Modelado en el nivel de concepto

Francisco J. Martín Mateos

Dpto. Ciencias de la Computación e Inteligencia Artificial
Universidad de Sevilla

Objetivos del nivel de concepto

- Especifica la estructura de la información y del conocimiento requerido por el SBC, sin hacer referencia a los detalles de implementación.
  - Determinar *qué* es lo que el sistema debe hacer, sin indicar *cómo* lo hace realmente.
- Incluye el desarrollo de dos modelos.
  - Modelo de conocimiento: Detalla los requisitos de conocimiento y razonamiento del SBC a desarrollar.
  - Modelo de comunicación: Detalla los requisitos de interacción entre el SBC y otros agentes.
- La entrada a ambos modelos es una tarea intensiva en conocimiento identificada en el estudio de viabilidad llevado a cabo en el nivel de contexto.
- Se especifica usando el Lenguaje de Modelado Conceptual (CML).

Estructura del modelo de conocimiento

- Tres categorías de conocimiento.
  - Conocimiento del Dominio: Terminología general y hechos de un dominio, descritos de forma independiente de una tarea particular.
  - Conocimiento sobre Inferencias: Procesos básicos de razonamiento que se realizan directamente sobre el conocimiento del dominio.
  - Conocimiento sobre Tareas: Terminología, modelos computacionales y hechos asociados con la ejecución de un tipo de tarea, sin centrarse en un dominio particular.

Conocimiento del Dominio

- En esta categoría se representa el conocimiento relevante del dominio de la aplicación.
  - Tipos y objetos de conocimiento descritos de forma independiente de la tarea.
- Se desarrolla en dos partes.
  - Esquema del Dominio: Terminología general utilizada para describir el conocimiento del dominio.
  - Base de Conocimiento: Hechos específicos de dicho dominio.

Conocimiento del Dominio

El conocimiento del dominio en CML

DOMAIN-KNOWLEDGE *identificador*;
    Descripción del esquema(s) del dominio;
    Descripción de la(s) base(s) de conocimiento;
END DOMAIN-KNOWLEDGE *identificador*;

Esquema del Dominio

- Descripción esquemática de la información y del conocimiento estáticos del dominio de la aplicación.
- Constructores básicos de modelado.
  - Conceptos: Definen colecciones de objetos que presentan características comunes.
  - Relaciones: Definen relaciones entre los conceptos.
  - Tipos de reglas: Definen relaciones entre expresiones lógicas sobre los valores de atributos de conceptos.

Esquema del Dominio

El esquema del dominio en CML

DOMAIN-SCHEMA *identificador*;

Dependencias entre esquemas de dominios;

Descripción de conceptos, relaciones y tipos de reglas;

END DOMAIN-SCHEMA *identificador*;

Conceptos

- Definen colecciones de objetos que presentan características comunes.
- Las características de un concepto se especifican mediante un conjunto de atributos.
- Cada atributo se especifica con un nombre que lo identifica y el tipo de valor que puede tomar.
  - Por defecto, los atributos toman un único valor, aunque se pueden definir explicitamente cardinalidades mayores (slot - multislot).
- Los tipos de valores de un atributo han de ser simples y no hacer referencia a otros objetos: natural, integer, float, symbol, string, ordinal.
  - Las referencias a otros objetos se especifican mediante relaciones.
- Se pueden indicar restricciones sobre los valores de los atributos.

