

ZABALA, Gonzalo. Analogía entre el diseño orientado a objetos y las inferencias lógicas utilizadas en el proceso de la investigación científica y los procesos de aprendizaje. En: Revista ieRed: Revista Electrónica de la Red de Investigación Educativa [en línea]. Vol.1, No.3 (Julio-Diciembre de 2005). Disponible en Internet: <<http://revista.iered.org>>. ISSN 1794-8061

Copyright © 2005 Revista ieRed.

Se permite la copia, presentación y distribución de este artículo bajo los términos de la Licencia Pública Creative Commons Attribution-NonCommercial-NoDerivs v2.0 la cual establece que: 1) se de crédito a los autores originales del artículo y a la revista; 2) no se utilicen las copias de los artículos con fines comerciales; 3) no se altere el contenido original del artículo; y 4) en cualquier uso o distribución del artículo se den a conocer los términos de esta licencia. La versión completa de la Licencia Pública Creative Commons se encuentra en la dirección de Internet: <<http://creativecommons.org/licenses/by-nc-nd/2.0/>>

ANALOGÍA ENTRE EL DISEÑO ORIENTADO A OBJETOS Y LAS INFERENCIAS LÓGICAS UTILIZADAS EN EL PROCESO DE LA INVESTIGACIÓN CIENTÍFICA Y LOS PROCESOS DE APRENDIZAJE

Gonzalo Zabala

gonzalo.zabala@vaneduc.edu.ar

Centro de Altos Estudios
Facultad de Tecnología Informática
Universidad Abierta Interamericana
Buenos Aires, Argentina

Los conceptos fundamentales del diseño y la programación orientada a objetos han sido desarrollados hace más de treinta años, pero en los últimos diez ha tenido un crecimiento vertiginoso en la industria de desarrollo de software, merced a los progresos en velocidad y capacidad de procesamiento, junto con la necesidad de creación de software de alta complejidad. A la luz de los pasos del hombre en sus métodos para la construcción del conocimiento, encontramos que la programación ha recorrido en los sesenta años de su existencia un camino similar. En este artículo presentamos características similares entre el método científico, los procesos de aprendizaje y el diseño orientado a objetos, como puntapié a una formalización más rigurosa de la metodología utilizada en la construcción de software.

1. CONCEPTOS BÁSICOS DE DISEÑO ORIENTADO A OBJETOS

“Sospecho, sin embargo, que no era muy capaz de pensar. Pensar es olvidar diferencias, es generalizar, abstraer.”

Funes, el memorioso, Jorge Luis Borges

En los últimos treinta años, la informática ha realizado avances notables con respecto al hardware, las comunicaciones y las capacidades multimediales de las computadoras. Sin embargo, en el plano del desarrollo de sistemas complejos –salvo contadas excepciones– las herramientas elaboradas no usan las capacidades de simulación y emulación que convierten una computadora en un laboratorio virtual de infinitas posibilidades de creación.

A fines de la década del 60, Alan Kay creó la Dynabook, convencido de que la simulación es una herramienta notable para la comunicación de ideas y que una computadora debería ser el contenedor de todos los medios de expresión en los que uno pudiera pensar, es decir, un meta-medio. La Dynabook debía de ser un dispositivo

portátil, con red inalámbrica y pantalla plana, entre otras cosas. Este dispositivo funcionaría como un "amplificador de la mente" y como lugar donde un usuario concentraría toda la información que consume y que genera. En un contexto en el que las computadoras eran grandes máquinas de costo altísimo, de uso privativo de grandes corporaciones, pensar en una computadora personal con estas características era dar un enorme salto hacia el futuro.

Guiados por esta visión, un grupo de referentes de la informática (Alan Kay, Dan Ingalls, Adele Goldberg, etc.) crean Smalltalk, un ambiente de objetos de donde surgen "la mayoría de las cosas buenas relacionadas con las computadoras personales (incluido el propio nombre)"¹.

