

## PROYECTO 2.4.: Editor e Intérprete de Diálogos

**Responsable:** Tine Stalmans

**Colaborador:** Joel Espinosa Longi

### 1. Reporte de implementación del editor de diálogos interactivos

#### v\_0.0.1

- Desarrollo inicial donde se explora la carga y guardado de archivos, con las herramientas que brinda HTML 5
- Se puede abrir, editar y guardar el contenido de un archivo XML

#### v\_0.0.2

- Incorporación de la biblioteca AlloyUI para el manejo de la interfaz
- Se agregó un editor de textos más completo para editar el contenido de los XML
- Se agregó una mejor interfaz para la creación de documentos nuevos, abrir y guardar archivos

#### v\_0.0.3

- Estructuración del código en clases y archivos separados para mejorar el desarrollo y mantenimiento

#### v\_0.0.4

- Cambio de la biblioteca AlloyUI por JQuery, ya que AlloyUI es más extensa de lo necesario, mientras que JQuery tiene las características necesarias
- Sustitución del editor de textos por una área para la edición de los diálogos de forma gráfica
- Se pueden abrir diálogos y mostrar preguntas y los atributos del dialogo

#### v\_0.0.5

- Prueba del plug-in *layoutengine* de JQuery. Para crear el layout de la interfaz grafica del editor

### v\_0.0.6

- Se sustituyo el plug-in *layouengine* y simplemente se usan reglas de CSS para el layout de la interfaz grafica del editor
- Se muestran los atributos del dialogo y las preguntas, permitiendo su modificación

### v\_0.0.7

- Cambio en el diseño gráfico, principalmente los iconos de los botones
- Se muestran los atributos del dialogo, las preguntas y las respuestas, permitiendo su modificación
- Para las preguntas se muestra en forma de tooltip, el texto de la pregunta
- Para las respuestas se muestra en forma de tooltip, el texto de la retroalimentación
- En el atributo de `texto` en las preguntas y el atributo `retroalimentacion` en las respuestas, se presenta un editor de texto enriquecido para su edición
- Se modifiko el código para guardar en el XML entidades HTML

### v\_0.0.8

- Reescritura y reestructuración del código
- Cambio del uso de XML como formato de archivo a JSON, ahora abre y guarda diálogos en JSON
- Modificación de los nombres de los atributos al inglés para estandarizarlos
- Se agrego la funcionalidad de agregar y borrar respuestas

### v\_0.1

- Cambio en la estructura del archivo de definición del dialogo, en lugar de usar un archivo JSON real se utiliza un archivo JavaScript, donde se le asigna a la variable `dialogo` la configuración
- Corrección en la posición de las preguntas, ya que ignoraban la posición del scroll
- Adición de un panel para los micromundos



## v\_0.2

- Realizando la primera incorporación con node-webkit, para la creación de una aplicación de escritorio
- Incorporación de una barra de menú para node-webkit

## v\_0.3

- Cambio del editor de textos de JQuery-TE a ckeditor

## v\_0.4

- Agrupación de atributos para mantener una mayor coherencia
- Cambio en el diseño para que sea más atractivo y funcional
- Adición del botón agregar pregunta a la barra de herramientas
- Adición del botón agregar micromundo a la barra de herramientas
- Corrección en el tipo de datos del atributo `nrRequireResponses`
- Se cambio el nombre por omisión de un nuevo dialogo
- Se quito el botón `close` de la barra de herramientas
- Se redujo el tamaño de letra para ajustar las dimensiones y desplegar de forma más efectiva la información
- Las preguntas se puede colapsar
- Cuando se agrega una pregunta ya no se agrega en el origen, si no que se agrega respecto al desplazamiento del scroll
- Al mostrar el identificador de la pregunta aparecía una letra `P` mayúscula, se elimino esto
- Los mensajes de confirmación mostrados con la función JavaScript `confirm` se sustituyeron por paneles modales de JQuery
- El botón agregar respuesta ya tiene apariencia de botón
- Cuando una respuesta no tenía una clave corta aparecía el texto `without_shortkey`, ahora ya no aparece nada
- Se agrego el nuevo atributo `description` a las respuestas
- Se agrego el nuevo atributo `toNextQuestionOnClose` a los micromundos
- Se agrego el nuevo atributo `deactivateChat` a los micromundos
- Se quitaron los tooltips de las preguntas y respuestas
- Se cambio el campo de texto de estilo `globos` por un menú
- Los campos de texto que corresponden a valores numéricos se cambiaron por `spinners`

- Se agruparon varios atributos en los paneles de atributos
- Se cambiaron varios textos de presentación en algunos atributos
- Se agregaron los vínculos entre respuestas y preguntas

## v\_0.5

- Adición del editor de expresiones radial (*ver 1b abajo*)
- Adición del atributo `firstQuestion` como atributo del dialogo