Conceptos en CML

CONCEPT cliente;
  DESCRIPTION:
    "Datos personales del cliente";
  ATTRIBUTES:
    nombre: STRING;
    domicilio: STRING;
    edad: NATURAL;
    ...
  AXIOMS:
    edad $\geq$ 18;
END CONCEPT cliente;

Conceptos en CML

CONCEPT solicitud;
DESCRIPTION:
"Solicitud de asesoramiento";
ATTRIBUTES:
usuario: infantil, adolescente, joven, maduro;
uso: científico, multimedia, diseño, ...;
CARDINALITY: 3;
inversión: NATURAL;
...
AXIOMS:
$400 < \text{inversión} \leq 1000$;
END CONCEPT solicitud;

Conceptos

- Se pueden establecer jerarquías entre los conceptos.
  - Subtipos: Conceptos hijos.
  - Supertipos: Conceptos padre.
- Entre los conceptos hijos se pueden indicar propiedades semánticas.
  - Completa: Todas las instancias del supertipo son a su vez instancias de algún subtipo.
  - Disjunta: Cada instancia del supertipo es a su vez instancia de como mucho uno de los subtipos.
- Posibles especializaciones en los subtipos.
  - Añadir nuevos atributos específicos del subtipo.
  - Restringir el conjunto de valores de un atributo heredado.
  - Reducir la cardinalidad de los atributos heredados.

Jerarquías de conceptos en CML

CONCEPT ordenador;
  DESCRIPTION:
    "Ordenador personal";
  SUPER-TYPE-OF: portátil, sobremesa;
  SEMANTICS:
    DISJOINT: YES;
    COMPLETE: YES;
  ATTRIBUTES:
    marca: STRING;
    modelo: STRING;
END CONCEPT ordenador;

Jerarquías de conceptos en CML

CONCEPT portátil;
  DESCRIPTION:
    "Ordenador personal portátil";
  SUB-TYPE-OF: ordenador;
  ATTRIBUTES:
    peso: FLOAT;
    autonomía: FLOAT
END CONCEPT portátil;

Jerarquías de conceptos en CML

CONCEPT sobremesa;
  DESCRIPTION:
    "Ordenador personal fijo";
  SUB-TYPE-OF: ordenador;
END CONCEPT sobremesa;

Conceptos

- Se pueden establecer relaciones del tipo *parte de* entre conceptos.
  - Un concepto puede estar formado por otros conceptos (componentes).
  - Un concepto puede ser parte integrante de otro concepto.
- Por defecto cada componente aparece una única vez, podemos indicar que una misma componente puede aparecer varias veces modificando la cardinalidad de dicha componente en la descripción de la misma.

Relaciones *parte de* en CML

CONCEPT ordenador;
  DESCRIPTION:
    "Ordenador personal";
  HAS-PARTS: cpu, memoria, ...;
  ATTRIBUTES:
    marca: STRING;
    modelo: STRING;
END CONCEPT ordenador;

Relaciones *parte de* en CML

CONCEPT memoria;
  DESCRIPTION:
    "Módulo de memoria RAM";
  PART-OF: ordenador;
    CARDINALITY: 1+;
  ATTRIBUTES:
    tipo: STRING;
    capacidad: NATURAL;
    velocidad: NATURAL;
END CONCEPT memoria;

Relaciones *parte de* en CML

CONCEPT cpu;
  DESCRIPTION:
    "Procesador";
  PART-OF: ordenador;
  ATTRIBUTES:
    marca: STRING;
    modelo: STRING;
    velocidad: NATURAL;
END CONCEPT cpu;

Relaciones

- Definen relaciones entre los conceptos.
- Permiten especificar relaciones más complejas que las de jerarquía o del tipo “parte de”.
- Una relación se define por medio de sus argumentos.
  - Los argumentos de una relación son conceptos u otras relaciones.
  - Se pueden definir relaciones con cualquier número de argumentos.
  - Las relaciones pueden tener atributos propios distintos de los conceptos relacionados.
- La cardinalidad de un argumento en una relación indica el número de veces que una instancia de dicho objeto puede aparecer en instancias de la relación.
  - La cardinalidad por defecto es uno, aunque se puede modificar de forma explícita.