Una lista, no completa, de los aportes de este proyecto al desarrollo de la informática son:

- El concepto de la Computadora Personal
- El paradigma de objetos
- Smalltalk
- Interfaces de usuario gráficas
- Uso del mouse
- Drag & drop (Arrastrar y arrojar)
- Menús despegables

Curiosamente, el proyecto y la filosofía Smalltalk surgieron con la misma idea: crear un entorno en el cual el proceso de simulación en una computadora fuera equivalente al proceso de pensamiento, de comprensión, de modelización humana. Si, finalmente, el aprendizaje no es más que una modelización mental acotada de un Universo infinitamente complejo y caprichoso, la computadora tenía que presentar un ambiente similar para poder utilizarla como un laboratorio virtual altamente manipulable. Es decir, entendemos al proceso de construcción de software como un proceso de modelización análogo al proceso que realiza un científico al tomar un espacio acotado del universo en el que vivimos, en un conjunto acotado de objetos y propiedades, que permita encontrar un modelo que luego pueda ser aplicado no sólo al conjunto de datos que utilizó para su estudio, sino también a otros casos posteriores que se encuentren dentro de la cota de su estudio.

En el proyecto Smalltalk aparecen conceptos que hoy en día están presentes en todos los ambientes de creación de software: objetos, atributos, clases, mensajes y métodos. Objetos, como aquellos elementos que entran en juego en el sistema que estoy modelando (por favor, pensar en sistemas como algo más amplio que un programa en una computadora). Atributos, como las características que me interesa representar de cada objeto para el recorte que hago del universo. Clases, como el agrupamiento de objetos con características y comportamientos comunes. Mensajes, como la forma de

1 PAPERT, Seymour. La familia conectada. Buenos Aires, Argentina: Emecé, 1997. ISBN 950-04-1727-8.

comunicarse y de solicitar colaboración que tienen los objetos entre sí. Y métodos, como la descripción de la respuesta de un objeto a un determinado mensaje.

Volviendo al concepto de clases, cuando el diseñador se encuentra con un conjunto de objetos que comparten los mismos atributos y el mismo comportamiento (por ejemplo, cada uno de los empleados de una empresa donde se está construyendo un sistema de jornales; cada uno de los productos que produce una fábrica donde se está realizando un sistema de control de stock), utiliza el mecanismo de clasificación para abstraer estas propiedades y comportamiento. De esta forma, la construcción de la clase le permite generar una suerte de “plantilla” a partir de la cual puede construir los objetos que forman parte del sistema. Esta “plantilla” es un modelo recortado que toma únicamente las propiedades y comportamiento que le interesan para el sistema que está modelando en ese momento.

En el proceso de clasificación nos encontramos frecuentemente con clases que comparten algunas características comunes. Por este motivo, en los ambientes de objetos podemos construir clases superiores o abstractas, que encapsulan el comportamiento común, para que el conjunto de clases con características comunes hereden de la clase abstracta dichas propiedades. Al mecanismo por el cual una clase recibe comportamiento y propiedades de una clase superior, se le llama Herencia. Por ejemplo, en el caso de la clasificación de los vertebrados, podemos encontrarnos con la clase Mamíferos. Todas las subclases de Mamíferos heredan las propiedades y el comportamiento que le corresponde a los objetos que pertenecen a la clase Mamíferos. Por lo tanto, los Carnívoros, los Roedores, los Primates, por nombrar algunas de las subclases de Mamíferos, heredan la propiedad de tener mamas, de tener sangre caliente, etc. A su vez, la clase Carnívoro tiene la subclase Perro, la cual hereda los atributos tanto de Carnívoro como de Mamífero. Es decir, cada subclase es una especialización de la clase superior.

Habitualmente, en el diseño frecuente de modelos similares, comenzamos a encontrar estructuras de clases análogas entre los diferentes modelos. A partir del ejercicio de diseño, podemos desarrollar un conjunto de clases vinculadas entre sí, que con pequeñas modificaciones, nos permitan desarrollar soluciones para problemas en ámbitos similares. A esta estructura de clases se la denomina Framework. Sólo el diseño reiterado de modelos similares nos permite encontrar este tipo de estructuras. Por ejemplo, si un desarrollador realiza un modelo del tateti, del ajedrez, del sokoban, del teg, probablemente encuentre un nivel de abstracción superior, los juegos de tablero, que le permita construir un conjunto de clases que con una pequeña adaptación permita diseñar cualquier juego de tablero. Ha encontrado un framework para juegos de tablero. Existen frameworks para sistemas contables, para sistemas de control de producción, para reglas de negocios, etc.

