

Un Ambiente para Especificar Aplicaciones Pedagógicas de Aprestos Matemáticos

Francisco Cárdenas
Isabel Peirano
Miguel Nussbaum
Escuela de Ingeniería

Ricardo Rosas
Escuela de Psicología

P. Universidad Católica de Chile
Casilla 306, Santiago 22, Chile
FAX 56 2 6865881
fjcarden@ing.puc.cl

Resumen

El presente trabajo, trata acerca de la implementación de un modelo de desarrollo de Sistemas Tutoriales Inteligentes (ITS) que se basa en emplear reutilización a nivel de conocimiento, encapsulándolo bajo la forma de estructuras¹, sobre las cuales un educador le adiciona nuevo conocimiento por medio de un editor de alto nivel. Se generan actividades de entrenamiento para el alumno, y se realiza una autorregulación para ajustarse a sus necesidades. Particularmente el prototipo consiste en un sistema para la adquisición del concepto de número en preescolares, utilizando aprestos matemáticos por medio de computador. Se distinguen tres tipos de usuario: el experto, de quien se “extraen” las estructuras; el profesor, que aporta contenido instruccional a las estructuras definidas por el experto, y el alumno, quien usa el ambiente de simulación para realizar su entrenamiento, monitoreado por el sistema experto. El prototipo usa reglas parametrizadas, y está desarrollado para plataforma PC en Visual Basic. Consta de los siguientes módulos: Edición de Contenidos, Definición de Actividades, el Simulador y un Módulo Habilitador. Se deja indicado en el diseño de la arquitectura además, la existencia de un Módulo de Afinamiento de Parámetros, y otro de Analisis de Resultados. Se trata de demostrar que un profesor no necesita tener grandes conocimientos de computación para generar un ITS dentro de un dominio específico, siendo las principales fortalezas de esta herramienta su flexibilidad y sencillez. Esta basado en los paradigmas instruccionalista y constructivista, empleando lo lúdico para capturar la atención del alumno.

Palabras Clave: Encapsulación y reutilización de conocimiento, edición de actividades y contenidos, autorregulación, ITS, apresto matemático.

¹ Nos referimos a estructuras de ejercicios genéricas, a la que posteriormente se le adicionará contenido instruccional dentro de un contexto específico.

1. Introducción

Desde hace un par de décadas los investigadores del área de inteligencia artificial estudian la aplicación de sistemas expertos en el ámbito de la educación. Ellos han desarrollado múltiples modelos de sistemas computacionales que ayudan, tanto a niños como adultos, a adquirir nuevo conocimiento.

Una de las primeras contribuciones de la inteligencia artificial a la educación lo constituyen los denominados Sistemas Tutoriales Inteligentes (ITS, del inglés Intelligent Tutoring Systems). Estos sistemas esencialmente buscan imitar la interacción persona a persona que se produce entre un educador, que es experto en cierto dominio del conocimiento, y un alumno que desea aprender tales conceptos y contenidos. En apariencia, los ITS se asemejan a los Sistemas de Instrucción Apoyados por Computador (CAI, Computer-Aided Instruction), que propician la ejercitación por medio de preguntas y respuestas. Pero existe una diferencia fundamental: el núcleo de un ITS es su Sistema Experto, el cual le permite acompañar al alumno en cada paso de la construcción de la solución del problema, detectando los conceptos o contenidos que el alumno desconoce o no utiliza adecuadamente, pudiendo así introducir las modificaciones al plan instruccional que considere necesarios. Habitualmente la forma de representación del conocimiento al interior de un ITS viene a estar dado por un modelo del alumno, un modelo instruccional y un modelo del dominio [Rickel89].

Generalmente, se realiza desarrollo de ITS en base a prototipos y éstos son realizados de manera incremental a través de sucesivos refinamientos; por lo tanto, el tiempo y costo de desarrollo se reduce grandemente si se reutiliza conocimiento. La base para generar software de ITS reusable es el diseño modular de sus componentes, y este diseño no sólo involucra reusabilidad, sino también que los componentes sean compatibles y transportables². [Arruarte96], da una clasificación acerca de la reutilización de conocimiento en los siguientes niveles:

- Bases de conocimiento/estructuras de datos.
- Biblioteca de recursos de enseñanza.
- Biblioteca de estrategias.
- Ontologías estructuradas.
- Reusabilidad de módulos.
- Reusabilidad de arquitecturas.

Donde el nivel más bajo, está dado por reutilizar la base de conocimiento (del dominio), y el nivel más alto, por reutilizar la arquitectura completa de un ITS para desarrollar un sistema nuevo. Considera además las siguientes formas de alcanzar reusabilidad:

- Mediante reutilización directa de conocimiento.
- Mediante herramientas que promueven la reusabilidad.
- Mediante herramientas para desarrollar aplicaciones.

Otros estudios en el desarrollo de arquitecturas de ITS, indican que existe un *continuum* en el grado de generalidad que éstas (las arquitecturas de ITS reusable) permiten en el sistema resultante. [Blument96] con su proyecto LEAP, señala que los *ITS shells*³ son o muy genéricos, o pedagógicamente débiles, o demandan una gran cantidad de esfuerzo, comparados con los ITS individuales confeccionados a medida, aspecto similar a lo que ocurrió con los *ES shells* (expert systems). En un punto intermedio entre estos extremos, están los ITS que se restringen a un dominio específico o a un tipo particular de tarea.

En las secciones siguientes se comenzará por describir el modelo propuesto de desarrollo de sistemas instruccionales, explicando los diferentes roles involucrados y su relación con el concepto de reutilización, que está dado a nivel de reutilización de conocimiento del dominio, mediante la edición de contenido. Posteriormente, se verá una

² Compatibles, en el sentido de que el mismo conocimiento pueda ser usado por varios sistemas implementados en diferentes lenguajes de programación y transportable, en el sentido que pueda ser instalado en distintas plataformas.

³ Shell para ITS, es una herramienta para construcción de ITS.

implementación en base a una arquitectura que soporta los conceptos explicados.

2. Modelo Propuesto

2.1 Características Generales

El objetivo principal nuestro, es generar una herramienta que pueda ser puesta a disposición de educadores, quienes sin ser expertos en computación, sean capaces de desarrollar, en corto tiempo, aplicaciones tutoriales adecuadas a sus necesidades. El educador, para hacer uso del conocimiento encapsulado, debe agregar elementos al sistema en base a su experiencia y/o percepción del entorno específico dentro del cual el sistema tutorial será utilizado. Como consecuencia de esta combinación de información experta respecto del dominio, el sistema es capaz de generar automáticamente ejercicios, escoger contenidos apropiados y presentarlos al alumno durante la sesión de ejercitación.

Desde el punto de vista de la reutilización de conocimientos, el modelo propuesto presenta una restricción en cuanto a la dependencia del sistema tutorial con un dominio específico del conocimiento. Se logró para determinado dominio del conocimiento, ciertas *estructuras* que adoptan las diferentes formas de ejercitación posibles de efectuar dentro de ese dominio. De este modo, múltiples formas de ejercitación se pueden hacer converger a unas pocas *estructuras* que determinan las características generales y la funcionalidad de las actividades que serán empleadas por el sistema tutorial para la ejercitación. Al relacionar cada estructura con un contenido específico y con información referente a la complejidad objetiva que presenta determinado ejercicio, se obtiene una *actividad*, de carácter menos genérica que la *estructura* en la cual se basa, pero que admite múltiples instancias, distintas unas de otras, las que son presentadas bajo la forma de *ejercicios* al alumno.

En estricto rigor, es el sistema experto el responsable del funcionamiento del sistema instruccional al interior de la sesión de ejercitación. Éste se encarga de relacionar la información existente respecto del alumno y su nivel de conocimientos actual, con las actividades que han sido definidas previamente por el educador, ajustando el nivel de complejidad de los ejercicios a cada alumno. Además evalúa el desempeño experimentado por el alumno en los ejercicios previos, para mantener actualizado el modelo de conocimientos de éste.