Relaciones en CML

```
RELATION venta;
    ARGUMENTS:
        ordenador;
            CARDINALITY 0-1;
        cliente;
            CARDINALITY ANY;
        comercial;
            CARDINALITY ANY;
    ATTRIBUTES:
        fecha-de-venta: DATE;
END RELATION venta;
```

Relaciones

- Las relaciones más frecuentes son las binarias.
- En una relación binaria se pueden especificar propiedades como reflexividad, simetría o transitividad.
- Una relación binaria puede ser dirigida, en cuyo caso se tiene que indicar el nombre de la relación binaria inversa (que no es necesario especificar).

Relaciones

Relaciones binarias en CML

RELATION comprado-por;
    INVERSE: comprar;
    ARGUMENT-1: ordenador;
        CARDINALITY 0-1;
    ARGUMENT-2: cliente;
        CARDINALITY ANY;
    [**REFLEXIVE** | **SYMMETRIC** | **TRANSITIVE**]
END RELATION comprado-por;

Tipos de reglas

- Especifican relaciones entre expresiones lógicas sobre los valores de atributos de conceptos.
  - Relaciones de implicación entre dos tipos de objetos.
- Permiten modelar en un único tipo un conjunto de reglas que comparten una estructura y funcionalidad similar.
- Componentes:
  - Un antecedente y un consecuente, en los que se indican los conceptos sobre cuyas instancias se definirán las expresiones lógicas.
  - Un símbolo de conexión, que será utilizado para definir las instancias del tipo de regla.

Tipos de reglas

Tipos de reglas en CML

RULE-TYPE sugerencia;
DESCRIPTION:
"Relación entre las preferencias de un usuario y las características del ordenador"
ANTECEDENT: solicitud;
CONSEQUENT: ordenador;
CONNECTION-SYMBOL sugiere;
END RULE-TYPE sugerencia;

Tipos de reglas

Tipos de reglas en CML

RULE-TYPE abstracción;
  DESCRIPTION:
    "Relación entre los datos de una solicitud y su abstracción"
  ANTECEDENT: solicitud;
  CONSEQUENT: solicitud;
  CONNECTION-SYMBOL se-abstrae-en;
END RULE-TYPE abstracción;

Modularización y reutilización

- El conocimiento del dominio se puede especificar por medio de la definición de distintos esquemas del dominio.
- Las dependencias entre los diferentes esquemas del dominio se indican mediante la construcción **USES**.
  - Un esquema del dominio puede importar todo el contenido de otro esquema del dominio.
  - Un esquema del dominio puede importar algunas construcciones definidas en otro esquema del dominio.
- Las construcciones importadas de otros esquemas del dominio se utilizan indicando el esquema del dominio del que proceden y el nombre de la construcción.

Modularización y reutilización en CML

DOMAIN-SCHEMA asesoramiento-ordenador;
    USES datos-clientes;
    USES:
        cpu, memoria **FROM** componentes-ordenador;
    ...
END DOMAIN-SCHEMA asesoramiento-ordenador;

La base de conocimiento

- En el esquema del dominio se definen los distintos tipos de conocimiento que podemos encontrar en un dominio de aplicación.
- Las instancias de dichos tipos de conocimiento se especifican en las bases de conocimiento.
- La especificación de una base de conocimiento consta de dos partes.
  - Indicar de qué tipo de conocimiento son las instancias que se están definiendo.
  - Definir las instancias concretas que componen la base de conocimiento.
- Usualmente se utilizan para definir instancias de tipos de reglas, aunque también se pueden incluir instancias de conceptos y relaciones.

La base de conocimiento

Bases de conocimiento en CML

KNOWLEDGE-BASE conocimiento-sugerencias;
  USES:
    sugerencia **FROM** asesoramiento-ordenador;
  EXPRESSIONS:
    datos-solicitud.uso = multimedia;
      SUGIERE
    memoria.capacidad $\geq$ 2Gb;
END KNOWLEDGE-BASE asesoramiento-ordenador;

Conocimiento sobre Inferencias

- Describe cómo se utilizan las estructuras estáticas descritas en el conocimiento del dominio para llevar a cabo procesos de razonamiento.
- Elementos de los que se dispone para especificar el Conocimiento sobre Inferencias.
  - Inferencias: Unidades básicas de procesamiento de información.
  - Roles de Conocimiento: Etiquetas abstractas que indican el papel que juega el conocimiento del dominio en el proceso de razonamiento.
  - Funciones de transferencia: Procesos que transmiten elementos de conocimiento entre el agente de conocimiento y el mundo exterior.