Yendo a un nivel de abstracción más alto aún, en todos los desarrollos de sistemas hay un conjunto de problemas que se presentan en forma sistemática. Luego de años de trabajo, los desarrolladores han encontrado estructuras de clases que permiten presentar una solución efectiva, económica y probada para estos problemas. A estas soluciones se las denomina Patrones de diseño.

Habiendo presentado (en forma harto resumida) los conceptos fundamentales del diseño orientado a objetos, podremos mostrar donde encontramos los métodos de inferencia tradicionales de la ciencia en el proceso de construcción del software.

2. MÉTODOS DE INFERENCIA

En el modelo clásico del proceso de investigación, nos encontramos con dos posibles puntos de partida para el proceso. En el caso de los empiristas, el proceso comienza por la observación, una observación inocente, no contaminada por teorías, a partir de la cual, en el hallazgo de comportamientos comunes de los hechos observados, se inducen una o más reglas que definen la teoría. En el caso de los teoricistas o aprioristas, los científicos predicen un conjunto de reglas sin ningún tipo de fundamentación a partir de las cuales se realizan verificaciones que corroboren o falseen las reglas.

En el primer caso tenemos un modelo que hace hincapié en procesos de inferencias inductivos, donde a partir de un conjunto de hechos similares, defino una regla común que explica todos estos hechos. En el segundo caso, los teoricistas ponen el acento en procesos deductivos, que a partir de la regla o teoría, se define el comportamiento de los hechos observables.

A simple vista podemos ver que ninguno de los dos caminos es posible: nunca realizamos observaciones en un marco "aséptico", sin vivencias y conocimientos previos que obliguen a la observación a dirigirse a ciertos aspectos y no otros. Y tampoco generamos mágicamente teorías a partir de la nada, que luego sean verificadas en los hechos. En sí, ninguno de los dos es un punto de partida: la combinación de prototeorías y protoobservaciones son las que nos permiten, en un camino oscilante de uno a otro, generar el conocimiento científico. A este conocimiento básico, que contiene en forma "primitiva" a la teoría y la observación, Ladriere la llamó precomprensión modelizante. A partir del mismo, mediante dos nuevas formas de inferencia, partimos de este conocimiento en formación o génesis y nos acercamos a lo que llamamos estructura. Estas formas de inferencia son la analogía y la abducción.

La analogía nos permite transpolar conocimientos de objetos o campos distintos a nuestro campo de estudio. Por ejemplo, en investigaciones sobre visión global en ambientes dinámicos llevados a cabo en nuestro centro, partimos de la idea gestáltica del proceso por el cual nuestro cerebro completa la imagen aún teniendo información incompleta de la misma. De esa manera, comenzamos nuestras investigaciones con algoritmos estadísticos que, con poca información de visión, permitían reconstruir los objetos en un tiempo mínimo.

La abducción nos permite inferir una hipótesis interpretativa de la causa del rasgo encontrado al relacionarlo con cierta regla que ya poseemos. Por ejemplo, si en nuestra detección de figuras en una imagen utilizando métodos estadísticos, nos encontramos que ante determinada toma de datos el error de detección es alto, arriesgaremos por abducción que la elección de los datos para el reconocimiento ha sido errónea. En este caso, el rasgo (error alto en la detección de la figura) junto a la Regla (con una muestra de distribución uniforme del 3% de los puntos de un cuadrado detectamos la posición de la figura con un porcentaje de error bajo) nos permiten abducir que el Caso (la

muestra tomada de la imagen) ha tenido un problema de sesgo, es decir, una muestra con distribución no uniforme o un número menor al 3% de los puntos del cuadrado.

