

## **De Redes Conceptuales a Ontologías, una Experiencia Aplicada a la Enseñanza de la Programación**

Aldo Algorry, Carlos Alberto Bartó, Laura Cecilia Díaz

[aalgorry@gmail.com](mailto:aalgorry@gmail.com), [cbarto@gmail.com](mailto:cbarto@gmail.com), [lcd\\_ic@yahoo.com.ar](mailto:lcd_ic@yahoo.com.ar)

Laboratorio de Educación Virtual, Departamento de Computación

F. C. E. F. y N. - U. N. C.

### **Resumen**

Este trabajo se inserta en la estrategia general del grupo de investigación del LEV-FCEFyN orientado a la implementación de Sistemas Tutores Inteligentes para ayudar en el proceso de aprendizaje de los estudiantes de la materia inicial de programación para carreras de Ingeniería, en el lenguaje Python.

Uno de esos sistemas es ASPIRE que requiere de la definición de la ontología del cuerpo de conocimiento a ser transmitido.

La formulación de ontologías constituye la herramienta fundamental para la descripción sistemática del conocimiento susceptible de ser tomado e interpretado por una herramienta automatizada como ASPIRE.

Otra forma de describir conocimiento es mediante mapas conceptuales que son representaciones gráficas con una estructura jerárquica.

Como existe una estrecha relación entre mapas conceptuales y ontologías, es posible la generación automática de ontologías a partir de un mapa conceptual aunque el mapa conceptual tiene algunas características que le dan un mayor nivel de complejidad que las ontologías utilizadas por ASPIRE.

Aquí se presenta una experiencia donde a partir del desarrollo de las Redes Conceptuales diseñadas con CMapTools sobre unidades de la materia se generaron las ontologías para ser utilizadas en la herramienta ASPIRE.

Para alcanzar este objetivo se detectaron los elementos comunes entre ambas formas de modelar el conocimiento y sus modos de representación, a partir de lo cual se fijaron reglas sintácticas adicionales en la formulación de los mapas conceptuales en CMapTools a fin de identificar unívocamente los elementos del mapa y asignarlos correctamente a los correspondientes componentes de las ontologías en ASPIRE. Luego se desarrolló la herramienta propia que realiza la generación automática de las ontologías.

Se describe la experiencia de construir la herramienta, la sintaxis, los conceptos que la fundamentan y los resultados de la misma.

**Palabras clave:** Ontologías, Sistemas Tutores Inteligentes, Mapas conceptuales, ASPIRE

## **Introducción**

Entendemos por ontología al conjunto de conceptos, sus propiedades y relaciones que constituyen un determinado dominio de conocimiento.

Las ontologías representan una forma de encapsular conocimiento para su posterior implementación y reuso, permitiendo describir el conocimiento propio de un dominio.

La formulación de ontologías constituye la herramienta fundamental para el desarrollo de la Web semántica por su capacidad de definir dominios específicos, otra de sus aplicaciones es la de contar con una descripción sistemática del conocimiento susceptible de ser tomado e interpretado por una herramienta automatizada, como ASPIRE, para presentar esos conocimientos a un estudiante guiándolo en su proceso de aprendizaje mediante la indicación de los errores detectados en la solución de problemas, por la violación de las reglas de restricciones generadas. Esta última aplicación es la de interés para el presente proyecto.

Los mapas conceptuales, otro modo de describir conocimiento, son representaciones gráficas para organizar y mostrar conceptos usualmente encerrados en círculos o cajas del mismo tipo, y relaciones entre conceptos indicados por una línea uniendo dos conceptos.

Texto sobre la línea, significa palabras o frases conectoras especificando relaciones entre dos conceptos. Por otro lado los mapas conceptuales permiten establecer una estructura jerárquica.

Como vemos existe una estrecha relación entre mapas conceptuales y ontologías por lo que es posible la generación automática de ontologías a partir de un mapa conceptual.

El mapa conceptual tiene algunas características que le dan un mayor nivel de complejidad que las ontologías utilizadas por ASPIRE, como ser la capacidad de “navegar” entre conceptos para formar proposiciones.

El equipo de investigación tiene como objetivo el desarrollo de tutores en diferentes herramientas a fin de evaluarlas, por lo que se justifica la realización de las ontologías en una única herramienta, en este caso CMapsTools para luego exportarla a otra.

## **Desarrollo**

Para generar la herramienta de transformación desde los mapas conceptuales de Cmap a las ontologías que utiliza ASPIRE lo primero que se realizó fue un estudio a nivel de metamodelo de la capacidad sintáctica de la herramienta ASPIRE para modelar las ontologías, luego se analizó la capacidad expresiva de los mapas conceptuales generados por CMapTools y se establecieron las reglas sintácticas a ser usadas en los mapas conceptuales y que permiten a la herramienta de traducción identificarlos correctamente para su implementación en ASPIRE.