Conocimiento sobre Inferencias

Conocimiento sobre inferencias en CML

INFERENCE-KNOWLEDGE *identificador*;
    Descripción de inferencias, roles de conocimiento y funciones de transferencia;
END INFERENCE-KNOWLEDGE *identificador*;

Inferencias y roles de Conocimiento

- Una inferencia describe un proceso primitivo de razonamiento.
  - Cada inferencia queda completamente descrita mediante una especificación declarativa de sus entradas y salidas.
  - El proceso interno de la inferencia no tiene interés desde el punto de vista del modelado de conocimiento.
- Las inferencias se identifican mediante un proceso de descomposición funcional de las tareas.
  - El proceso de descomposición funcional termina cuando con las inferencias obtenidas podemos obtener una traza comprensible del razonamiento efectuado por el sistema.
  - Un SBC debe ser capaz de explicar el proceso de razonamiento que lleva a cabo.

Inferencias y roles de Conocimiento

- Una inferencia hace referencia al conocimiento del dominio indirectamente a través de los roles de conocimiento.
- Los roles de conocimiento son etiquetas que indican el papel que juega el conocimiento del dominio en el proceso de razonamiento.
  - Roles dinámicos: Constituyen las entradas y salidas del proceso de inferencia. Cambian en diferentes invocaciones de la inferencia.
  - Roles estáticos: Especifican el conocimiento del dominio que se utiliza en el proceso de razonamiento indicado en la inferencia. Se mantienen estables durante todas las invocaciones de la inferencia.
- La metodología CommonKADS ofrece un catálogo de 18 tipos de inferencia distintos.

Inferencias y roles de Conocimiento en CML

INFERENCIA abstraer;
OPERATION-TYPE: ABSTRACT
ROLES:
INPUT caso;
OUTPUT caso-abstraido;
STATIC conocimiento-abstracción;
SPECIFICATION:
"Genera una transformación en los datos de entrada que produce una descripción más cualificada de un caso";
END INFERENCE abstraer;

Inferencias y roles de Conocimiento

Inferencias en CML

INFERENCIA diagnosticar;
OPERATION-TYPE: COVER
ROLES:
INPUT fallo;
OUTPUT hipótesis;
STATIC modelo-causal;
SPECIFICATION:
"Genera el conjunto de todas las situaciones que pueden causar el fallo introducido como entrada";
END INFERENCE diagnosticar;

Inferencias y roles de Conocimiento

- La descripción de la inferencia no incluye referencias a elementos del dominio de aplicación.
  - Una inferencia sólo puede hacer referencia al conocimiento del dominio indirectamente a través de los roles de conocimiento.
- Una vez especificadas las inferencias hay que especificar los roles de conocimiento indicando con qué elementos del dominio se corresponden.

Inferencias en CML

KNOWLEDGE-ROLE caso;
  TYPE: DYNAMIC;
  DOMAIN-MAPPING: solicitud;
END KNOWLEDGE-ROLE caso;

KNOWLEDGE-ROLE caso-abstraido;
  TYPE: DYNAMIC;
  DOMAIN-MAPPING: solicitud;
END KNOWLEDGE-ROLE caso-abstraido;

KNOWLEDGE-ROLE conocimiento-abstracción;
  TYPE: STATIC;
  DOMAIN-MAPPING:
    abstracciones **FROM** asesoramiento-computador;
END KNOWLEDGE-ROLE conocimiento-abstracción;

Funciones de transferencia

- Procesos que transmiten elementos de conocimiento entre el agente de conocimiento y el mundo exterior.
- Se especifican mediante un nombre, un tipo y los elementos de entrada y salida.
- Los detalles específicos de estas funciones se establecen en el modelo de comunicación.