En síntesis, la deducción, la inducción, la abducción y la analogía son los métodos de inferencia esenciales (pero no los únicos) sobre los que se basa la construcción del conocimiento en el proceso científico. Presentados estos, veamos la vinculación existente entre ellos y el proceso de diseño de software orientado a objetos.

3. LOS PROCESOS DE INFERENCIA EN EL DISEÑO ORIENTADO A OBJETOS

A continuación tomaremos cada uno de los métodos de inferencia presentados, y mostraremos en qué momentos del diseño podemos encontrarnos con cada uno de ellos. Tal como en el proceso científico, estos mecanismos no se dan compartimentados. Aunque los presentemos como momentos separados, todos los métodos de inferencia actúan engarzadamente para llevar adelante la construcción del sistema.

Previamente debemos aclarar que el diseñador de un sistema recibe un conjunto de documentos realizados por los analistas. Esta documentación refleja en un lenguaje no formal los requerimientos de construcción del “sistema virtual” realizados por el experto, es decir, aquella o aquellas personas que conocen el “sistema real” y que transmiten su conocimiento para simular en un ambiente virtual el mundo real o imaginario que ellos conocen.

a) Deducción

Probablemente sea el menos constructivo pero a su vez el proceso más utilizado en el diseño. Cuando creo un objeto de una clase, al pertenecer dicho objeto a la clase, puedo deducir cuáles son las propiedades y mensajes a los cuales responde el objeto. Es decir, la definición de la clase es la Regla de mi proceso de inferencia, el objeto que yo creé es el Caso, y los rasgos son las propiedades y comportamiento que el objeto me ofrecerá. Veamos un ejemplo: tenemos la clase Alumno, que tiene las propiedades apellido, nombre, dirección, teléfono, notas, presentismo, etc; además le puedo enviar el mensaje nota informándole en qué materia se sacó la nota, cuándo y cuál es la nota. Cuando creamos un objeto de esa clase, sabemos que el objeto tiene las propiedades y responde a los mensajes que su clase tiene definidos. Aún más, si modificamos la definición de la clase, automáticamente todos los objetos creados de la clase sufrirán la misma modificación.

b) Inducción

Cuando el diseñador realiza la primera aproximación con la documentación del análisis, comienza por la búsqueda de los objetos que dicha documentación describe. En su lectura, encuentra que cada objeto presenta ciertas propiedades y comportamientos, y que muchos de ellos los comparten. Por ejemplo, en la descripción del análisis de un sistema de administración de alumnos de una facultad, encontramos un alumno que se inscribe a un determinado tipo de curso. En otro momento de la descripción encontramos a otro alumno que paga su cuota. De esta manera comienzo a concluir que hay un conjunto de objetos que comparten ciertos atributos y que responden a los

mismos mensajes: he descubierto una Clase. Esta clase no presenta todas las características de los objetos del mundo real; sólo modelará aquellas que sean de incumbencia para el sistema que se está construyendo.

En este caso, hemos realizado un proceso inductivo. A partir de los diferentes Casos (los objetos que describe el análisis) y de los rasgos que cada objeto ha presentado, construimos la Regla, es decir, nuestra Clase.

c) Analogía

Los dos procesos anteriores serían imposibles si partimos de la nada. Así como un arquitecto no construye una casa a partir de arena y piedra, los diseñadores no construyen desde clases atómicas. Siempre que el diseñador lee la documentación del análisis, tiene en su mente el bagaje de las lecturas previas y de los diseños anteriormente realizados. Es más, todos los ambientes de programación orientada a objetos, ofrecen al diseñador un conjunto muy importante de clases ya creadas e implementadas, las cuales puede utilizar para heredar o para componer. Los procesos de herencia y de composición mismos se presentan como una analogía a lo ya construido.

