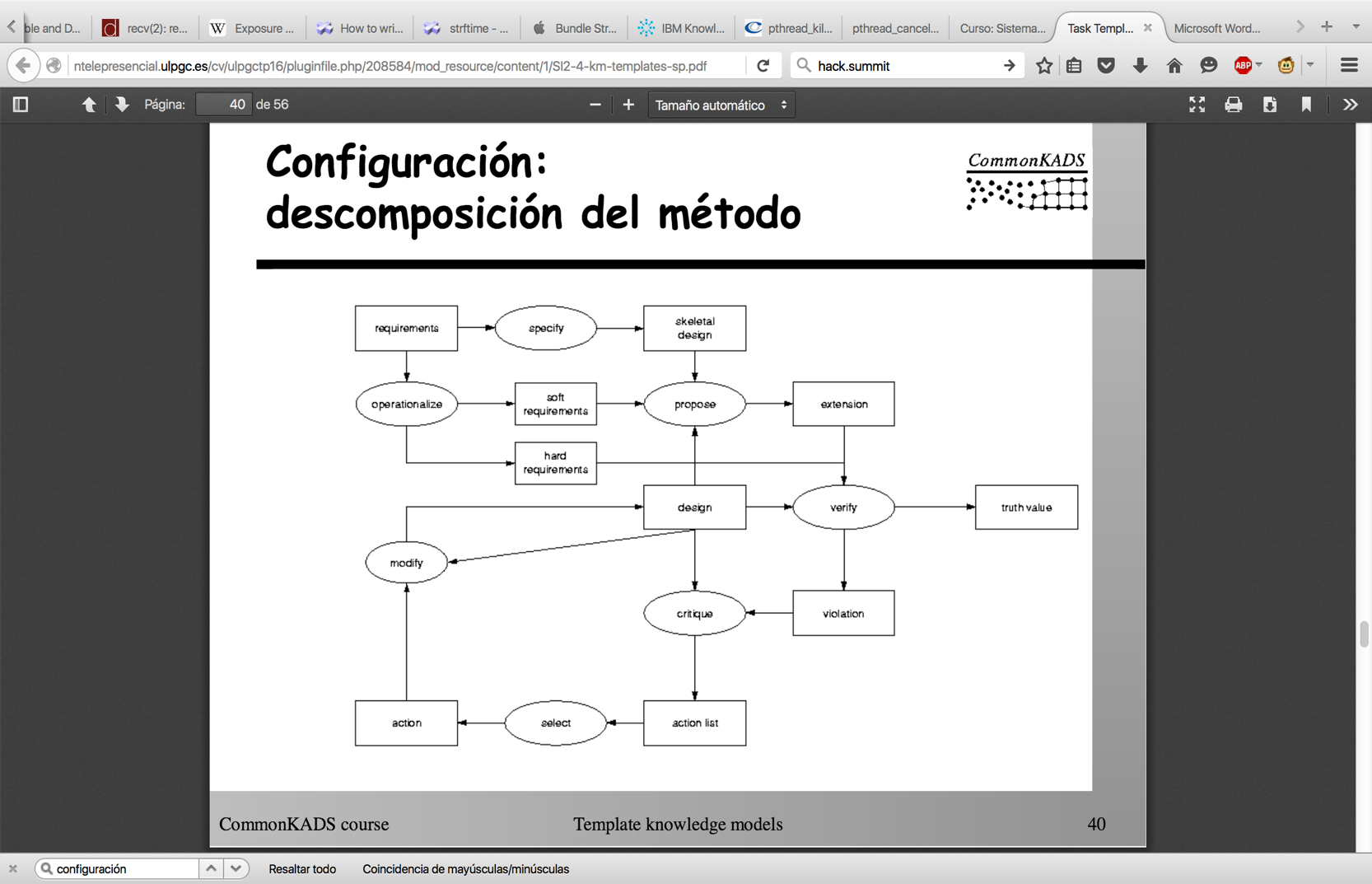
**Grupo: Laura del Pino Díaz y Aitor del Pino Saavedra Hernández**

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 laboratorio son de dos horas (Pasará a ser límite duro).
5. Solo existen tres laboratorios. (Limite duro)
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 Horario

DOMAIN-KNOWLEDGE Horario

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

DOMAIN-SCHEMA ds-horario

CONCEPT recinto

DESCRIPTION:

HAS-PARTS:

ATTRIBUTES:tipo{laboratorio,aula},id

END CONCEPT recinto

CONCEPT:profesor

DESCRIPTION:

HAS-PARTS:

ATTRIBUTES: ID, departamento

END CONCEPT profesor

CONCEPT:asignatura

DESCRIPTION:

HAS-PARTS:

ATTRIBUTES: tipo{troncal,no\_troncal}, nhorasTeoría, nhorasPráctica,departamento

END CONCEPT asignatura

CONCEPT:horario

DESCRIPTION:

HAS-PARTS: casillas

SUB-TYPE-OF:

ATTRIBUTES: validez

END CONCEPT horario

CONCEPT: casilla

DESCRIPTION:

HAS-PARTS: asignatura,professor,clase,recinto

ATTRIBUTES: hora-inicio, hora-fin, dia, validez

SUB-TYPE-OF:

END CONCEPT:casillas

CONCEPT:curso

DESCRIPTION:

HAS-PARTS: asignatura

SUB-TYPE-OF:

END CONCEPT:curso

CONCEPT:clase

DESCRIPTION:

HAS-PARTS:

SUB-TYPE-OF:

ATTRIBUTES: tipo{teoría, práctica}

END CONCEPT:clase

RELATION asignación info casillas

ARGUMENTS-1: profesor

CARDINALITY:1

ARGUMENTS-2:asignatura

CARDINALITY:1

ARGUMENTS-3:clase

CARDINALITY-3:1

ARGUMENTS-4:recinto

CARDINALITY-4:1

END RELATION asignación info casillas

RELATION verificación asignación

ARGUMENTS-1: horario

CARDINALITY:1

ARGUMENTS-2:casillas

CARDINALITY:\*

ARGUMENTS-3:asignatura

CARDINALITY-3:\*

END RELATION verificación 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-horario**

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

**KNOWLEDGE-BASE kb-horarios**

USES:

Requisitos-de-asignación FROM (ds-horarios)

EXPRESIONS:

“cond-inic1: si profesor.departamento != asignatura.departamento ESTABLECE casilla.validez = falso

“cond-inic2: si la clase es de teoría entonces recinto es del tipo aula.

clase.teoria ESTABLECE recinto.aula

“cond-inic3: si la clase es de práctica entonces recinto es del tipo laboratorio”

clase.practica ESTABLECE recinto.laboratorio

AND

casilla.hora-fin = casilla.hora-inicio + 2

“ cond-inic4: si las horas de clase de teoría o práctica son menores que las que debe tener, el horario no es válido”

ESTABLECE horario.validez = falso;

“cond-inic5: si dos clases de práctica se imparten en el mismo momento entonces se deben dar en distintos laboratorios”

(casilla1.hora-inicio == casilla2.hora-inicio and casilla1.dia == casilla2.dia and casilla1.recinto.id == casilla2.recinto.id ESTABLECE horario.validez = falso)

“cond-inic6: si se imparte dos horas de clase de teoría estas deben ser consecutivas”

(casilla1.asignatura.id == casilla2.asignatura.id and casilla1.dia == casilla2.dia and casilla1.hora-fin ¡= casilla2.hora-inicio ESTABLECE horario.validez = falso)

“cond-inic7: las asignaturas troncales deben estar consecutivas a partir de la primera hora de la mañana”

**END KNOWLEDGE-BASE kb-horarios**

**END DOMAIN-KNOWLEDGE**

**INFERENCE-KNOWLEDGE horarios**

KNOWLEDGE-ROLE requirements

TYPE: DYNAMIC

DOMAIN-MAPPING: requerimientos para el diseño del horario

END KNOWLEDGE-ROLE requirements

KNOWLEDGE-ROLE soft-requirements

TYPE: DYNAMIC

DOMAIN-MAPPING: Preferencias

END KNOWLEDGE-ROLE soft-requirements

KNOWLEDGE-ROLE hard-requirements

TYPE: DYNAMIC

DOMAIN-MAPPING: Restricciones

END KNOWLEDGE-ROLE hard-requirements

INFERENCE clasificar requerimientos (operationalize)

ROLES:

INPUT: requirements

OUTPUT: soft-requirements,hard-requirements

SPECIFICATION:

A partir de los requerimientos clasificarlos entre preferencias y restricciones

END INFERENCE clasificar requerimientos

KNOWLEDGE-ROLE skeletal design

TYPE: STATIC

DOMAIN-MAPPING: diseño esquelético del horario.

END KNOWLEDGE-ROLE skeletal design

INFERENCE creación de horarios (propose)

ROLES:

INPUT: soft requirements,skeletal design

OUTPUT: extension

SPECIFICATION:

A partir de las preferencias y el diseño esquelético proponemos un horario

END INFERENCE creación de horarios

KNOWLEDGE-ROLE violation

TYPE: STATIC

DOMAIN-MAPPING: no se cumple restricciones (hard-requirements)

END KNOWLEDGE-ROLE violation

KNOWLEDGE-ROLE truth value

TYPE: STATIC

DOMAIN-MAPPING horario correcto.

END KNOWLEDGE-ROLE truth value

INFERENCE verificación de horario (verify)

ROLES:

INPUT: extension

OUTPUT: truth value || violation

SPECIFICATION:

A partir del diseño de un horario indicar si es correcto o no.

END INFERENCE verificación de horario

KNOWLEDGE-ROLE design

TYPE: DYNAMIC

DOMAIN-MAPPING Nuevo diseño de horario.

END KNOWLEDGE-ROLE design

INFERENCE modificación de horario (modify)

ROLES:

INPUT: extension,violation

OUTPUT: design

SPECIFICATION: nuevo diseño de horario corrigiendo las restricciones no cumplidas.

END INFERENCE modificación de horario

**END INFERENCE-KNOWLEDGE horarios**