Para llevar a cabo estas tareas, el sistema experto encapsula el conocimiento por medio de reglas del tipo: *si se cumple cierta condición, ejecutar determinada acción*. Distinguimos diversas familias de reglas, dependiendo de la función que cumplen al interior del sistema:

- Reglas que determinan *qué actividades presentar* al alumno, dependiendo del nivel de conocimientos de éste. Es deseable presentar actividades cuyas destrezas aún no han sido logradas por el alumno, que empleen estructuras que han presentado aplicaciones exitosas en otras actividades, etc.
- Reglas que determinan la *complejidad del ejercicio* que va a ser presentado. Los valores de los parámetros que inciden en la complejidad pueden depender de los últimos valores registrados por el alumno en ejercicios previos basados en la misma actividad, o bien basarse en actividades que trabajan la misma destreza o hacen uso de la misma estructura, o bien tomar valores por defecto si no hay registros previos.
- Reglas que determinan cuándo inducir en el alumno *cambios de actividad*, para que la progresión sea gradual y no se alcance el umbral de frustración o excesiva simplicidad de un ejercicio.
- Reglas que activan *niveles de ayuda*, al detectar comportamientos erráticos o errores cometidos por el alumno. Ésta puede ir desde repetir el enunciado hasta dar indicios de qué elementos no están siendo considerados por el alumno en su análisis.
- Reglas que determinan la *forma en que son evaluados* los ejercicios, dependiendo de la estructura en la cual se basan. Determinan qué es considerado una respuesta correcta y

qué tipos de respuesta incorrecta son admisibles, gatillando acciones correctivas y *feedback* al alumno.

- Reglas que manejan *interrupciones*, que pueden ser provocadas por el alumno, el educador o incluso el mismo sistema. Por ejemplo qué hacer si el alumno presiona un botón para terminar un ejercicio que aún no ha sido resuelto.
- Reglas que detectan *comportamientos* anormales o que interfieren en el proceso de aprendizaje. Indican qué hacer, por ejemplo, ante la inactividad del usuario, o ante un comportamiento impulsivo.

Las reglas son parametrizables, lo que quiere decir que están expresadas en función de variables o parámetros, los cuales son instanciados al momento de evaluar el cumplimiento de la regla. Con esto se consigue poder modificar y afinar su funcionamiento sin tener que modificar el código con el cual ellas han sido escritas. Por ejemplo, una regla que indica:

***“Si pasan 30 segundos y el alumno no ejecuta movidas válidas,
entonces repetir el enunciado del ejercicio”***

resulta más general si dice:

***“Si pasan ‘s’ segundos y el alumno no ejecuta movidas válidas,
entonces repetir el enunciado del ejercicio”***

Son las mismas reglas, por medio de las acciones que gatillan, las que determinan el cambio de valor de los parámetros. Finalmente, no resulta conveniente que la funcionalidad del sistema tutorial se limite a ofrecer instancias de ejercitación. En la medida en que el sistema experto logra configurar un perfil del alumno con el que ha tomado contacto, puede compartir dicha información con el educador, de modo de enriquecer aún más el conocimiento que éste posee respecto de la cantidad y profundidad de conocimientos que maneja el alumno. De esta forma, es posible delegar en un experto humano la aplicación de estrategias o el desarrollo de habilidades de orden superior, que resultan difíciles de encapsular en un sistema computacional de instrucción. Asimismo, el sistema debe poder ser objeto de una evaluación crítica por parte del educador o experto humano, la cual se puede traducir en un afinamiento de variables y parámetros que inciden en el funcionamiento interno del sistema.

2.2 Perfiles de Usuario que Interactúan con el Sistema Instruccional

El sistema instruccional, tal y como ha sido diseñado, contempla la existencia de 3 tipos o perfiles de usuario claramente diferenciados. El alumno constituye el primer perfil de usuario. Su participación dentro del contexto del sistema instruccional resulta evidente, puesto que es él quien desea adquirir nuevo conocimiento como consecuencia de su interacción con el sistema. Su participación se circunscribe principalmente a la sesión de ejercitación, durante la cual le serán presentadas múltiples alternativas de ejercicios, entre las cuales éste podrá optar.

Sin embargo, se debe tener en consideración que el educador probablemente desea trabajar no sólo con un alumno, sino con un grupo de éstos, por lo que el sistema tutorial debe manejar el concepto de *usuario*. Asimismo, para un mismo usuario, dado que se produce una progresión en la cantidad y calidad de los conocimientos que éste domina a lo largo del tiempo, es deseable que exista una administración de *sesiones*, que registre aquellas variables relevantes para cada usuario, que den cuenta del momento en que se produce la adquisición de nuevas habilidades y destrezas.

Un segundo perfil de usuario lo constituye el educador. Éste ha sido considerado al interior de este modelo como un participante activo, quien tiene a su cargo el desempeño de dos roles: en una primera instancia desempeña el rol de desarrollador (ver Fig.1): posee *background* suficiente como para generar ejercicios, utilizando el sistema instruccional como herramienta de autoría, instanciando los contenidos y brindando semántica a las actividades que posteriormente le serán presentadas al alumno, enriqueciendo el sistema instruccional.

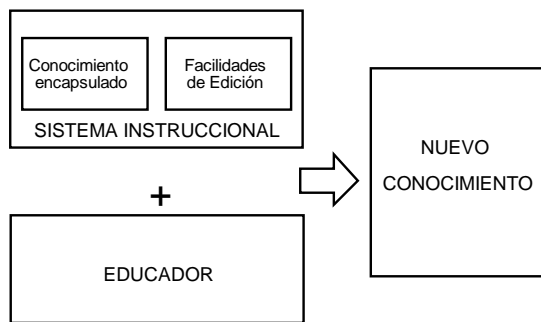


Fig.1: El Educador como "Productor"

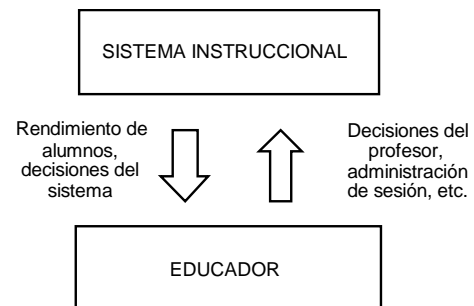


Fig.2: El Educador como "Usuario"

El otro rol que cumple el educador dentro del sistema instruccional, es el de usuario (ver Fig.2): durante la sesión de ejercitación, el sistema tutorial le informa acerca de las condiciones o acciones del alumno en las que se ha basado para tomar decisiones instruccionales. Con posterioridad al desarrollo de la sesión, el educador podrá efectuar un análisis de lo que ha sido el desempeño de sus alumnos, tanto en forma individual como agregada. De esta forma, el sistema tutorial es capaz de transmitir al educador la información relevante que ha adquirido luego de su interacción con los alumnos, y además el educador podrá aclarar las dudas del alumno que no puedan ser resueltas por el sistema.

Una ventaja implícita en el hecho de considerar explícitamente al educador como parte del sistema instruccional, está dada por la necesidad que éste experimenta por comprometerse en el desarrollo y empleo del sistema tutorial. [VanderM95] sugiere la existencia de un problema de compromiso relacionado con la utilización de software educativo al interior de las escuelas, donde resulta necesario convencer previamente al usuario intermedio (el educador) de su valor pedagógico, siendo difícil que tanto educadores como los mismos institutos educacionales se comprometan en su utilización. Por su parte, si se hace uso del editor de sistemas instruccionales, es el educador quien en última instancia puede organizar los contenidos según la forma que le resulte más conveniente. El sistema tutorial se convierte así en una herramienta que queda subordinada a sus necesidades.

Como una última consideración a este respecto, es posible señalar que lo que se pretende es poner a disposición del educador una nueva herramienta, no reemplazarlo. No es posible desconocer el hecho de que cualquier interacción con un educador humano, empleando el lenguaje natural en vez de un lenguaje artificial, constituye una situación más deseable que la interacción con una máquina; que el educador dispone de mucho más información de contexto que le permite evaluar las respuestas del alumno en forma más precisa. Pero tampoco se puede desconocer que la situación ideal "un educador - un alumno" no suele darse con frecuencia y que lo más probable es que un grupo de alumnos "compartan" un único educador.

El tercer y último perfil de usuario está dado por la presencia de un experto tanto en el dominio de aplicación en que se desea que el alumno aprenda, como en el empleo del sistema tutorial propiamente tal. Se ha considerado que este rol puede ser desempeñado por un tutor, educador o psicólogo educacional que haya sido instruido en el funcionamiento del sistema, y que conozca en profundidad el modelo subyacente en él. Este usuario tiene la facultad de efectuar modificaciones que alteran el funcionamiento interno del sistema, afectando indirectamente el sistema de reglas en que se basa el sistema tutorial. Estas modificaciones se efectuarán empleando un lenguaje de alto nivel, por medio de menús y gráficas, evitando así la necesidad de alterar el código fuente con el que se ha implementado el sistema tutorial.