Por ejemplo, cuando nos encontramos con la necesidad de manipular un conjunto de objetos, sabemos que en las clases ya implementadas tenemos algunas de ellas que nos permiten trabajar con colecciones: colecciones indexadas y no indexadas, bolsas, conjuntos, colecciones con índices de dos dimensiones, etc. A partir de este conocimiento, decidiré si utilizo directamente alguna de ellas, si creo una subclase de una de dichas clases modificando y especificando su comportamiento, o si me conviene realizar una clase completamente desde cero (lo cual es muy poco común).

d) Abducción

Como postula Aníbal Bar en², la abducción en muchos casos tiene un fuerte vínculo con la analogía. Cuando en el diseño de objetos, nos encontramos con algunos objetos que presentan determinadas características y comportamientos, y conociendo las clases preexistentes que me proporciona el ambiente, es habitual que utilicemos la abducción. Es decir, a partir de ciertos rasgos del objeto, y la definición de una determinada clase (Regla), suponemos que ese objeto pertenece a la clase (Caso) y por lo tanto, automáticamente el objeto se enriquece con las demás propiedades y comportamiento de la clase. Puede ocurrir que en ese momento nos demos cuenta de que otras características de la clase no coinciden con las características del objeto, con lo cual la hipótesis abductiva queda descartada. Como vemos, este proceso está íntimamente vinculado con el mecanismo de la analogía.

Esta es una breve analogía entre los métodos de inferencia y el proceso de diseño de software orientado a objetos. Pero no es en el único modo en el que ambos procesos se relacionan. A continuación presentaremos otras similitudes entre ellos.

2 BAR, Aníbal. Abducción. La Inferencia del Descubrimiento. En: Cinta de Moebio [online]. Santiago, Chile: Facultad de Ciencias Sociales. Universidad de Chile. 2001, Nro 12. Disponible en <http://rehue.csociales.uchile.cl/publicaciones/moebio/12/>. ISSN 0717-554X.

4. OTRAS CARACTERÍSTICAS COMUNES ENTRE EL PROCESO DE LA INVESTIGACIÓN CIENTÍFICA Y EL DISEÑO ORIENTADO A OBJETOS

La lectura de³ revela importantes características comunes entre el proceso de investigación y el diseño orientado a objetos. “La comprensión es, entonces, un acto imaginario o simbólico de producción, en el sentido en que algo queda comprendido o explicado para un sujeto cuando los datos del hecho real pueden ser puestos en correspondencia con representaciones internas que corresponden a los esquemas de producción habituales para esa cultura”. De la misma manera, el diseñador crea un modelo simbólico del sistema que está construyendo, y este modelo es correcto (es verificado), cuando puede ser puesto en funcionamiento con los datos de la realidad (o de la fantasía) desde los cuales se hizo el primer análisis. Por lo tanto, el proceso de construcción de software no es ni más ni menos que un proceso de comprensión y modelización de un determinado conjunto de objetos de la realidad (o la fantasía)⁴ que se desea llevar a un sistema de simulación en una computadora.

En otra parte del mismo texto, el Dr. Samaja describe a los sistemas formales como “simulacros o ‘mapas’ (modelos) que permiten una drástica simplificación y generalización de los mecanismos que –por hipótesis- gobiernan el comportamiento de las cosas de manera que operando luego sobre tales mapas logramos orientarnos en el manejo de los procesos reales”. Esta descripción de los sistemas formales no es ni más ni menos que la descripción de un software realizado en un ambiente de objetos. En la puesta en marcha del sistema sobre los procesos reales, iremos modificando nuestro diseño de manera tal que vayamos reflejando con mayor exactitud la realidad. Posteriormente, y una vez que el sistema está validado, corroborado, podremos modificar variables para proyectar y estudiar comportamientos posibles ante sucesos que no se han dado en el mundo real.

En el diseño orientado a objetos, la tarea de exploración, la construcción de la “precomprensión modelizante” es una etapa fundamental. Curiosamente, dado que esta metodología es relativamente nueva en el mercado de desarrollo de software, muchas veces los líderes de proyecto ven con malos ojos este proceso, porque consideran que la programación es una actividad más “apriorista”: lanzarse a “construir” hipótesis (modelos, programas) y luego verificar si estos coinciden con la realidad. Es habitual la discusión entre gerentes y desarrolladores, porque estos últimos “no han escrito nada”, sólo se detuvieron a diseñar y a comprender el modelo, sólo eso...