## **ASPIRE, una herramienta para construir tutores inteligentes**

ASPIRE es un ambiente para la creación e implementación de Sistemas de Tutores Inteligentes orientado a los docentes para su uso en sus propios cursos con el objeto de mejorar el proceso de aprendizaje.

ASPIRE específicamente es un sistema de tutoría basado en restricciones semánticas y sintácticas

El proceso de creación de un tutor basado en restricciones es un proceso semi-automático en ASPIRE. Este proceso consta de las siguientes etapas

1. Modelar la estructura del dominio;
2. Componer la ontología de dominio;
3. Modelar las estructuras del problema y soluciones;
4. Diseñar la interfaz del alumno;
5. Agregar problemas y soluciones;
6. Generar restricciones sintácticas;
7. Generar restricciones semánticas;
8. Implementar el dominio.

En el segundo paso el autor desarrolla una ontología sobre el dominio instruccional, la que juega un papel fundamental en el proceso de generación de las restricciones que son la base del conocimiento de ASPIRE.

Una ontología describe la estructura de un dominio, mostrando sus conceptos básicos,

sus propiedades y las relaciones entre los conceptos. Una definición ampliamente aceptada es que una ontología es una especificación de una formalización (Gruber, 1993), en otras palabras, es una especificación explícita, formal del vocabulario de dominio que presenta una comprensión común de los temas que se pueden comunicar entre usuarios y aplicaciones.

Una ontología permite de este modo a todas las personas involucradas hablar el mismo idioma.

Las ontologías juegan un papel crucial en ASPIRE. Como el objetivo de ASPIRE es hacer posible que los profesores desarrollen sus tutores para sus cursos, el proceso de creación con el apoyo de ASPIRE se basa en la ontología del dominio de conocimiento. En lugar de pedirle al docente que codifique manualmente el dominio mediante un lenguaje de representación del conocimiento específico, el autor debería describir el dominio de instrucción especificando la ontología del mismo. Esto es un requisito mucho más simple, ya que el autor no tiene que aprender el lenguaje de representación del conocimiento y los detalles de un enfoque particular a la utilización de modelos de dominio. Además de especificar la ontología de dominio, el autor proporciona ejemplos de los problemas y sus soluciones y sus relaciones con los conceptos definidos en la ontología. ASPIRE-author analiza las tres fuentes de conocimiento (ontología, problemas y soluciones), e induce el modelo de dominio representándolo en términos de un conjunto de restricciones.

### **ASPIRE y su capacidad expresiva para modelar ontologías.**

Para la construcción de la ontología del dominio, ASPIRE cuenta con una interfaz online con un editor gráfico que permite la generación de grafos jerárquicos donde los conceptos, simbolizados como rectángulos con etiqueta, se vinculan entre sí mediante líneas de conexión que indican relaciones del tipo “es un” siendo este el único de tipo de relación gráfica soportado por la herramienta, constituyendo una taxonomía.

Un concepto puede ser definido como “abstracto” o sea que no puede ser instanciado en un problema o solución sino a través de sus sub-conceptos, representando generalizaciones.

Para definir otros tipos de relaciones o propiedades la herramienta permite que a un concepto se le agreguen “slots” que pueden ser propiedades de un tipo predefinido (Any, Boolean, Float, Integer, String, Symbol) o relaciones (Relationship) con otros conceptos presentes en la ontología. Los slots tienen un nombre que es un descriptor de la relación y

permiten definir la cardinalidad de las relaciones; una cardinalidad se puede marcar como “opcional” equivalente a que permite valor nulo o “múltiple” que indica que puede haber mas de uno, también se puede indicar una cardinalidad específica.

Por otro lado en los casos de las propiedades se pueden establecer valores máximo y mínimo que puede tomar el valor de la propiedad.

Un tipo especial de propiedad es el Symbol, donde el autor debe especificar la lista de valores que puede tomar la propiedad.

Las relaciones con otros conceptos puede ser establecidas con un o con una lista de conceptos, este último caso se presenta cuando se quiere modelar una relación donde una instancia de un concepto puede asociarse a una instancia cuyo concepto esta incluido en una lista de conceptos, aqui se marca la relación como “list” y se indican cuales son los conceptos de los que puede ser la instancia de la relación. Para mayor detalle ver el manual de ASPIRE [6].

## **CMapTools y los mapas conceptuales**

CMapTools es un poderosa herramienta para modelar gráficamente ontologías que permite realizar mapas conceptuales completos que en definitiva se pueden ver como ontologías. Los mapas conceptuales desarrollados en CMapTools se los llama Cmaps.

