

MELISSA: Methods and Tools for Natural-Language Interfacing to Standard Software Applications

Organismo financiador: European Commission (DG III – F6), ESPRIT 22252

Grupos participantes en el proyecto:

Software AG España, S.A. (SAGE)	Coordinador
Anite Systems Plc., International Division (Anite)	Partner
Institut für Angewandte Informationsforschung (IAI)	Partner
SEMA GROUP (SEMA)	Partner
Organización Nacional de Ciegos de España (ONCE)	Usuario
NETCologne - Gesellschaft für Telekommunikation mbH	Usuario

Persona de contacto: José Antonio Frías. Software AG España, S.A. Ronda de la Luna, 4. Tres Cantos, E-28760 Madrid Tel: +34 91 8079411 - fax: +34 91 8079447 email: jfrias@sag.es

Resumen

El objetivo de MELISSA se define de forma tan sencilla como ambiciosa: acceder a la funcionalidad de cualquier aplicación utilizando como interfaz el lenguaje propio del usuario.

Para la consecución de este objetivo, es necesario resolver tres problemas principales:

- Interpretación de lenguaje natural de una forma eficiente y flexible, dada la multitud de diferentes usuarios y contextos de ejecución que se puede encontrar MELISSA.
- Análisis semántico capaz de manejar los hechos semánticos obtenidos en el análisis lingüístico y convertirlos en las acciones posibles sobre el modelo de la aplicación.
- Representación genérica de aplicaciones que permita modelar diversos tipos de aplicaciones software de un modo sencillo, y que a su vez sea el soporte de los diversos algoritmos lingüísticos

1. Arquitectura

La arquitectura que MELISSA plantea para resolver el problema de la generalización lingüística y el acceso a la aplicación, es la creación de una **Base de Conocimientos** durante la fase de adaptación de la aplicación a MELISSA, que se realiza mediante el **Subsistema de Desarrollo**, de tal forma que el **Subsistema de Ejecución** es capaz de resolver el acceso a la aplicación mediante algoritmos genéricos e independientes de la misma.

Los tipos de aplicación sobre los que opera MELISSA están estructurados y representados bajo dos grandes bloques: basadas en el paradigma de OO y en programación clásica.

Las primeras son las ideales para una modelización genérica y acceso sencillo, la estructuración en objetos permite una delimitación de la funcionalidad más clara en general que en las aplicaciones clásicas. Por otra parte, los métodos de acceso a la funcionalidad en las primeras, gracias a

CORBA/DCOM, pueden hacerse de manera más estándar que en las aplicaciones clásicas con interfaces basadas en menús, en las que para realizar la conexión se atenderá a técnicas de wrapping para acceder con CORBA/DCOM o a través de su interfaz de usuario, simulando un operador automático.

Se describen a continuación los componentes principales de MELISSA.

1.1. Subsistema de ejecución

1.1.1. Intérprete de Lenguaje Natural

La primera fase de procesamiento de MELISSA es el análisis lingüístico, que recibe la frase en lenguaje natural como entrada y la traduce a una descripción factual consistente en una lista de hechos semánticos que describen las relaciones existentes entre los elementos derivados de la frase de entrada.

El análisis lingüístico de MELISSA hace uso de los algoritmos y recursos lingüísticos definidos sobre la plataforma ALEP (Advanced

Language Engineering Platform). Están comprendidas dos fases: la primera es un tratamiento inicial del texto dónde se etiquetan distintas palabras, se detectan tipos de datos genéricos y dependientes del dominio, y la segunda, un análisis morfo-sintáctico que a su vez construye la descripción factual resultante. La descripción de los recursos lingüísticos (léxico, cobertura de fenómenos lingüísticos, tipos de datos, ...) se ha modularizado para permitir una gestión más fácil y configurable.

La naturaleza de producto comercial de MELISSA implica para este módulo especial atención a la 'flexibilidad', dada la gran variedad de usuarios potenciales, siendo permisivo a errores gramaticales, admitiendo relajación en la concordancia entre palabras y manejando información morfo-sintáctica específica de la aplicación (codificación de términos, multi-word-units, formas 'habituales' de expresión de los usuarios, ...).

1.1.2. Análisis Semántico

El objetivo de este módulo de análisis es encontrar la correspondencia entre la representación semántica de una entrada del sistema y las funcionalidades de la aplicación destino. Esta interpretación se basa en el uso del formalismo de grafos semánticos que permiten representar tanto el contenido de la frase en lenguaje natural como cada funcionalidad lógica presente en el sistema.

La descripción de las funcionalidades se construye enunciando el modelo conceptual de la aplicación dónde cada concepto presente se detalla con relaciones sinonímicas y colección de atributos con sus tipos y valores. Este modelo sirve de base para la descripción de funciones como acciones sobre los conceptos existentes. El algoritmo de análisis equipara el grafo semántico construido a partir de la salida del módulo lingüístico, con los grafos de las funciones lógicas.

La modelización permite que el algoritmo detecte encajes parciales y actúe en consecuencia mediante mecanismos de elipsis o anáfora, enriquecimiento automático del modelo conceptual (aprendizaje de sinónimos) o tratamiento de ambigüedades.

1.1.3. Acceso a la Aplicación

La modelización de las aplicaciones tiene como elemento básico la idea de función

lógica. Una función lógica es una unidad funcional de la aplicación identificable por una frase/petición del usuario. El módulo de acceso, basándose en las estructuras homogéneas proporcionadas desde la herramienta de desarrollo, traduce la función lógica a sus correspondientes funciones físicas, completando si es necesario la información de la misma mediante preguntas al usuario o con la historia de ejecución en casos de elipsis.

Una vez generada la secuencia de comandos o funciones físicas de la aplicación, se realiza el acceso concreto para cada 'tipo' de aplicación accedida.

1.2. Subsistema de desarrollo

Pieza clave para la estandarización de aplicaciones, proporciona herramientas para la modelización de la aplicación concreta, de forma que esta pueda ser accedida por el Interface de Lenguaje Natural. Para ello recoge información en diversos niveles: funcionalidad de la aplicación (funciones lógicas, parámetros asociados, etc.), cobertura lingüística (términos con los que se hace referencia a los elementos de la aplicación); relaciones semánticas entre diversos elementos; y por último la implementación física de cada funcionalidad.

2. Plataformas

Finalizado el proyecto, el núcleo de MELISSA run-time ejecutará en plataformas WNT y SUN Solaris, y su cliente en W9x y WNT. El Subsistema de desarrollo (en Java), únicamente requiere de una base de datos relacional con acceso ODBC.

3. Conclusión

MELISSA es una herramienta que se basa en tres características esenciales:

- una potente modelización de aplicaciones en sus funciones lógicas,
- la adaptación de los módulos lingüísticos al dominio de la aplicación,
- un analizador semántico capaz de traducir entre ambas representaciones.

Con estas bases, MELISSA se plantea como una solución para el uso de aplicaciones en empresas en las que los gastos de formación, el continuo cambio en las aplicaciones y la inexperiencia informática del personal, representan un coste excesivo.