2.3 Paradigmas Educativos que sirven de Base al Modelo

Resulta de interés efectuar un cuestionamiento respecto del paradigma educacional en que se basa el modelo propuesto [Cunning96]. Durante el último tiempo, se ha visto un vuelco desde aquellos paradigmas de tipo instruccionalista, hacia concepciones de tipo constructivista.

El paradigma instruccionalista centra el proceso de enseñanza en el educador, quien define el conjunto de conocimientos y habilidades, así como el orden en que éstos deben ser presentados a los alumnos. La enseñanza suele encontrarse desvinculada del contexto en que dichos conocimientos suelen ser aplicados en la vida cotidiana, lo que dificulta su comprensión y valoración por parte de los alumnos. Entre sus exponentes es posible mencionar a B. Skinner y R. Gagné.

El paradigma constructivista, por su parte, centra el proceso de aprendizaje en el alumno, el cual es expuesto a situaciones o ambientes de aprendizaje que le permiten lograr una construcción interna del conocimiento. Se espera que el ambiente de aprendizaje sirva de contexto y facilite en el alumno la integración de las nuevas situaciones con la experiencia previa que tiene al respecto. Se suele relacionar estos planteamientos con las investigaciones realizadas por J. Piaget y L. Vygotsky.

Tradicionalmente el diseño de sistemas instruccionales asume ambos paradigmas como antagónicos, por lo que parece ser necesario optar por uno de ellos a la hora de desarrollar un modelo. Sin embargo, en [Rosas96] se analiza la forma de integrar ambos paradigmas en un mismo modelo de sistema instruccional, para así potenciar las fortalezas individuales, procurando minimizar sus debilidades.

El modelo de desarrollo de software educativo propuesto por Rosas se basa en los planteamientos de Vygotsky, quien afirma que el desarrollo de estructuras de conocimiento supone la existencia de dos momentos diferenciables a lo largo del tiempo. El primer momento consiste en una exposición al contenido a ser aprendido, en que el significado de éste es dado externamente al alumno; mientras que el segundo momento está dado por una construcción interna de significados, a partir de la experiencia vivida por el alumno. Ambas etapas presentan características y requerimientos específicos, que hacen recomendable el empleo de un paradigma educacional por sobre el otro.

En nuestro sistema instruccional, se hace uso de ejercicios que son confeccionados y evaluados en tiempo real, de acuerdo con las circunstancias específicas del alumno que está empleando el sistema, siendo capaz de mediar con éste para así lograr el máximo aprovechamiento de la sesión de ejercitación. El control del proceso de aprendizaje se encuentra distribuido entre el alumno, el educador y el sistema. El alumno puede escoger las actividades que desea realizar, dentro de un subconjunto especificado por el educador, mientras que el sistema gradúa la complejidad de los ejercicios que le son presentados al alumno, en base a su desempeño. En un primer momento, el control es administrado principalmente por el sistema en forma externa al alumno, el que actúa bajo las indicaciones del educador. A medida que el alumno logra mayor experiencia en el empleo de los contenidos y conceptos, adquiere un mayor grado de control sobre el simulador, pudiendo dirigir la secuencia de aprendizaje de acuerdo con sus intereses particulares.

3. La Arquitectura del Modelo

En la presente sección, se plantean módulos conceptuales, y luego se realiza un proceso de refinamiento para detallar la arquitectura propuesta. Esta se explica a continuación.

3.1 Módulos Conceptuales

La siguiente arquitectura contempla el diseño general de sistemas instruccionales que administran diferentes roles (experto, educador, alumno), con reutilización de conocimiento mediante la edición de contenidos (del dominio).

Con estas consideraciones, se toman originalmente las ideas de *editar*, *entrenar* y *evaluar*. Como consecuencia de éstas ideas, surge luego la función de *registrar*. Se quiere entonces, una arquitectura que soporte un sistema instruccional flexible (editar), orientado al alumno (entrenar), que permita una retroalimentación al educador (evaluación), y que administre adecuadamente información relevante acerca del desempeño (registrar). Así, conceptualmente la arquitectura del sistema consta de estos módulos: *de Edición*, *de Simulación*, *de Evaluación* (de desempeño), y *Memoria del sistema*. La interacción entre éstos es indirecta (Fig.3), a través de la memoria del sistema. Estas se explican a continuación:

El Editor: Mediante este módulo, se permite al educador editar “actividades de entrenamiento”, brindando la semántica adicional que requieren las estructuras para convertirse en actividades, transfiriendo de esta forma su propio conocimiento del dominio del problema, al sistema. Este módulo debe ser lo suficientemente amigable al educador, y el lenguaje para capturar el conocimiento del educador debe ser gráfico y sencillo.

El Simulador: Módulo dirigido al alumno. En este módulo se lleva a cabo el aprendizaje en el sentido de ejercitación de destrezas. En [Self91] se señala que existen diferentes grados de control del aprendizaje, y que según Vygotsky, las alternativas que el sistema le ofrezca (al alumno), tienen que estar dentro de su ZDP⁴. En el modelo propuesto, el control del aprendizaje viene a estar repartido entre el sistema y el alumno mismo, quien dirige su ritmo de enseñanza de acuerdo a las alternativas presentadas por el sistema (en base al modelo del alumno). Las alternativas instruccionales presentadas al alumno son creadas en forma dinámica, dada la configuración realizada por el educador en base a las estructuras. La presentación de información es llevada a cabo en un contexto familiar al alumno.

El Evaluador: Presenta resultados respecto al desempeño de un alumno en particular o un grupo de alumnos, en forma gráfica para un mejor análisis, para una evaluación por parte del educador, o del experto.

La Memoria del Sistema: Aquí se mantienen almacenados el conocimiento del experto, y el conocimiento que aporta el educador. Además, sirve para la comunicación entre los módulos del sistema, mediante el almacenamiento de valores de parámetros del sistema, y se mantiene en constante consulta/actualización por parte de los módulos anteriormente mencionados.

⁴ Vygotsky define la ZDP como “la distancia entre el nivel real de desarrollo, determinado por la capacidad de resolver independientemente un problema, y el nivel de desarrollo potencial, determinado a través de la resolución de un problema bajo la guía de un adulto o en colaboración con otro compañero más capaz” (Citado en [Rosas96]).

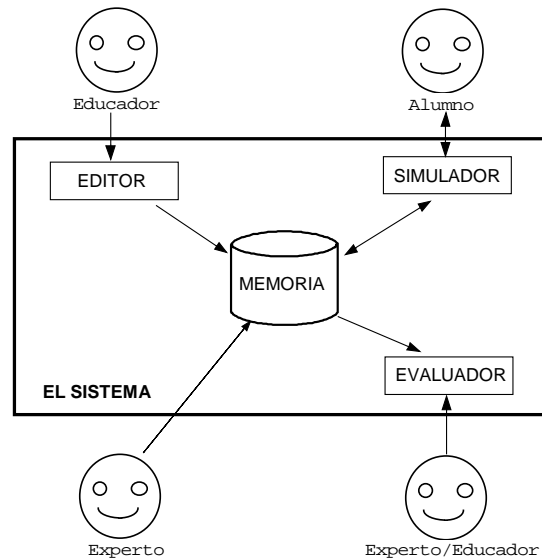


Fig.3: Módulos conceptuales

3.2 Arquitectura Propuesta

Cuando se refinan los módulos conceptuales para plantear una arquitectura, aparece de manera natural cierta funcionalidad hasta el momento implícita. Genéricamente, se ha hablado de las funciones de *editar*, *entrenar*, *evaluar* y *almacenar*. Pero ya más específicamente, el sistema instruccional que se plantea debe poder administrar diferentes *usuarios* (alumnos), y diferentes *sesiones* para cada uno de ellos. La memoria del sistema debe ser capaz además de recordar estos hechos. Asimismo, juega un rol importante la *interfaz* del sistema, pues debe ser intuitiva para el alumno, e informativa a su vez para el educador.

Otra función que esta implícita en el sistema, y que no se ha mencionado hasta el momento, es la concerniente al ajuste del sistema a un conjunto de alumnos. Esto se logra mediante el afinamiento de los valores de los parámetros del sistema, labor orientada al experto, quien en base al desempeño observado en la evaluación, debe ser capaz de modificar esos valores.

Si se agrupan todas estas funciones en los módulos conceptuales descritos, para definir especializaciones, se obtiene el siguiente esquema (ver Fig.4).