Como dijimos, la idea de este trabajo es la de poder generar reutilizar mapas conceptuales desarrollados en CMapTools para generar ontologías de ASPIRE.

Los mapas conceptuales se crean en un ambiente gráfico donde los conceptos de indican como rectángulos y se vinculan a través de líneas que indican las diferentes relaciones entre ellos, la relaciones se nominan y pueden tener información asociada y complementaria.

A diferencia de ASPIRE que no permite repetir nombres de los conceptos y de sus slots, CMapTools no tiene restricciones de ese tipo permitiendo generar nombres repetidos ya que mantiene identificadores únicos internamente.

## **Reglas sintácticas en los Cmaps para poder correlacionarlos con las ontologías de ASPIRE**

Para poder cumplimentar con el objetivo de portar Cmaps a ontologías de ASPIRE se debieron generar reglas sintácticas en los Cmaps que permitieran identificar las diferentes

características de los conceptos, propiedades y relaciones según los elementos establecidos por ASPIRE.

Se analizaron varias alternativas para fijar las reglas sintácticas adicionales, como el uso de diferentes colores, tipos de líneas o el uso de estilos prefijados para representar las diferentes posibilidades y sus combinaciones permitidas por ASPIRE.

Al final se decidió utilizar la capacidad de CMapTools de adjuntar a las relaciones entre conceptos información adicional, en este caso texto.

Para ello se utiliza una especificación mediante el uso de tags y atributos que la herramienta de traducción interpreta para generar los slots de ASPIRE.

Se prefijaron conceptos “reservados” que representan los tipos definidos por ASPIRE para las propiedades: Any, Boolean, Float, Integer, String y Symbol. De modo que una conexión que llega a un concepto etiquetado con alguno de estos nombres reservados será interpretado por la herramienta de traducción como un slot de propiedad en ASPIRE.

Referencia sintáctica:

**Tag: concept**

Indica un concepto, esta tag es opcional, solo se usa cuando se quiere marcar un concepto como abstracto.

Attributes:

Nombre	TipoEsquemaXML	Requerido	Descripción
abstract	string	No	“Yes” si es abstracto, “No” si no lo es

Ej: `<concept abstract="Yes" />`

**Tag: is\_a**

Indica conexión de tipo jerárquica “es un” que se representa gráficamente en ASPIRE, se aplica en las cajas de los conceptos.

Attributes:

No posee

Ej: `<is_a/>`

**Tag: relationship**

Se usa para indicar slots de relaciones con otros conceptos, se aplica en las cajas

conectoras de los Cmaps.

Attributes:

Nombre	TipoEsquemaXML	Requerido	Descripción
optional	string	No	“No” si es mandatorio, “Yes” si no lo es
multiple	string	No	“No” si es unico, “Yes” si pueden ser varios valores
at_least	string	No	Cardinalidad mínima
at_most	string	No	Cardinalidad maxima
free_text	string	No	“Yes” si es texto libre, “No” si no lo es

Ej: `<relationship multiple="Yes" at_least="2"/>` (indica un relación entre conceptos que debe tener dos o más instancias de concepto asociadas)

**Tag: property**

Se usa para indicar slots de propiedades en conexiones con conceptos con nombres reservados ( Any, Boolean, Integer, Float o Symbol), se aplica en las cajas conectoras de los Cmaps.

Attributes:

Nombre	TipoEsquemaXML	Requerido	Descripción
optional	string	No	“No” si es mandatorio, “Yes” si no lo es
multiple	string	No	“No” si es unico, “Yes” si pueden ser varios valores
at_least	string	No	Cardinalidad mínima
at_most	string	No	Cardinalidad maxima
unique	string	No	Valor por defecto
min	string	No	Valor mínimo, solo para conexiones con Integer, Float o String
max	string	No	Valor maximo, solo para conexiones con Integer, Float o String
list	string	No	Lista de simbolos separados por caracter pipe ( ), solo para conexiones con Symbol

Ej: `<property optional="Yes" at_most="1"/>` (indica una propiedad de un concepto que puede tomar o no un valor y no mas de un valor)

Estas tags y sus atributos deben ser agregadas usando el botón secundario del mouse sobre el nombre de la relación y seleccionando “Agregar información...”, en la seccion “Info visible”, siguiendo las reglas de XML.

En caso de tratarse de la relación de tipo “relationship” con multiples conceptos asociados (“list”) prevista en los slots en ASPIRE, las conexiones de los elementos del Cmap deben tener el mismo nombre para que sean unificadas por la herramienta de traducción en un solo slot.