## v\_1.0

- Adición de las restricciones de activación de variables en la interfaz gráfica
  - El atributo `audio` en preguntas y respuestas solo está habilitado cuando el atributo `withAudio` esta activado y el atributo `tts` esta desactivado
  - Al estar sin activar el atributo `withAudio` el atributo `tts` está inhabilitado
  - Cuando el atributo `nrRequiredResponses` tiene su valor por omisión (1) los atributos `prompt` y `nextQuestion` están inhabilitados
  - Cuando el atributo `front` del micromundo no está activado los atributos `autoDisplay` y `toNextQuestionOnClose` deben estar inhabilitados
- Corrección en el guardado del valor del atributo `weight`
- Corrección en el guardado del valor del atributo `toNextQuestionOnClose`
- Ahora se muestra el valor del atributo `description` de las respuestas en lugar de la clave corta
- Se sustituyeron los campos de texto de los atributos `nextQuestion` y `firstQuestion` por menús cuyas opciones son los identificadores de las preguntas



## 1b. Reporte de implementación del editor de expresiones RADial

### Julio 2013

Primera versión del editor donde las expresiones lógico-semánticas se construyen mediante la adición progresiva de operadores y claves que aparecen con los paréntesis requeridos.

Funcionalidad:

- Código de color para distinguir el nivel de anidación de los paréntesis.
- Los operadores se agregan mediante botón, las claves se agregan escribiendo. Una clave se puede cambiar simplemente haciendo clic en ella y escribiendo la corrección.
- Cada clave semántica tiene una flecha abajo para desplegar una lista con todas las palabras y expresiones relacionadas (conexión con DB semántica en servidor arquimedes) en la cual se seleccionan los lemas que se deben incluir en la expresión extendida.
- Se pueden escribir lemas o frases que no están en la lista para incluirlos en la expresión lógico-semántica.
- A partir de la expresión sencilla y los lemas seleccionados o agregados en la lista, se genera de manera automática una expresión extendida.
- Botón *undo* y *redo*.

### Agosto 2013

- Posibilidad de visualizar la expresión lógico-semántica como árbol.

### Septiembre 2013

- Corrección bug: Al insertar elementos en una expresión, las flechas posteriores para consultar el diccionario no se colocaban bien (no se recorrían).
- Agregar expresiones fijas SÍ, NO y DUDA.
- Posibilidad de borrar segmentos de la expresión.
- Posibilidad de cambiar un operador con un clic.
- Implementación de restricciones inherentes de algunos operadores:
  - CAU: sólo puede unir 2 expresiones
  - NEG: sólo puede tener una expresión simple entre sus paréntesis (sin ningún otro operador)
  - Al agregar operador detrás de parentesis de cierre, en caso de ya haber un operador a ese mismo nivel de paréntesis, sólo se activa ese mismo operador en el menú

- Diferenciar entre distintas instancias de una misma palabra (en lista semántica que se despliega al hacer clic en la flecha correspondiente)
- Adición de un botón y correspondiente espacio para poner a prueba la expresión extendida.
- Reconstrucción de la expresión en el editor (y las correspondientes listas con palabras seleccionadas y escritas) a partir de la expresión corta y extendida guardada en dialogo.json
- Diferenciar si las palabras agregadas son palabras simples, locuciones fijas (con operador \_), expresiones compuestas (operador AND) o antónimos (operador NEG)

### Noviembre 2013

- Incorporación de módulo de comunicación con el editor de diálogos
- Cambios gráficos en la interfaz para adecuarlo al diseño del editor de diálogos
- Adición de un loader para indicar que se está reconstruyendo la expresión y las listas correspondientes
- Adición de un botón para editar manualmente la clave extendida



## 2. Reporte de desarrollo del intérprete de diálogos (herramienta de ejecución)

*Nota: se trata de un trabajo de depuración y ampliación del código existente.*

### Julio 2013

- Inclusión del parámetro “peso” en los nodos de respuesta, y los parámetros “respReq”, “prompt” y “proxPreg” en los nodos de pregunta para aquellas preguntas que posiblemente requieren de más de una respuesta (parcial) para poder seguir a la siguiente pregunta.
- Posibilidad de que los globos del chat sean imágenes con 9-scale-grid (atributo “fondoGlobos=img/css” en nodo principal del xml)
- Nuevo atributo “textoLiga” en nodos de micromundos, para poder especificar el texto que debe aparecer en la liga para volver a visualizar cada micromundo.

### Agosto 2013

- Script “getLemas.php” que se llama una sola vez para obtener los lemas de las palabras (con 1er nivel de corrección de palabras inexistentes), antes de realizar el parseo semántico.