La comunicación entre los módulos del sistema (Simulador, Evaluador, Editor) se realiza principalmente de forma indirecta a través de la memoria del sistema. A continuación, se detallan las funciones agrupadas por módulo:

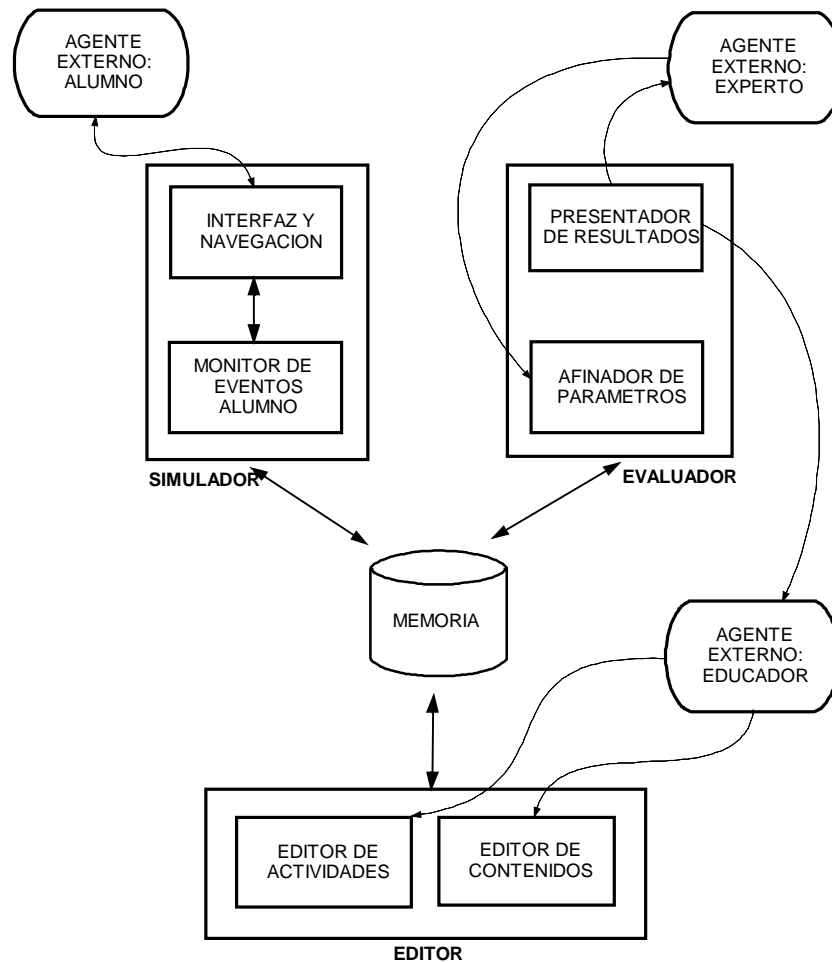


Fig.4: Los módulos del sistema y agentes externos

| | |
|---------------------|---|
| El Editor | • Almacenar <i>contenido</i> instruccional del educador. |
| | • Editar <i>actividades</i> , a partir del conocimiento encapsulado del experto (<i>estructuras genéricas</i>), con los contenidos dados por el educador. |
| El Simulador | • Administración de diferentes usuarios (alumnos) y sesiones por usuario. |
| | • Presentador de la interfaz y de la secuencia de presentación de contenidos (<i>navegación</i>). |
| | • Monitorear los eventos que realiza el alumno. |
| | • Tomar decisiones con la información de los eventos. • Guardar información histórica en la memoria del sistema. |
| El Evaluador | • Mostrar desempeño de un alumno, o de un grupo de alumnos, con distintas alternativas de presentación. |
| | • Permitir modificar valores de los parámetros para ajustar el desempeño del sistema hacia necesidades más específicas (función exclusiva del experto). |

Dentro de esta arquitectura, la parte inteligente del sistema instruccional, el sistema experto propiamente dicho, está dada por el componente que monitorea las acciones del alumno, toma decisiones, y ordena al resto del sistema qué acción tomar a continuación. En

el esquema de arquitectura propuesto, el sistema experto está constituido por los siguientes submódulos:

- Un monitor interactividades: Que analiza el comportamiento general del alumno y su evolución en el aprendizaje, al entrenarse en las diversas actividades generadas por el sistema. Esto lo logra mediante el empleo de reglas activas durante la sesión.
- Un monitor intraactividades: Que analiza el desempeño del alumno al interior de una actividad, también utilizando reglas, pero particulares por actividad, permaneciendo activas durante el tiempo de vida de la actividad. Esto corresponde al periodo desde que comienza la actividad, hasta que el alumno termina, o es interrumpida por éste o por el educador.

Interactuando con el sistema experto, al interior del simulador, está el módulo de I/O, que realiza el despliegue de interfaz, y la captura de acciones (eventos) que realiza el usuario (alumno). Los módulos inteligentes son los que graban información en la memoria del sistema. Entre las interacciones (ver Fig.5) que se realizan en el sistema experto con los otros componentes del simulador, se tienen las siguientes:

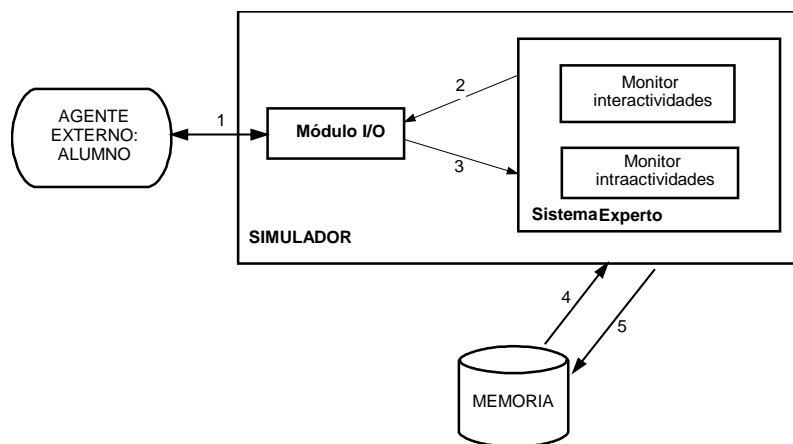


Fig.5: El Sistema Experto dentro del Simulador

1. La interacción entre el alumno y el módulo de I/O, en ambos sentidos, en la que se le presentan contenidos, y se reciben las acciones efectuadas por el alumno. Aquí parte del control lo tiene el alumno.
2. Las órdenes efectuadas por el sistema experto en su conjunto (es decir, por ambos módulos inteligentes) hacia el módulo de I/O, ajustando la información a presentar de acuerdo a las capacidades del alumno. Parte del control ahora lo tiene el sistema.
3. Las acciones del alumno que trasmite el módulo de I/O hacia el sistema experto, las que se analizan para tomar nuevas decisiones.
4. La información sobre sesiones anteriores del alumno, y su historia, que recibe el sistema experto de la memoria del sistema.
5. La información histórica y de desempeño del alumno que almacena el sistema experto en la memoria del sistema.

El sistema experto, y el módulo de I/O, son los que en conjunto en el esquema conceptual del inicio de la sección se les denomina como *Simulador*.

Otro aspecto del modelo, es que dado que un alumno puede desconocer cierto elemento del contenido, o no entender una actividad (del dominio en estudio), el sistema experto puede tomar la decisión de desactivar algunas de estas entidades (una actividad, o parte del contenido), para posteriormente ser habilitado en forma expresa por el educador.

4. Implementación del modelo

A continuación se describirá una aplicación del modelo de desarrollo de sistemas instruccionales a un dominio concreto de enseñanza: la generación de actividades de entrenamiento de habilidades matemáticas básicas, para niños en edad preescolar.

4.1 El Dominio Escogido

El modelo descrito anteriormente, si bien no impone a priori restricciones respecto del tipo de dominios que pueden ser apoyados, presenta ciertas características que hacen aconsejable su empleo para el entrenamiento de ciertas habilidades por sobre otras. Es así como su utilización es recomendada principalmente en aquellos dominios que hacen uso de ejercitación y práctica como un medio de consolidar los conocimientos adquiridos.

Aunque la tendencia actual apunta a dejar de lado la ejercitación como técnica de aprendizaje, ésta sigue siendo válida en el desarrollo de aquellas habilidades básicas que requieren respuestas automáticas, con el fin de poder construir sobre ellas otras habilidades de orden superior. De esta forma, las personas consiguen liberar su atención de estas labores triviales, pudiendo concentrarse en aquellos aspectos complejos del problema que desean resolver [Norman96].

