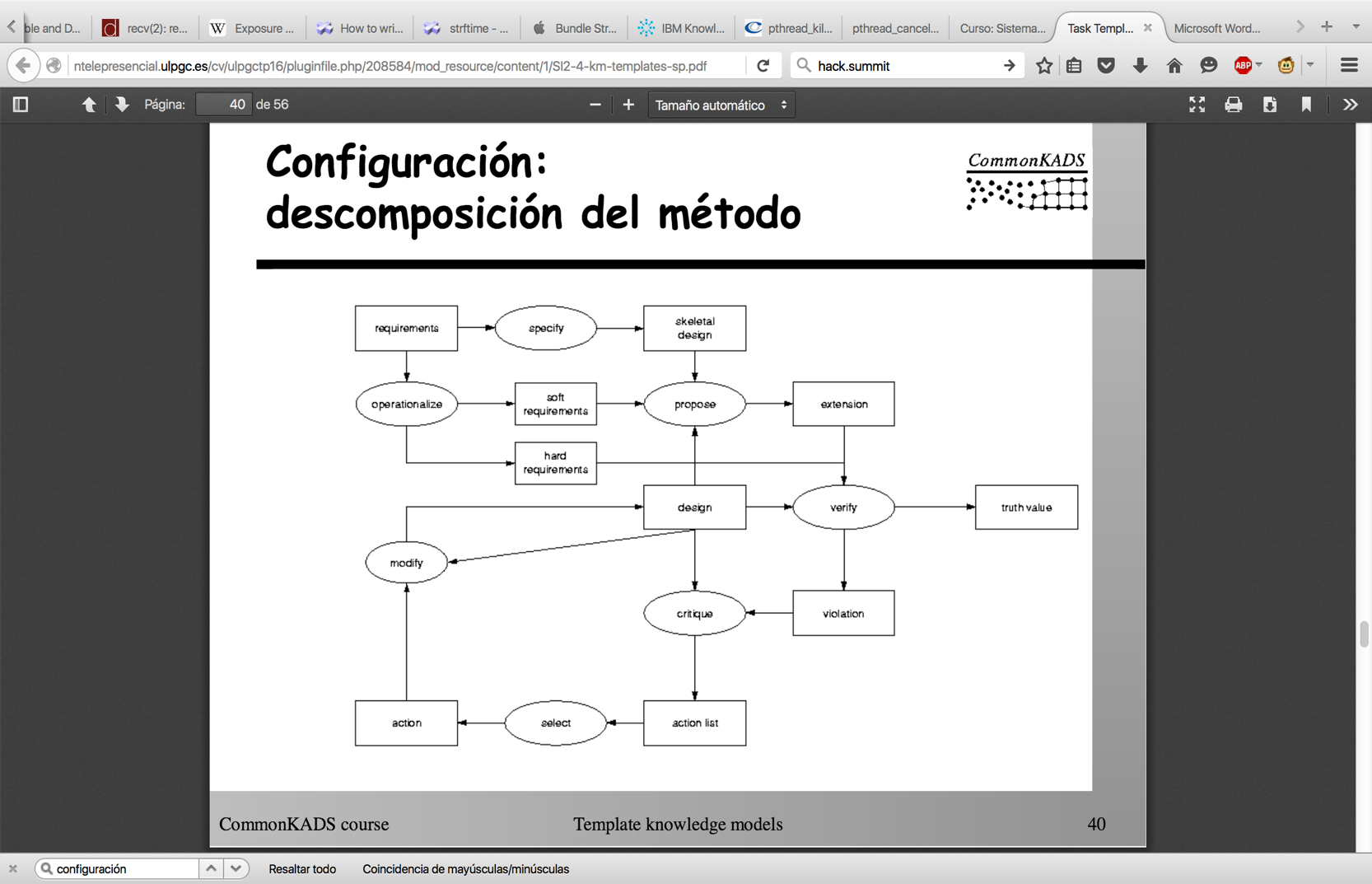
**Grupo: Laura del Pino Díaz y Aitor del Pino Saavedra Hernández**

1. **Identificar y justificar el tipo genérico de tarea según la clasificación de CommonKAD.**

Es una tarea sintética del tipo configuración porque no tiene restricciones temporales.