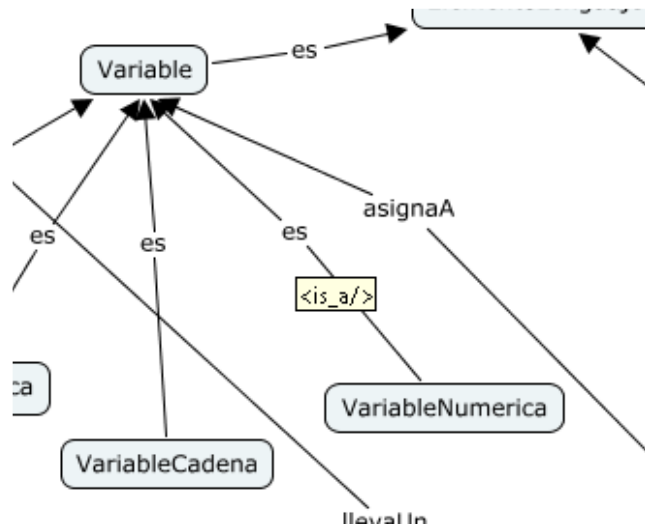


Fig. 1 : Implementación de una relación jerárquica en CmapTools.

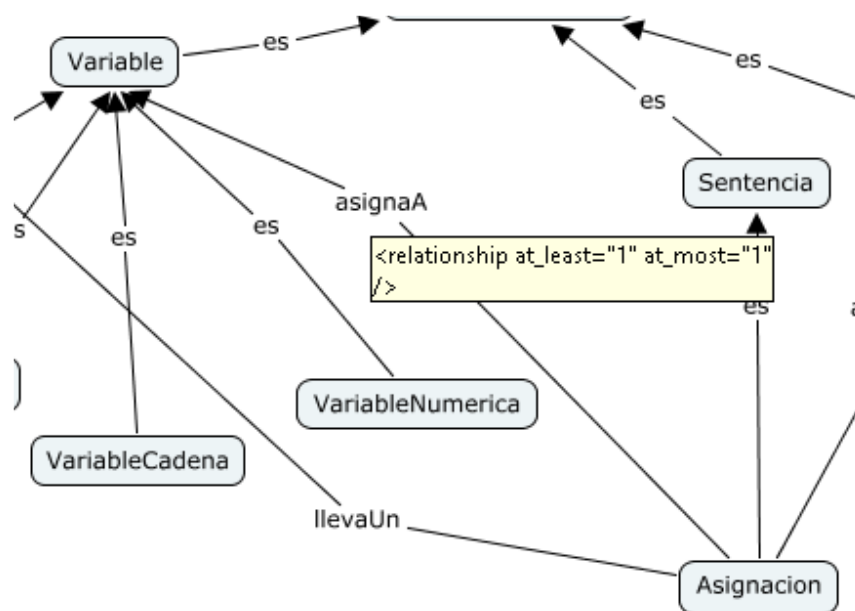


Fig. 2: Implementación de una relación entre conceptos en CmapTools.

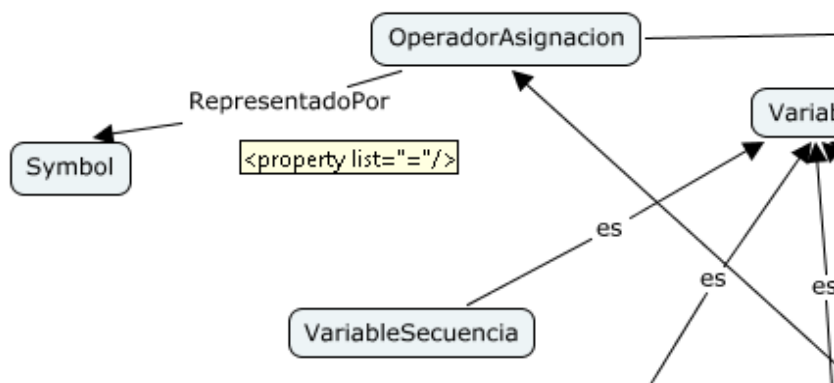


Fig. 3: Implementación de una propiedad Symbol en CmapTools.



## La herramienta de traducción

Se desarrolló una herramienta, denominada CMapToAspire, la que en base a las reglas descriptas permite que los Cmaps sean implementados en ASPIRE-author .

La herramienta utiliza archivos generados por CMapTools de extensión CXL, un lenguaje XML que describe el contenido de los Cmaps, , cuyo esquema cmap.xsd se encuentra disponible.

CMapToAspire analiza el archivo CXL, detecta inconsistencias y en caso de no haberlas genera una salida que permite ser ingresada a ASPIRE. Es de destacar que actualmente ASPIRE-author no cuenta con la posibilidad de importar archivos con las ontologías de dominio por lo que la herramienta no genera una salida en archivo sino una salida de pantalla. Es idea trabajar con los responsables de ASPIRE para desarrollar la capacidad de importar archivos y evitar retrabajo.