Es así que para efectos de demostrar la aplicación del modelo, se ha escogido como dominio el desarrollo de habilidades matemáticas básicas, relacionadas con la adquisición del concepto de número en preescolares. Este dominio cumple con el requisito de emplear como una posible metodología de trabajo la ejercitación y práctica. El sistema tutorial, en consecuencia, ofrecerá a los niños actividades de entrenamiento, las que son definidas mediante la edición de conocimientos, utilizando instancias de los módulos definidos en las secciones anteriores.

Es reconocida la existencia de ciertas características propias de los niños en edad preescolar, a las que se puede apelar al diseñar planes instruccionales orientados a este grupo: el instinto de propiedad, el gusto por coleccionar objetos, el amor por la repetición, el deseo de observar y la necesidad de orden [Bandet68].

La utilidad de una herramienta computacional de apoyo que adecue la presentación de contenidos al ritmo de cada niño es avalada por la educadora [Mira89], quien señala que si bien el orden de sucesión de adquisiciones es constante, no necesariamente es posible definir una cronología asociada.

Existe bastante bibliografía respecto al uso de computadores como herramienta de apoyo en educación escolar. En particular, [Kafai94] realiza un análisis crítico de algunos software disponibles en el mercado para niños en edad preescolar (pertenecientes a lo que ella denomina la *Barney's Generation*). En [Moss96] se menciona el desarrollo de habilidades en niños mediante la simulación de la realidad (juegos), la instrucción asistida por computador y la preparación de los alumnos para entrar a una nueva era de innovación tecnológica.

4.2 Los Roles Instanciados

Con relación a los roles descritos previamente (alumno, educador y experto), dentro del contexto del dominio de las habilidades matemáticas básicas, se tiene que:

- los *alumnos* son niños en edad preescolar (de entre 3 y 6 años aproximadamente), que aún no saben leer ni escribir, pero tienen la habilidad psicomotora suficiente como para manejar un *mouse*. Este supuesto se confirma con observación visual.
- el *educador* es un profesor con conocimientos de educación preescolar. Él crea las actividades adecuadas para desarrollar en sus alumnos cierto conjunto de destrezas, acción que realiza por medio de estructuras. El educador acompaña al niño durante el transcurso de la sesión, pero no precisa dedicación exclusiva a esta tarea. También es

informado por parte del sistema, una vez concluida la sesión de ejercitación, respecto de los acontecimientos relacionados con el alumno, pudiendo efectuar análisis agregados, empleando la información del grupo curso.

- el *experto* es un psicólogo educacional o una persona con conocimientos acerca del dominio de las matemáticas básicas y del modelo de desarrollo de sistemas instruccionales.

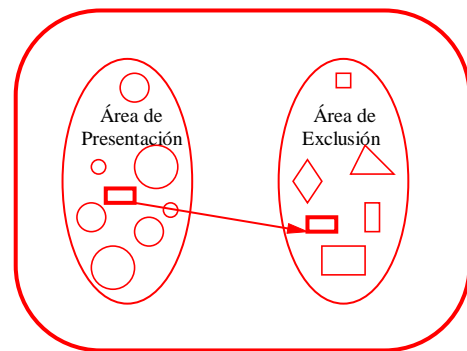
4.3 Ejemplo de Conceptualización de un Dominio Específico

Por restricciones relacionadas con la implementación fueron seleccionadas y descritas en detalle 3 estructuras, pudiendo definirse otras como una prolongación natural del trabajo.

Estructura 1: Excluir elementos que no pertenecen a determinada Categoría (nombre abreviado: Excluir en Categorías)

Descripción: Se presenta al alumno un conjunto mixto de elementos, entre los cuales se encuentran algunos que cumplen cierto criterio (pertenecen a la categoría base), mientras que el resto son distractores que el niño debe detectar.

Objetivo: Al niño se le indica cuál es el criterio que define la categoría base. El niño analiza uno a uno los elementos que son dispuestos en el área de presentación. Si el elemento pertenece a la categoría base, lo deja en el lugar, mientras que si es un distractor lo debe llevar al área de exclusión.



Parámetros: Número de elementos que pertenecen a categoría base y distractora, lista de categorías para escoger categoría base.

Ejemplo: Sacar los animales que no son marinos a la superficie del océano para que no se ahoguen.

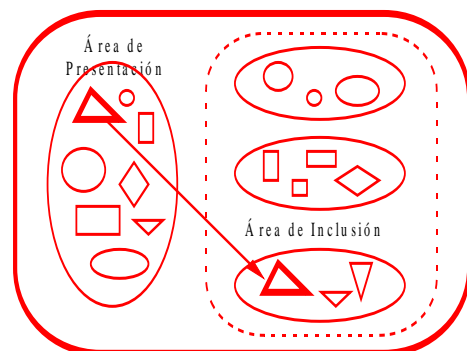
Estructura 2: Colocar elementos formando Categorías (n. a.: Colocar en Categorías)

Descripción: Se presenta al alumno un conjunto mixto de elementos, pertenecientes a varias categorías. Al niño se le da un ejemplo de cada una de las categorías.

Objetivo: El niño debe clasificar cada uno de los objetos que le han sido presentados, en alguna de las categorías disponibles.

Parámetros: Número de categorías a presentar, número de elementos de cada categoría, lista de categorías entre las que escoger.

Ejemplo: Ordenar abarrotos en el almacén, según forma y tamaño.



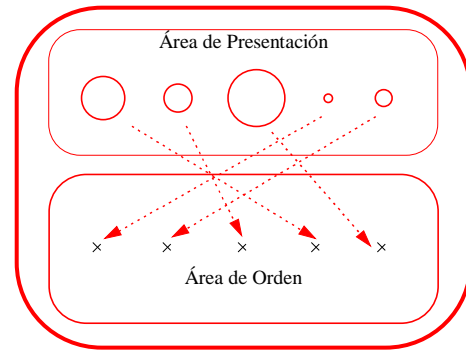
Estructura 3: Formar Secuencias Objetivas (n. a.: Formar Secuencias)

Descripción: Se presenta una serie de imágenes o cuadros que son susceptibles de ser ordenados en forma única según un criterio objetivo. Se indica además al niño la dirección del ordenamiento (p. ej.: de más grande a más chico).

Objetivo: El niño debe mover cada imagen para conformar la secuencia ordenada.

Parámetros: Largo de la secuencia, grado de desorden de los elementos a presentar, granularidad con que se escogen elementos de la secuencia, lista de categorías para escoger.

Ejemplo: Ordenar cronológicamente los cuadros que describen la caída de un florero al suelo (florero sobre la mesa, florero cayendo, florero en el suelo quebrado).



Un aspecto de particular importancia se refiere al manejo del control durante el desarrollo de la sesión. El alumno debe ser capaz de escoger las actividades de entrenamiento que le resulten más atractivas, de entre el conjunto de actividades que el educador ha especificado y puesto a disposición del sistema. Para abordar este aspecto, que es denominado “*navegación*” dentro del contexto de este informe, es posible utilizar como mecanismo auxiliar la definición de una nueva estructura, que como parte de su funcionamiento se encarga de presentar alternativas al usuario, en base al nivel de conocimientos alcanzado por éste, y de los recursos (actividades) disponibles en el sistema.

Tipos de Elementos con que Operan las Estructuras

Las estructuras operan sobre un conjunto de parámetros, uno de los cuales se refiere a los contenidos empleados por la estructura en la confección de los ejercicios. Del análisis de las estructuras antes descritas se determina la necesidad de contar con tres tipos de elementos u objetos diferentes: los elementos simples, y las secuencias de elementos.

1. Las *categorías de elementos simples* están compuestas por conjuntos de objetos, cuya naturaleza es una imagen, donde el contenido visual determina el elemento que se quiere representar (p.ej.: una gallina, un pollito, un conejo, un pato, un ganso, un chanco y un caballo pertenecen a la categoría *animales de la granja*), y una descripción de los mismos (que suele corresponder al nombre del elemento: “*gallina*”).
2. Las *secuencias de elementos* son categorías que agrupan series de elementos cualitativamente similares, los cuales poseen un orden relativo según algún criterio cuantitativo (p.ej.: una secuencia de botellas de menor a mayor tamaño). La naturaleza de estos elementos también está conformada por una imagen, y una descripción que es común a toda la secuencia (p.ej.: “*secuencia de botellas según tamaño*”).