1. **Describir el control de la tarea, roles e inferencias para la tarea seleccionada.**



Se plantea una modificación del esquema superior de forma que solo haya una única acción en la lista de acciones que sea deshacer y que la inferencia “modify” siga un esquema de búsqueda en el espacio de estados.

Los **roles** identificados son los siguientes:

* Requerimientos
* Requerimientos suaves y duros.
* Diseño esquelético.
* Extensión
* Estado de verdad
* Especificación de violación de requerimientos.
* Diseño.

Los requerimientos son:

1. Máximo de dos horas de teoría por día. (Pasará a ser límite duro).
2. Las asignaturas trocales son las primeras se dispondrán consecutivamente cada día.(Límite duro)
3. Se prefiere que las asignaturas troncales se den lo más temprano posible. (Preferencia, límite suave)
4. Todas las clases de laboratorioson de dos horas (Pasará a ser límite duro).
5. Solo existen tres laboratorios. (Limite furo)
6. Solo se pueden simultanear las horas de prácticas.
7. Un profesor debe dar solo un grupo de teoría.
8. Se prefiere que se pongan las horas de teoría antes que las de práctica.

El diseño esquelético:

Es el horario con sus franjas horarias.

Las extensiones son:

Cada una de las versiones del horario con un conjunto de franjas ocupadas.

Las especificaciones de violaciones son las extensiones que no cumplan las restricciones con la señalización de cuáles son las franjas que no cumplen las restricciones.

El estado de verdad es un valor que indica si el diseño es correcto, es decir, no viola ninguna de las restricciones.

**Las inferencias** de esta tarea son:

* “specify”/especifica: a partir de los requerimientos se genera el diseño esquelético.
* “operationalize”/operacionaliza:dividir los requerimientos en las clases suave y duro.
* “propose”/propone: genera una extensión a partir de los requerimientos suaves y el diseño esquelético o el diseño.
* “verify”/verifica:verificar que se cumplen los límites duros y genera un valor de verdad o un violación de restricciones.
* “modify”/modifica: deshace la última extensión y prueba con otro valor.

**El control** de esta tarea es:

operacionaliza(requirimientos ->requerimientos suaves+ requerimientos fuertes);

especifica(requiremientos -> diseño esquelético);

*mientras* solución\_nueva propone(diseño esquelético + diseño + requisitos suaves

-> extensión) hacer

diseño := extensión unión diseño

verifica(diseño + requerimientos duros -> estado de verdad y especificación de violación);

*mientras* especificación de violación no sea nula hacer

modifica(diseño + acción -> diseño); //deshace

verify(design + hard-reqs -> estado de verdad y especificación de violacion);

*fin mientras*;

*fin mientras;*

1. **Desarrollar el conocimiento del dominio en CML especificando los conceptos, relaciones, type‐rule (tipos de reglas) y las bases de conocimiento.**

KNOWLEDGE-MODEL pseudo\_EII

**DOMAIN-KNOWLEDGE Horario**

/\* esquemas del dominio (definiciones) \*/

**DOMAIN-SCHEMA ds-crepe**

CONCEPT clase

DESCRIPTION:

HAS-PARTS:

END CONCEPT clase

CONCEPT laboratorio

DESCRIPTION:

HAS-PARTS:

SUB-TYPE-OF: clase

PART-OF:

ATTRIBUTES:

END CONCEPT laboratorio

CONCEPT:aula

DESCRIPTION:

HAS-PARTS:

SUB-TYPE-OF: clase

END CONCEPT aula

CONCEPT:profesor

DESCRIPTION:

HAS-PARTS:

END CONCEPT professor

CONCEPT:asignatura

DESCRIPTION:

HAS-PARTS:

END CONCEPT asingatura

CONCEPT:troncal

DESCRIPTION:

HAS-PARTS:

SUB-TYPE-OF: asignatura

END CONCEPT truncal

CONCEPT:no troncal

DESCRIPTION:

HAS-PARTS:

SUB-TYPE-OF: asignatura

END CONCEPT no truncal

CONCEPT:horario

DESCRIPTION:

HAS-PARTS: casilla

SUB-TYPE-OF:

END CONCEPT horario

CONCEPT:casilla

DESCRIPTION:

HAS-PARTS: asignatura,professor,clase

ATTRIBUTES: hora y dia

SUB-TYPE-OF:

RELATION asignación

ARGUMENTS-1: profesor

CARDINALITY:1

ARGUMENTS-2:Asignatura

CARDINALITY:1

ARGUMENTS-3:Clase

CARDINALITY-3:1

END RELATION asignación

/\* tipos de regla\*/

RULE-TYPE Requisitos-de-asignación

ANTECEDENT: diseño esquemático

CARDINALITY: 1

CONSEQUENT:horario

CARDINALITY:1

CONECTION-SYMBOL: establece

END RULE-TYPE Requisitos-de-asignación

**END DOMAIN-SCHEMA ds-crepe**

/\* bases de conocimiento (instancias y reglas) \*/

**KNOWLEDGE-BASE kb-crepe**

USES:

Requisites-de-concesión FROM (ds-crepe)

EXPRESIONS:

“cond-inic1: comprobamos que el cliente suficiente solvencia para hacer frente a la cuota y además cuenta con un margen para hacer frente a gastos imprevistos”

(( info-ingresos.total – info-gastos.total) > 120% info-crédito.cuota)

AND

( info-crédito.cuota < 35 % info-ingresos.total)

ESTABLECE

Solvencia.verificado = true

“ cond-inic2: relaja las condiciones de “cond-inic1” para solteros que viven en casa familiar”

(( info-ingresos.total – info-gastos.total) > 110% info-crédito.cuota)

AND

( info-crédito.cuota < 50 % info-ingresos.total)

AND

( info-personal.estado-civil = soltero/a)

AND

(info-personal.residencia-familiar =true)

ESTABLECE

Solvencia.verificado = true

**END KNOWLEDGE-BASE kb-1**

**END DOMAIN-KNOWLEDGE**

KNOWLEDGE-MODEL CrePe

**DOMAIN-KNOWLEDGE dk-Créditos\_Personales**

/\* esquemas del dominio (definiciones) \*/

**DOMAIN-SCHEMA ds-1**

CONCEPT c

DESCRIPTION:

SUB-TYPE-OF:

SUPER-TYPE-OF:

HAS-PARTS:

PART-OF:

ATTRIBUTES:

(nombre-atributo): (tipo: number, integer, natural, real, image, string, boolean, universal, date, text, ref. a value-type ) + (rango(min, max))

(cardinalidad)

DEFAULT-VALUE:

AXIOMS:

END CONCEPT c

VALUE-TYPE vt

TYPE:

VALUE-LIST:

END VALUE-TYPE vt

BINARY-RELATION br

DESCRIPTION:

ARGUMENT-1:

CARDINALITY:

ARGUMENT-2:

CARDINALITY:

ATTRIBUTES:

END BINARY-RELATION br

RELATION r

ARGUMENTS:

END RELATION r

/\* tipos de conocimiento \*/

RULE-TYPE rt1

ANTECEDENT:

CARDINALITY:

CONSEQUENT:

CARDINALITY:

CONECTION-SYMBOL:

END RULE-TYPE rt1

**END DOMAIN-SCHEMA ds-1**

/\* bases de conocimiento (instancias y reglas) \*/

**KNOWLEDGE-BASE kb-1**

USES:

(rule-type y) FROM (domain-schema x)

EXPRESIONS:

(reglas)

**END KNOWLEDGE-BASE kb-1**

**END DOMAIN-KNOWLEDGE**

KNOWLEDGE-MODEL CrePe

**DOMAIN-KNOWLEDGE dk-Créditos\_Personales**

/\* esquemas del dominio (definiciones) \*/

**DOMAIN-SCHEMA ds-1**

CONCEPT c

DESCRIPTION:

SUB-TYPE-OF:

SUPER-TYPE-OF:

HAS-PARTS:

PART-OF:

ATTRIBUTES:

(nombre-atributo): (tipo: number, integer, natural, real, image, string, boolean, universal, date, text, ref. a value-type ) + (rango(min, max))

(cardinalidad)