La herramienta ha sido utilizada para transformar Cmaps en ontologías de ASPIRE sobre contenidos de unidades de la materia Informática de la FCEEyN de la UNC.

## Ejemplo

En las figuras siguientes se muestra un ejemplo de una ontología simplificada del dominio de conocimiento del tema “asignación de una variable”, parte del contenido de la materia elaborado por el equipo de investigación.

En la fig. 4 se ve el Cmap generado con CMapsTool donde coexisten relaciones jerárquicas, relaciones entre conceptos y propiedades, se visualiza el agregado de información para su transformación a ASPIRE. En el ejemplo el concepto “Asignacion” tiene relaciones con varios conceptos: “Variable”, “OperadorAsignacion” y “Expresion” a la vez que es una “Sentencia”. Mientras que el concepto “OperadorAsignacion” implementa una propiedad llamada “RepresentadoPor” de tipo Symbol.

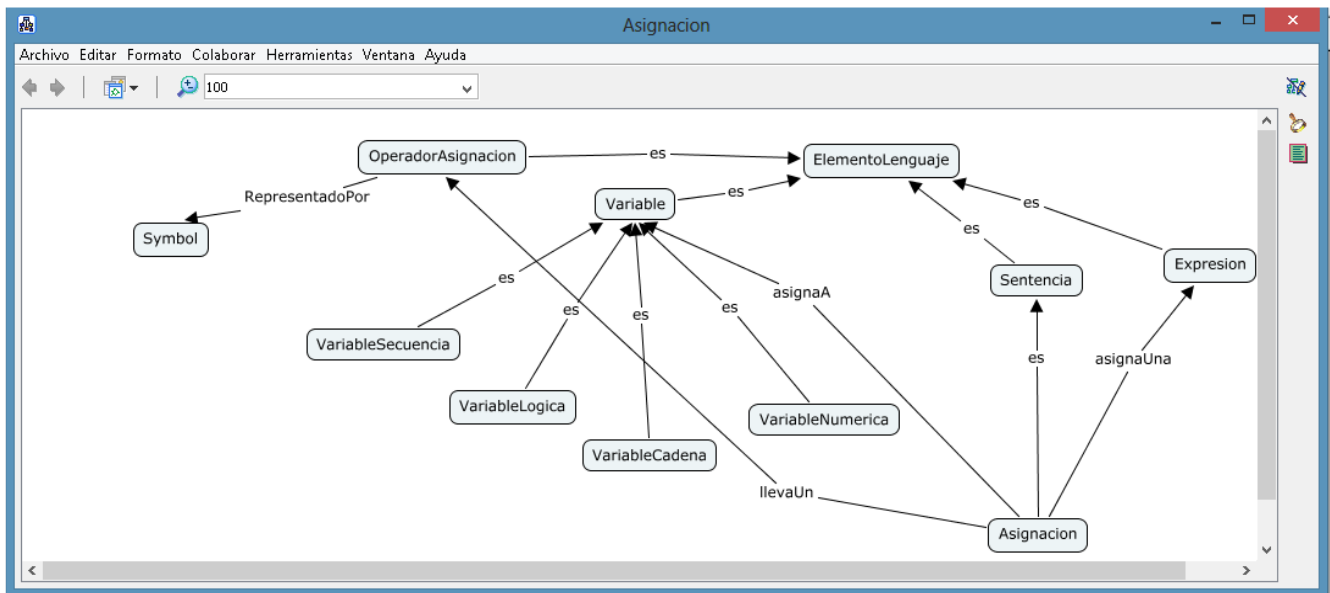


Fig. 4: Implementación completa de la ontología en CMapTools

En la fig. 5 se presenta el archivo .CXL cargado en CMapToAspire y en las figs. 6 y 7 se presenta la salida de la herramienta donde del lado izquierdo se ve la jerarquía de conceptos y del lado derecho se muestran los slots generados del concepto seleccionado para ser implementados en ASPIRE.

```

<map width="921" height="337">
  <concept-list>
    <concept id="1LHR629CB-2BBFJVM-TT" label="ElementoLenguaje" short-comment="&lt;concept abstract=&quot;Yes&quot;/">