4.4 Los Módulos Instanciados

Desde el punto de vista de la arquitectura del sistema, para el dominio escogido, se definen los siguientes módulos en la implementación del modelo:

- Un módulo configurador de categorías de elementos, denominado *Clasificador* (ver Fig.6), cuyo usuario es el educador. Su misión es organizar en la memoria del sistema (la base de datos), las categorías simples y secuencias. Su interfaz es sencilla e intuitiva, manteniendo independencia con respecto a las estructuras.
- Un módulo editor, que se llamó *Editor de Actividades* (ver Fig.7), en el cual el educador

puede instanciar una estructura brindándole contexto a través de un enunciado y un fondo, el que luego es asociado con un conjunto de tipos de datos, para definir así las actividades.

- Un módulo llamado *Simulador*, encargado de presentar las actividades definidas por el educador al niño. Éste efectúa la administración de I/O y el sistema experto, realizando además de la autorregulación, la creación de un entorno personalizado para cada alumno.
- Un módulo *Habilitador*, mediante el cual el educador activa elementos, categorías de elementos y/o actividades que previamente han sido deshabilitadas por el sistema, cuando éste estima que el niño ya es capaz de entenderlos. Este módulo no tiene un equivalente explícito en el modelo conceptual.
- Un módulo de *Análisis de Resultados*. Bajo éste módulo, tanto educador como el experto, pueden analizar el desempeño de un niño o un grupo de niños.
- El módulo de *Afinamiento*, cuyo acceso se encuentra restringido al experto, quien puede ajustar los parámetros del sistema para realizar una mejor adecuación de acuerdo a las necesidades particulares del grupo de niños en entrenamiento.

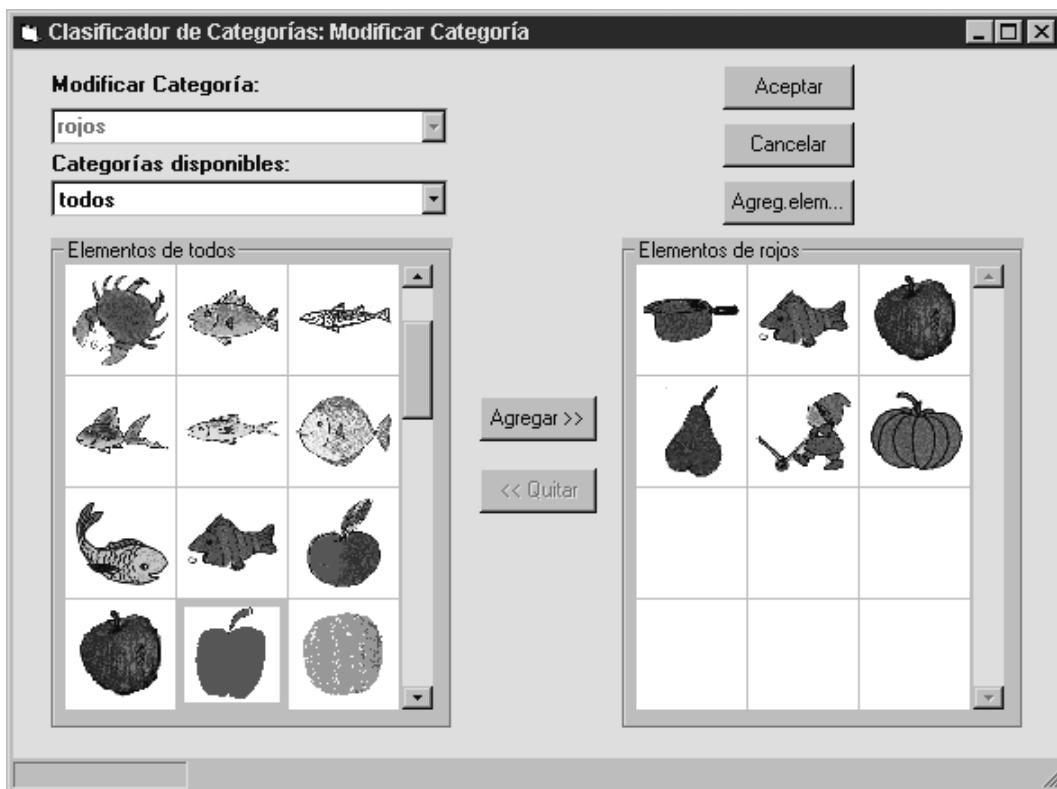


Fig.6: Ejemplo de modificar Categoría

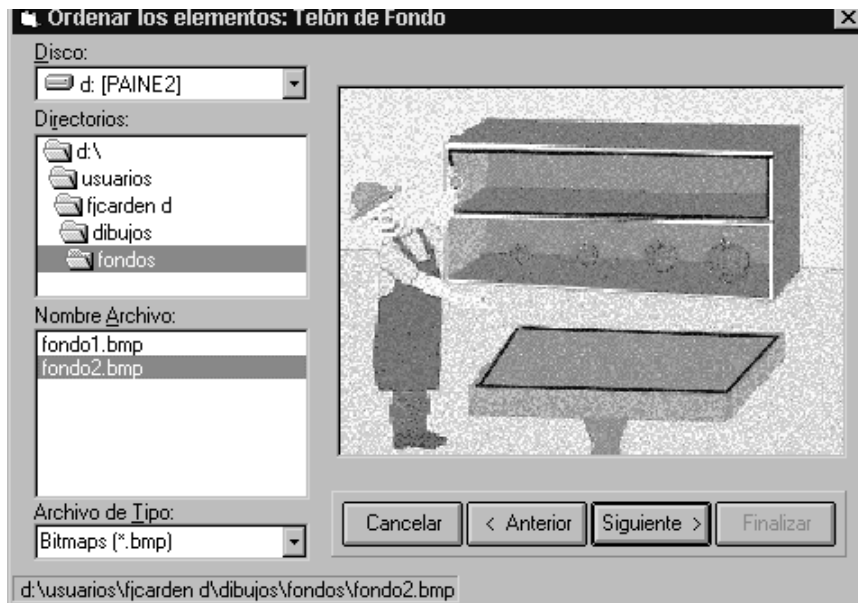


Fig.7: Elección telón de fondo en la Edición de Actividades

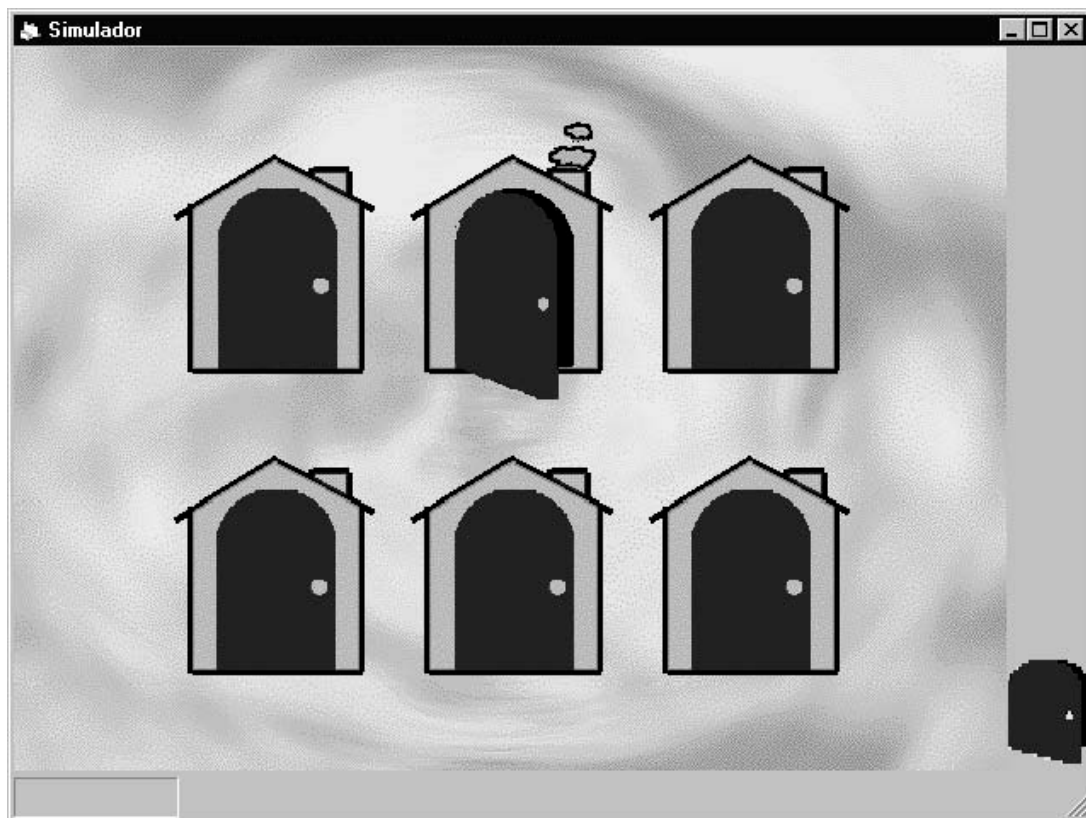


Fig.8: Simulador, pantalla de Navegación.