Como dijimos anteriormente, el diseño es necesariamente un proceso de simplificación, de reducción. Como bien comenta Borges en Funes, el memorioso, pensar (¿diseñar? ¿modelar? ¿hacer ciencia?) es olvidar diferencias, generalizar, abstraer... “[...] cualquier intento de conocer la realidad está obligado a operar de manera inevitable una drástica reducción de su infinita complejidad mediante una operación que, de manera básica, consiste en proponer cuáles serán los elementos o componentes relevantes que se tomarán en cuenta y qué aspectos de ello serán atendidos a la hora

3 SAMAJA, J. (2002) La ciencia como proceso de investigación. En: DEI, Daniel. Pensar y hacer en investigación. Bs. As, Argentina: Editorial Docencia y Fundación Hernandaria. p. 153-235.

4 Se resalta el aspecto de la fantasía para dar cabida a la construcción de juegos, modelos matemáticos, etc.

de su descripción” escribe Samaja (op.cit). De la misma manera, un diseñador trabaja como un Anti – Funes: olvida, generaliza, abstrae, para construir su modelo. Como un científico...

Podríamos encontrar otros espacios comunes entre el trabajo de un científico y de un diseñador con respecto a la construcción del objeto modelo, a la formalización y a la experimentación. Pero queremos hacer hincapié en la última etapa del proceso científico, el punto más alto, al cual no todos los investigadores llegan: la creación o definición de una teoría. ¿Existe algún proceso similar en el diseño de software? Como comentamos en el primer punto, cuando un diseñador se encuentra una y otra vez con sistemas similares que debe modelar, comienza a encontrar un conjunto de clases comunes a todos los sistemas de ese tipo. Es a esto lo que llamamos framework. Cuando un diseñador puede construir un framework para un determinado tipo de sistemas, ha llegado a un punto de comprensión y de modelización que cubre horizontalmente un tipo de problema. A partir de ese framework, futuros diseñadores podrán comprender en forma genérica el marco del problema, simplemente personalizar, modificar levemente el mismo, para que se ajuste a las necesidades específicas del sistema que están realizando.

Y encontramos un paso más en esta búsqueda del “patrón universal”, del “número de Dios”: los patrones de diseño. Estos patrones presentan soluciones a problemas universales, a conflictos de diseño que se encuentran en cualquier sistema que uno está construyendo. Son un conjunto de ideas iluminadoras que cubren todo tipo de problemas. Sólo hay un puñado (no más de 20 pequeños modelos), pero conocerlos le permite al diseñador resolver en forma segura (porque son modelos muy probados) y velozmente cientos de los conflictos habituales de su diseño.

Muchísimos elementos comunes entre el proceso científico y el de diseño quedan por describir. Es intención del autor que este material sirva como punto de partida para una “epistemología” de la informática y de la formalización del proceso de construcción del software. Es una tarea que nos debemos para poder entrar en el Olimpo de las ciencias.

5. VINCULACIÓN ENTRE EL PROYECTO SMALLTALK Y LOS PROCESOS DE APRENDIZAJE

En la obra cuya cita da comienzo a este artículo, Borges presenta a un personaje de nombre Funes, que por un accidente recuerda absolutamente toda su vida, detalle por detalle: “Dos o tres veces había reconstruido un día entero; no había dudado nunca, pero cada reconstrucción había requerido un día entero.” Funes captaba en un golpe de vista todos los detalles del mundo que lo rodeaba, y los almacenaba en forma absoluta. Para Funes, cada elemento del Universo es un elemento distinto a otro, sin características comunes. Más aún, en cada momento un mismo objeto se diferencia de sí mismo en un tiempo anterior o futuro. Y absolutamente todo se almacena en su memoria. Sin poder desechar información inútil. Sin poder encontrar leyes comunes de comportamiento o de cualidades de los elementos. Y por lo tanto, como bien dice el narrador del cuento, Funes se convierte en un ser incapaz de pensar. Funes, al no generalizar, no puede encontrar patrones comunes, y por lo tanto, no puede clasificar.

Un hipotética enciclopedia de Funes sería casi infinita... ¿Qué diferencia a Funes de un niño en sus etapas de aprendizaje?