      long-comment=""/>
    <concept id="1LHQZJ2T0-12SYKXH-KM" label="Variable" short-comment="&lt;concept abstract = &quot;Yes&quot;/">
      long-comment=""/>
    <concept id="1LHR01F5H-259NFRR-N3" label="VariableLogica"/>
    <concept id="1LJ9957JJ-37D35Q-BB" label="Symbol"/>
    <concept id="1LJ993DWD-209JC8T-8L" label="OperadorAsignacion"/>
    <concept id="1LHR675HN-7XDF5J-VH" label="Sentencia" short-comment="&lt;concept abstract=&quot;Yes&quot;/">
      long-comment=""/>
    <concept id="1LHR01P81-1VZ6QV3-NQ" label="VariableCadena"/>
    <concept id="1LHR02JQ7-GWB11S-Q2" label="VariableSecuencia"/>
    <concept id="1LHR67NVV-1XLQLRK-WF" label="Expresion"/>
    <concept id="1LHR67C46-27S3CB9-VY" label="Asignacion"/>
    <concept id="1LHR021Y9-DXQWB3-PC" label="VariableNumerica"/>
  </concept-list>
  <linking-phrase-list>
    <linking-phrase id="1LHR0D6KR-1LR8HVR-SS" label="es" short-comment="&lt;is_a/">
      long-comment=""/>
    <linking-phrase id="1LHR6D4RZ-QZKT6W-158" label="es" short-comment="&lt;is_a/">
      long-comment=""/>
    <linking-phrase id="1LJFS4TD5-1X6ZLKB-FD" label="RepresentadoPor" short-comment="&lt;property list=&quot;=&quot;/">
      long-comment=""/>
  </linking-phrase-list>
</map>
  
```

Fig. 5: El archivo cxl cargado en la herramienta CMapToAspire.

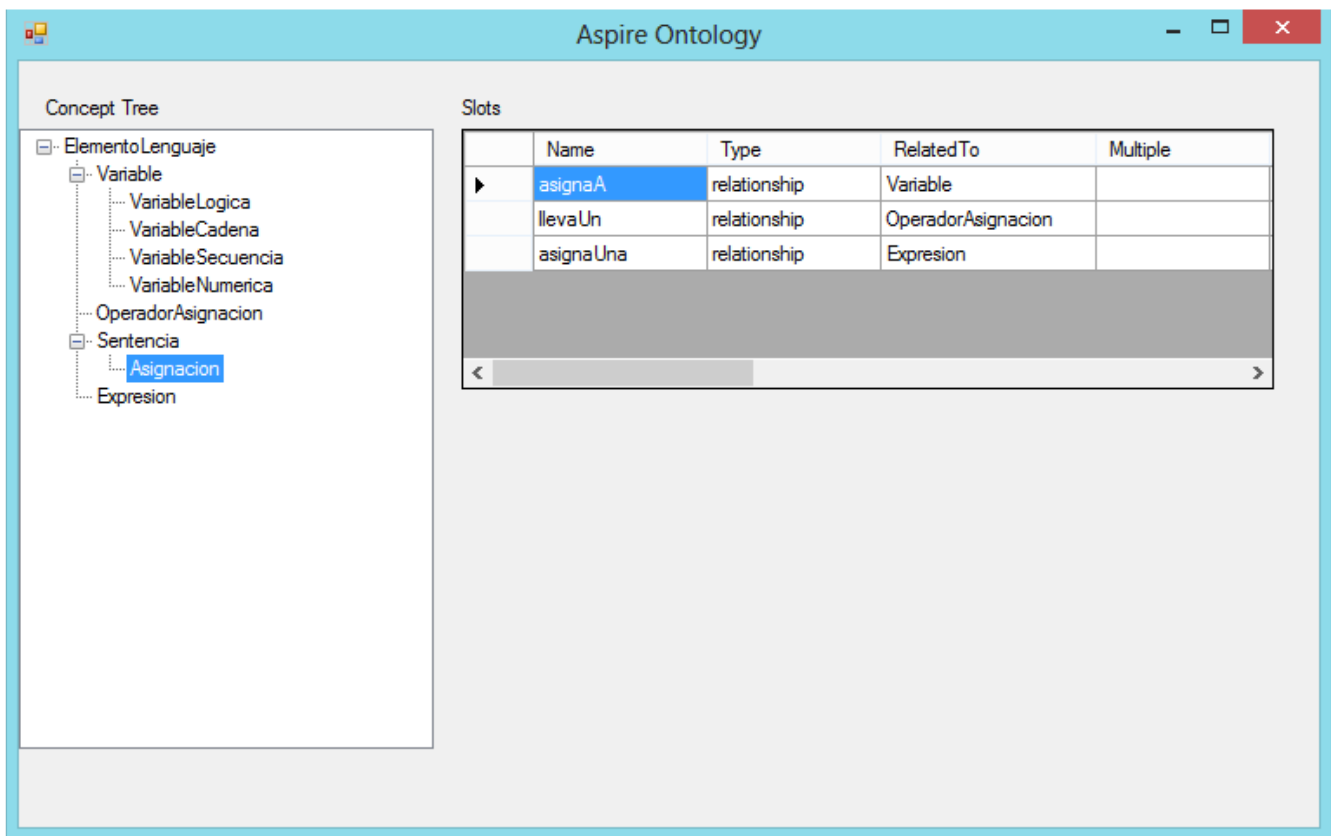


Fig. 6: La salida de la herramienta CMapToAspire con un concepto seleccionado con relaciones.