DEFAULT-VALUE:

AXIOMS:

END CONCEPT c

VALUE-TYPE vt

TYPE:

VALUE-LIST:

END VALUE-TYPE vt

BINARY-RELATION br

DESCRIPTION:

ARGUMENT-1:

CARDINALITY:

ARGUMENT-2:

CARDINALITY:

ATTRIBUTES:

END BINARY-RELATION br

RELATION r

ARGUMENTS:

END RELATION r

/\* tipos de conocimiento \*/

RULE-TYPE rt1

ANTECEDENT:

CARDINALITY:

CONSEQUENT:

CARDINALITY:

CONECTION-SYMBOL:

END RULE-TYPE rt1

**END DOMAIN-SCHEMA ds-1**

/\* bases de conocimiento (instancias y reglas) \*/

**KNOWLEDGE-BASE kb-1**

USES:

(rule-type y) FROM (domain-schema x)

EXPRESIONS:

(reglas)

**END KNOWLEDGE-BASE kb-1**

**END DOMAIN-KNOWLEDGE**

KNOWLEDGE-MODEL CrePe

**DOMAIN-KNOWLEDGE dk-Créditos\_Personales**

/\* esquemas del dominio (definiciones) \*/

**DOMAIN-SCHEMA ds-1**

CONCEPT c

DESCRIPTION:

SUB-TYPE-OF:

SUPER-TYPE-OF:

HAS-PARTS:

PART-OF:

ATTRIBUTES:

(nombre-atributo): (tipo: number, integer, natural, real, image, string, boolean, universal, date, text, ref. a value-type ) + (rango(min, max))

(cardinalidad)

DEFAULT-VALUE:

AXIOMS:

END CONCEPT c

VALUE-TYPE vt

TYPE:

VALUE-LIST:

END VALUE-TYPE vt

BINARY-RELATION br

DESCRIPTION:

ARGUMENT-1:

CARDINALITY:

ARGUMENT-2:

CARDINALITY:

ATTRIBUTES:

END BINARY-RELATION br

RELATION r

ARGUMENTS:

END RELATION r

/\* tipos de conocimiento \*/

RULE-TYPE rt1

ANTECEDENT:

CARDINALITY:

CONSEQUENT:

CARDINALITY:

CONECTION-SYMBOL:

END RULE-TYPE rt1

**END DOMAIN-SCHEMA ds-1**

/\* bases de conocimiento (instancias y reglas) \*/

**KNOWLEDGE-BASE kb-1**

USES:

(rule-type y) FROM (domain-schema x)

EXPRESIONS:

(reglas)

**END KNOWLEDGE-BASE kb-1**

**END DOMAIN-KNOWLEDGE**

**INFERENCE-KNOWLEDGE if-1**

KNOWLEDGE-ROLE abstracted-case

TYPE: DYNAMIC

DOMAIN-MAPPING:solicitud

END KNOWLEDGE-ROLE abstract-case

KNOWLEDGE-ROLE Requierement

TYPE: STATIC

DOMAIN-MAPPING:

Requisitos-de-concesión FROM (ds-crepe)

END KNOWLEDGE-ROLE kr-2

/\* definiciones de las inferencias utilizadas \*/

/\* incluir sólo las modificaciones respecto a la estructura estandarizada \*/

INFERENCE se especifican los cambios realizados en la plantilla

ROLES:

INPUT: abstracted-case,norm

OUTPUT: norm value

STATIC: requierement

SPECIFICATION:

“deducimos el valor de los criterior a partir de la información del usuario”

END INFERENCE evaluate

**END INFERENCE-KNOWLEDGE**

/\* tareas \*/

/\*No es necesario realizarlo \*/

**TASK-KNOWLEDGE tk-1**

TASK t-1

DOMAIN-NAME:

GOAL:

ROLES:

INPUT:

OUTPUT:

END TASK t-1

TASK-METHOD tm-1

REALIZES:

DECOMPOSITION:

TASKS:

ROLES:

INTERMEDIATE:

CONTROL-STRUCTURE:

END TASK-METHOD tm-1

**END TASK-KNOWLEDGE tk-1**

END KNOWLEDGE-MODEL