4.5 Descripción de Procesos en el Sistema

Los procesos se inician con la edición (ver Fig.9). El educador dispone de un *Ambiente de Edición* conformado por los módulos *Clasificador* y *Editor de Actividades*. El *Clasificador* es empleado para ingresar al sistema el contenido instruccional bajo la forma de categorías y secuencias. Aquí el educador incorpora imágenes y sonidos⁵, dándole

⁵ La edición de imágenes y sonidos se realiza en forma externa al sistema.

descripciones adecuadas. Posteriormente, el educador utiliza el módulo *Editor de Actividades*, donde crea actividades para el entrenamiento del niño. Esto lo realiza escogiendo una *estructura* de funcionamiento (de entre las disponibles en el sistema, que previamente han sido conceptualizadas por el experto). El editor le solicita un nombre apropiado, una imagen que sirva de contexto (fondo), un enunciado para la actividad, y una lista de categorías de elementos, de entre las existentes en el *Clasificador*. El educador puede crear prácticamente infinitas actividades a partir de las estructuras proporcionadas por el sistema, limitadas sólo por espacio.

Una vez definidas algunas actividades, ya es posible llevar a cabo la ejercitación por parte del niño. Para esto, el educador debe iniciar una sesión de trabajo en el *Simulador*, identificando al niño. Si el niño ha empleado con anterioridad el sistema, éste carga la historia disponible respecto de ese niño; en caso contrario, inicia un nuevo registro para el alumno. En caso de existir alguna nota recordatoria para el educador (por ejemplo, indicación de un elemento deshabilitado), ésta es mostrada al educador, quien puede emplear el módulo *Habilitador* para efectuar ajustes.



Fig.9: Relación de Usuarios y Módulos en el Sistema

Luego comienza la sesión de trabajo. Se le presenta al niño una pantalla que lo sitúa en un contexto que le resulte familiar, dándole la oportunidad de explorar. En ella hay iconos o botones que actúan como “puertas” hacia “lugares” donde le serán presentadas actividades, de acuerdo con su nivel de desarrollo. Una vez que el niño ingresa a un lugar (utilizando el mouse), se despliega un breve enunciado que explica al niño en qué consiste el “juego”. Ahora toca el turno de actuar al niño, quien puede mover objetos, hacer *click* en botones, dependiendo del tipo de actividad. Sus acciones son permanentemente registradas y evaluadas por el *Simulador*, el que toma decisiones empleando las reglas y los valores vigentes de los parámetros. De esta forma los ejercicios son evaluados, se activa o desactiva la ayuda, se modifican los valores de los parámetros y se generan mensajes al educador. Durante la sesión, el sistema hace uso de una porción de la pantalla donde despliega mensajes al educador, sin distraer al niño. Una vez que por decisión del sistema, el educador o el mismo niño se sale de una actividad, se regresa al contexto de navegación antes mencionado, donde además existe la opción de terminar la sesión. En caso de ocurrir el término de la sesión, el sistema despliega al educador un resumen de lo que ha sido la sesión de trabajo, informándole los hechos más relevantes que han ocurrido (consecución de alguna destreza, deshabilitación de algún ítem, etc.).

Con posterioridad al desarrollo de la sesión, el educador puede hacer uso del módulo de *Análisis de Resultados*, empleando la información contenida en la base de datos del sistema para evaluar el nivel de logro alcanzado por un alumno o un grupo de alumnos.

El experto también puede hacer uso del módulo de análisis, pero esta vez con la finalidad de diagnosticar comportamientos erróneos al interior del sistema (por ejemplo niveles de exigencia demasiado altos o bajos, tiempos de espera muy cortos, etc.). En caso de detectarlos, puede introducir modificaciones en los parámetros internos del sistema, como una forma de “calibrar” su funcionamiento, haciendo uso del módulo de *Afinamiento*.

4.6 La herramienta a utilizar

[Davies] menciona los tipos de herramientas de software para desarrollar CBL (Computer Based Learning). Los clasifica en dos dimensiones: en sus requerimientos de autoría (programación/no programación) y estilo de desarrollo (script/no script). Dentro de esta clasificación, la herramienta de autoría elegida para desarrollar el sistema es Visual Basic, que como se aprecia en la Fig.10, tiene un nivel de programación y uso de scripts. Es un lenguaje de programación visual, y con este enfoque, la programación convencional se combina con técnicas de autoría basadas en forms (formularios) para proveer un ambiente de desarrollo poderoso y accesible, en ambiente Windows-PC.

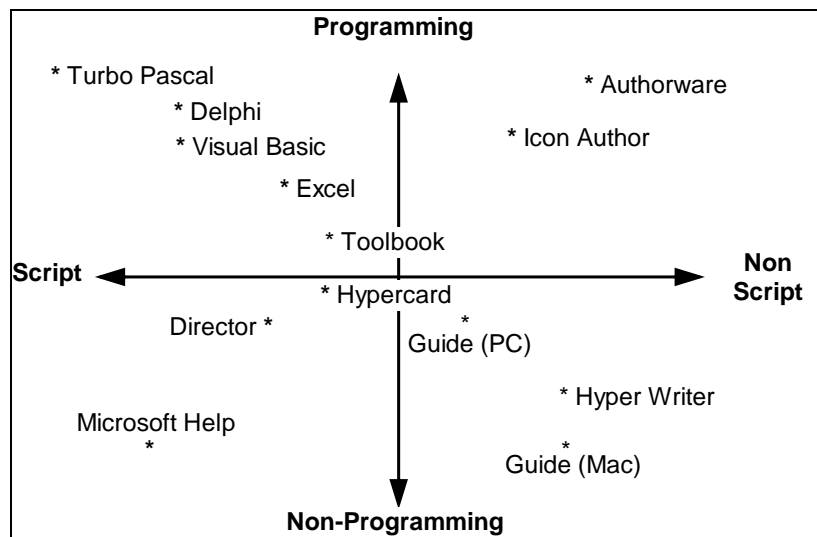


Fig.10: Clasificación de herramientas de autoría

VB es un lenguaje basado en eventos, y parcialmente orientado a objetos, y tiene una buena librería de controles y facilidades para el diseño de CBL, como excelentes capacidades gráficas y de multimedia, así como la funcionalidad de bases de datos. Estas características, unidas al hecho de que la plataforma (PC) se ha vuelto de bajo costo, disponible en muchos colegios, es que se ha elegido VB 4.0 en su versión de 32 bits (para Windows95), dado su relativo menor costo comparado con otras herramientas. Windows95 se ha vuelto a su vez un entorno gráfico y sistema operativo muy popular. Como memoria del sistema, se ha elegido una base de datos en MS Access, que es la base de datos nativa de VBasic.

La Interfaz

El sistema, tiene tres tipos de usuarios, de los cuales, uno (el alumno), es un caso muy particular. El caso de los niños en edad preescolar, impone ciertas restricciones de diseño [Piernot], ya que:

- Los niños no son capaces todavía de leer.
- Los niños tienen un razonamiento simbólico limitado.
- Los niños tienen habilidades de planificación aún limitadas, pudiendo tener problemas

llevando a cabo tareas que requieran varios pasos.

- Los niños tienen un rango de atención menor que los adultos.

En [Rosas96] y [Piernot], se menciona que la interfaces para el caso de niños debe ser lúdica, de tal forma que el contenido instruccional sea entretenido, apreciado por el niño bajo la forma de juegos. Para el presente sistema, se adoptan ambas ideas, una interfaz diferenciada para el profesor y el psicólogo, de lo que es para el alumno. Mientras que para los primeros se definen ventanas y cuadros de diálogo, con el uso tanto del mouse como de teclado, para el último (el alumno: niño) se adopta una interfaz en la que se presentan las actividades como juegos, con una presentación multimedial entretenida. En [Sealy95], se mencionan la habilidad del empleo de multimedios para representar situaciones de la vida real mejor que cualquier otro medio.