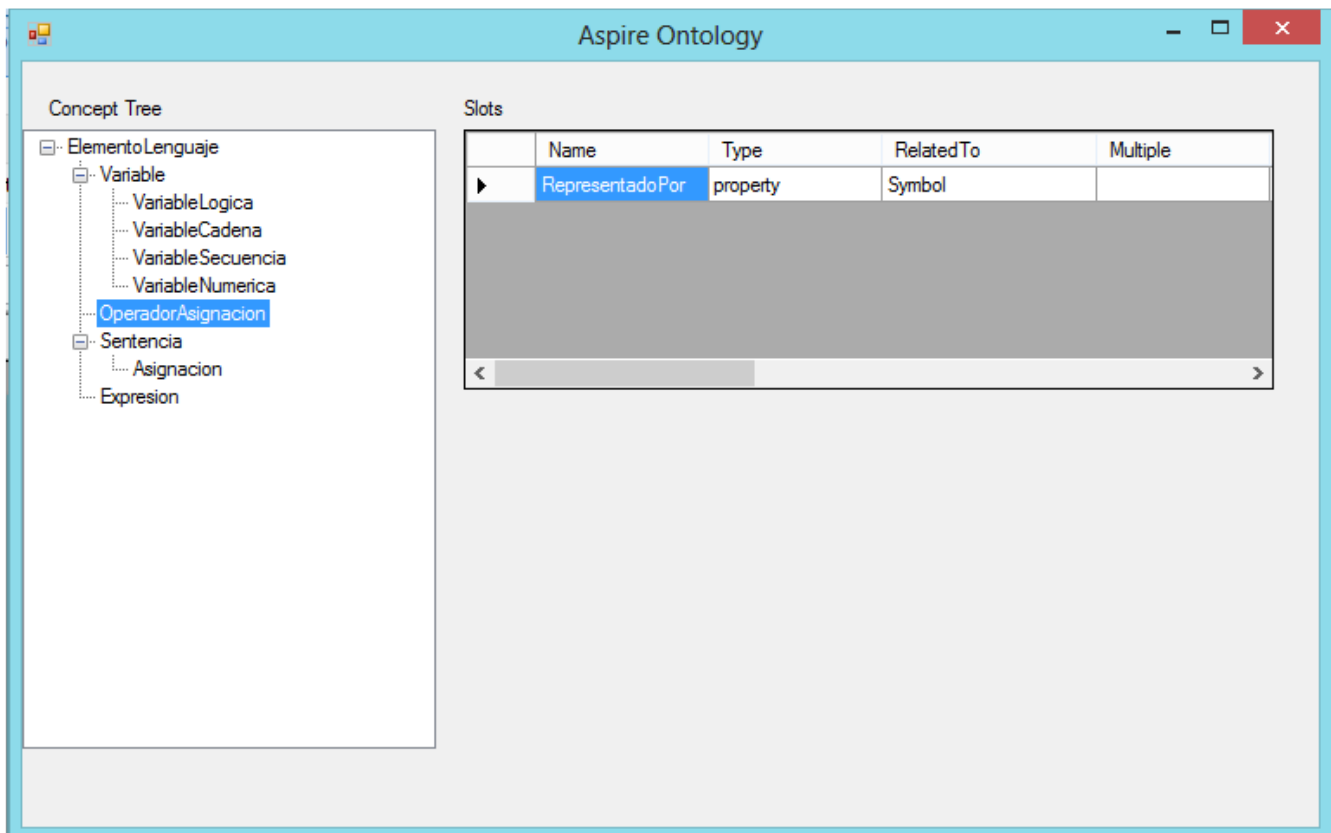


Fig. 7: La salida de CMapToAspire con un concepto seleccionado con un propiedad de tipo Symbol.

Por último se muestra esta ontología cargada en ASPIRE en la fig. 8.

