

Una Taxonomía de Cambios para el Grafo de Espacios de Nombres Contextuales para Sistemas Multi-Agente en Contextos Múltiples

Sergio A. Gómez

Facultad de Informática

Universidad Nacional de La Plata

50 y 115

1900 - La Plata

ARGENTINA

E-mail: sergiog@ada.info.unlp.edu.ar

Abstract

Los agentes pueden definirse como componentes de software (programas) activos que perciben, razonan y se comunican a través de un protocolo predefinido. Muy generalmente, como en el caso de la World Wide Web, los agentes no son desarrollados en aislamiento sino como parte de un sistema multi-agente; en este caso, un agente invocará servicios de otros agentes y, a su vez, servirá a otros agentes que requieran sus servicios.

Para la interoperabilidad, los agentes deben ser capaces de comunicarse con otros agentes. Surge entonces la necesidad de una lingua franca, una solución es el formato de intercambio de información (Knowledge Interchange Format, KIF). El compartir conocimiento en un grupo de agentes demanda un acuerdo acerca del significado de los símbolos. Dos problemas pueden surgir, incompatibilidad, cuando símbolos diferentes significan lo mismo, e inconsistencia, cuando los símbolos iguales significan cosas diferentes. Una solución a dichos problemas, el grafo de espacios de nombres de contextos para sistemas multi-agentes en contextos múltiples (NSCG), puede encontrarse en la literatura de agentes.

En este artículo, propongo una taxonomía de cambios para el NSCG; también especifico un modelo para la semántica de los cambios de contextos basado en un conjunto de invariantes y un conjunto de reglas. Finalmente, comparo mi trabajo con los trabajos anteriores.

1. Introducción

Debido a la proliferación de las computadoras y las redes, los deseos de interconexión y la necesidad de accesibilidad a los datos en todo momento y lugar, los ambientes de información modernos se han vuelto grandes, abiertos y heterogéneos [Huhns 97]. Dichos ambientes están compuestos principalmente por componentes distribuidos y de legado. Se dice que son abiertos porque dichos componentes pueden agregarse y quitarse dinámicamente. Enfoques recientes introducen a los agentes de software para lidiar con las características recién mencionadas [Aarsten 96, Cheong 96, Cutkosky 97, Shoham 97].

Los agentes pueden definirse como programas que interactúan enviándose imperativas (*performatives*) en KQML con contenido en KIF [Finin 91, Finin 94, Genesereth 92, Patil 97]. KIF es un lenguaje orientado a las computadoras para el intercambio de información entre programas de computadora dispares llamados agentes. KIF tiene semántica declarativa, es decir, el significado de las expresiones en la representación puede entenderse sin necesidad de apelar a un intérprete para manipular dichas expresiones. KIF es lógicamente comprensivo, es decir, permite la expresión de fórmulas arbitrarias en el cálculo de primer orden; permite además la representación de metaconocimiento, es decir, conocimiento acerca del conocimiento. También, permite la representación de reglas de deducción no monótonas, y permite la definición de objetos, funciones y relaciones.

Además, KIF incluye un sublenguaje para declarar axiomas de categoría y definición para constantes (por ejemplo, "Los triángulos son polígonos de 3 lados."). Los axiomas definitorios son verdaderos *por definición* en una manera que los distingue de las fórmulas que expresan propiedades

contingentes del mundo. Las definiciones en KIF pueden ser *completas*, especificando que una expresión es equivalente a una constante, o *parciales*, especificando que un axioma que restringe las posibles denotaciones de una constante.

2. Contextos y el NSCG

En inteligencia artificial, los contextos han sido propuestos como un medio de definir la verdad de una sentencia relativa a una situación. Muchos conceptos que usa un agente están limitados a un contexto particular, dicho contexto puede involucrar muchas suposiciones que son relevantes a la verdad de una proposición hecha en un contexto particular. Son de interés los trabajos de [McCarthy 96, Giunchiglia 93a, Giunchiglia 93b].

Los agentes son programas que interoperan usando un lenguaje de comunicación. Esta visión asume que “... *los agentes coordinarán sus actividades basados en el contenido de sus comunicaciones. Este uso de un lenguaje de comunicación asume que algún agente receptor será capaz de entender el significado de la expresión independientemente del emisor (semántica independiente de los agentes). Aquí aparece el problema de los espacios de nombres: la semántica independiente de los agentes requiere que todos los agentes acuerden un espacio común de nombres. Esta es una hipótesis poco realista cuando se consideran las interacciones potenciales a la escala de Internet. Una hipótesis más realista es que dentro de un contexto dado, los agentes van a estar de acuerdo sobre el significado de un símbolo. Así, el significado de un símbolo estará ligado al contexto en que se lo comunica. Un contexto de espacios de nombres permite a los agentes, a la escala de la Internet, particionar el espacio de nombres global en espacios de nombres más pequeños y manejables. Además, las comunidades de agentes pueden evolucionar dinámicamente mediante la creación de nuevos contextos a partir de viejos contextos usando el grafo contextual de espacios de nombres (NSCG)*” como se explica en [Singh 95].

De esta manera, un nuevo contexto C , llamado un *subcontexto*, puede “colgarse” de otro contexto S , llamado *supercontexto* de C , haciendo que los símbolos definidos en S lo estén también en C . Así, el conjunto de contextos estará relacionado en una manera ancestro-descendiente, la cual se asemeja a la relación de herencia en los lenguajes de programación orientados a objetos.

Entonces, como el NSCG está pensado para soportar la comunicación de información entre agentes en contextos diferentes, recordemos algunas definiciones útiles. Un símbolo local a un contexto es un símbolo descontextualizado, por ejemplo, *apple*. Un símbolo contextualizado indica su contexto explícitamente, y es de la forma *símbolo@contexto*. Un símbolo contextualizado puede referirse sin ambigüedad en cualquier contexto. Un símbolo descontextualizado puede mapearse en un símbolo contextualizado concatenando el contexto al símbolo. Por ejemplo, el símbolo *apple* en el contexto *computers* será *apple@computers* mientras que *apple* en el contexto *fruit* será *apple@fruit*. Para más detalles, referirse a [Singh 95].

Además, según [Singh 95], la presencia de un símbolo en un contexto, demanda un acuerdo en su significado por parte de todos los agentes que lo utilizan; por lo tanto, el NSCG no hace un manejo explícito de los significados de los símbolos. Por esto, cuando símbolos diferentes refieren al mismo concepto, el conocimiento de este hecho es obligación de los agentes intervinientes en la comunicación.

La sintaxis de KIF extendida con información de contexto puede observarse en los ejemplos siguientes, como se puede encontrar en [Singh 95]:

```
(<@kif (size@reebok shoe@reebok) (9@kif))
```

que establece que el tamaño del zapato es menor a 9 (con respecto a sus contextos relativos). Y,

```
(< (size shoe1) (size@LAGear shoe2@LAGear))
```

que establece que el tamaño del primer zapato es menor al tamaño del segundo zapato, mientras que menor (<), el (primer) tamaño y el primer zapato se toman relativos al contexto corriente y el (segundo) tamaño y el segundo zapato están explícitamente contextualizados.

En la Figura 1 se muestra un pequeño grafo de contextos donde se pueden ver los contextos