Requisitos de Infraestructura y Capacitación

La utilización de una herramienta como la descrita en el presente artículo, no demanda mucha capacitación ni grandes despliegues técnicos, considerando esos puntos como una fortaleza. Se requiere tener instalado un PC con mouse y tarjeta de sonido. Como sistema operativo, Windows95[®]. Éste requiere que el PC sea un 486 o superior, con al menos 8 megabytes de memoria RAM. Por otro lado, la herramienta requiere un espacio libre en disco suficiente como para albergar el banco de imágenes, siendo un estimado razonable 20 megabytes. Ésta es una configuración no muy exigente en estos días.

En el aspecto humano, durante la operación normal del sistema, se asume la presencia de un educador (o varios) encargado de las sesiones, pero no se requiere su dedicación en forma exclusiva a un alumno, sino la supervisión del ambiente completo por turnos. Adicionalmente, resulta conveniente contar con un dibujante o diseñador, quien proporcione al educador un banco de contenidos (imágenes), para la confección de actividades. Finalmente (en forma opcional, aunque deseable), se requiere que un educador o psicólogo educacional se interiorice en el funcionamiento del sistema para llevar a cabo el ajuste de parámetros.

El uso del software por parte del niño sólo requiere que éste tenga la capacidad motriz suficiente para manejar el mouse. En el caso del educador, supone conocimientos generales no como programador, sino como usuario, sobre todo en el entorno Windows: saber iniciar Windows, hacer clicks a iconos, efectuar selecciones en menús, etc. Adicionalmente, el educador debe familiarizarse con el empleo de la herramienta.

Por este motivo, durante el periodo de introducción de la herramienta en el ámbito escolar, al igual que ocurre con cualquier producto de software desarrollado para ser usado en una organización, debe llevarse a cabo la puesta en marcha del sistema. Ésta contempla, en primer término, la instalación del software y la base de datos empleada por éste. Paralelamente, debe efectuarse la capacitación de educadores en el funcionamiento del sistema, abordando específicamente la forma de editar actividades, agrupar y emplear contenidos, intervenir durante el desarrollo de una sesión y efectuar análisis de los resultados obtenidos por los alumnos. Para esto, resulta adecuado contar con un manual que sirva como referencia y esté a disposición de los usuarios.

Como parte de una política de mantenimiento, el sistema debe ser objeto de evaluación durante su operación normal. Ésta es responsabilidad de aquel educador o psicólogo especializado en el sistema, el cual requerirá algunas sesiones adicionales de instrucción. Para servir de apoyo a esta tarea, es recomendable elaborar un manual específico que instruya sobre los parámetros internos del sistema y la forma en que pueden ser intervenidos, para obtener modificaciones en el comportamiento de la herramienta.

5. Conclusiones

En base a la experiencia adquirida durante el presente trabajo, es posible mencionar

las siguientes conclusiones:

Se ha observado que mientras mayor es la cantidad de conocimiento encapsulado específico al problema que se está intentado resolver, menor es el aporte que se requiere por parte del educador, por lo cual se simplifica el proceso de transmisión de información. Por otro lado, si el conocimiento encapsulado es mínimo, el educador debe aportar todo el conocimiento que no está presente, aumentando la cantidad de cosas que tiene que hacer. En el caso del prototipo, se optó por dejar al educador las labores más importantes, delegando algunas acciones para que las realice el sistema, tratando de conseguir un balance adecuado entre flexibilidad y facilidad de uso.

Respecto a la modelación del conocimiento del dominio, durante la investigación fue posible conceptualizar más estructuras de las que han sido descritas, lo que hace suponer que aún es posible extender el conjunto de estructuras que conforman el sistema experto.

También cabe destacar otros aspectos. Uno es el hecho de utilizar la potencia de combinar los multimedios con los sistemas expertos. El atractivo de los multimedios brinda una dimensión que no se consigue con lápiz y papel, y los sistemas expertos otorgan autoregulación necesaria para un aprendizaje más personalizado. Otro punto importante es lo relacionado al educador como productor de conocimiento utilizando editores de alto nivel.

Finalmente, adoptamos un concepto que no es nuevo: el hecho de la integración de aspectos instruccionistas y constructivistas en un mismo sistema instruccional.

6. Trabajo Futuro

Con el desarrollo del presente sistema, se trata de demostrar la posibilidad que con el modelo propuesto, un educador pueda crear actividades de entrenamiento sin conocer lenguajes de programación, siendo esta implementación una herramienta útil. Para demostrar esta hipótesis, es necesario poner a prueba el sistema. Sin embargo, debemos reconocer que quedan cosas por hacer.

En primer lugar, tenemos que recordar que la presente implementación es un prototipo, y no pretende ser un producto final, por lo que está sujeto a ampliación, revisión y depuración. Así, existen dos módulos que están considerados en el modelo pero no en la implementación; uno es el referido al afinamiento de parámetros, donde el experto podrá calibrar el sistema de acuerdo a cómo se vaya evaluando el desempeño del mismo, y el otro es un módulo de análisis de resultados. Por el momento, ambas labores requieren del auxilio del experto y de un programador.

7. Referencias

- [Arruarte96]. Arruarte, A.; Elorriaga, J.A.; Fernández-Castro, I.; Ferrero, B. "Knowledge Reusability: Some Experiences in Intelligent Tutoring Systems". Position Paper for ITS'96 Workshop on Architectures and Methods for Designing Cost-Effective and Reusable ITS's., 1996.
<http://advlearn.lrdc.pitt.edu/its-arch/papers/arruarte.html>.
- [Bandet68]. Bandet, J.; Mialaret, G.; Brandicoutrt, R.; "Los Comienzos del Cálculo", Buenos Aires: Kapelusz, 1968.
- [Blument96]. Blumenthal, R.; Meiskey, L.; Dooley, S.; Sparks, R. "Reducing Development Costs With Intelligent Tutoring Systems Shells". Position Paper for ITS'96 Workshop on Architectures and Methods for Designing Cost-Effective and Reusable ITS's., 1996.
<http://advlearn.lrdc.pitt.edu/its-arch/papers/blumenthal.html>

- [Cunning96]. Cunningham, D.; "Differentiate between instructionism, constructivism and constructionism" Del curso "Learning and Cognition in Education - P540", http://education.indiana.edu/~cep/courses/p540/p540_1.html, 1996.
- [Davies]. Davies, P.; Brailsford, T.; "Development Tools". Bio-Informatics Research Group, Department of Life Science, University of Nottingham.
- [Kafai94]. Kafai, Y. y Soloway, E. "Computational Gifts for the Barney Generation". *Communications ACM*, Vol. 37, N° 9, pp.19-22, Septiembre 1994.
- [Mira89] Mira, M.; "Matemática Viva en el Parvulario", Barcelona: CEAC, 1989.
- [Moss96]. Moss Point High School. "Computers in Education", Octubre 1996. <http://www.mphs.edu/internet/final.html>
- [Norman96] Norman, D. & Spohrer, J. (1996, Abril) Learner-Centered Education. *Communications of ACM*, Vol. 39, N° 4, pp. 24-27.
- [Piernot]. Piernot, P.; Felciano, R.; Stancel, R.; Marsh, J.; Yvon, M. "Designing the PenPal: Blending Hardware and Software in a User-Interface for Children". Stanford University.
- [Rickel89]. Rickel, J. "Intelligent Computer-Aided Instruction: A Survey Organized Around System Components". *IEEE Transactions on Systems, Man and Cybernetics*, Vol. 19, N° 1, pp. 40-57, Enero / Febrero 1989.
- [Rosas96]. Rosas, R.; Strasser, K.; Nussbaum, M.; Buzeta, R.; Moulian, M. "Enseñando al Homo Ludens: Propuesta de un Modelo para el Diseño de Sistemas Instruccionales Lúdicos Autorregulados". Pontificia Universidad Católica de Chile, 1996.
- [Sealy95]. Sealy, B. and Phelan, P. "Pedagogy for Multimedia Design", Paper presented at the conference Networked Multimedia: the Interactive Future, INESC, Portugal 1995. <http://yang.inescn.pt/peoplephelan/pedagogy.html>
- [Self91]. Self, J. and Cumming, G., "Learner Models in Collaborative Intelligent Educational Systems". AAI/AI-ED Technical Report No.55 Computer Department Lancaster University, 1991.
- [VanderM95]. Van der Mast, Ch. "Developing Educational Software: Integrating Disciplines and Media". Thesis Technische Universiteit Delft, 1995.