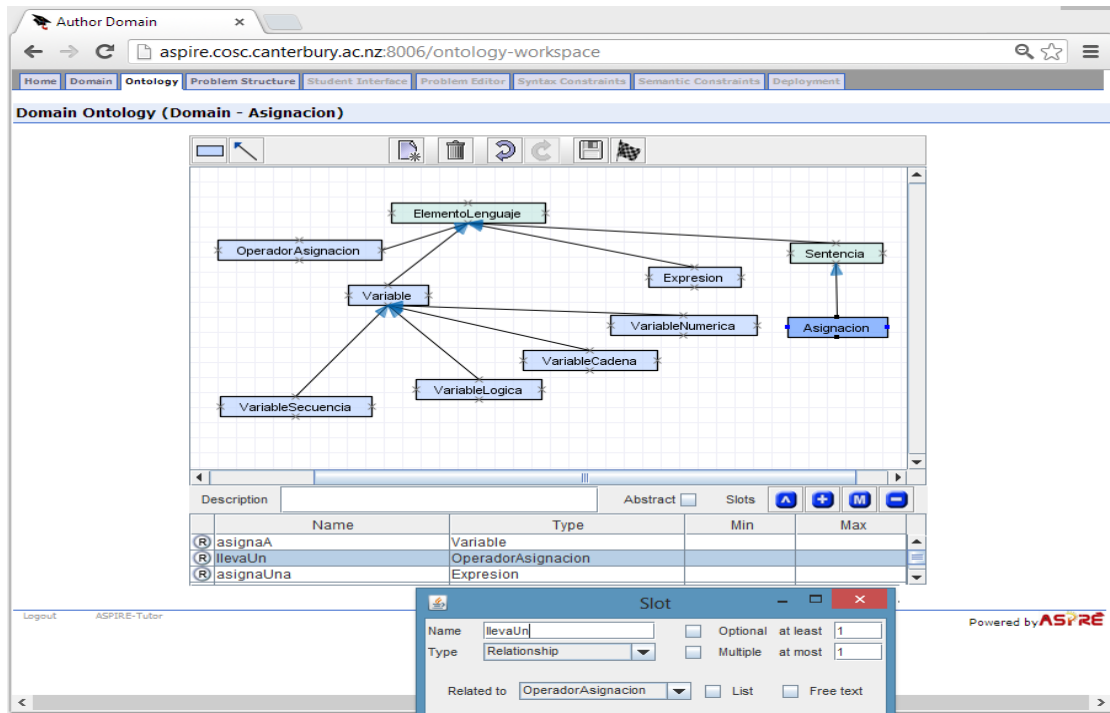


Fig. 8: La ontología del ejemplo en ASPIRE con un concepto seleccionado y editando un slot.

## Conclusiones

En este trabajo se pone en evidencia que es posible establecer correlaciones entre diferentes modos de representar el conocimiento y permitir su portabilidad entre herramientas, en este caso entre CMapsTools y ASPIRE-author.

## Futuros pasos

Como tarea pendiente se destaca la necesidad de permitir que las ontologías traducidas puedan ser importadas directamente en ASPIRE mediante un formato standard, para lo cual es necesario trabajar con el equipo de desarrollo de ASPIRE para que se habilite esa funcionalidad.

Por otro lado este trabajo solo abarca la parte de la formulación de la ontología, como indicamos mas arriba el proceso de autoría de un tutor en ASPIRE tiene pasos subsiguientes, entre los que se destaca la formulación de problemas y soluciones, los que se vinculan a los conceptos para que luego se generen las restricciones automáticamente. Se debería trabajar para ver de que manera se puede usar CMapsTools para implementar los

problemas y soluciones y sus vinculaciones, ampliando la herramienta de conversión para portarlos a ASPIRE.

## Referencias

- (1) C. A. Bartó, Member, IEEE and L. C. Díaz. *Intelligent Systems Applied to Computer Engineering Teaching*. En revista IEEE Latin American Transactions, 2012.
- (2) L.C. Díaz, A. M. Algorry, M. Eschoyez, R. Marangunic and C. A. Bartó, Member, IEEE. *Actions towards the application of intelligent systems in computer education*. En revista IEEE Latin American Transactions, 2012.
- (3) V.H. Forte. *Mapas Conceptuales, La gestión del conocimiento en la didáctica*. Alfaomega Grupo Editor, S.A. de C.V. 2005
- (4) *CMapsTools*. Institute for Human and Machine Cognition <http://cmap.ihmc.us/>
- (5) P. Suraweera, A. Mitrovic, B. Martin, J. Holland, N. Milik, K. Zakharov, N. McGuigan. *Using Ontologies to Author Constraint based Intelligent Tutoring Systems* Intelligent Computer Tutoring Group University of Canterbury Christchurch, New Zealand 2007
- (6) A. Mitrovic B. Martin, P. Suraweera, N. Milik, J. Holland, K. Zakharov. *ASPIRE User Manual*. Intelligent Computer Tutoring Group University of Canterbury Christchurch, New Zealand 2007
- (7) A. Mitrovic B. Martin, P. Suraweera, N. Milik, J. Holland, K. Zakharov. *ASPIRE: An Authoring System and Deployment Environment for Constraint-Based Tutors* Intelligent Computer Tutoring Group University of Canterbury Christchurch, New Zealand 2009