por meio de uma estrutura sintática e um conjunto de conceitos, relações e atributos, que são (Montesco & Moreira, 2001):

- UW (*Universal Word*), que representa o significado de uma palavra ou conceito.
- Rótulo de relação, que representa uma relação entre *Universal Words*.
- Rótulo de atributo, que contém alguma informação adicional ou definição que se acrescenta à *Universal Word* e que está presente na mensagem.

4. Especificação da linguagem

4.1 UW (Universal Word)

A *Universal Word* é a unidade mínima de representação de um conceito que, em conjunto, denotam um significado específico em uma mensagem. Sua representação pode ser feita por palavras em inglês ou de qualquer outra linguagem, ou mesmo por símbolos, sendo que o mais importante é o significado ou conceito que ela envolve.

Na **Figura 1**, pode se observar que cada UW definida deve ter um identificador *id*. Este identificador é formado por uma cadeia de caracteres alfanuméricos e serve como único identificador do conceito em toda a mensagem. Este identificador pode ser referenciado em qualquer outra parte da mensagem, mas não pode ser redefinido.

O rótulo *head* corresponde ao lugar onde o conceito é definido. Os conceitos utilizados sempre estão relacionados à ontologia em uso, e é neste ponto que a linguagem UCL se conecta com a ontologia de um domínio específico de conhecimento.

```
<uw id="uw02" head="area">
  <icl direction="to">
    <uw head="place"/>
  </icl>
  <reference attribute="indef"/>
</uw>
<uw id="uw03" head="strategic"/>
<uw id="uw04" head="designate">
  <icl direction="to">
    <uw head="do"/>
  </icl>
  <focus attribute="entry"/>
  <viewpoint attribute="may"/>
</uw>
<uw id="uw05" head="read">
  <icl direction="to">
    <uw head="do"/>
  </icl>
</uw>
<uw id="uw06" head="home"/>
```

Figura 1 Definição de conceitos em um documento XML

Neste exemplo, pode-se observar a definição de cinco conceitos, cada um rotulado como UW. Cada conceito tem um identificador único incluído como um atributo (*id*), e cada conceito está incluído com o atributo *head*.

4.2 Rótulos de relação

A linguagem UCL representa mensagens que possuem um determinado significado envolvendo conceitos. Esta composição de conceitos é representada por um conjunto de relações binárias que permitem distinguir as diferentes relações envolvendo os conceitos.

4.3 Rótulos de atributo

Os rótulos de atributos foram introduzidos para limitar o significado dos conceitos. Com os rótulos de atributos pode-se particularizar o significado de uma UW. As representações destes atributos estão agrupadas de forma tal que se possa identificar as informações de tempo verbal, o aspecto do conceito ou estrutura da mensagem, entre outros. Estes rótulos de atributo podem ser inseridos dentro da abrangência do rótulo UW para limitar o significado.

5. Implementação do protótipo do interpretador UCL

Na implementação da linguagem, utilizou-se a meta-linguagem XML para definir as mensagens. A estrutura destas mensagens está baseada em um documento DTD. Utilizou-se o interpretador de documentos XML - *Java API for XML Processing* (JAXP) Version 1.1 da Sun. Como a representação semântica da linguagem requer uma ontologia que represente os conceitos envolvidos na mensagem, adotou-se a ontologia do software *ThoughtTreasure*. Além disso, se implementou um protótipo que permitiu gerar e manipular as mensagens UCL considerando a utilização dos recursos mostrados anteriormente.

6. CONCLUSÕES

Durante a definição da linguagem *Universal Communication Language* (UCL) procurou-se abranger todos os conceitos teóricos da linguagem *Universal Networking Language* (UNL), no intuito de preservar o poder representativo de sentenças desta linguagem.

Os vários requisitos de uma linguagem para comunicação entre agentes, foram unificados por meio dos recursos para a comunicação entre agentes na internet definidos anteriormente. Sendo esta linguagem definida para ser utilizada na comunicação entre agentes, foi importante ressaltar como a mensagem iria representar conceitos do mundo real, e neste caso optou-se por utilizar ontologias como meio de representação. Dentre as bases de dados ontológicas analisadas (WordNet, Cyc, *ThoughtTreasure*, dentre outras), a do software *ThoughtTreasure* foi a mais adequada aos requisitos de disponibilidade, de não ser proprietária e de oferecer uma interface de acesso à base de dados. Além disso, a base ontológica do *ThoughtTreasure* contém, juntamente com os conceitos definidos dentro dela, várias características e atributos para representar linguagem natural que pode ser processada através da sua API TT.

6.1. Contribuições

Este trabalho contribui com a especificação da linguagem UCL, dando ênfase à importância da semântica da mensagem a ser representada com esta linguagem. A UCL facilita a representação de sentenças em linguagem natural utilizando-se de ontologias, reduzindo, desta forma, confusões conceituais e terminológicas que podem existir na construção da sentença original em linguagem natural. Além disso, a utilização de ontologias na linguagem auxilia na interoperabilidade entre diferentes agentes que trocam mensagens. O uso do padrão XML, na implementação, contribuiu para que a linguagem UCL seja facilmente integrável com a Internet e com outras aplicações também baseadas neste padrão. A implementação do codificador da linguagem UCL serviu como subsídio para experimentação e também testar a viabilidade da proposta apresentada. Esta implementação foi desenvolvida como um interpretador de sentenças (mensagens), utilizando a linguagem Java e o programa TT e está disponível como open software (licença GNU).

BIBLIOGRAFIA

- (Connolly, 2000), Connolly, D. *Extensible Markup Language (XML)*. Fevereiro 2000. Disponível on-line: <http://www.w3.org/XML/>
- (Dignum, 2000), Dignum, Frank; Greaves, Mark. (ed.). *Issues in agent communication*. – (Lecture notes in computer science; Vol 1916: Lecture notes in artificial intelligence) Berlin; Heidelberg; New York; Barcelona; Hong Kong; London; Milan; Paris; Singapore; Tokyo: Springer, 2000.
- (Montesco & Moreira, 2001) Montesco C.E.; Moreira D.A. *Un Lenguaje de Comunicación para Agentes en Internet, Basado en Ontologías*, V Congreso Internacional Sudamericano de Ingeniería de Sistemas e Informática, Arequipa Peru, outubro 2001, CDROM 15 pp.
- (Montesco & Moreira, 2002) Montesco C.E.; Moreira D.A. *UCL - Universal Communication Language*, Proc. of the First International Workshop on UNL, other Interlinguas and their Applications, Las Palmas Spain, June 2002, pp. 33-37.
- (Moreira & Walczowski, 1997), Moreira, D. A.; Walczowski, L. T. *Using Software Agents to Generate VLSI Layouts*. IEEE Expert Intelligent Systems, p.26-32. Nov/Dez. de 1997.
- (Mueller, 1998), Mueller, Erik T. *Natural Language processing with ThughtTreasure*. New York: Signiform. Disponível on-line: <http://www.signiform.com/tt/book/>
- (Uchida, 1999), Uchida, H.; Zhu Meiyang; Della Senta, T. (1999). *Universal Networking Language: a gift for amillenium*. Tokyo: United Nations University. Disponível on-line: <http://www.unl.ias.unu.edu/publications/index.html>.