Cuando los niños se enfrentan por primera vez a un juego en una computadora, y dado que en general no leen los manuales de instrucciones de los mismos, para comprender cómo llegar al objetivo final, realizan las siguientes tareas:

- Identifican a los objetos que entran en juego: qué representan ellos, cuáles son los enemigos, cuáles son los colaboradores, qué objetos tiene el mundo en el cual están jugando.
- Determinan qué propiedades tiene cada uno de estos objetos: color, peso, habilidades, costo, forma, etc.
- Una vez que determinan los objetos existentes y sus propiedades, los dividen en grupos en los que los objetos tengan propiedades comunes: armas, enemigos, amigos, jugadores, fichas, etc. De esta manera, proceden a clasificar los objetos.
- Por último, descubren la manera de comunicarse con estos objetos (texto, acciones, secuencia de teclas, etc) y qué servicios les presta cada uno de ellos.

Este proceso no es lineal; se retroalimenta constantemente con nuevos descubrimientos y con modificaciones de lo ya definido. De esta manera, los chicos aprenden los elementos necesarios para intentar llegar a la meta u objetivo del juego: descubrir un objeto, no llenar un recipiente, matar a todos los enemigos, etcétera. Es decir, los niños son el anti-Funes en su proceso de aprendizaje: ellos sí piensan, porque olvidan diferencias, abstraen, generalizan... De la misma forma en la que aprendieron a manejarse en este mundo cuando fueron pequeños: preguntando, probando, encontrando similitudes que permitieran compactar una inmensa cantidad de información, en un conjunto de conceptos fundamentales que les permitiera modelar mentalmente al mundo que los rodea.

Vemos entonces claramente, el poderoso vínculo que hay entre los conceptos explicados en el primer punto de este artículo, y los procesos educativos. Tal es así, que en la década del 90, la empresa Disney contrató a Alan Kay y su equipo para desarrollar un ambiente de objetos pensado y diseñado para niños. Este proyecto se concretó en Squeak, un desarrollo gratuito y completamente abierto que permite experimentar con los conceptos que hemos desarrollado, en un espacio lúdico atractivo y multimedial. Sin entrar en detalle, queremos demostrar con esta vinculación entre los ambientes de objetos y el proceso educativo, que Smalltalk ofrece a los docentes una herramienta de creación, de representación del proceso de aprendizaje, que está mucho más allá de un uso transversal de la herramienta: Squeak es un laboratorio infinito, el más maravilloso de los laboratorios que hemos conocido, exceptuando al Universo mismo. Squeak es una herramienta para educar a los anti-Funes: alumnos pensantes, reflexivos de su propia tarea de aprender. Aunque escapa al alcance de este artículo, invitamos a todos los docentes e investigadores de la educación a conocerlo, visitando el sitio <http://www.small-land.org>.

¿Qué categorías usamos en nuestros procesos de clasificación? ¿Cómo creamos nuestros frameworks? ¿Qué patrones hemos encontrado en el mundo que nos rodea?

Finalmente, tal vez ese es el punto que une a las tres actividades mencionadas en este artículo: la búsqueda de patrones que nos permitan comprender el mundo de una forma sencilla, metódica, ordenada, previsible... Probablemente, esa sea una de las búsquedas fundamentales de la humanidad.

BIBLIOGRAFÍA

PAPERT, Seymour. La Familia Conectada. Buenos Aires, Argentina: Emecé, 1997. ISBN 950-04-1727-8

BAR, Aníbal. Abducción. La Inferencia del Descubrimiento. En: Cinta de Moebio [online]. Santiago, Chile: Facultad de Ciencias Sociales. Universidad de Chile. 2001, Nro 12. Disponible en Internet: <<http://rehue.csociales.uchile.cl/publicaciones/moebio/12/>>. ISSN 0717-554X.

SAMAJA, J. (2002) La ciencia como proceso de investigación. En: DEI, Daniel. Pensar y hacer en investigación. Bs. As, Argentina: Editorial Docencia y Fundación Hernandaria. p.p. 153-235.