### Septiembre 2013

- Traducción de la estructura XML del diálogo y las definiciones a JSON.
- Homogeneización de los atributos (todos en inglés) y su tipo.
- Ajustes correspondientes en el código del intérprete y en el script php para la creación del diccionario local (ver apartado 3 abajo)
- Otras adiciones y correcciones
  - Posibilidad de poner un ícono en el botón de cerrar de la ventana de los micromundos que aparecen encima del *chat*.
  - Corrección del error al pulsar ENTER 2 veces seguidas sin escribir nada en el campo de escritura.
  - Colocar la barra de scroll del chat hasta abajo cada vez que cambie el micromundo o cuando se agregan opciones múltiples a un globo del tutor.
  - Aviso cuando clave RADial tiene errores.
  - Hacer que la lista de opción múltiple se despliegue en el globo correcto, aun cuando el alumno escribe una respuesta que no se reconoce (corrección de error en código).
  - Desactivación del campo de escritura al finalizar el diálogo (mediante atributo “endDialogue=true”)

- En caso de que `withoutServer=true & tts=true` → checar si audio está disponible (por si el diálogo corre como `file://...`)

### Octubre 2013

- Agregar botón *Apagar/Activar audio* en caso de que se optó por escuchar audio (con tts o pregrabado)
- Nuevo atributo *toNextQuestionOnClose* para micromundos con `front=true`. Sirve para saltar de inmediato a la siguiente pregunta al cerrar el micromundo, sin necesidad de que el usuario escriba algo.
- Nuevo atributo *deactivateChat* para micromundos con `front=false`. Sirve para desactivar el campo de escritura en el chat mientras se muestra el micromundo en cuestión.
- Inclusión de ligas de texto que pueden llevar a micromundos (nuevos o ya vistos) mediante la función “*showMicroWorld*”
- Hacer que reconstrucción sólo se ofrece si NO se llegó al final del diálogo en la sesión anterior. Tomar en cuenta nuevo atributo *toNextQuestionOnClose* en la reconstrucción.
- Otras correcciones a raíz de pruebas con iOS y Android:
  - Programación de un componente de scroll para actualizar correctamente la posición de scroll del chat en Android
  - Cambios en archivos `index.html` y `dialogo.css` para uniformar el acomodo del campo de escritura en IE y demás browsers
  - Corrección problema de que se encimaban micromundos al visitar un micromundo en el momento en que se agregaba otro nuevo.

### Noviembre 2013

- Adición de atributo “`reset=true/false`” a micromundos para reiniciar micromundos al volverlos a visualizar

### Diciembre 2013

- Diferenciar entre rutas absolutas y relativas a los micromundos



### 3. Reporte de desarrollo de algoritmos lingüísticos (morfológicos y semánticos) para el intérprete de diálogos

#### Julio 2013

- Programación de un algoritmo que recursivamente busca todas las expresiones equivalentes a partir de un lema de input y los incluye en la lista de palabras relacionadas en el editor de expresiones lógico-semánticas. Creación de una base de datos “exprEq” con dichas expresiones (ésta se alimentará manualmente por los usuarios a través del editor de expresiones)
- Reprogramación del algoritmo de parseo de expresiones lógico-semánticas para permitir nodos con más de 2 *childnodes* (árbol NO binario). Esto para posteriormente permitir un despliegue visual más claro del árbol y hacer más sencillo el análisis de los operadores que requieren de medición de posiciones (MAS, CAU y SEG)
- Adición de una dimensión al array *\$raices* para permitir una medición más precisa de la posición de las palabras en la oración.
- Corrección en el operador “MAS” (se consideraba la posición de palabras que no aportaban al match=1 por pertenecer a un nodo sólo parcialmente correcto. Tampoco se consideraba la duplicación de palabras en la respuesta). Misma corrección en operador “CAU”.
- Correcciones en operador “NEG”. Ahora el match ya no se anula cuando una palabra es precedida por una negación que niega otro concepto.
- Adición del operador “SEG”, el cual indica el requerimiento de un orden estricto entre las claves unidas por él.
- Adición del operador “\_” para mejorar el análisis de expresiones y locuciones de más de una palabra (las palabras unidas por este operador deben ser consecutivas).
- Prueba del sistema de parseo lingüístico *off-line*, usando las claves lógico-semánticas extendidas de RADial y consultando los datos morfológicos requeridos para el diálogo en archivos txt (diccionario morfológico local).
- Elaboración de un algoritmo para crear los archivos txt con todos los datos morfológicos.

#### Agosto 2013

- Corrección en opciones de corrección ortográfica (se había descompuesto por algunos cambios hechos al código en julio, en particular, dimensión adicional para el array *\$raices*)
- Para poder usar las claves RADial en modo *off-line*:
  - Algoritmo (con index.html) para la creación de un diccionario local como objeto JSON con base en las claves extendidas contenidas en el archivo xml del diálogo
  - Traducción de php a javascript de los algoritmos para convertir la respuesta del usuario en una lista ordenada de lemas

- Programación en javascript de un algoritmo con funcionalidad más reducida para la corrección ortográfica
- Traducción de los algoritmos de parseo de expresiones lógico-semánticas y análisis semántico de php a javascript

### Septiembre 2013

- Algoritmo para la creación automática de la expresión extendida de las expresiones fijas (NEG), (AFIR) y (DUDA)
- Corrección de algunos errores del análisis semántico en las expresiones fijas (por ejemplo, “no sé” no se reconocía como duda por la presencia de la negación).
- Cambio de las expresiones (NEG) y (AFIR) a (NO) y (SÍ), respectivamente.
- Permitir combinaciones entre expresiones fijas y/o con otras claves.