Funciones de transferencia

- Cada función de transferencia tiene dos propiedades que la caracterizan:
  - ¿quién toma la inciativa de establecer la transferencia?
  - ¿quién posee la información objeto de la transferencia?
- Tipos de funciones de transferencia.
  - **OBTAIN**: El sistema solicita la información a una fuente externa.
  - **RECEIVE**: El sistema recibe información desde una fuente externa.
  - **PRESENT**: El sistema presenta información a una fuente externa.
  - **PROVIDE**: El sistema proporciona información a una fuente externa.

Funciones de transferencia

Inferencias en CML

TRANSFER-FUNCTION obtener;
    TYPE: OBTAIN
    ROLES:
        INPUT hallazgo-esperado;
        OUTPUT hallazgo-observado;
END TRANSFER-FUNCTION obtener;

Conocimiento sobre Tareas

- Especificación de los objetivos del proceso de razonamiento.
  - La tarea especifica el objetivo del proceso de razonamiento que se está intentando modelar en función de sus entradas y salidas.
  - El método de la tarea que indica cómo se puede llevar a cabo la tarea mediante la especificación de su descomposición en subtareas y el proceso de control sobre las mismas.

Conocimiento sobre Tareas

Conocimiento sobre tareas en CML

TASK-KNOWLEDGE *identificador*;
    Descripción de tareas y métodos de tareas;
END TASK-KNOWLEDGE *identificador*;

Tareas

- Una tarea define un proceso de razonamiento complejo.
- La tarea más alta de la jerarquía debe corresponderse con alguna de las tareas identificadas en el Modelo de Tareas.
- Una tarea se especifica mediante:
  - El nombre de la tarea.
  - El objetivo de la tarea.
  - Los roles de entrada y salida.
  - Una descripción en lenguaje natural de lo que hace la tarea.

Tareas

Tareas en CML

TASK decidir-caso;
  GOAL:
    "Aplicar criterios al caso para tomar una decisión en función de sus valores";
  ROLES:
    INPUT:
      caso-abstraido;
      cirterios-especificos;
    OUTPUT:
      decisión;
END TASK decidir-caso;

Métodos de las tareas

- El método de una tarea nos indica cómo se puede alcanzar el objetivo indicado en la tarea, mediante la descomposición en subtareas, inferencias y funciones de transferencia.
- Un método se especifica mediante:
  - La tarea a la que está asociado.
  - Las subtareas, inferencias y funciones de transferencia en las que se descompone.
  - Los roles intermedios utilizados para almacenar resultados temporales.
  - La estructura de control que describe cómo se desarrolla el método.

Métodos de las tareas

Métodos de las tareas en CML

TASK-METHOD metodo-decidir-caso;
REALIZES: decidir-caso;
DECOMPOSITION:
[**TASKS**: ...]
INFERENCES: especificar, seleccionar, evaluar, encajar;
TRANSFER-FUNCTIONS: obtener;
ROLES:
INTERMEDIATE:
criterios: "el conjunto de criterios a evaluar";
criterio: "un criterio de asesoramiento";
valor: "valor de un criterio de asesoramiento";
resultados: "criterios seleccionados junto con sus valores";
CONTROL-STRUCTURE:
Descripción en pseudocódigo del proceso a realizar en términos de las subteareas, inferencias y funciones de transferencia
END TASK-METHOD metodo-decidir-caso;

Bibliografía

- Schreiber G., Akkermans H., ...
  "Knowledge Engineering and Management: The CommonKADS Methodology", The MIT Press, 1999.
  - Cap. 5: "Knowledge Model Components"
  - Apéndice: "Knowledge-Model Language"
- Alonso A., Guijarro B., ...
  "Ingeniería del Conocimiento: Aspectos Metodológicos", Pearson Prentice Hall, 2004.
  - Cap. 5: "El modelo de Conocimiento"